中国科学技术大学
University of Science and Technology of China

数学建模
Mathematical Modeling

陈仁杰
中国科学技术大学

数据建模方法

聚类分析模型

Clustering Analysis


物以类聚
人以群分

蠓虫分类问题

（一）背景知识

两种蠓虫 Af 和 Apf 已由生物学家罗纳（w.L.Grogna）和维尔恩（W.W.Wirth）于 1981 年根据它们的触角长（mm）和翅长（mm）加以区分，6 只 Apf 和 9 只 Af 蠓虫的触长,翅长数据如下：

Apf：(1.14,1.78)，(1.18,1.96)，(1.20,1.86)，(1.26,2.00)，(1.28,2.00)，(1.30,1.96)；

Af：(1.24,1.72)，(1.36,1.74)，(1.38,1.64)，(1.38,1.82)，(1.38,1.90)，(1.40,1.70)，(1.48,1.82),(1.54,1.82)，(1.56,2.08)。

在生物学中，根据触角长和翅长来识别一只蠓虫标本是 Af 还是 Apf 是很重要的。

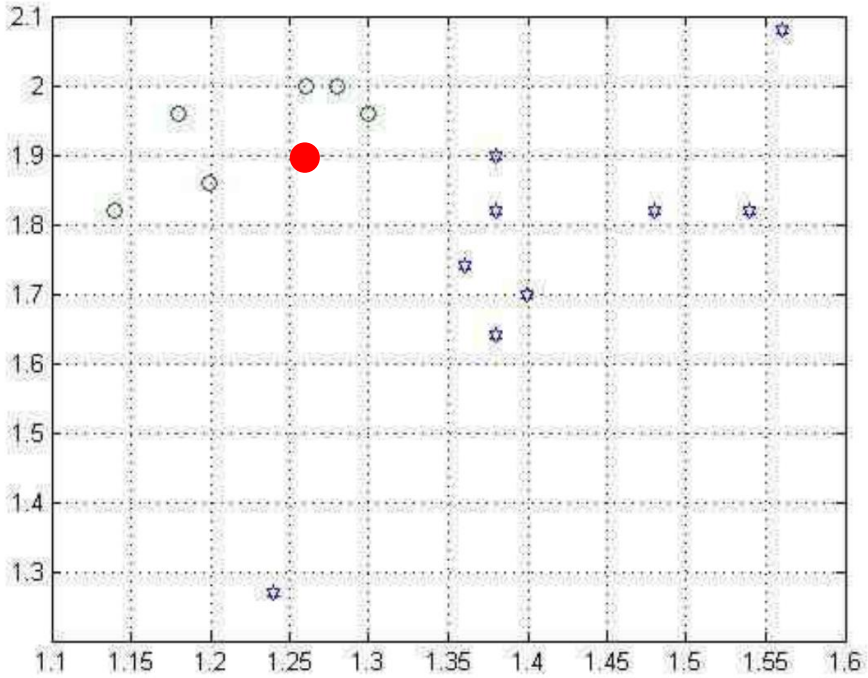
（二）要解决的问题

1、根据给定的数据，制定一种方法，正确区分两类蠓虫；

2、用我们的方法对触长、翅长分别为(1.24,1.80)、(1.28,1.84)、(1. 40,2.04)的三个样本进行识别；

3、假设 Af 是宝贵的传粉益虫，Apf 是某种疾病的载体，在这种情况下我们是否应该修改所用的分类方法，且如何修改。

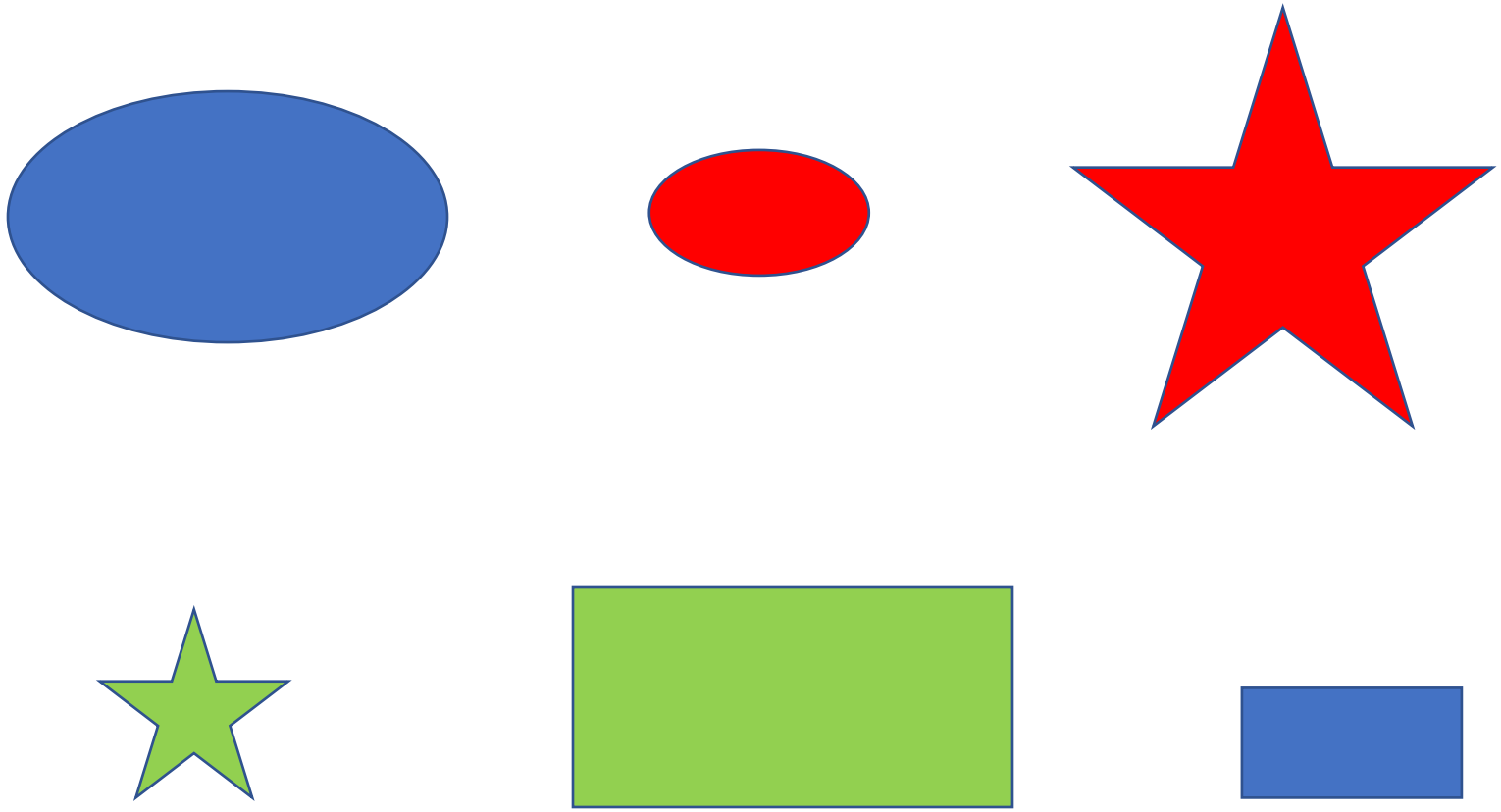
蠓虫分类问题：有监督分类



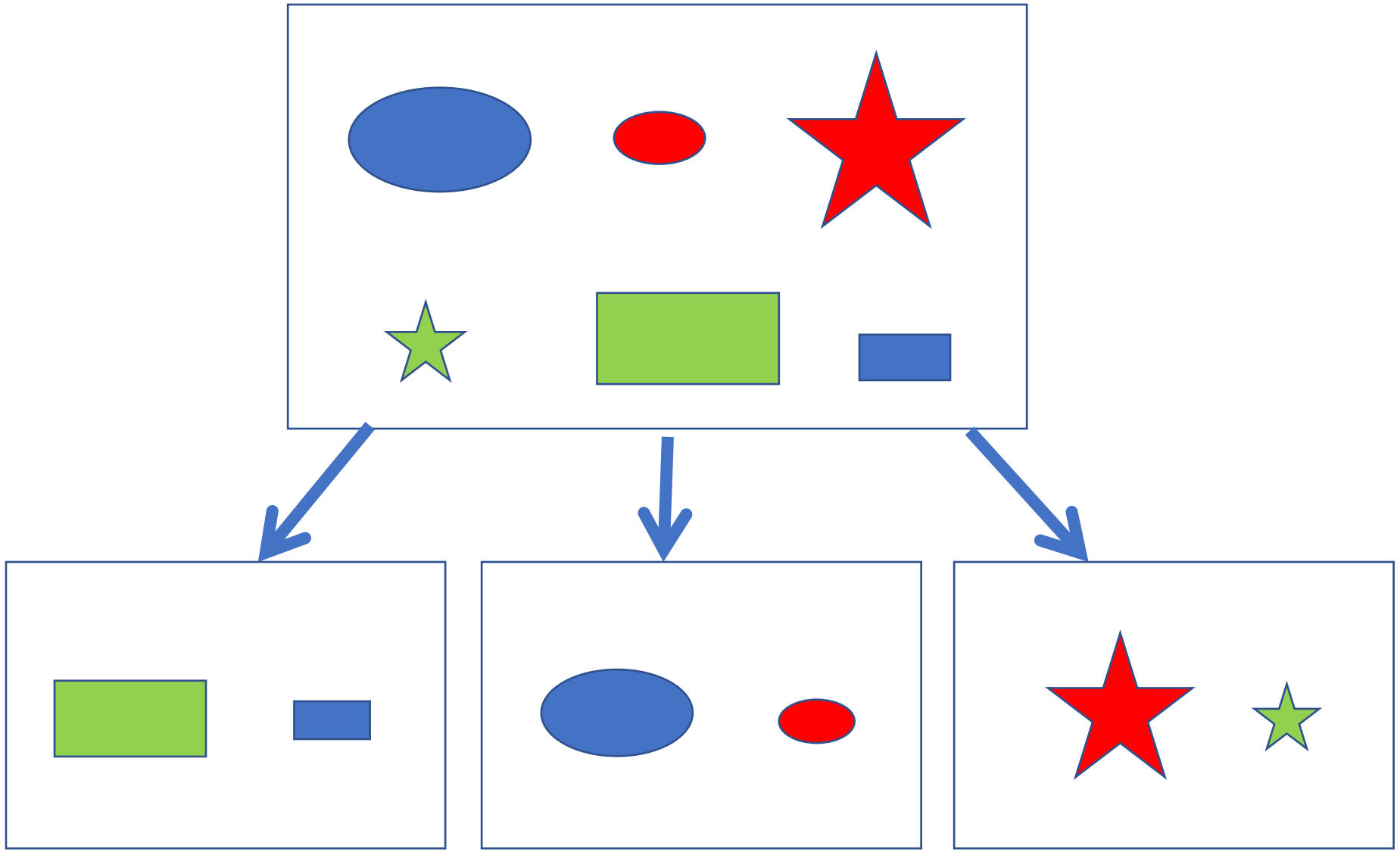
Af	1	2	3	4	5	6	7	8	9
触角长度	1.24	1.36	1.38	1.38	1.38	1.4	1.48	1.54	1.56
翼长	1.72	1.74	1.64	1.82	1.9	1.7	1.82	1.82	2.08

Apf	1	2	3	4	5	6
触角长度	1.14	1.18	1.20	1.26	1.28	1.30
翼长	1.78	1.96	1.86	2.00	2.00	1.96

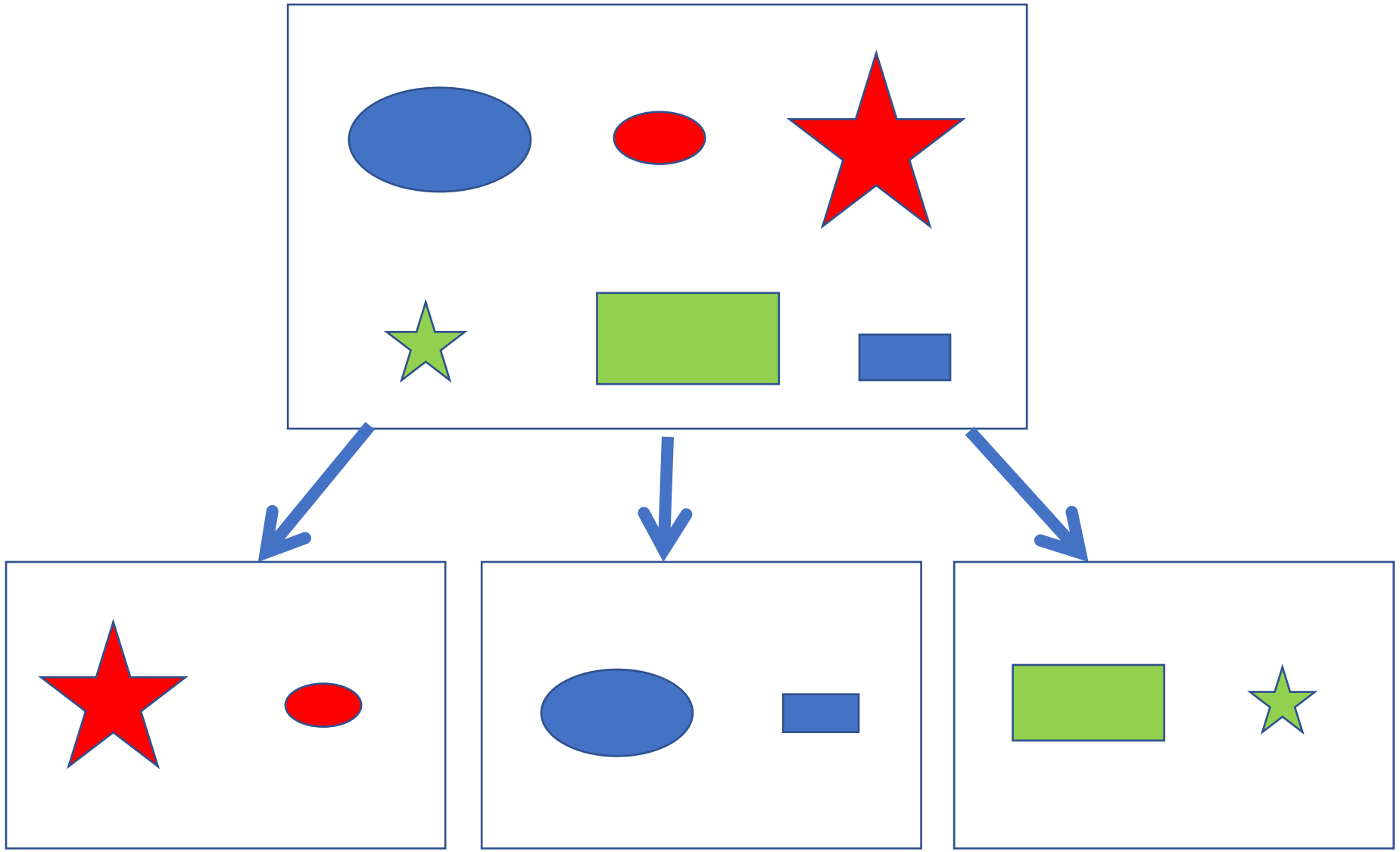
图形分类：无监督分类
下列图形能分成哪几类？



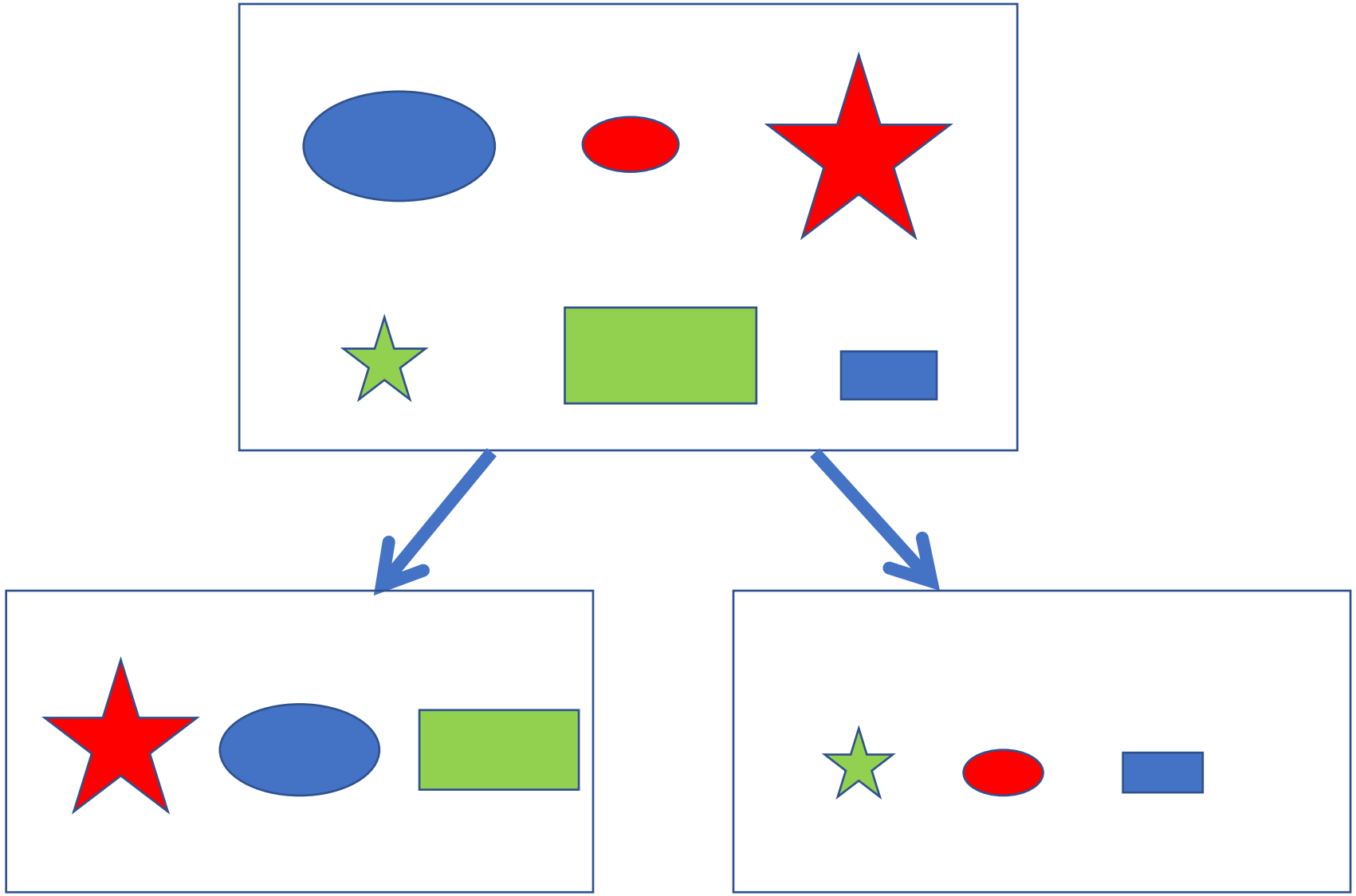
分类-1：按形状



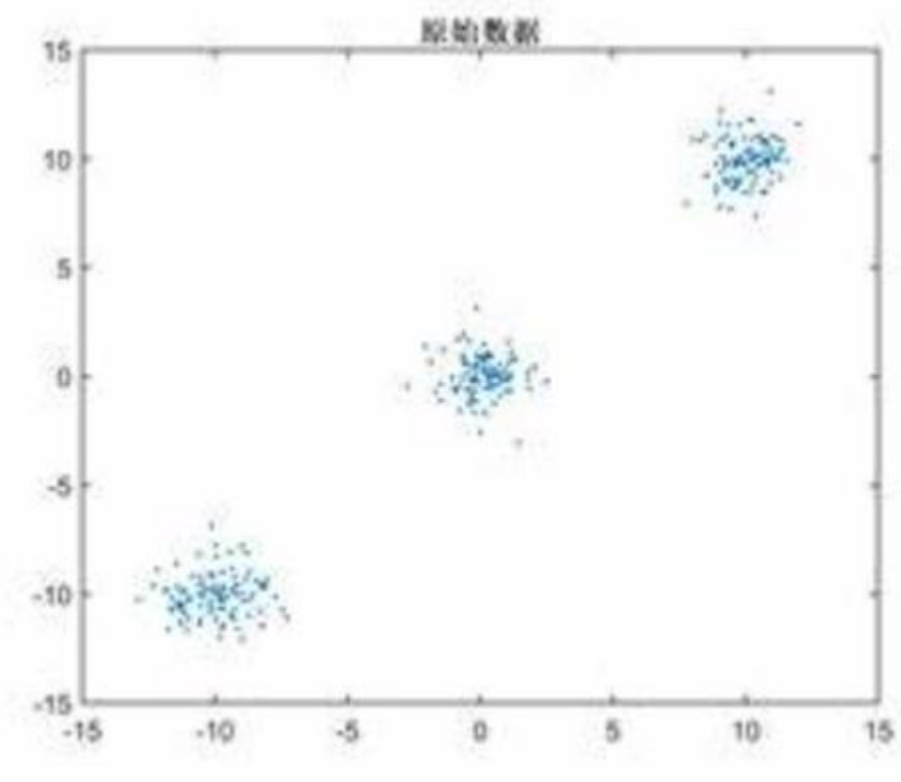
分类-2：按颜色



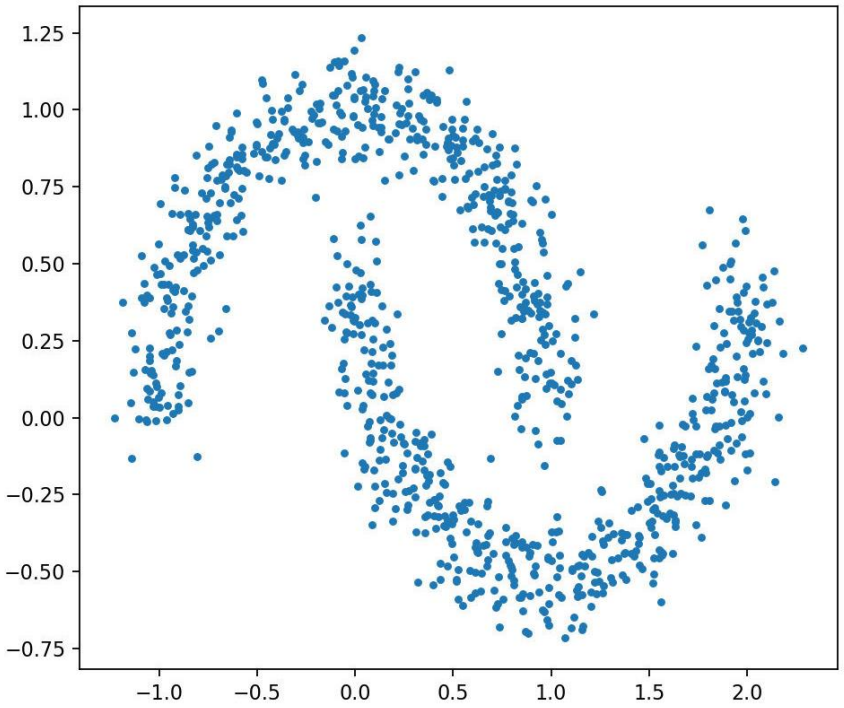
分类-3：按大小

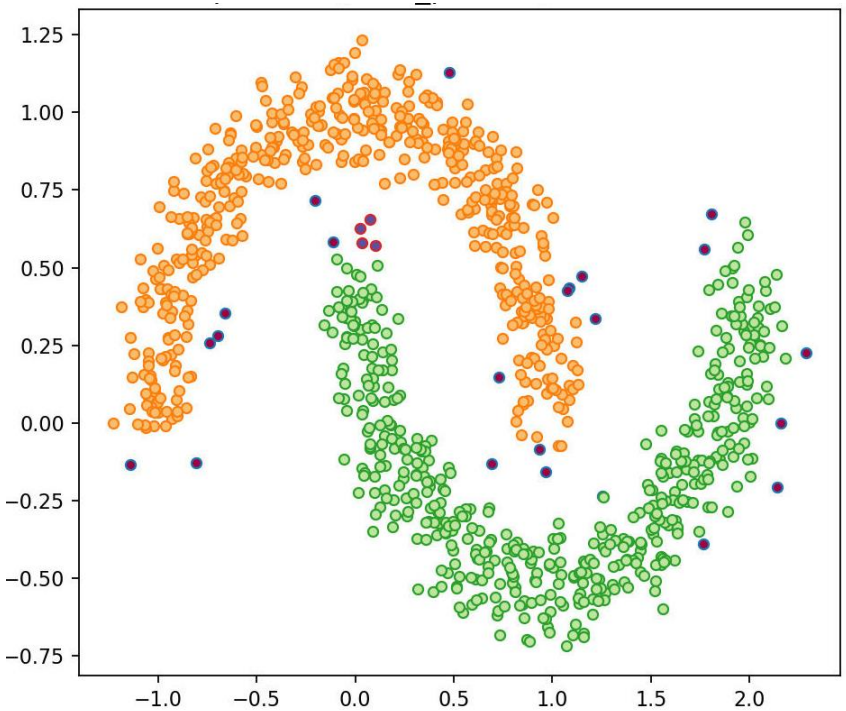


例子：分几类？

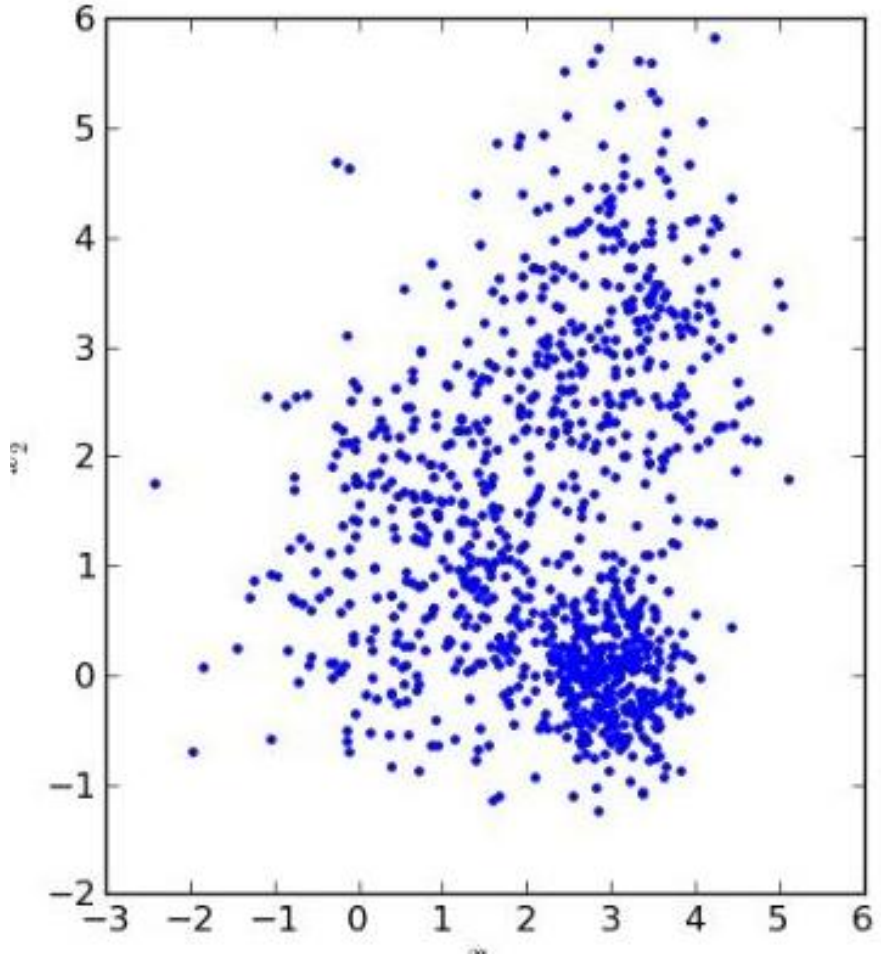


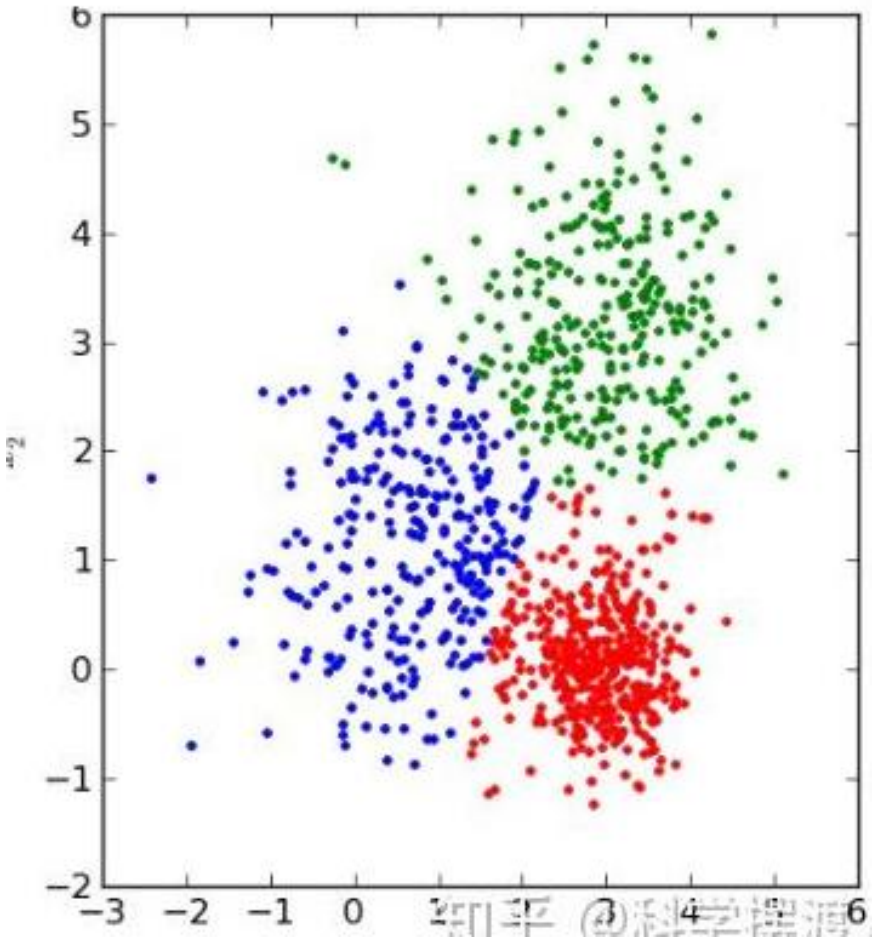
例子：分几类？


1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

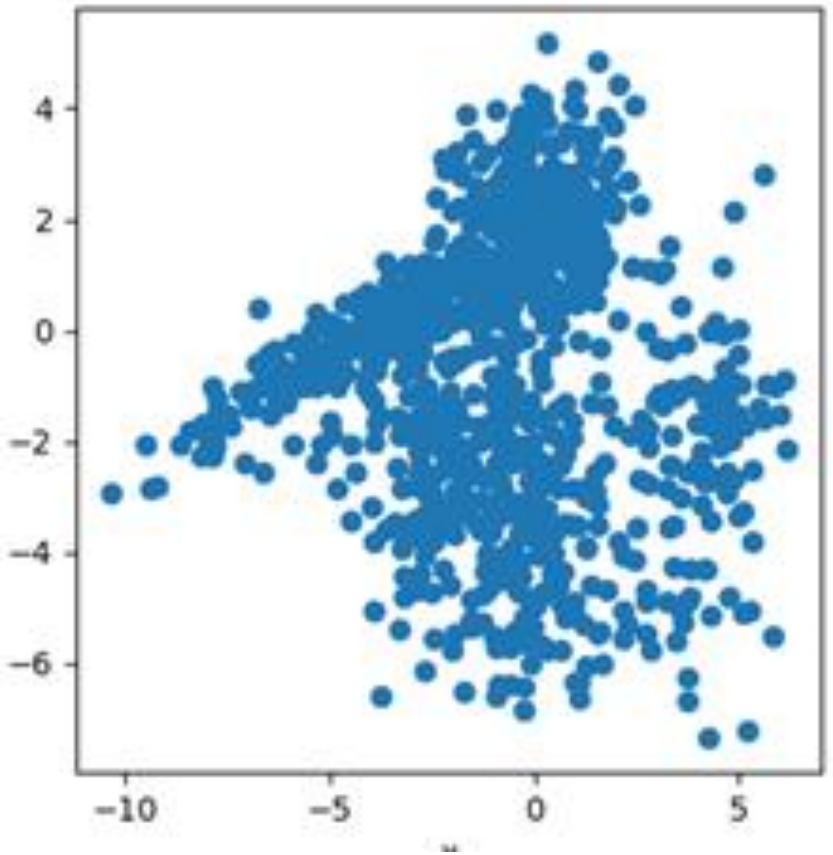

1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

例子：分几类？

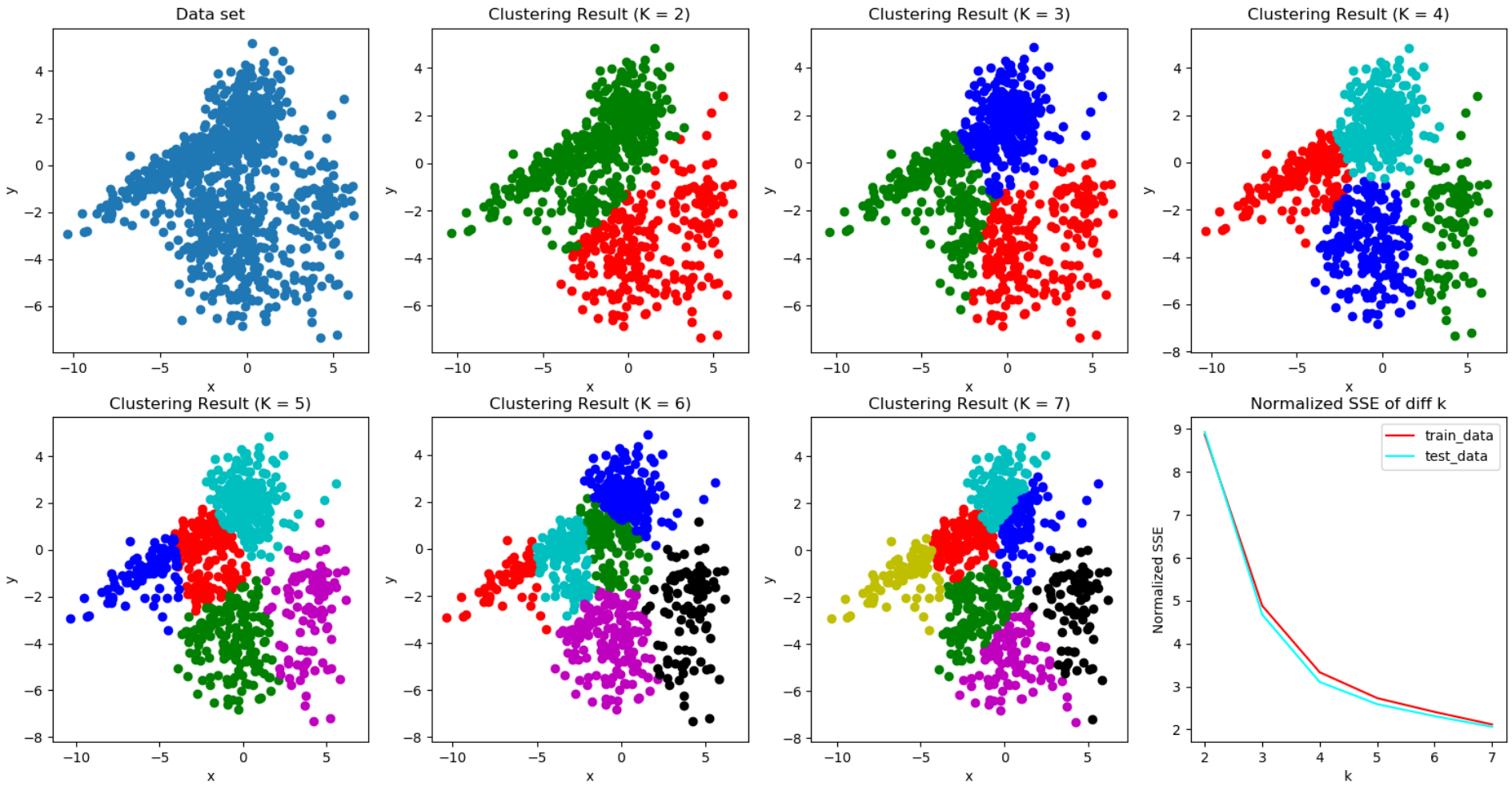




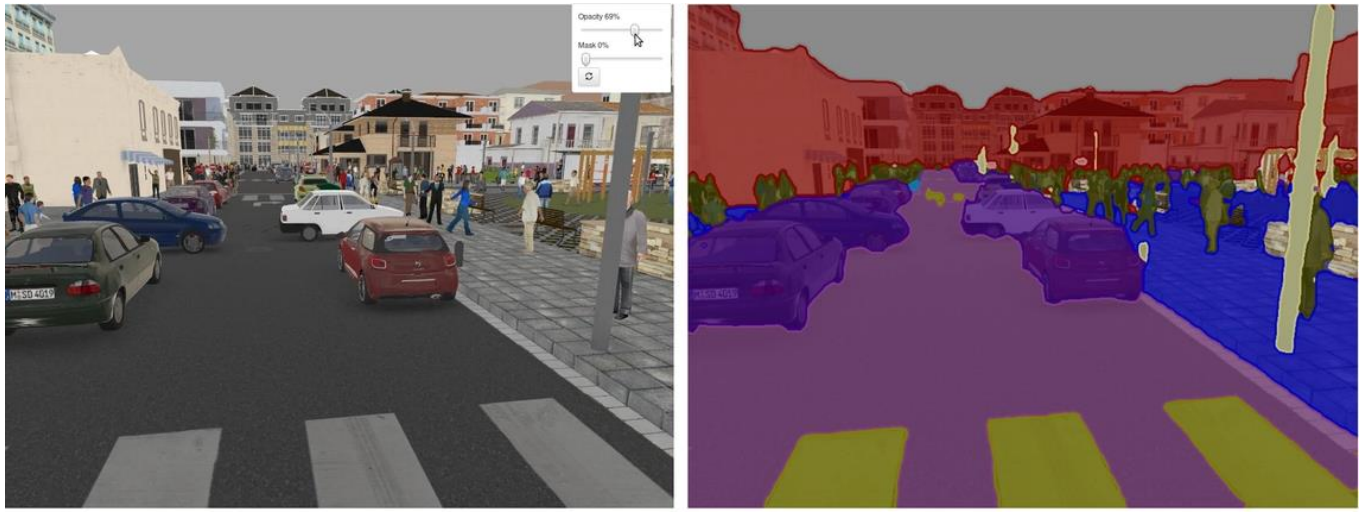
例子：分几类？

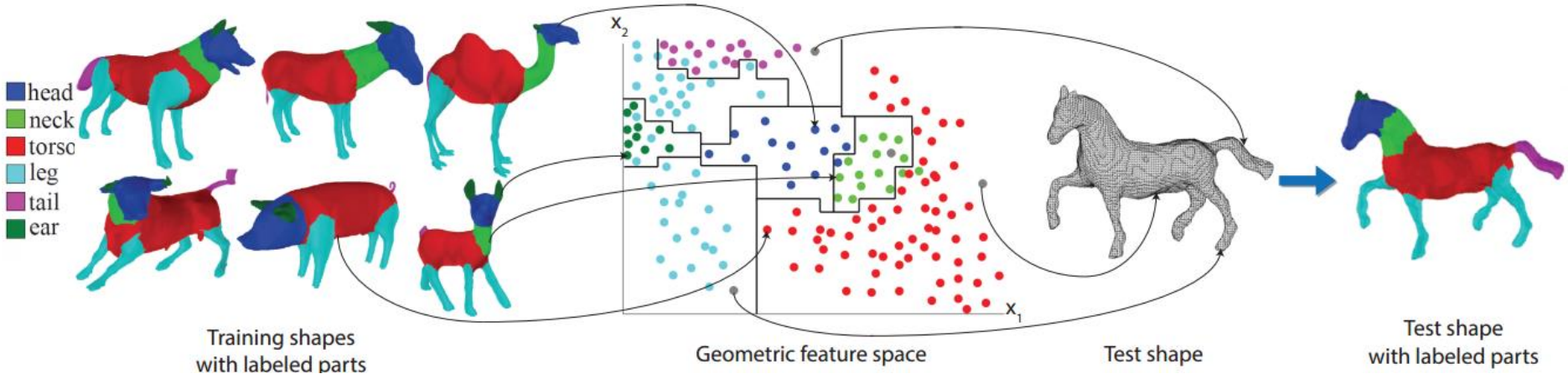


例子：分几类？


Data set
Clustering Result (K = 2)
Clustering Result (K = 3)
Clustering Result (K = 4)
Clustering Result (K = 5)
Clustering Result (K = 6)
Clustering Result (K = 7)
Normalized SSE of diff k
train_data
test_data
Normalized SSE
x
y
k

应用：图像分割与形状分割





聚类分析 (cluster analysis)

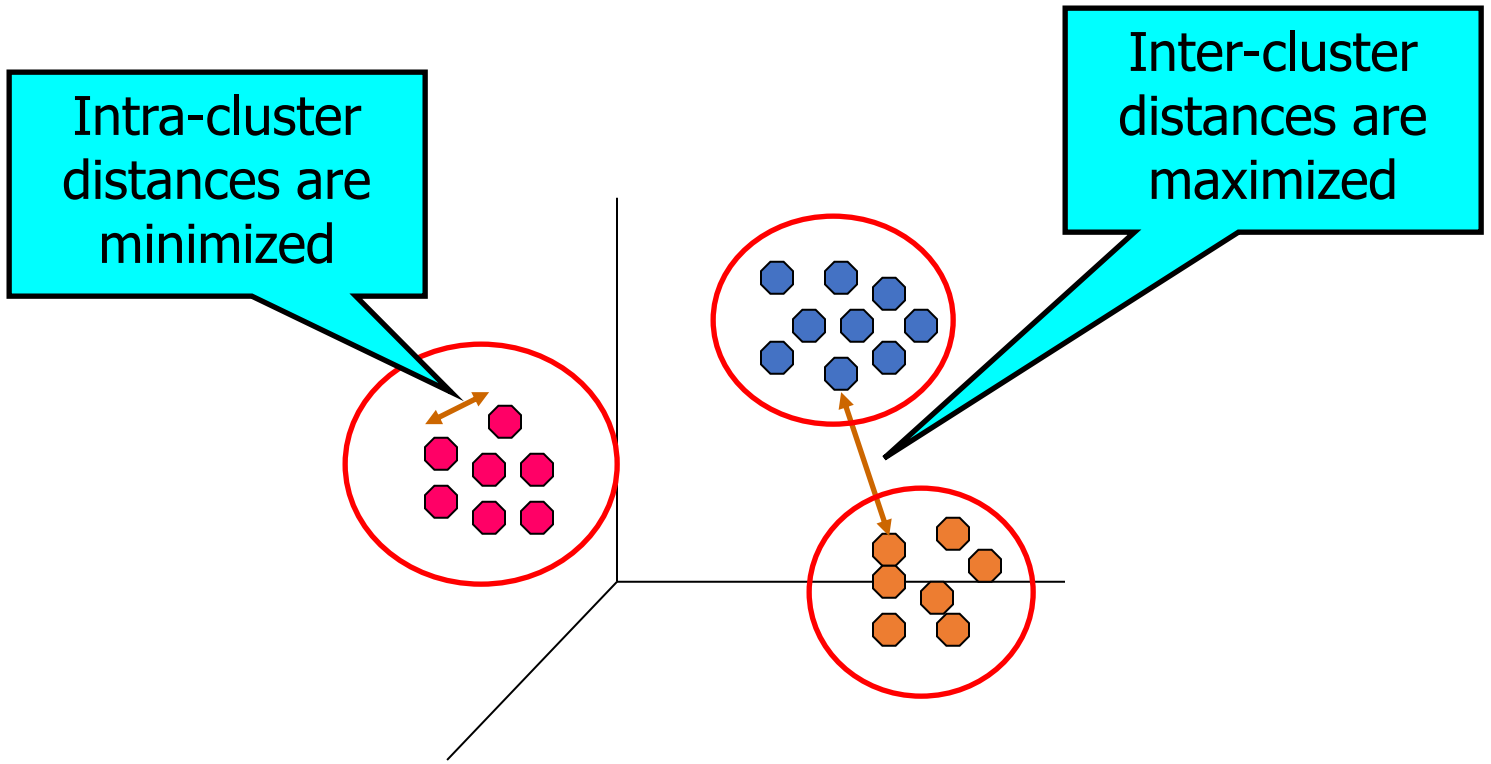
- 聚类分析指将物理或抽象对象的集合分组为由类似的对象组成的多个类的分析过程。

定义:

- 给定一个数据点集，聚类算法将每个数据点分类到一个特定的组中。
- 同一组数据点具有相似的性质或（和）特征，不同组数据点具有高度不同的性质或（和）特征。

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



What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

General Applications of Clustering

- Pattern Recognition
- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

聚类分析的应用

- 数据预处理过程中，对于复杂结构的多维数据可以通过聚类分析的方法对数据进行聚集，使复杂结构数据标准化。
- 用来发现数据项之间的依赖关系，从而去除或合并有密切依赖关系的数据项。
- 为某些数据挖掘方法（如关联规则、粗糙集方法），提供预处理功能。
- 在商业类问题上，聚类分析是细分市场的有效工具，被用来发现不同的客户群，并且它通过对不同的客户群的特征的刻画，被用于研究消费者行为，寻找新的潜在市场等等。

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

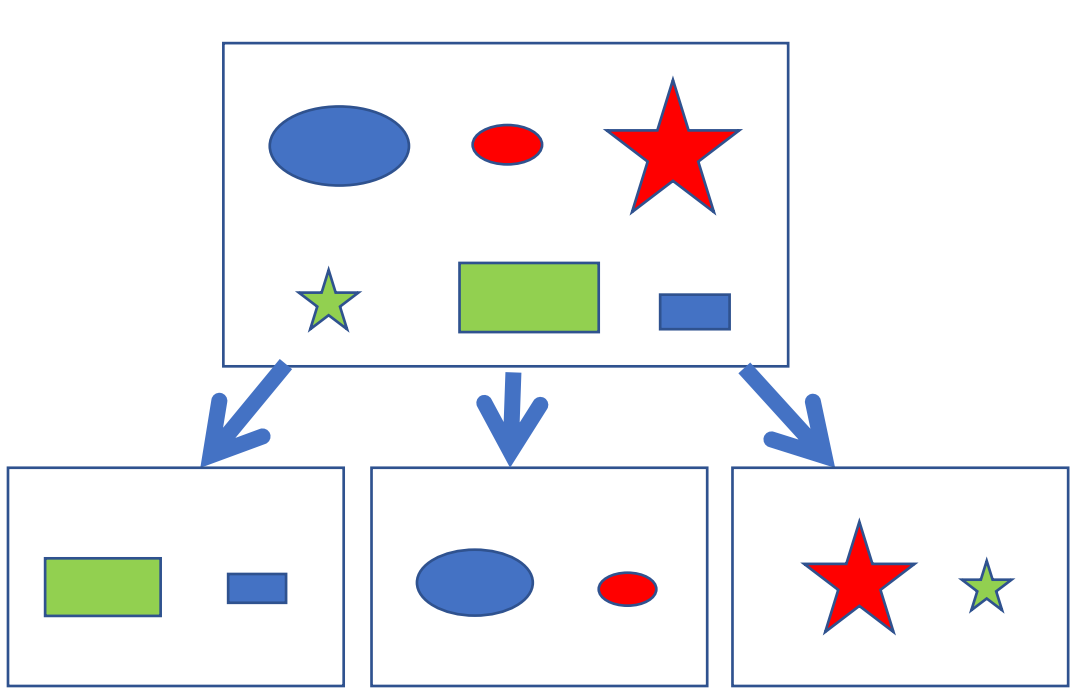
Two Clusters

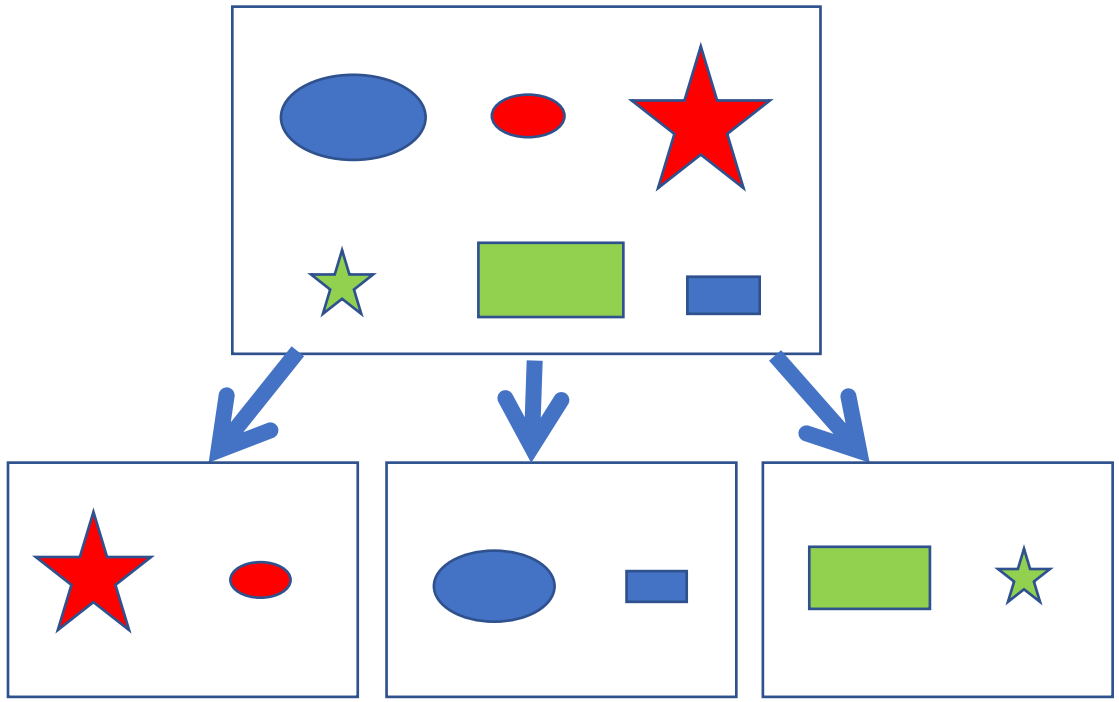
Four Clusters

What Is Good Clustering?

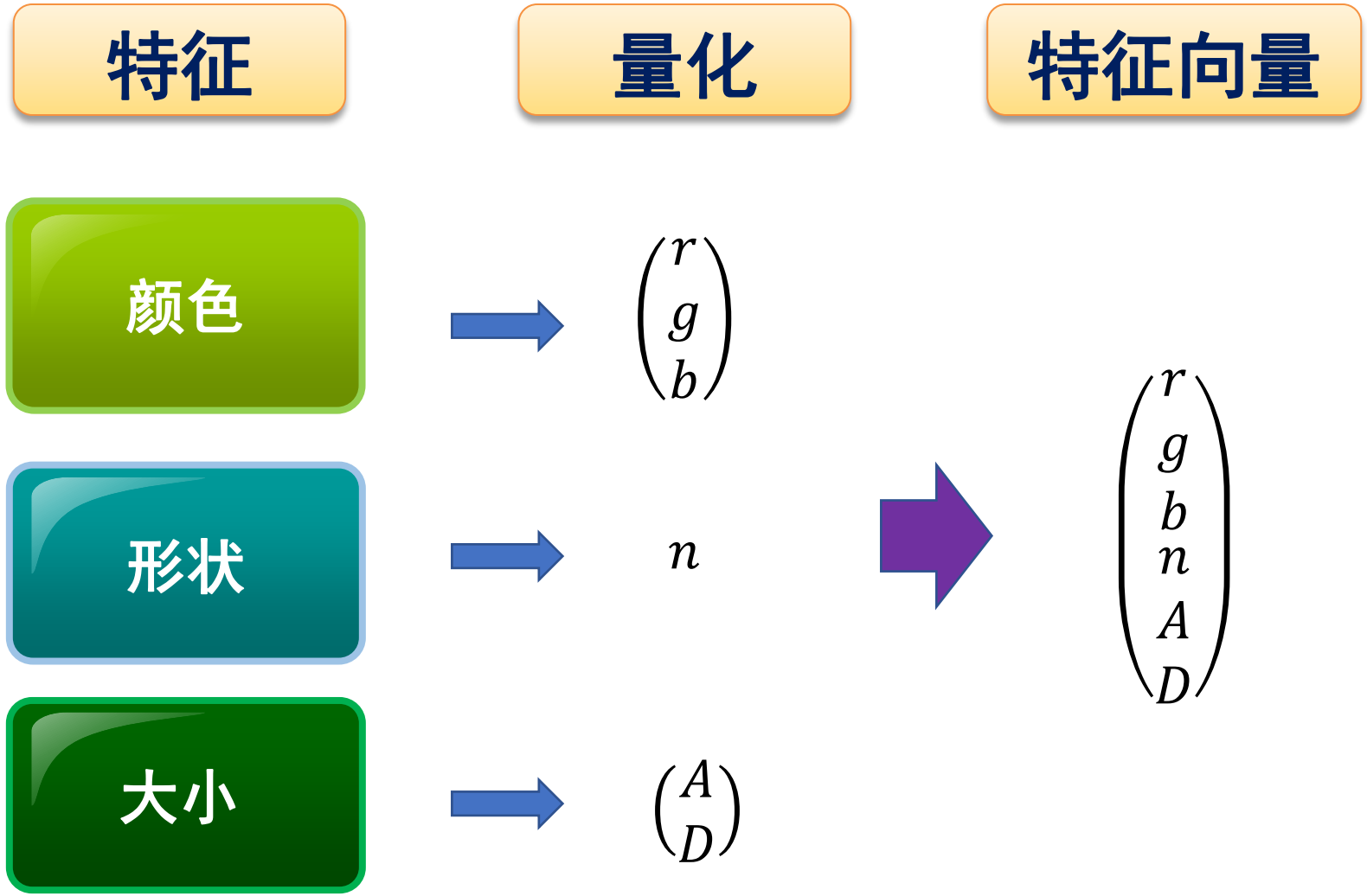
- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

不同的特征导致不同的分类结果





特征


特征
量化
特征向量
颜色
形状
大小
$\begin{pmatrix} r \\ g \\ b \end{pmatrix}$
n
$\begin{pmatrix} A \\ D \end{pmatrix}$
$\begin{pmatrix} r \\ g \\ b \\ n \\ A \\ D \end{pmatrix}$

聚类分析的过程

- 特征选择和提取
- 度量：距离和相似系数
- 聚类算法设计
- 聚类验证

距离

- 绝对值距离 $d_{ij}(1)=\sum_{k=1}^{p}\left|x_{ik}-x_{jk}\right|$
- 欧氏(Euclidean)距离 $d_{ij}(2)=\left[\sum_{k=1}^{p}\left(x_{ik}-x_{jk}\right)^{2}\right]^{1/2}$
- 切比雪夫(Chebychev)距离 $d_{ij}(\infty)=\max_{1\le k\le p}\left|x_{ik}-x_{jk}\right|$
- 明氏(Minkowski)距离 $d_{ij}(q)=\left(\sum_{k=1}^{p}\left|x_{ik}-x_{jk}\right|^{q}\right)^{1/q}$

相似系数

- 夹角余弦 $$\cos\theta_{st} = \frac{\sum_{i=1}^{n} x_{is} x_{it}}{\sqrt{\sum_{i=1}^{n} x_{is}^2 \cdot \sum_{i=1}^{n} x_{it}^2}}$$

- Pearson相关系数 $$r_{st} = \frac{\sum_{i=1}^{n} (x_{is} - \bar{x}_s)(x_{it} - \bar{x}_t)}{\sqrt{\sum_{i=1}^{n} (x_{is} - \bar{x}_s)^2 \cdot \sum_{i=1}^{n} (x_{it} - \bar{x}_t)^2}}$$

两种主要的聚类方法

- **层次聚类方法：Hierarchical Clustering** – final clusters are built following a distinct set of sequential steps

- **非层次聚类方法： Non-Hierarchical Clustering** – Clusters are built in such a way that if M clusters are built there is no guarantee that putting together two of the clusters would give rise to the same (M-1) clusters built separately by the method.

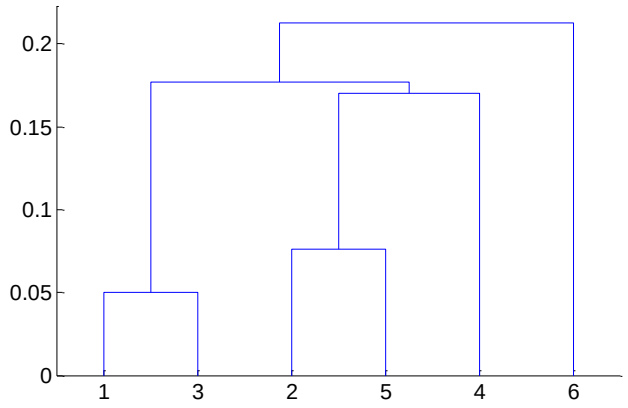
两种主要的聚类方法

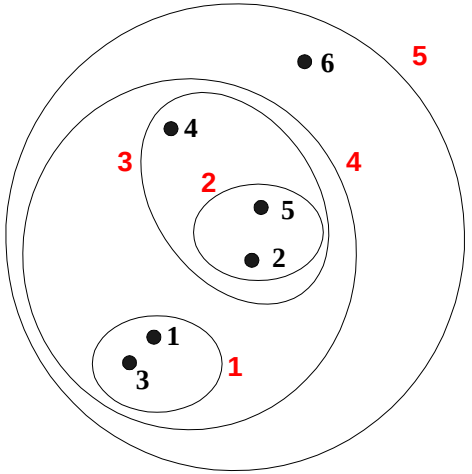
- **层次聚类方法：Hierarchical Clustering** – final clusters are built following a distinct set of sequential steps

- **非层次聚类方法： Non-Hierarchical Clustering** – Clusters are built in such a way that if M clusters are built there is no guarantee that putting together two of the clusters would give rise to the same (M-1) clusters built separately by the method.

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 merges or splits





Strengths of Hierarchical Clustering

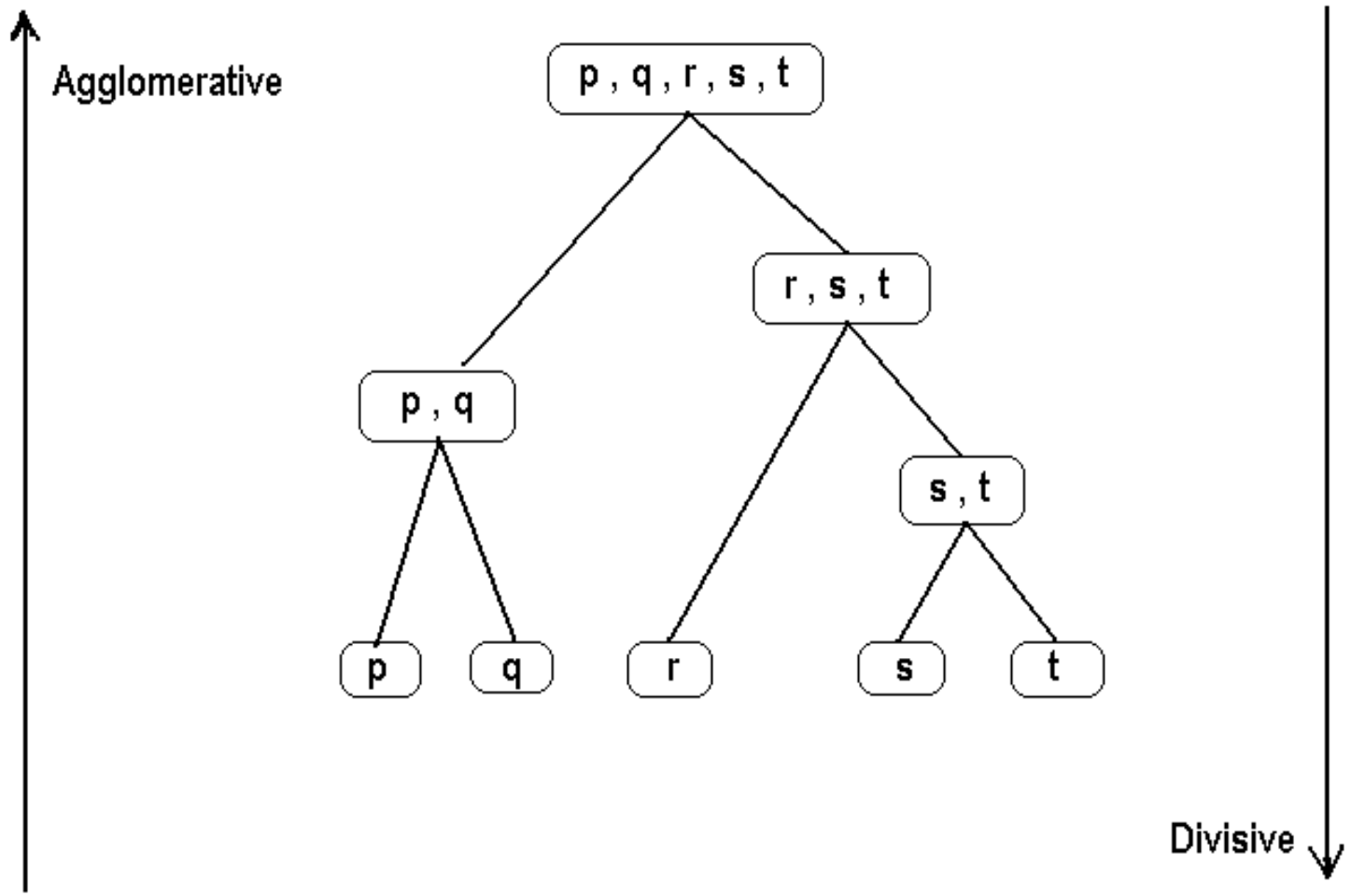
- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

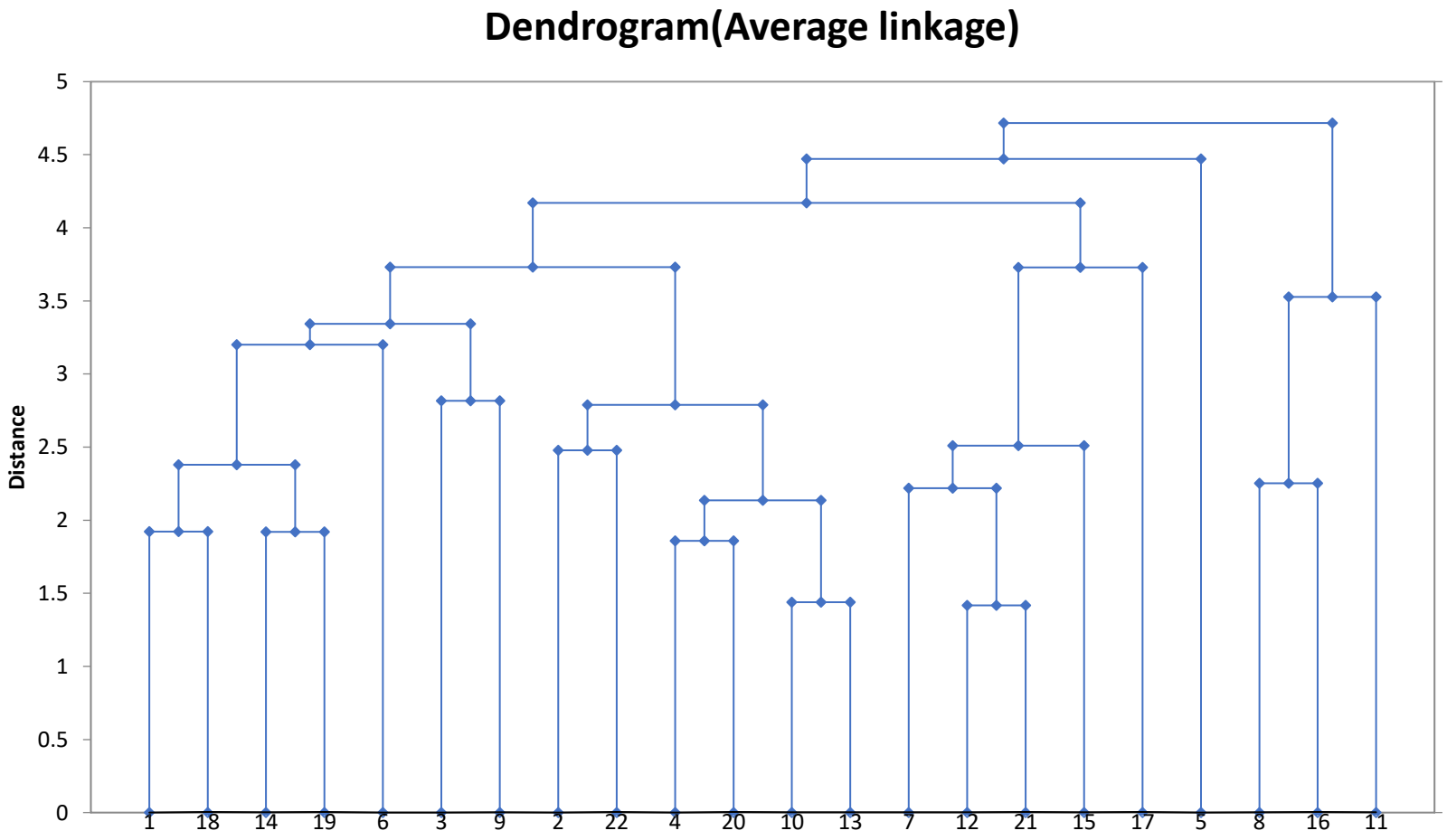
Hierarchical Clustering

- 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 versus **Divisive** Approaches



The Dendrogram of Hierarchical Clustering



Complexity of hierarchical clustering

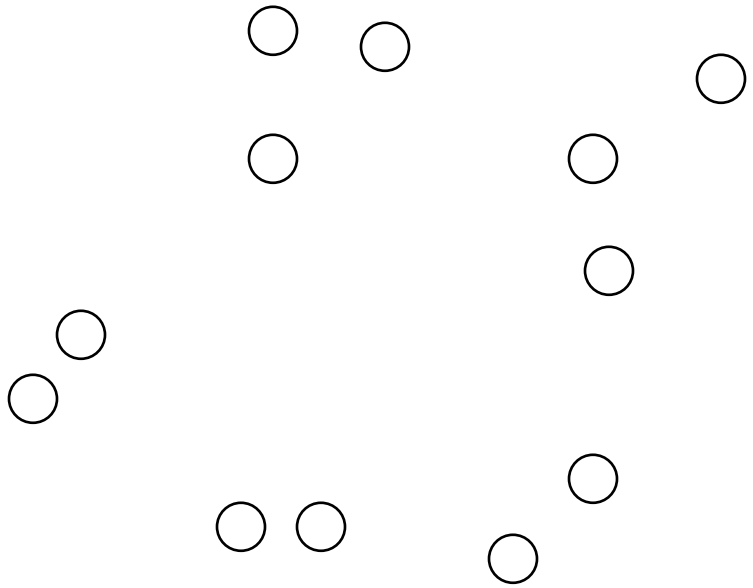
- Distance matrix is used for deciding which clusters to merge/split
- At least quadratic in the number of data points
- Not usable for large datasets

Agglomerative clustering algorithm

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

Input/ Initial setting

- Start with clusters of individual points and a distance/proximity matrix

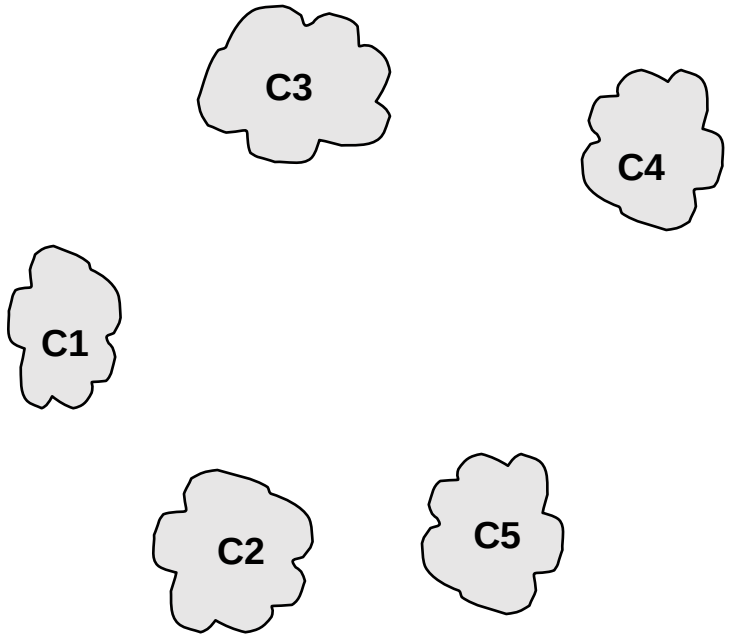


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						

Distance/Proximity Matrix

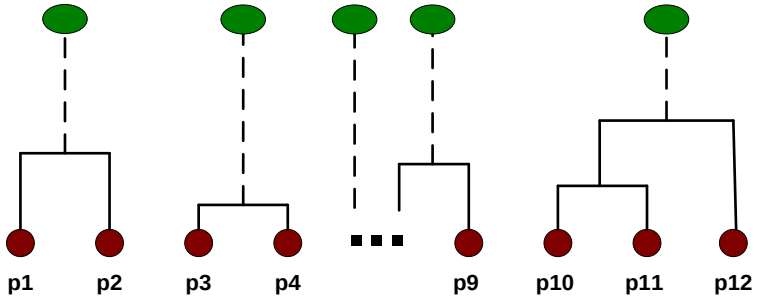
Intermediate State

- After some merging steps, we have some clusters



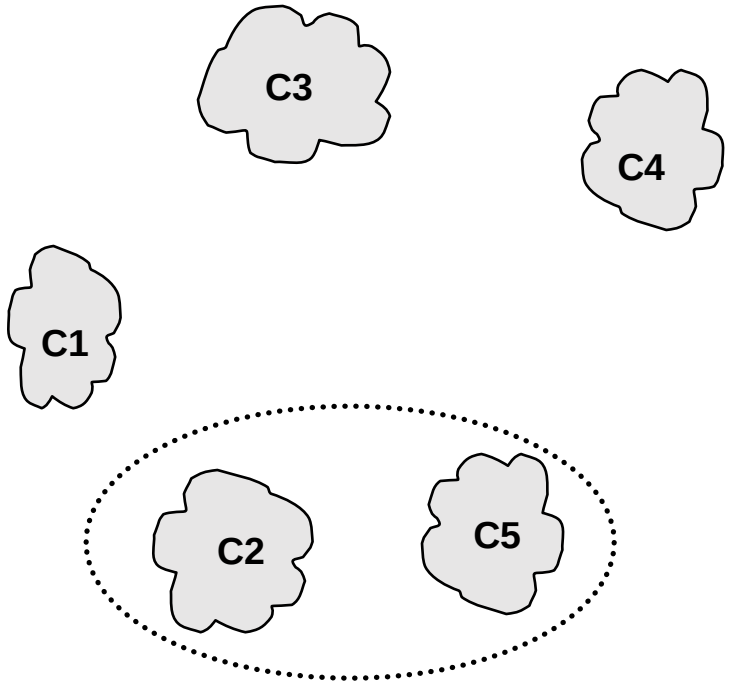
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix



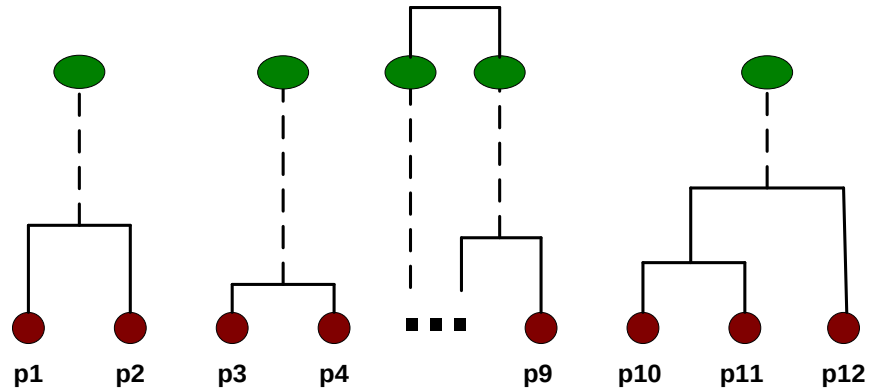
Intermediate State

- Merge the two closest clusters (C2 and C5) and update the distance matrix.



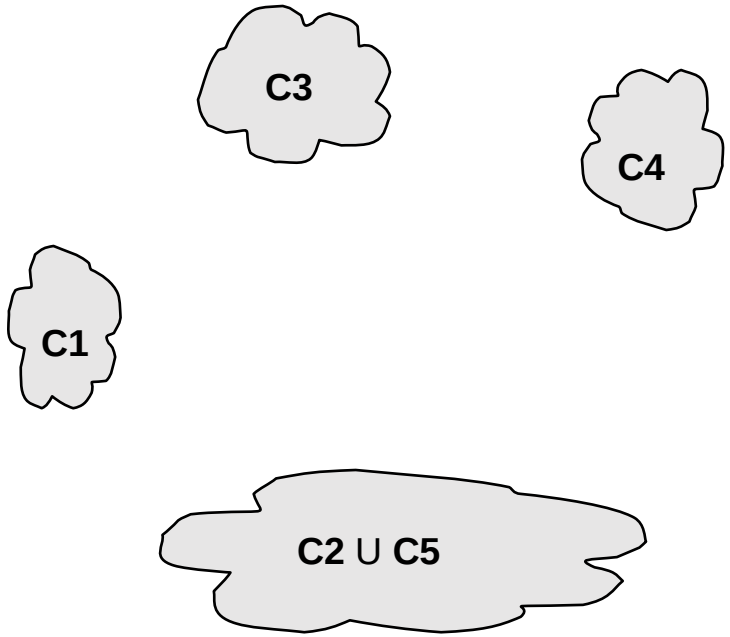
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

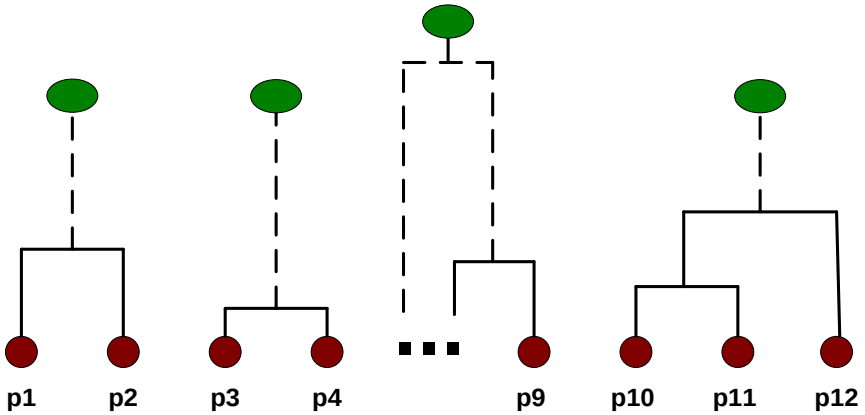


After Merging

- “How do we update the distance matrix?”



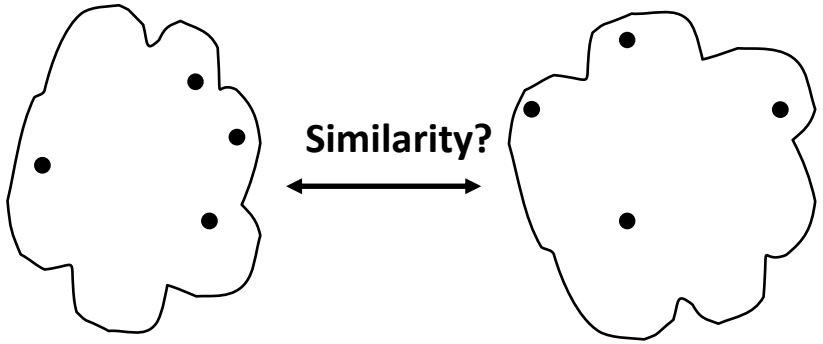
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

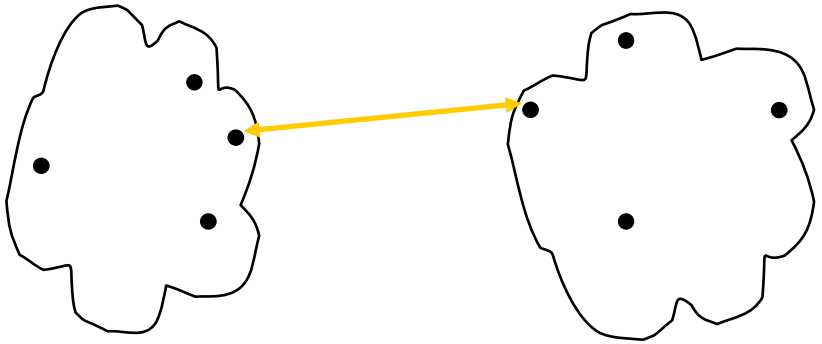


Distance between two clusters

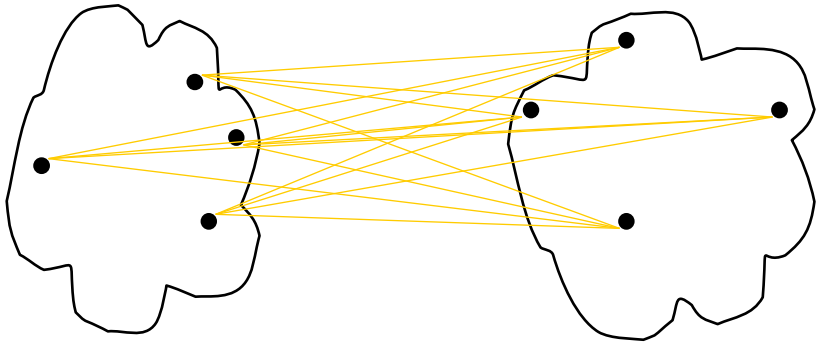
- Each cluster is a set of points

- How do we define distance between two sets of points
  - Lots of alternatives
  - Not an easy task

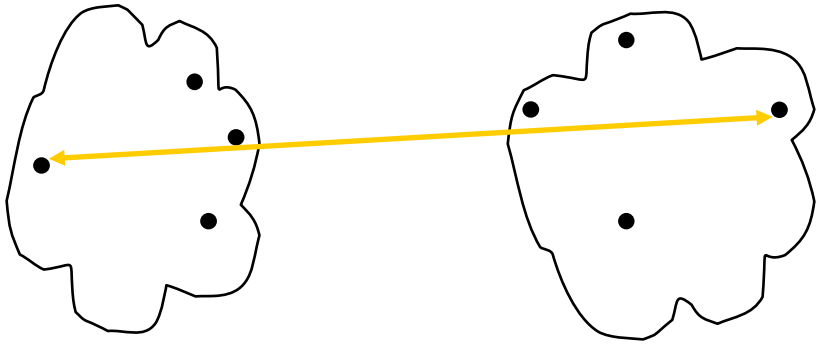




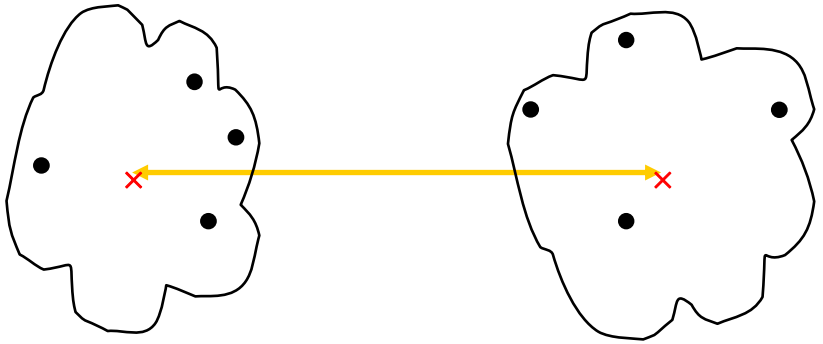
Min



Average



Max



Group

Distance between two clusters

- **Single-link distance** between clusters C_i and C_j is the ***minimum distance*** between any object in C_i and any object in C_j

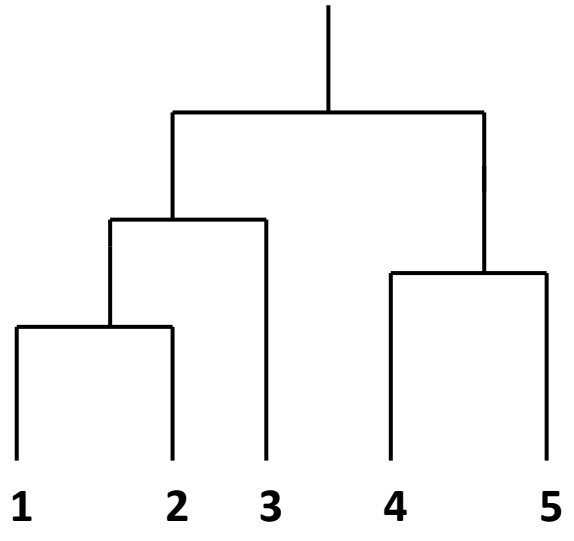
- The distance is **defined by the two most similar objects**

$$D_{sl}(C_i, C_j) = \min_{x,y} \left\{ d(x, y) \middle| x \in C_i, y \in C_j \right\}$$

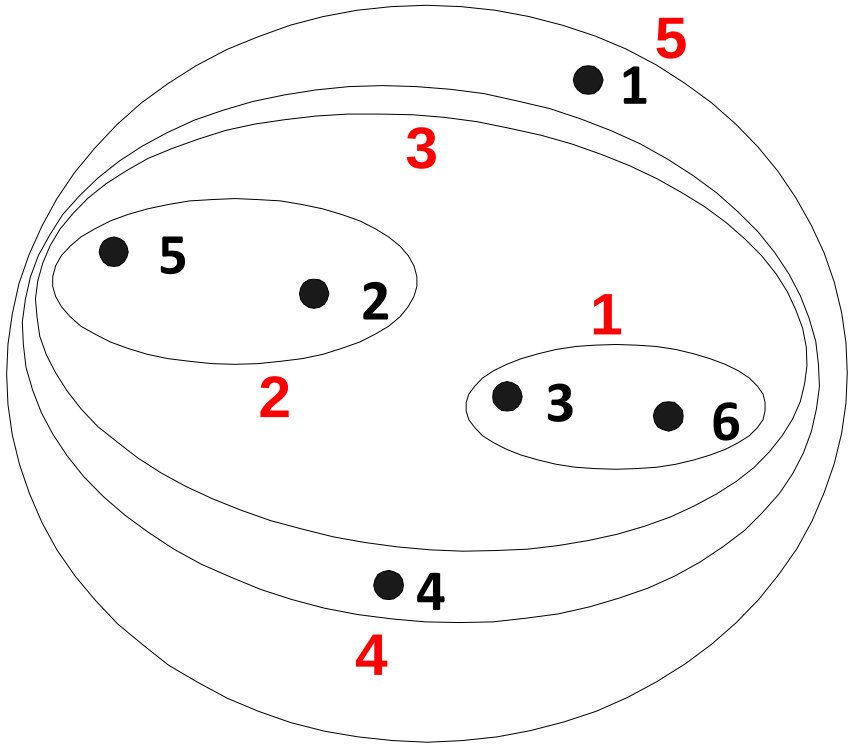
Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

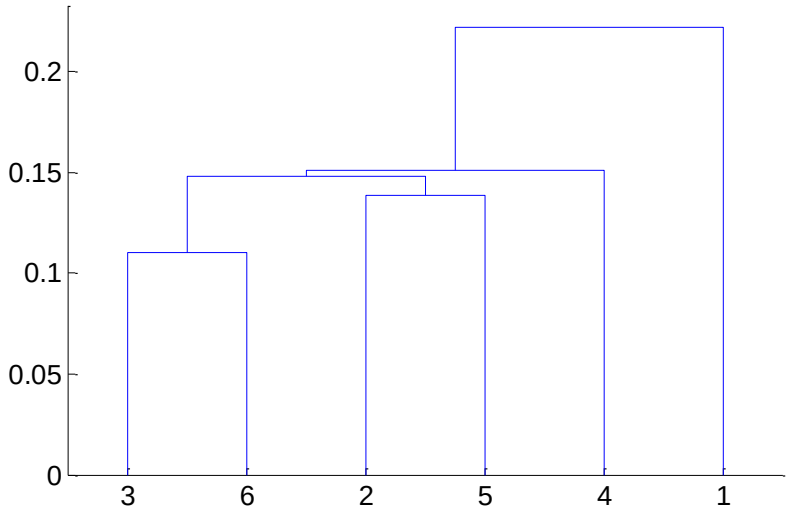
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Single-link clustering: example

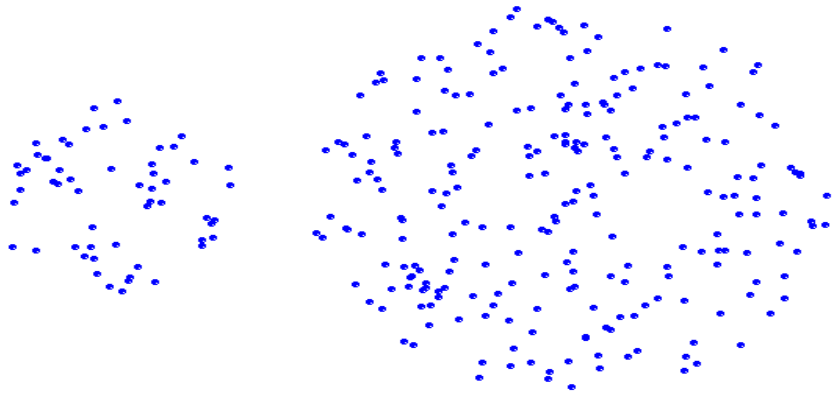


Nested Clusters

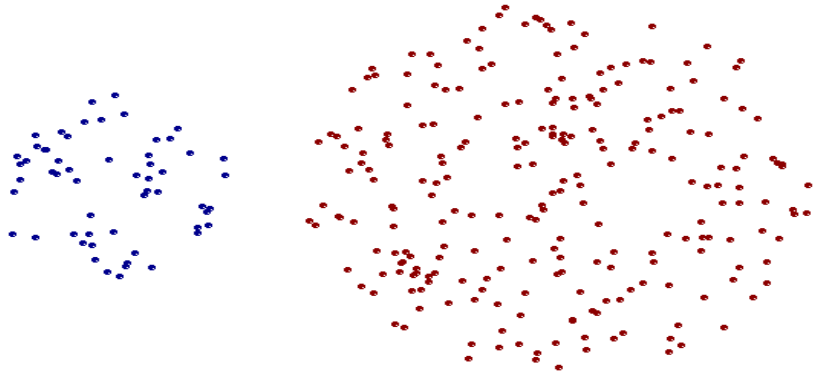


Dendrogram

Strengths of single-link clustering



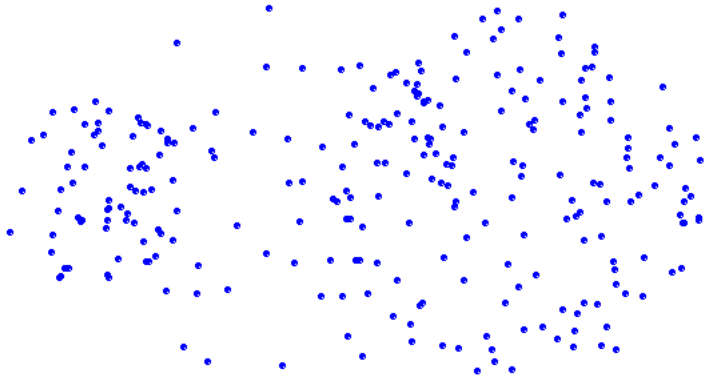
Original Points



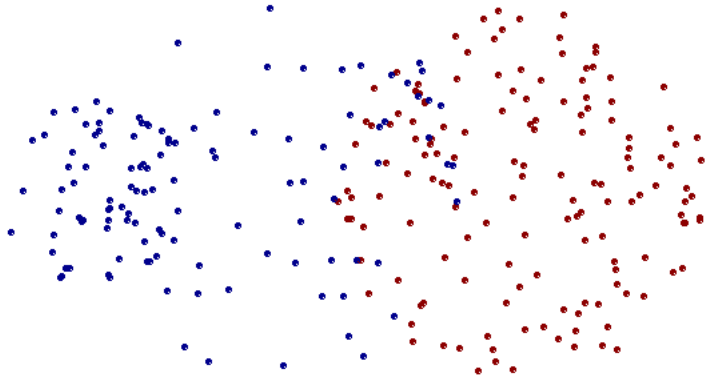
Two Clusters

- **Can handle non-elliptical shapes**

Limitations of single-link clustering



Original Points



Two Clusters

- **Sensitive to noise and outliers**
- **It produces long, elongated clusters**

Distance between two clusters

- **Complete-link distance** between clusters C_i and C_j is the ***maximum distance*** between any object in C_i and any object in C_j

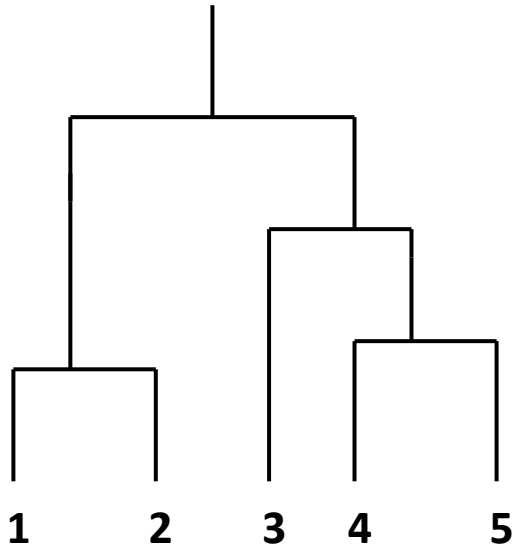
- The distance is **defined by the two most dissimilar objects**

$$D_{cl}(C_i, C_j) = \max_{x,y} \{ d(x, y) \mid x \in C_i, y \in C_j \}$$

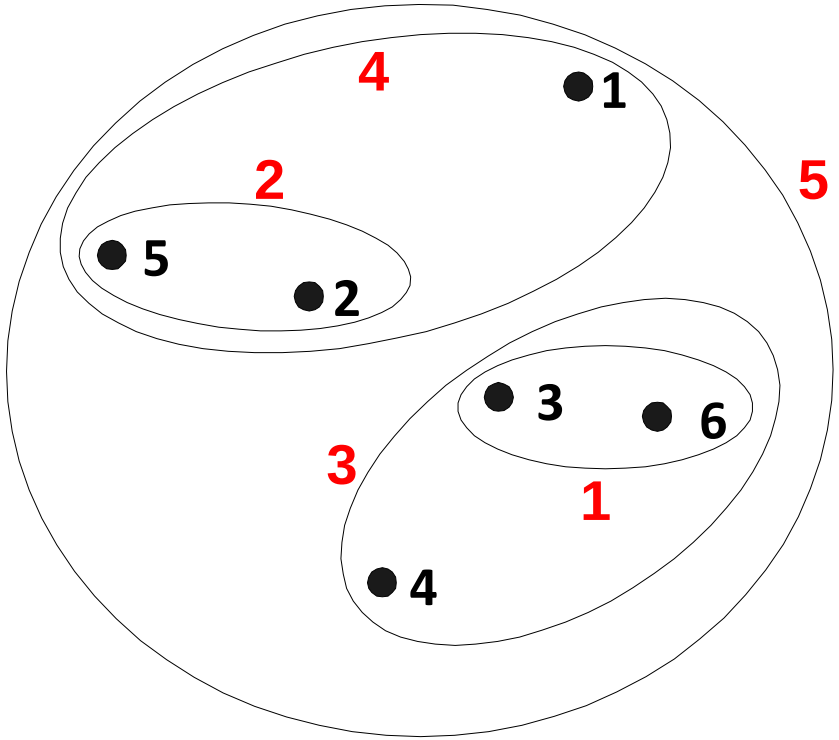
Complete-link clustering: example

- Distance between clusters is determined by the two most distant points in the different clusters

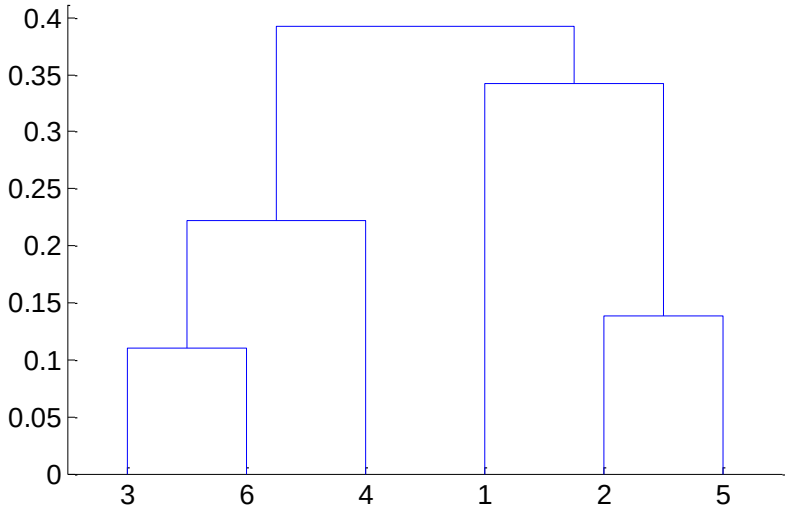
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Complete-link clustering: example

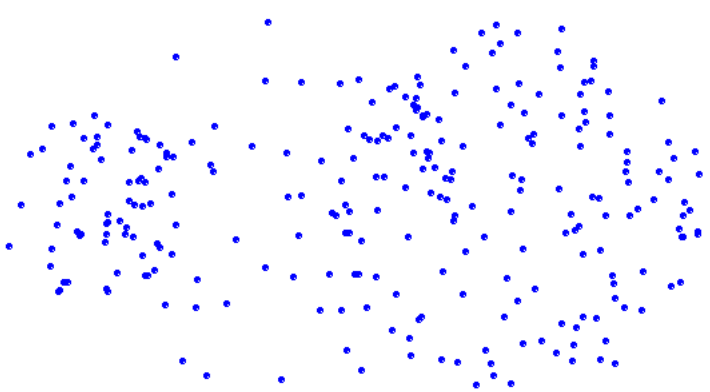


Nested Clusters

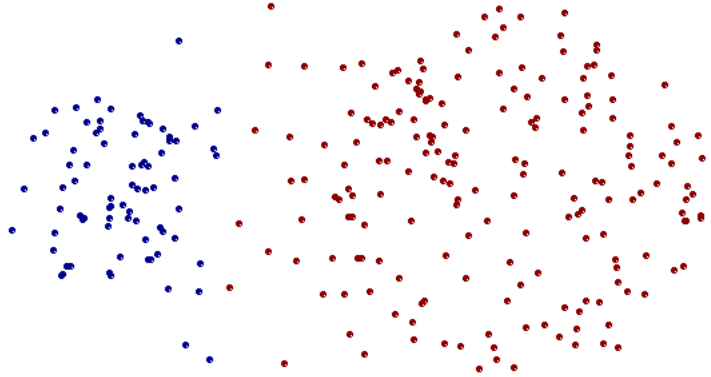


Dendrogram

Strengths of complete-link clustering



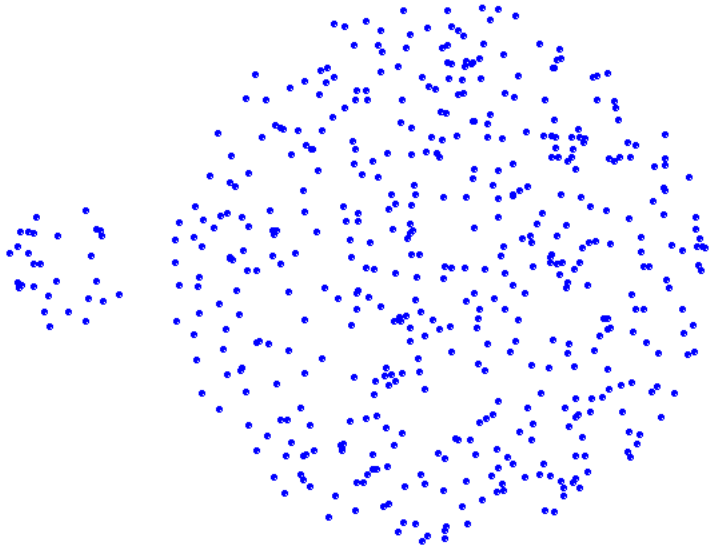
Original Points



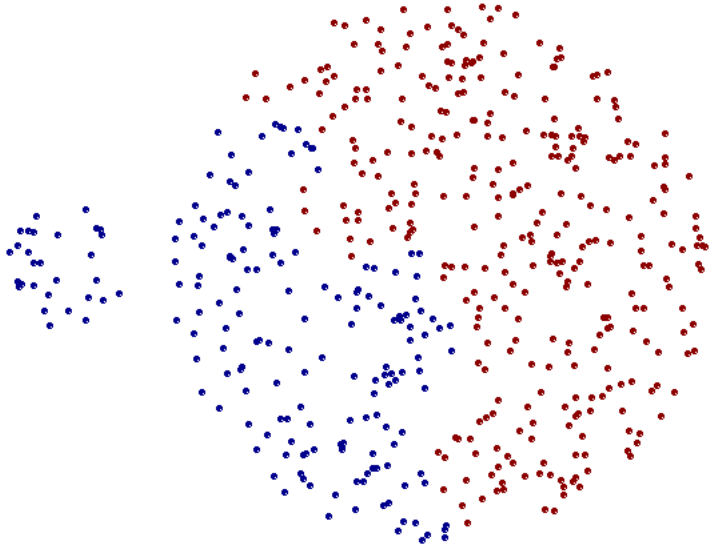
Two Clusters

- **More balanced clusters (with equal diameter)**
- **Less susceptible to noise**

Limitations of complete-link clustering



Original Points



Two Clusters

- **Tends to break large clusters**
- **All clusters tend to have the same diameter – small clusters are merged with larger ones**

Distance between two clusters

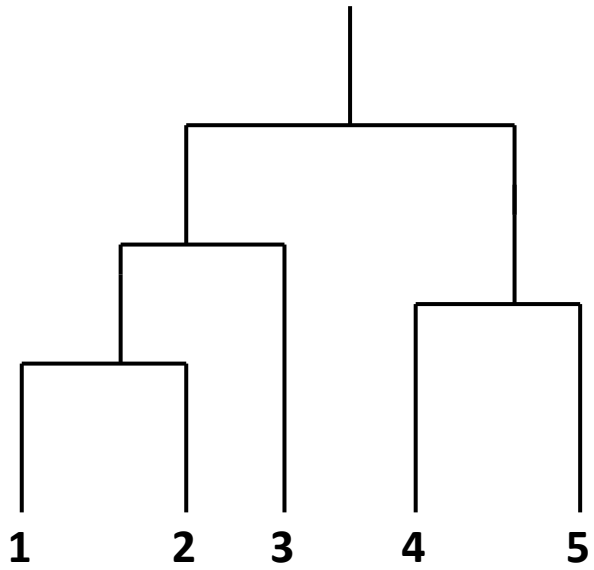
- **Group average distance** between clusters C_i and C_j is the ***average distance*** between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

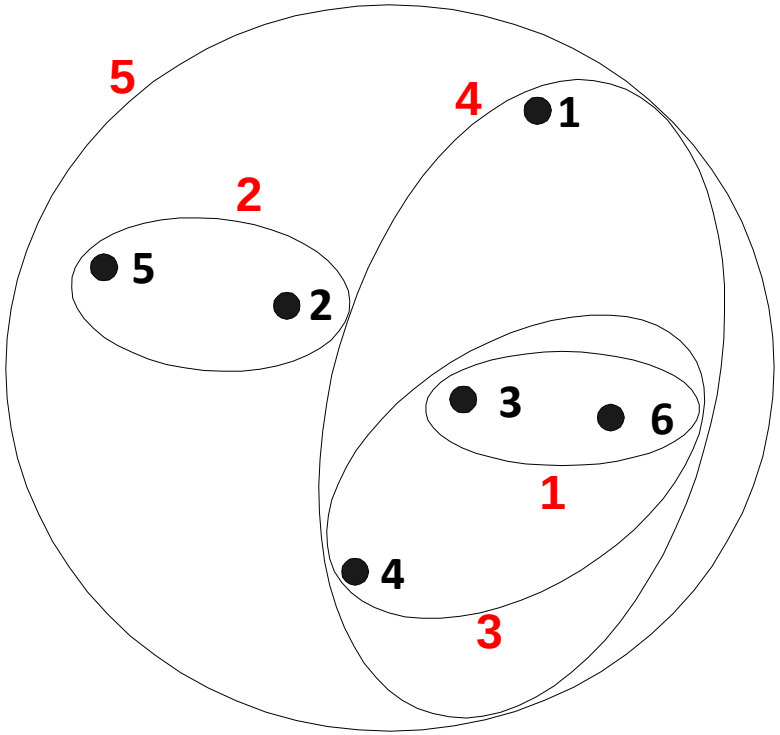
Average-link clustering: example

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

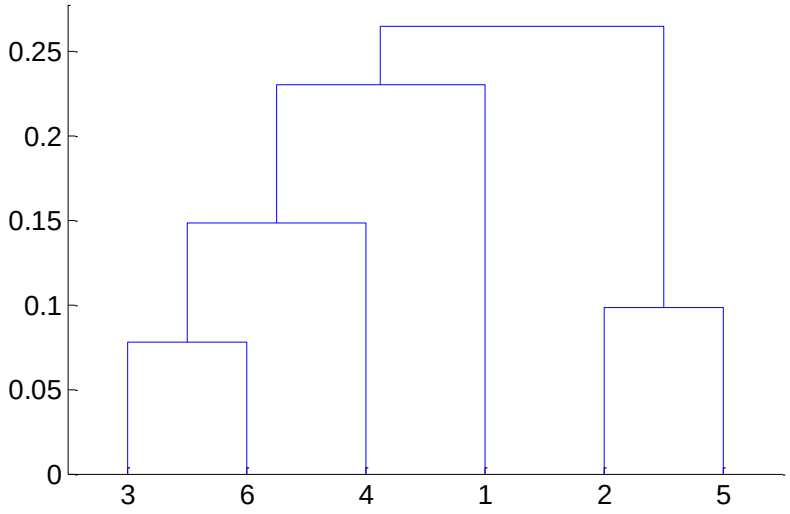
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Average-link clustering: example



Nested Clusters



Dendrogram

Average-link clustering: discussion

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Distance between two clusters

- **Centroid distance** between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

Distance between two clusters

- **Ward's distance** between clusters C_i and C_j is the ***difference*** between the ***total within cluster sum of squares for the two clusters separately***, and the ***within cluster sum of squares resulting from merging the two clusters*** in cluster C_{ij}

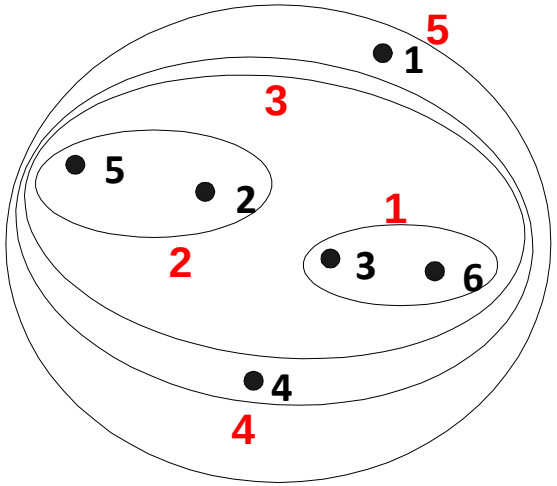
$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

- r_i: centroid of C_i
- r_j: centroid of C_j
- r_{ij}: centroid of C_{ij}

Ward's distance for clusters

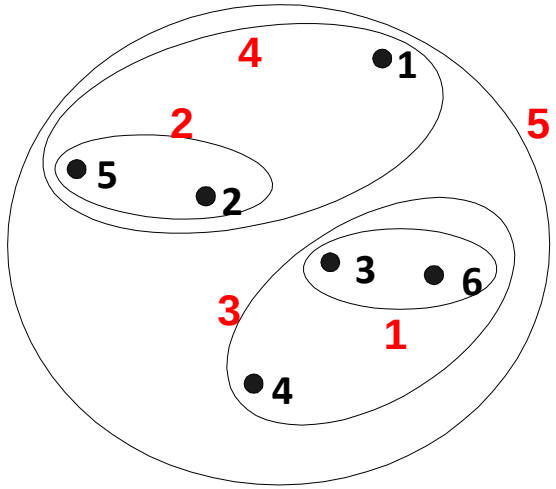
- Similar to group average and centroid distance
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
  - Can be used to initialize k-means

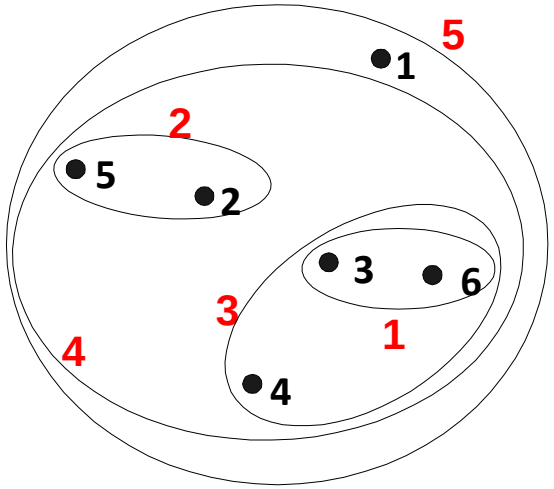
Hierarchical Clustering: Comparison



MIN

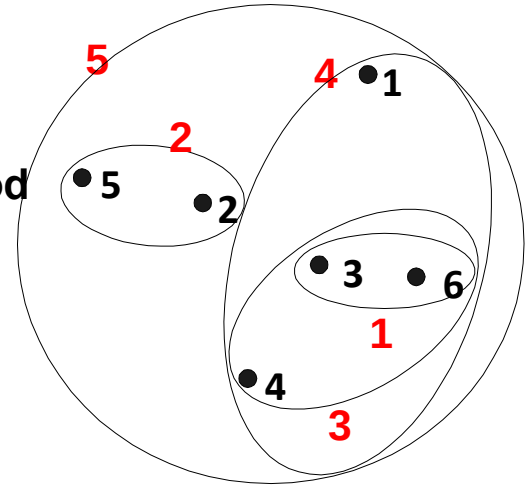
MAX





Group Average

Ward's Method



Hierarchical Clustering: Time and Space requirements

- For a dataset **X** consisting of **n** points
- **$O(n^2)$ space**; it requires storing the distance matrix
- **$O(n^3)$ time** in most of the cases
  - There are **n** steps and at each step the size **n^2** distance matrix must be updated and searched
  - Complexity can be reduced to **$O(n^2 \log(n))$** time for some approaches by using appropriate data structures

Divisive hierarchical clustering

- Start with a single cluster composed of all data points
- Split this into components
- Continue recursively
- *Monothetic* divisive methods split clusters using one variable/dimension at a time
- *Polythetic* divisive methods make splits on the basis of all variables together
- Any intercluster distance measure can be used
- Computationally intensive, less widely used than agglomerative methods

两种主要的聚类方法

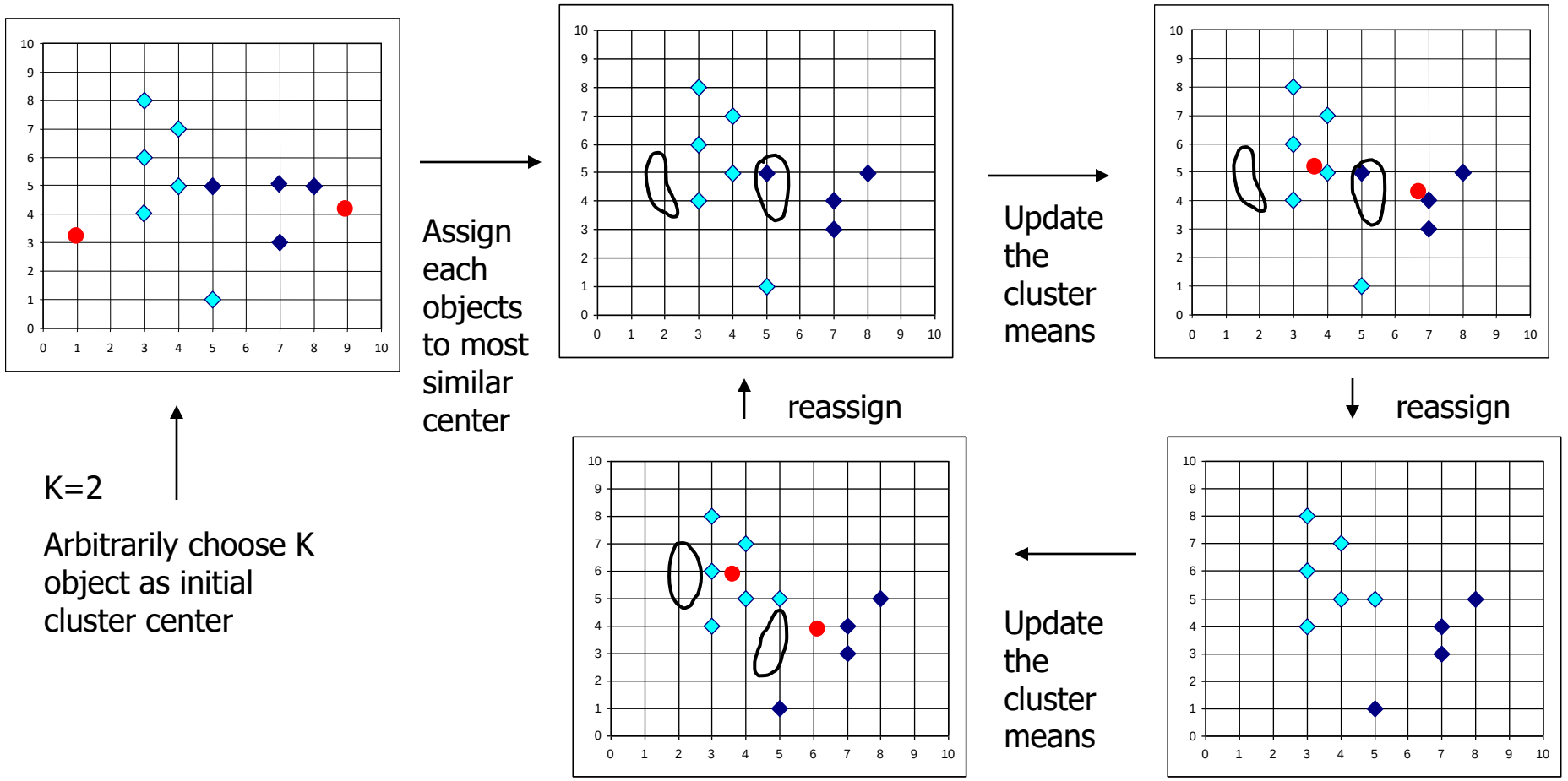
- **层次聚类方法：Hierarchical Clustering** – final clusters are built following a distinct set of sequential steps

- **非层次聚类方法： Non-Hierarchical Clustering** – Clusters are built in such a way that if M clusters are built there is no guarantee that putting together two of the clusters would give rise to the same (M-1) clusters built separately by the method.

K-Means Clustering

- K-Means Clustering is a **non-hierarchical method** in the sense that if one has 2 clusters, say, generated by pre-specifying 2 means (centroids) in the K-means algorithm and 3 clusters generated by pre-specifying 3 means in the K-means algorithm, then it may be the case that no combination of any two clusters of the 3 cluster group can give rise to the 2 cluster grouping.

The *K-Means* Clustering Method



Steps in the K-Means Clustering Approach

- Given a set of observations $(\mathbf{x}_1, \mathbf{x_2}, \cdots . \mathbf{x_n})$ where each observation is a d-dimensional real vector, then K-means clustering aims to partition the n observations into K sets (K < n), $\boldsymbol{S} = \{S_1, S_2, \cdots . S_K\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$arg \min_{\boldsymbol{S}} \sum_{i=1}^{K} \sum_{\boldsymbol{x_j} \in S_i} \left\| \boldsymbol{x_j} - \boldsymbol{\mu_i} \right\|^2 \qquad (1)$$

where $\boldsymbol{\mu_i}$ is the mean of the points in S_i . Now minimizing (1) can, in theory, be done by the **integer programming method** but this can be extremely time-consuming. Instead the **Lloyd algorithm** is more often used.

Lloyd Algorithm

- The steps of the **Lloyd algorithm** are as follows. Given the initial set of K-means $\boldsymbol{m}_{\boldsymbol{1}}^{(\boldsymbol{1})}, \cdots, \boldsymbol{m}_{\boldsymbol{K}}^{(\boldsymbol{1})}$ which can be specified randomly or by some heuristic, the algorithm proceeds by alternating between two steps:
- Assignment Step: Assign each observation to the cluster with the closest mean

$$S_i^{(t)} = \left\{ \boldsymbol{x_j}: \left\| \boldsymbol{x_j} - \boldsymbol{m_i^{(t)}} \right\| \leq \left\| \boldsymbol{x_j} - \boldsymbol{m_{i^*}^{(t)}} \right\| \right\} \text{ for all } i^* = 1,2,\cdots,K. \text{ (2)}$$

- Update Step: Calculate the **new means** to be the centroids of the observations in the clusters, i.e.

$$\boldsymbol{m_i^{(t+1)}} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{\boldsymbol{x_j} \in S_i^{(t)}} \boldsymbol{x_j} \text{ for } i = 1,2,\cdots,K. \qquad (3)$$

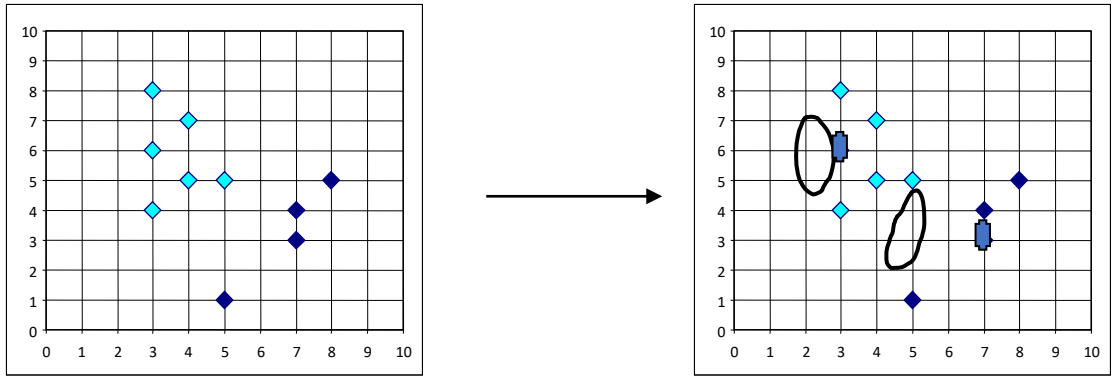
- Repeat the Assignment and Update steps until WCSS (equation (1)) no longer changes. Then the centroids and members of the K clusters are determined.
- **Note:** When using random assignment of the K-means to start the algorithm, one might try several starting point K-means and then choose the "best" starting point to be the random K-means that produces the smallest WCSS among all of the random starting points tried in the K-means procedure.
- Regardless of the clustering technique used, one should strive to choose clusters that are interpretable and make sense given the domain-specific knowledge that we have about the problem at hand.

Variations of the *K-Means* Method

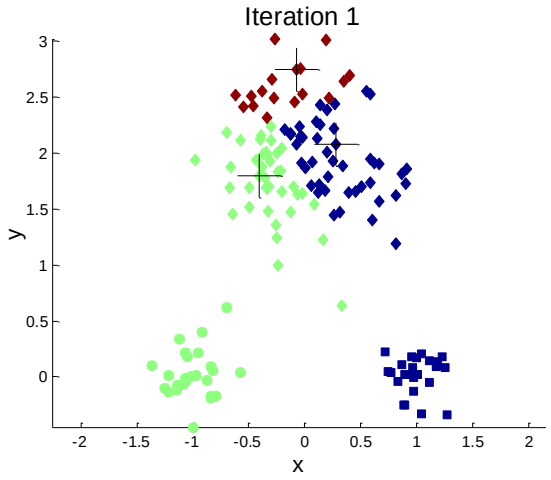
- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

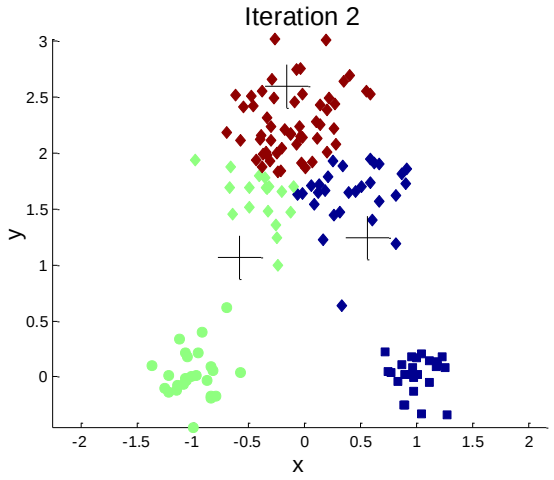
What is the problem of k-Means Method?

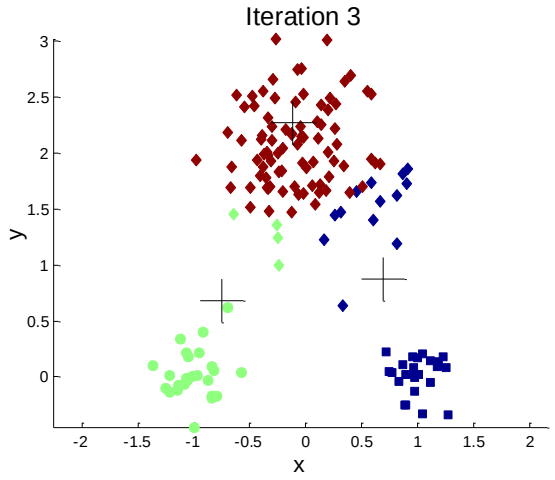
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

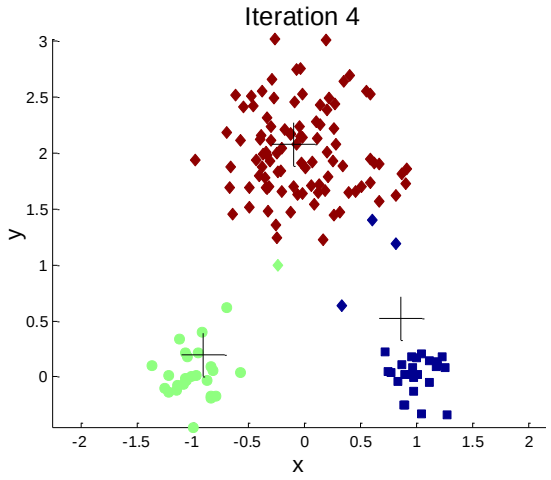


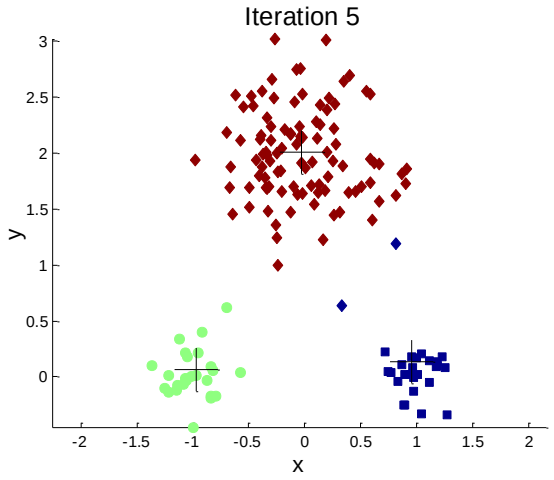
Importance of Choosing Initial Centroids

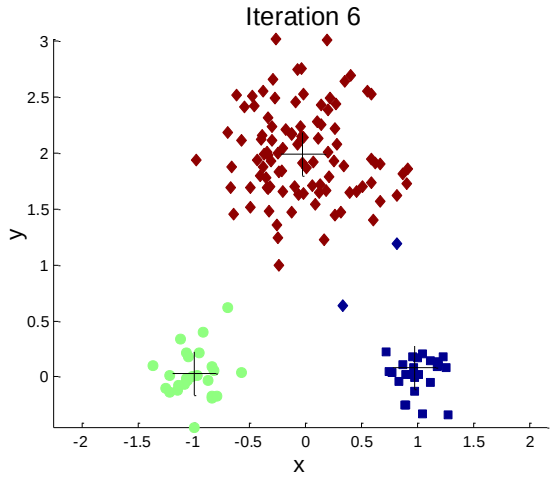








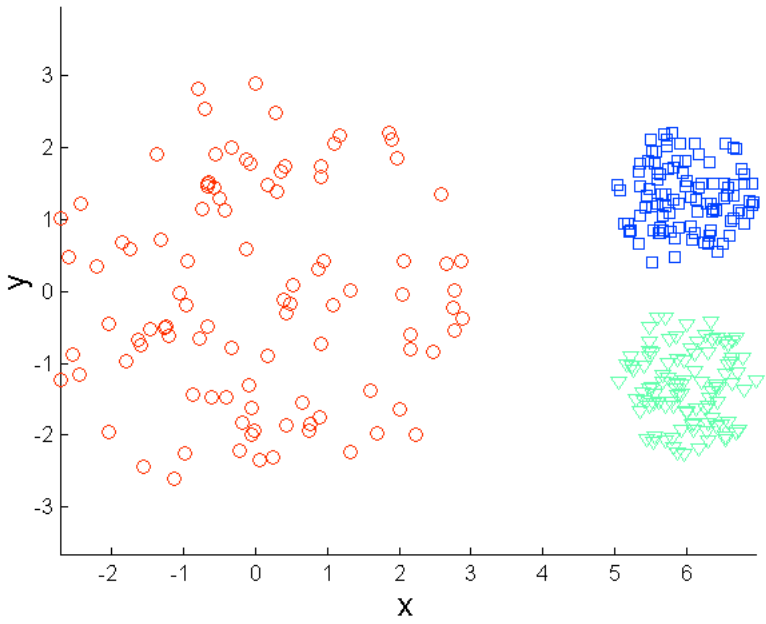




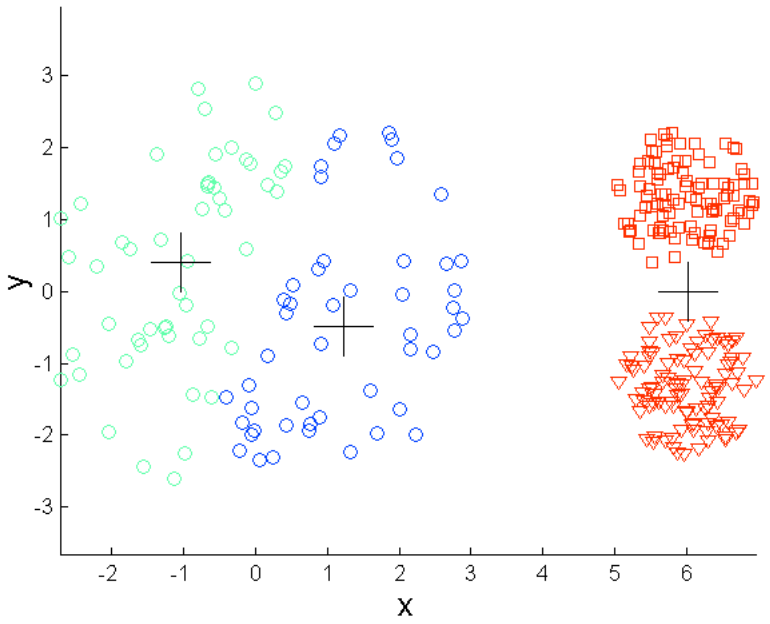
Solutions to Initial Centroids Problem

- 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

Limitations of K-means: Differing Density

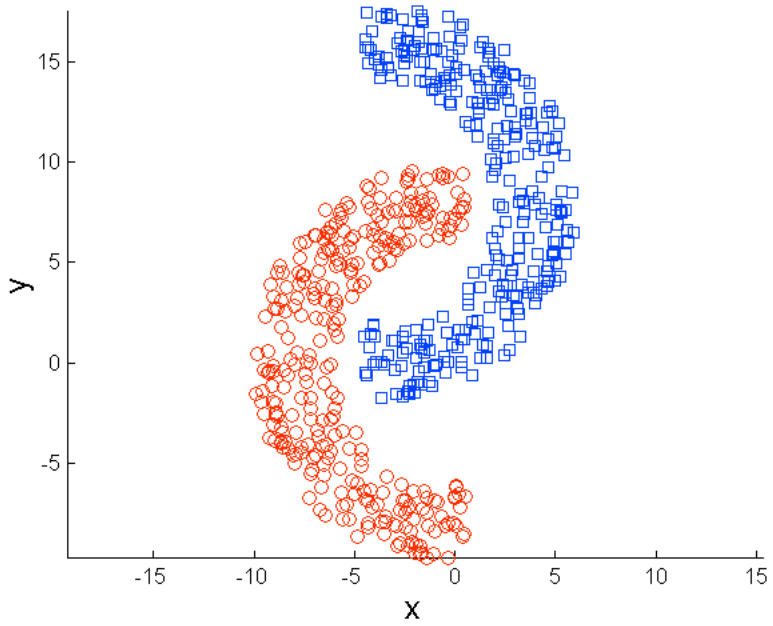


Original Points

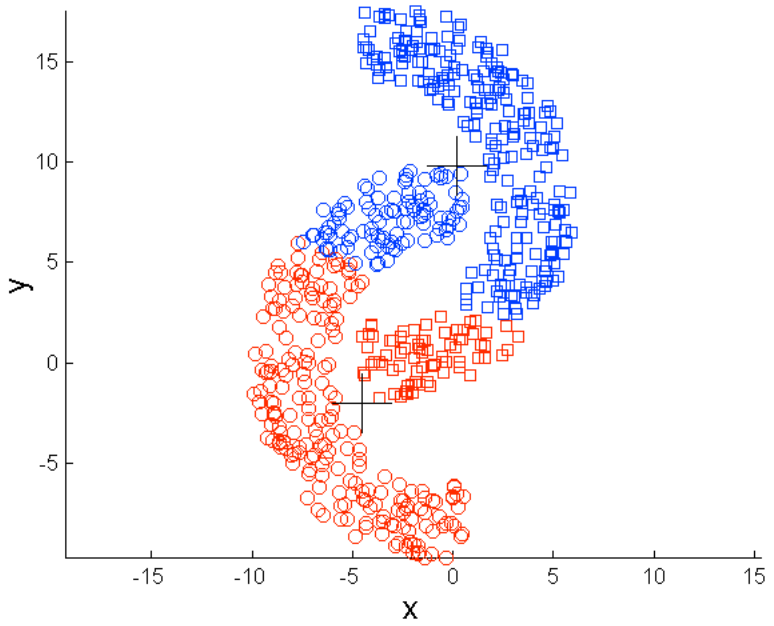


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

Model-based clustering

- Assume data generated from **k** probability distributions
- ***Goal:*** find the distribution parameters
- ***Algorithm:*** Expectation Maximization (EM)
- ***Output:*** Distribution parameters and a **soft** assignment of points to clusters

Model-based clustering

- Assume **k** probability distributions with parameters: $(\theta_1,\ldots, \theta_k)$
- Given data **X**, compute $(\theta_1,\ldots, \theta_k)$ such that $Pr(X|\theta_1,\ldots, \theta_k)$ [likelihood] or $\ln(Pr(X|\theta_1,\ldots, \theta_k))$ [loglikelihood] is maximized.
- Every point $x \epsilon X$ need not be generated by a single distribution but it can be generated by multiple distributions with some probability [soft clustering]

EM Algorithm

- Initialize **k** distribution parameters **(θ_1,…, θ_k)**; Each distribution parameter corresponds to a cluster center

- Iterate between two steps

  - **E**xpectation step: (probabilistically) assign points to clusters

  - **M**aximation step: estimate model parameters that maximize the likelihood for the given assignment of points

EM Algorithm

- Initialize **k** cluster centers
- Iterate between two steps
  - **E**xpectation step: assign points to clusters

$$\Pr(x_i \in C_k) = \Pr(x_i \mid C_k) \Big/ \sum_j \Pr(x_i \mid C_j)$$

$$w_k = \frac{\sum_i \Pr(x_i \in C_k)}{n}$$

  - **M**aximation step: estimate model parameters

$$r_k = \frac{1}{n} \sum_{i=1}^{n} \frac{\Pr(x_i \in C_k)}{\sum_k \Pr(x_i \in C_j)}$$

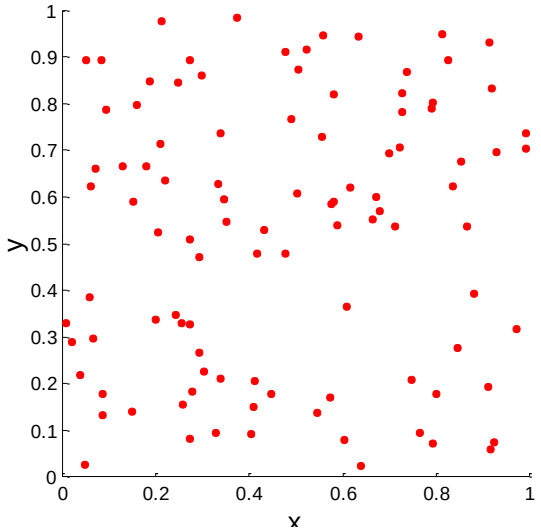
Cluster Evaluation

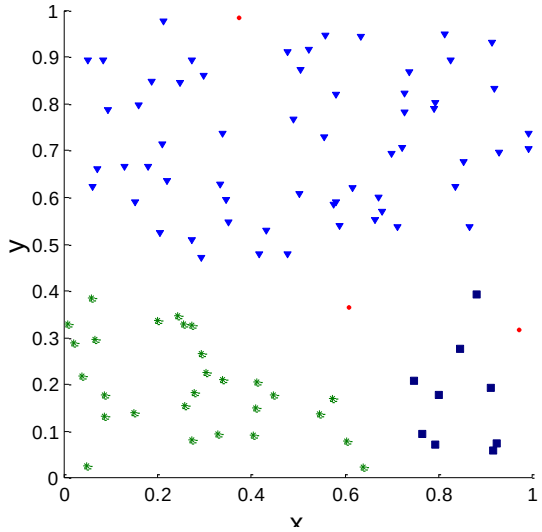
Cluster Validity

- 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”!
- 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

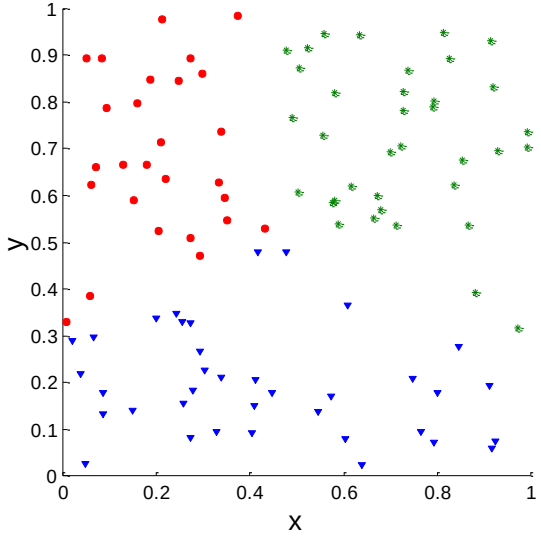
Random Points

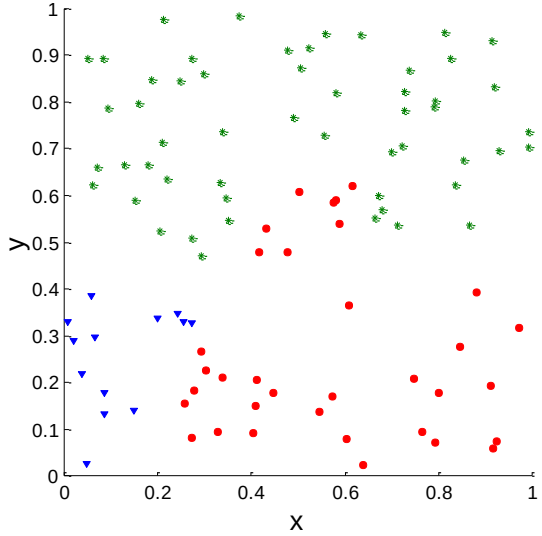




DBSCAN

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.

Measures of Cluster Validity

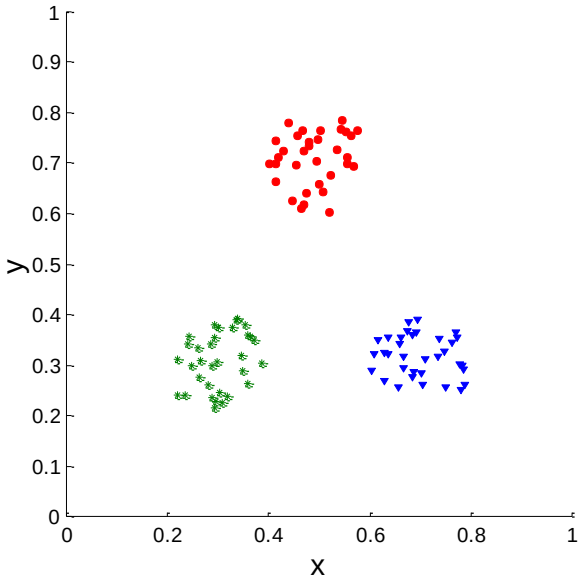
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness 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.

Measuring Cluster Validity Via Correlation

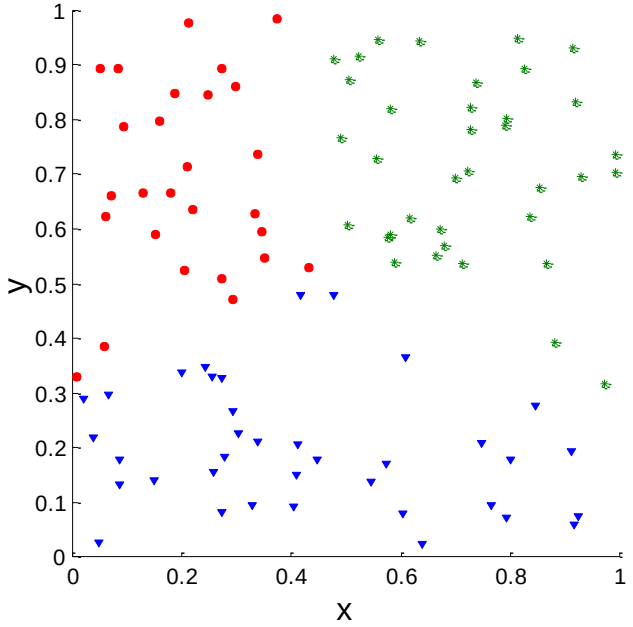
- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between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



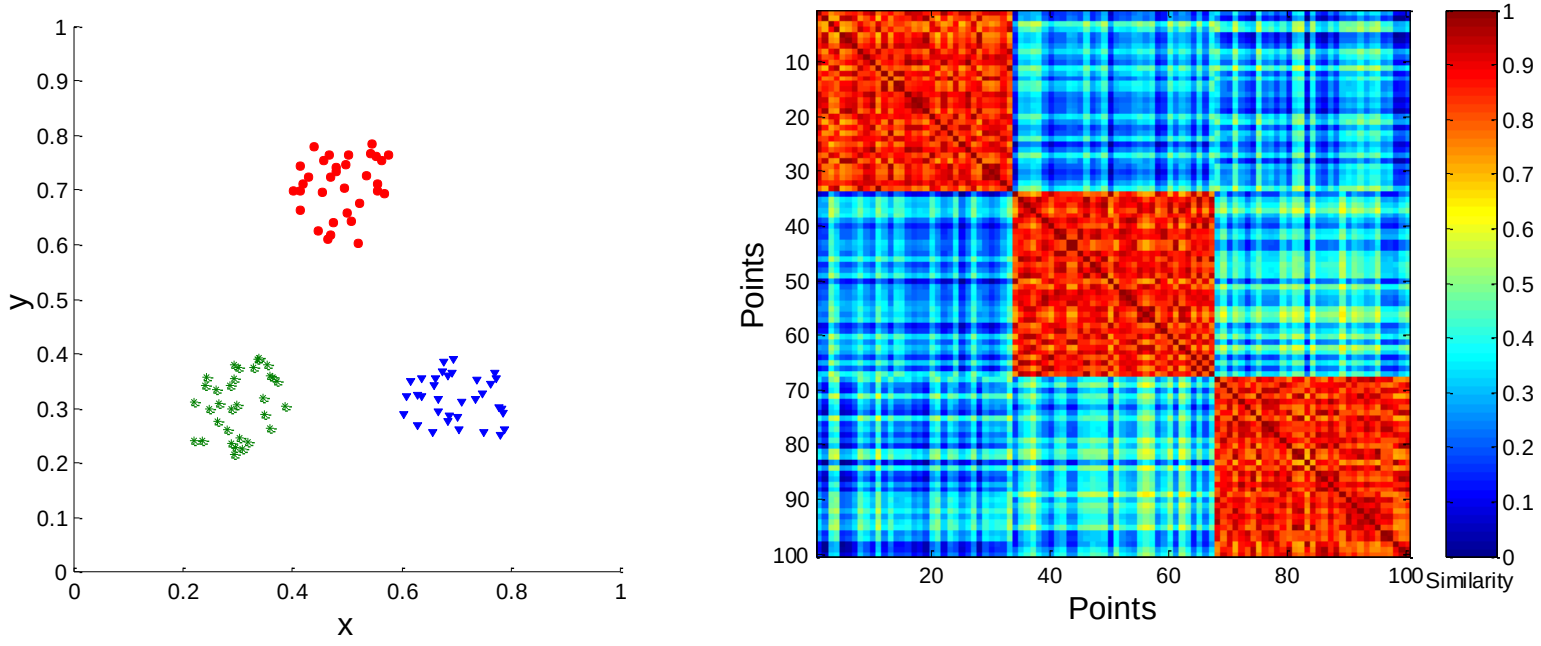
Corr = -0.9235



Corr = -0.5810

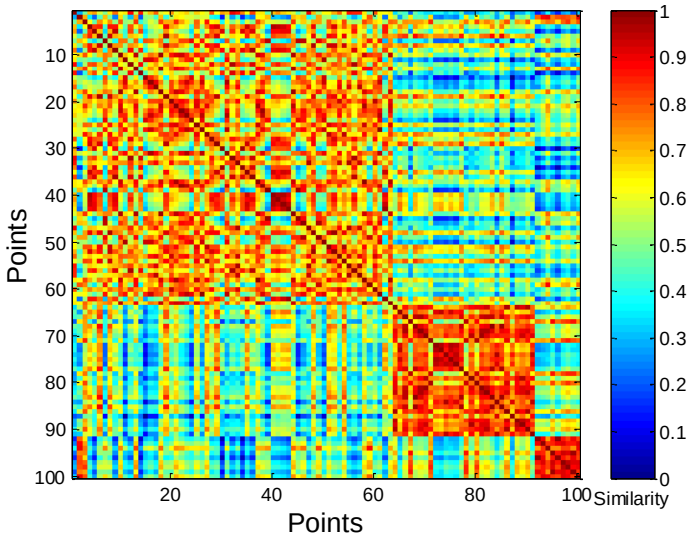
Using Similarity Matrix for Cluster Validation

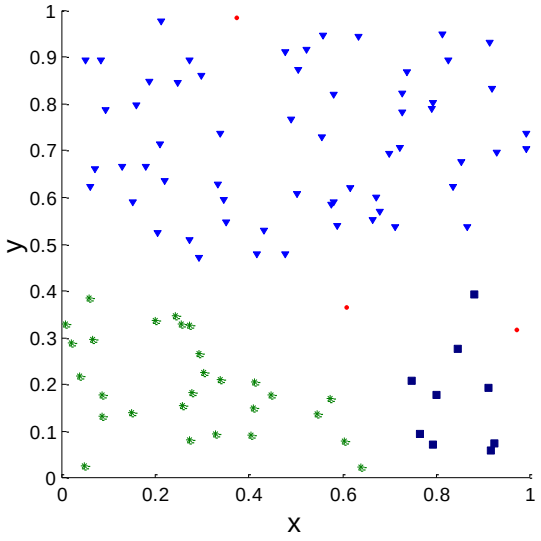
- Order the similarity matrix with respect to cluster labels and inspect visually



Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

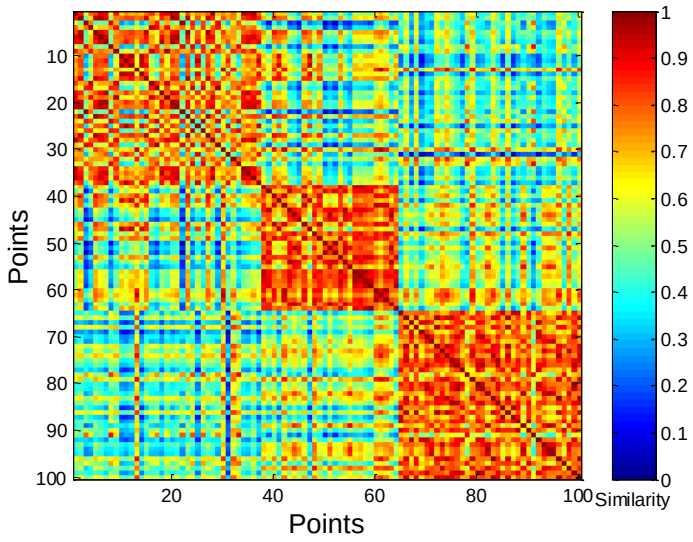


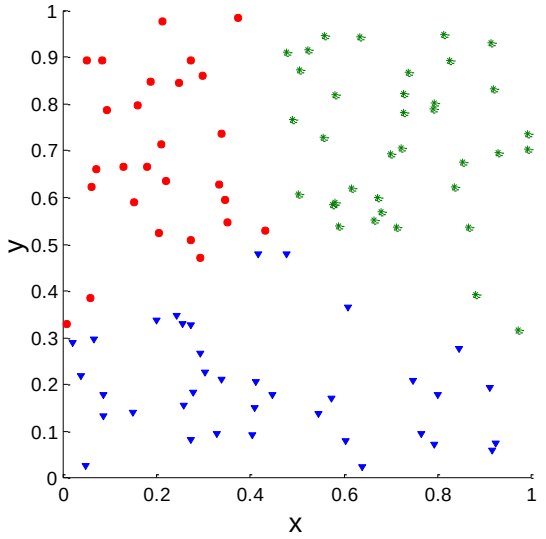


DBSCAN

Using Similarity Matrix for Cluster Validation

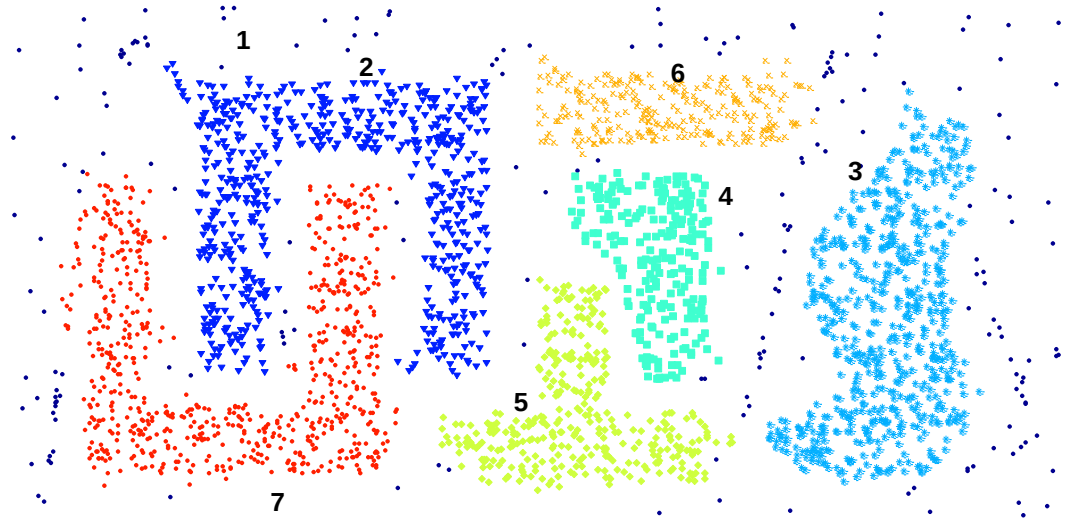
- Clusters in random data are not so crisp

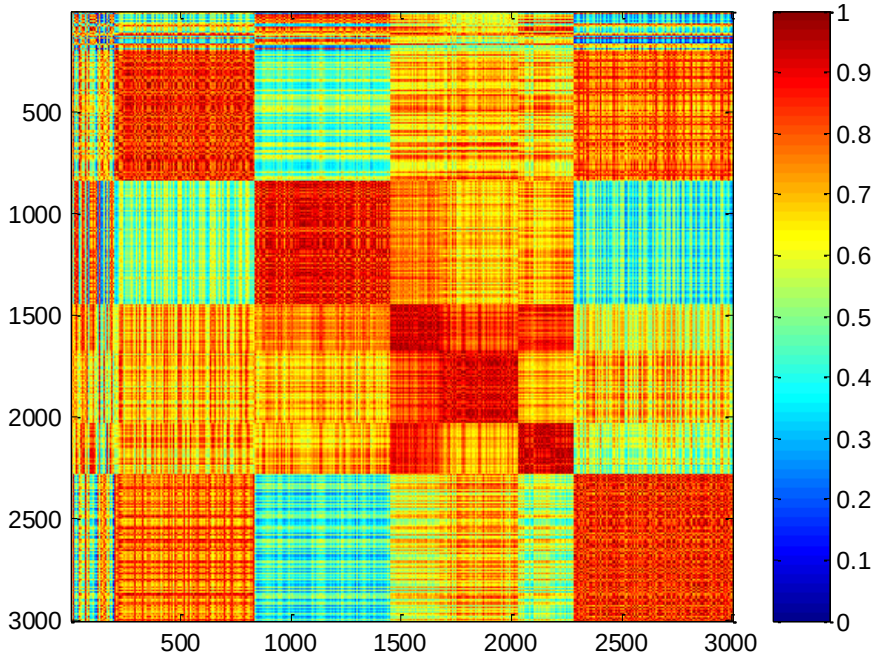




K-means

Using Similarity Matrix for Cluster Validation

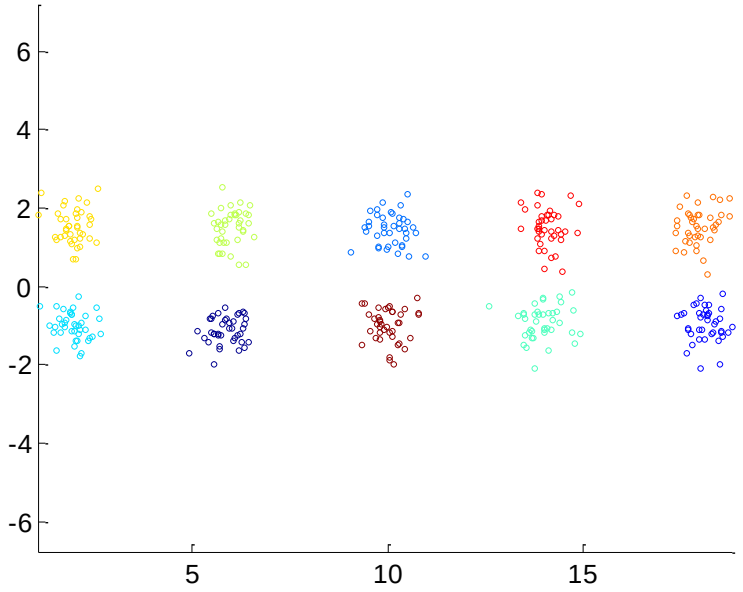


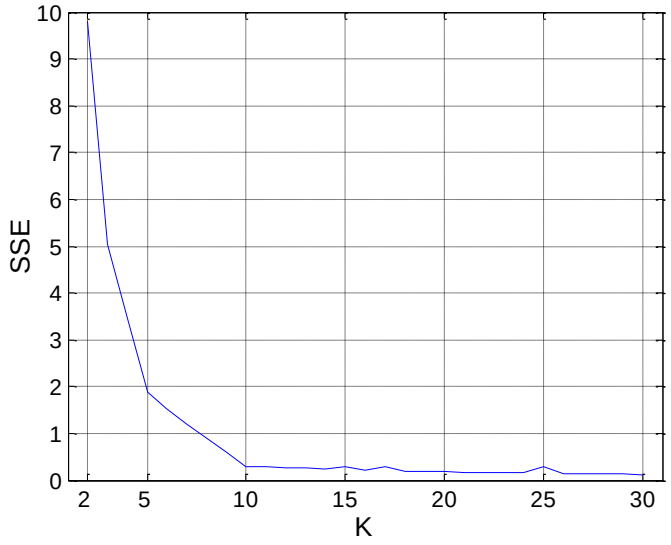


DBSCAN

Internal Measures: SSE

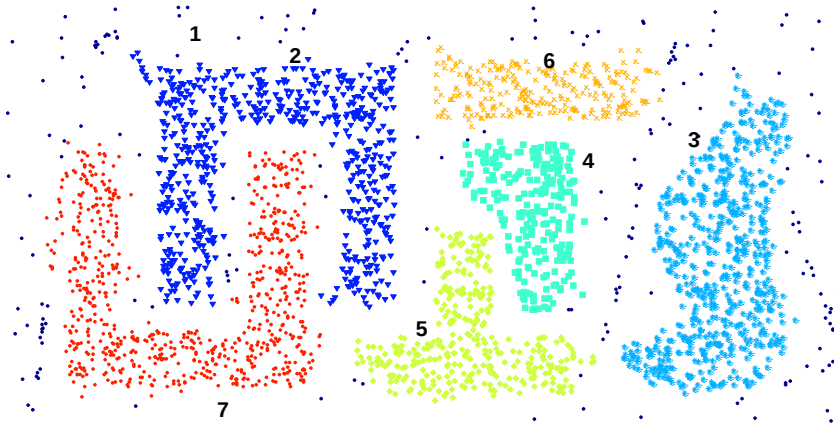
- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of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

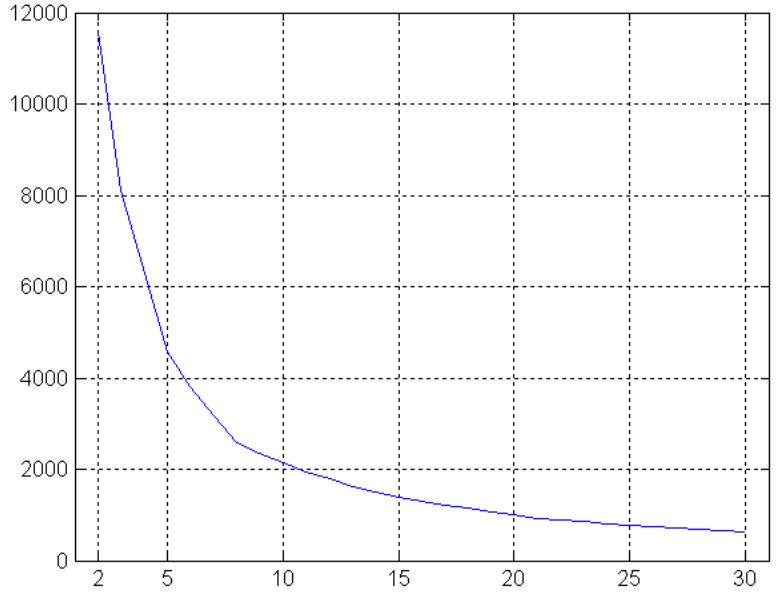




Internal Measures: SSE

- SSE curve for a more complicated data set





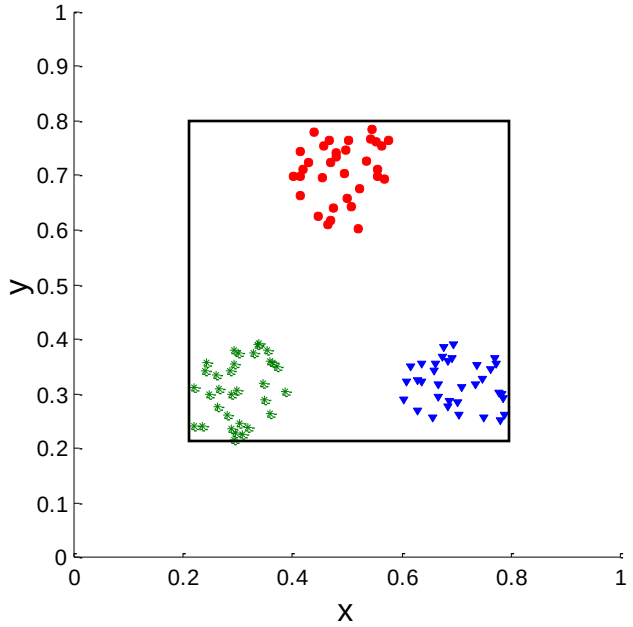
SSE of clusters found using K-means

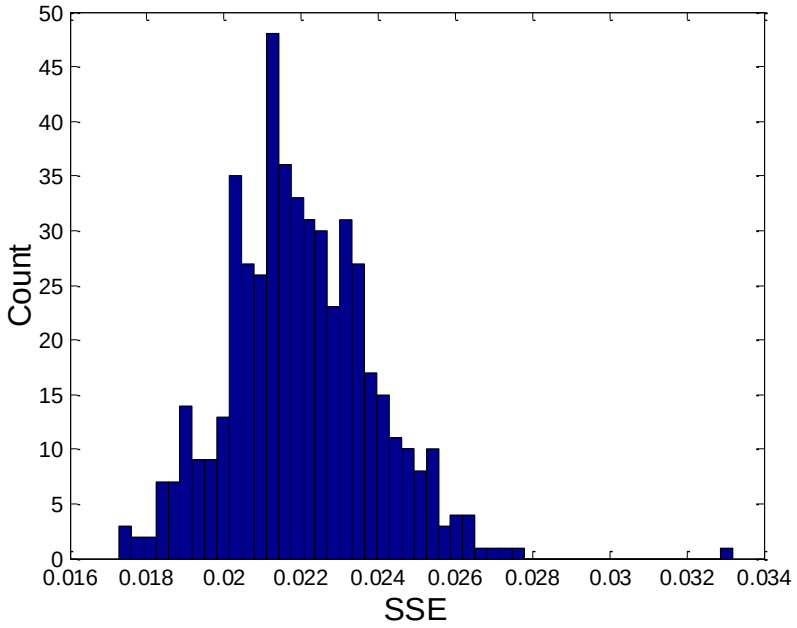
Framework for Cluster Validity

- 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

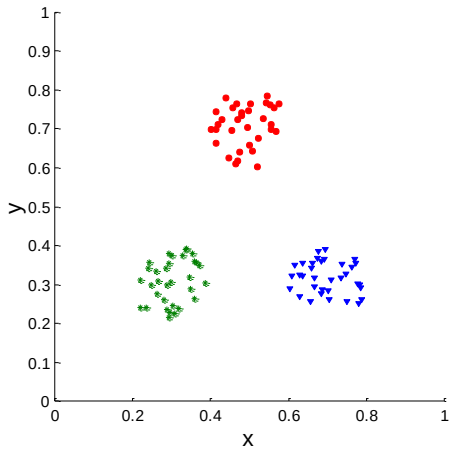
- Example
  - Compare SSE of 0.005 against three clusters in random data
  - Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

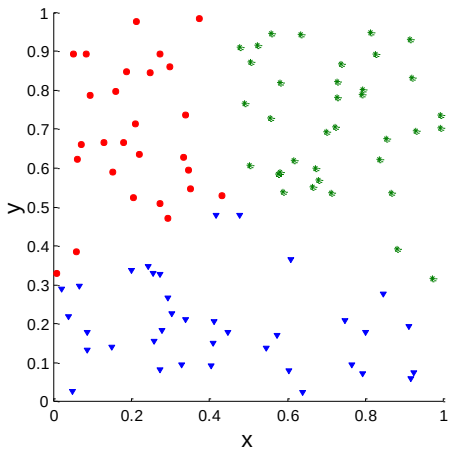


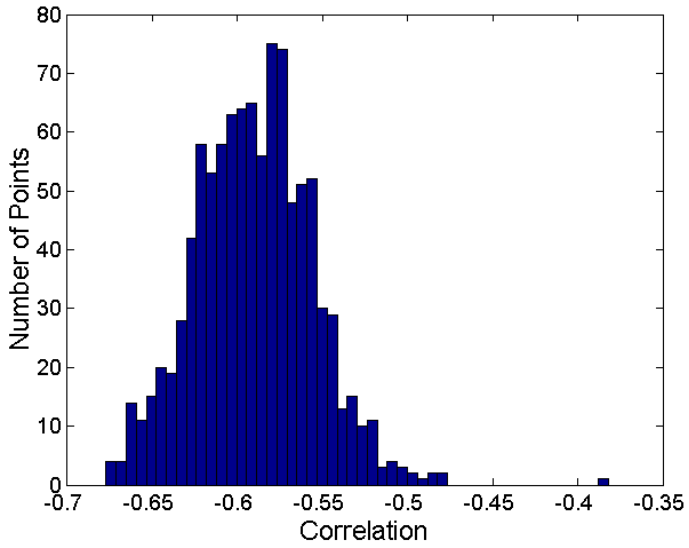


Statistical Framework for Correlation

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







Corr = -0.9235

Corr = -0.5810

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)

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

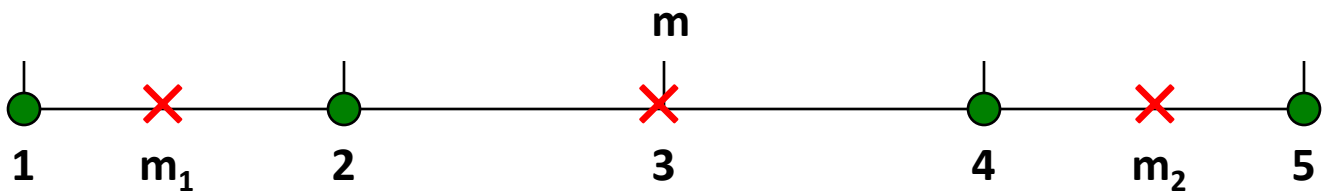
  - Separation is measured by the between cluster sum of squares

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

    - 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:

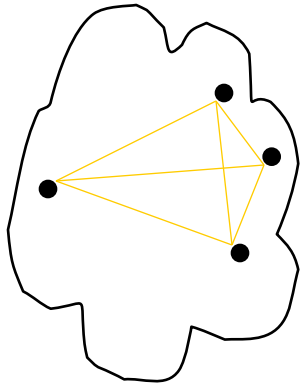
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

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

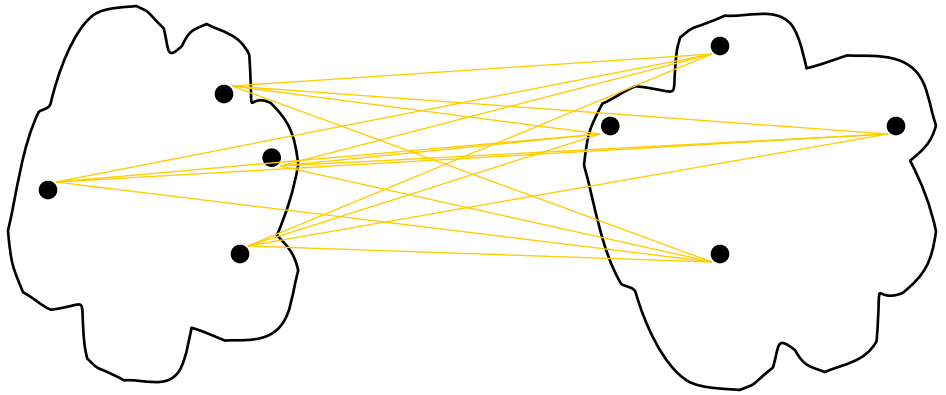
$$Total = 1 + 9 = 10$$

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



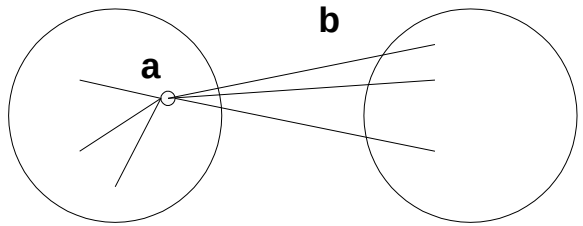
separation

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

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity

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

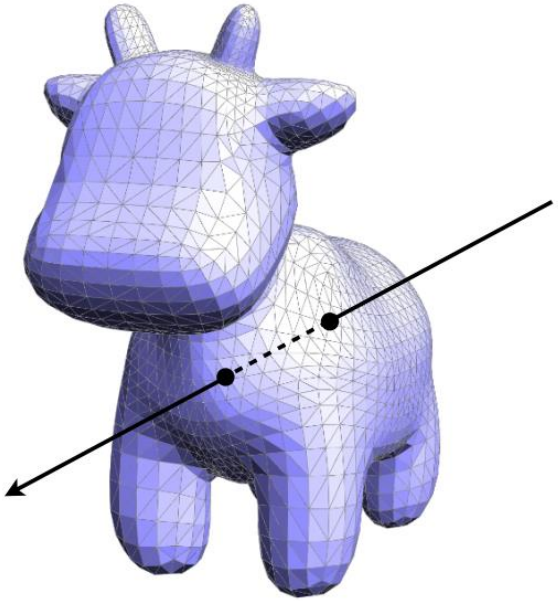
Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

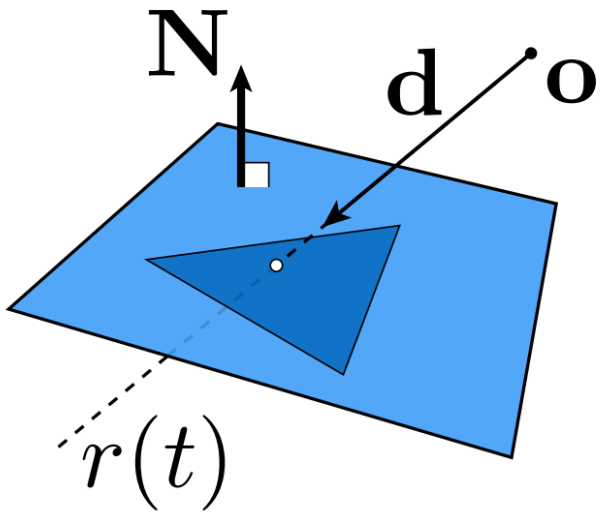
Algorithms for Clustering Data, Jain and Dubes

应用：搜索加速结构

大量搜索计算

- 点与三角网格的求交
- 点与三角形的求交

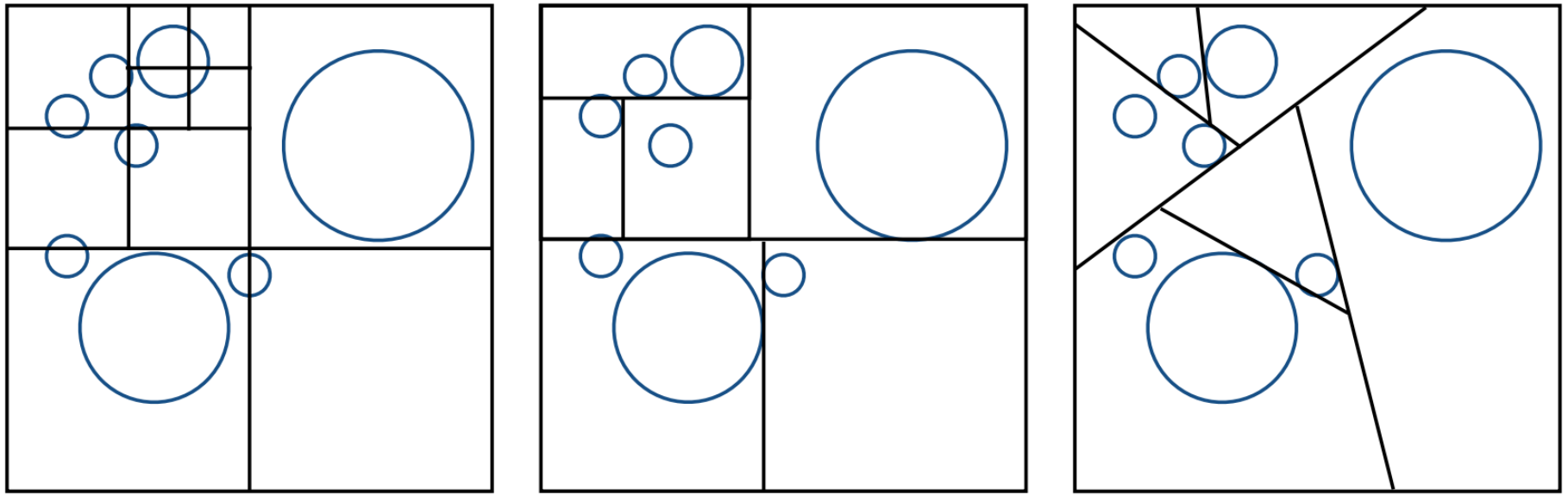




加速策略： Bounding Volumes

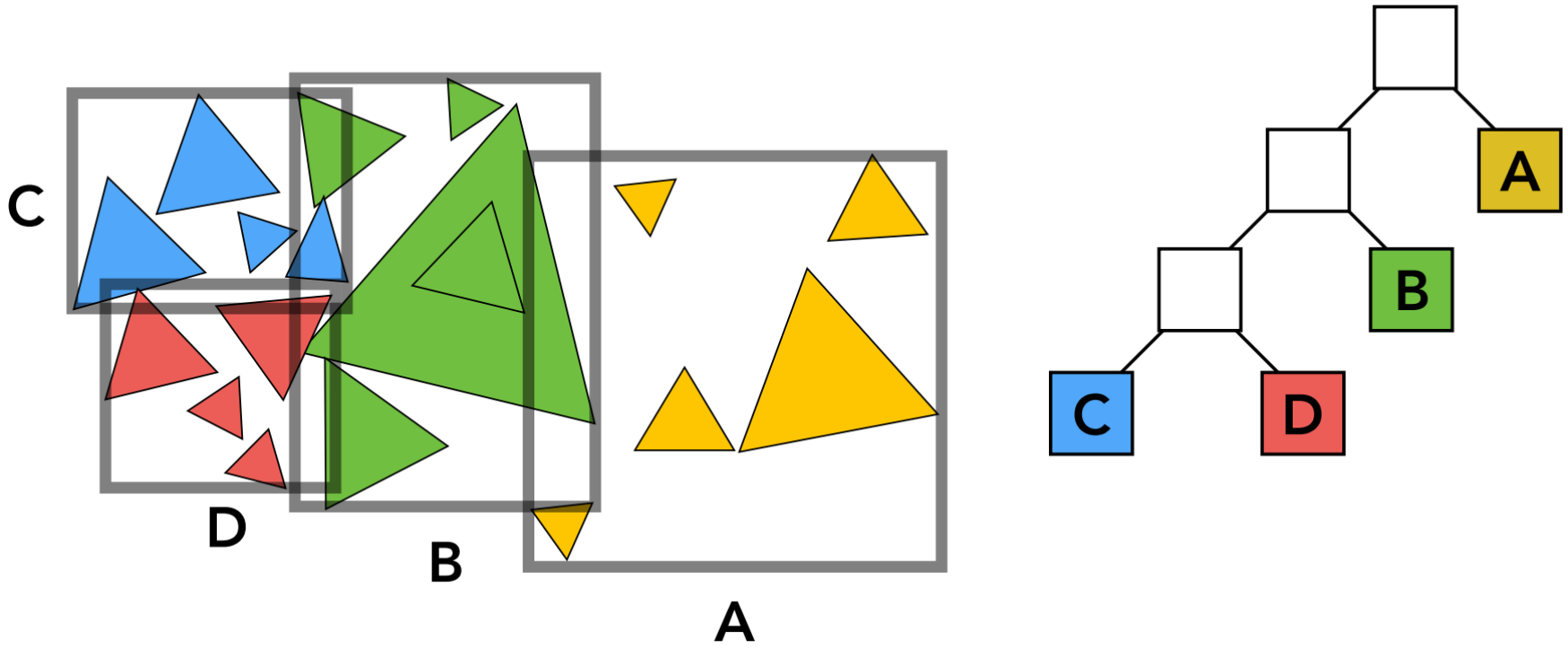
- Axis-Aligned Bounding Box (AABB) (轴对齐包围盒)
- 包围球

加速策略：空间组织



Oct-Tree　　**KD-Tree**　　BSP-Tree

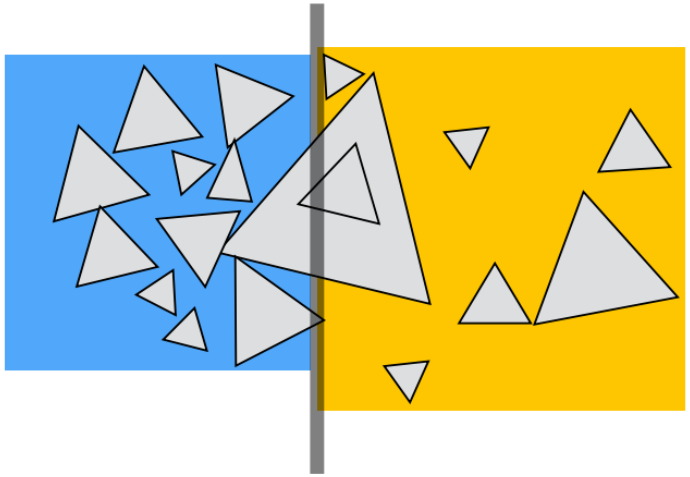
加速结构： Bounding Volume Hierarchy (BVH)



Spatial vs Object Partitions

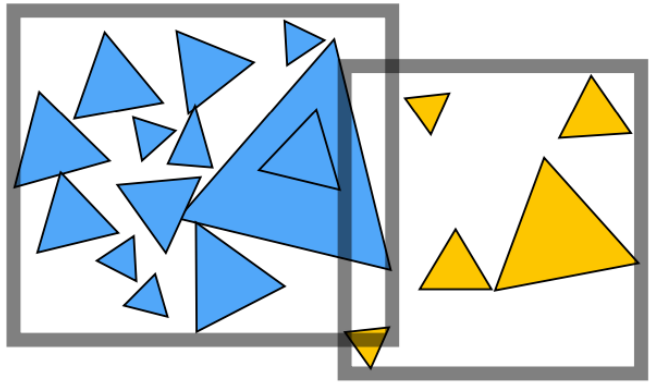
Spatial partition (e.g.KD-tree)

- Partition space into non-overlapping regions
- An object can be contained in multiple regions



Object partition (e.g. BVH)

- Partition set of objects into disjoint subsets
- Bounding boxes for each set may overlap in space

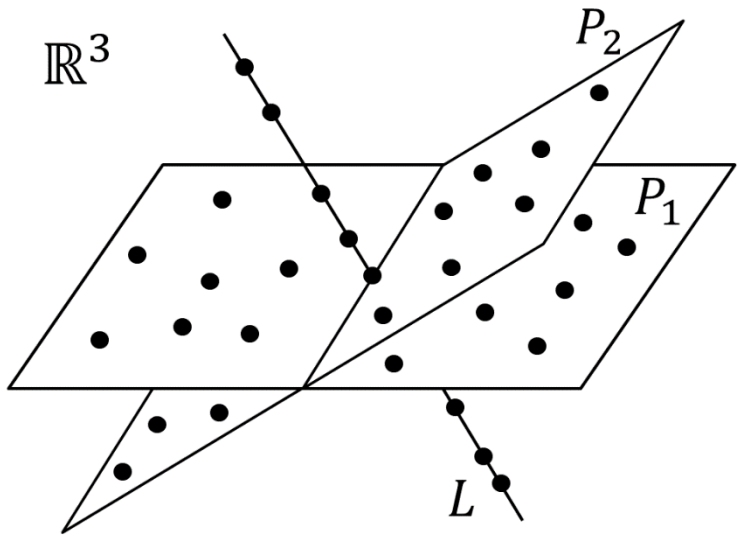


Clustering-II

复杂高维数据

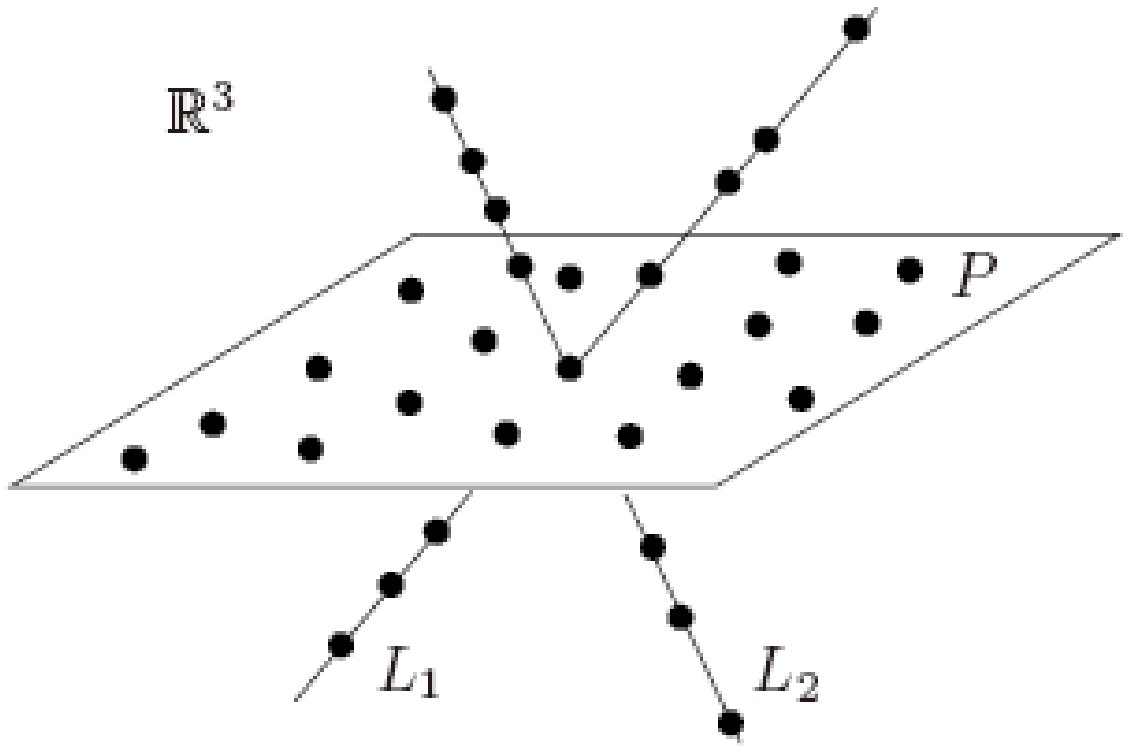
子空间聚类（Subspace clustering）

- Input:
  - high dimensional datasets having low intrinsic dimensions
  - $\{x_j\}_{j=1,\dots,N}, x_t \in \mathbb{R}^D$

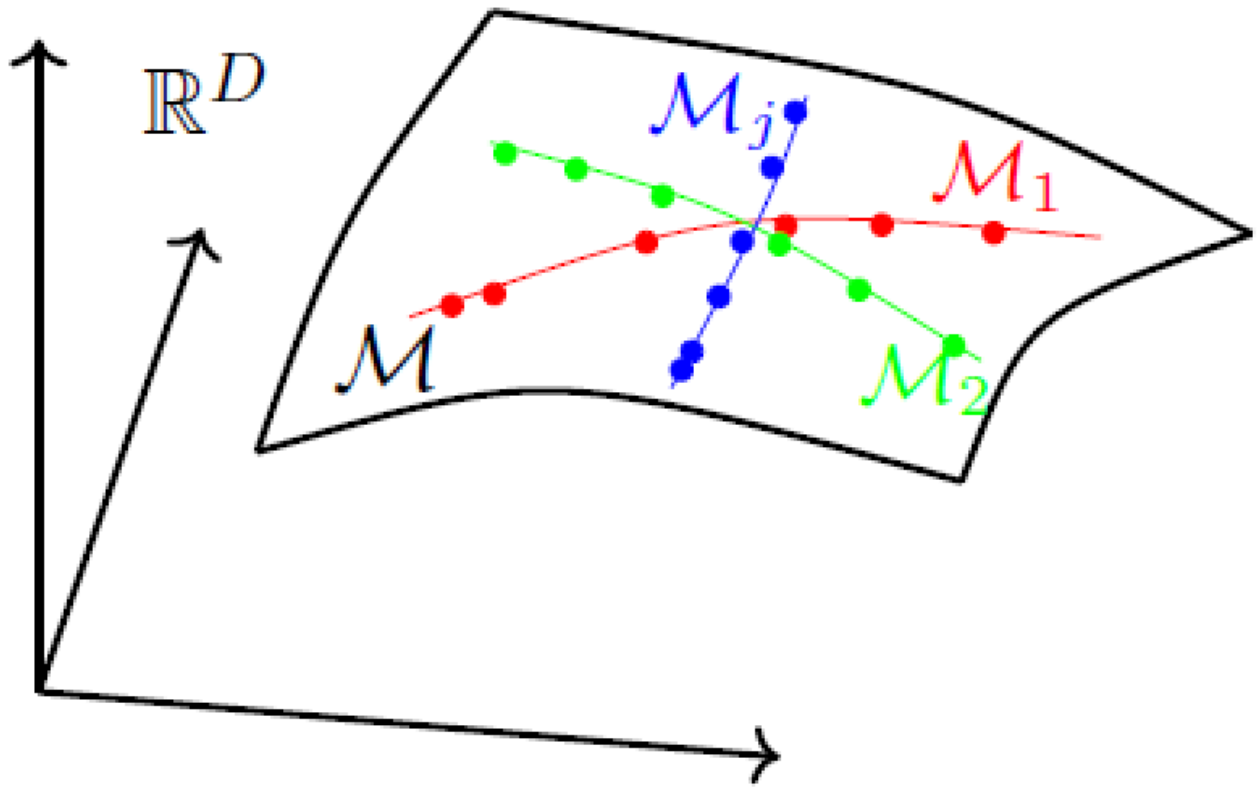


- Output:
  - multiple low-dimensional linear subspaces
  - L, P_1, P_2

Sub-manifolds



High-Dim Data with Mixed (multiple) Low-Dim Structures



Sparse subspace clustering(SSC)

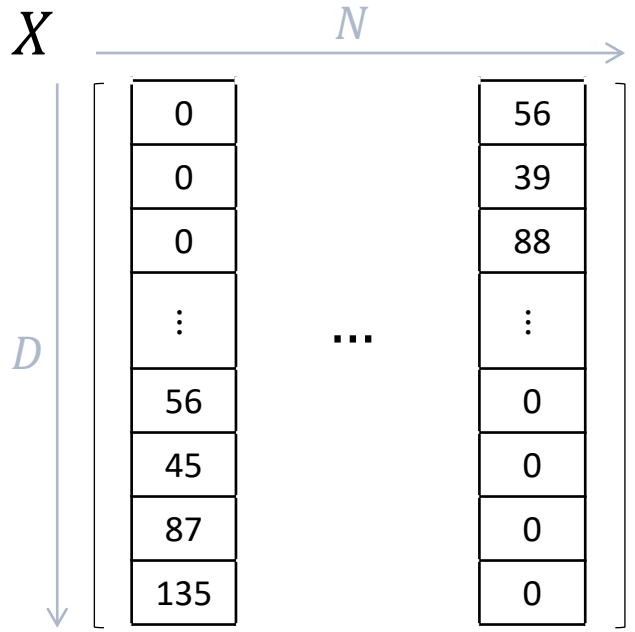
[Elhamifar and Vidal 2009]

- Based on the observation:
  - each point can always be represented as a linear combination of the points belonging to the same subspace

$$x_j = \sum_{i=1}^{N} w_{ij} x_i \, , j = 1, \dots, N$$

$$X = XW, \text{where}$$

$$X = (x_1, \dots, x_N) \in \mathbb{R}^{D \times N}$$



https://www.cis.jhu.edu/~ehsan/Downloads/SSC-CVPR09-Ehsan.pdf

Sparse subspace clustering(SSC)

[Elhamifar and Vidal 2009]

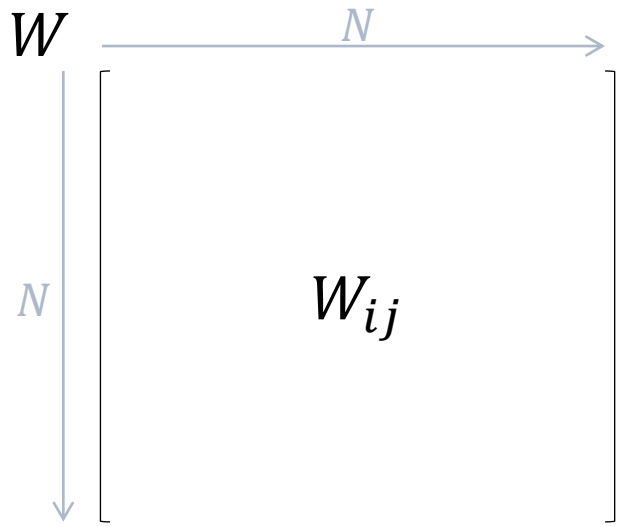
- Based on the observation:
  - each point can always be represented as a linear combination of the points belonging to the same subspace

$$x_j = \sum_{i=1}^{N} w_{ij} x_i \, , j = 1, \dots, N$$

$$X = XW, \text{where}$$

$$X = (x_1, \dots, x_N) \in \mathbb{R}^{D \times N}$$

$$W = (w_{ij}) \in \mathbb{R}^{N \times N}$$



Sparse subspace clustering(SSC)

[Elhamifar and Vidal 2009]

- Based on the observation:
  - each point can always be represented as a linear combination of the points belonging to the same subspace

$$\min_{W} \ \|W\|_{1,1} \quad \text{s.t.} \ \ X = XW, \ \operatorname{diag}(W) = 0$$

$$\Rightarrow \quad \min_{W} \ \|XW - X\|_F^{\,2} + \lambda\|W\|_{1,1} \quad \text{s.t.} \ \ \operatorname{diag}(W) = 0$$

where $\|W\|_{1,1} = \sum\|w_j\|_1 = \sum|w_{ij}|$

SSQP

[Wang et al. 2011]

$$\min_{W} \ \|XW - X\|_F^{\ 2} + \lambda\|W\|_{1,1} \quad \text{s.t.} \ \ \mathrm{diag}(W) = 0$$

$$\Rightarrow \quad \min_{W} \ \|XW - X\|_F^{\ 2} + \lambda\|W^T W\|_{1,1} \quad \text{s.t.} \ \ W \geq 0, \ \ \mathrm{diag}(W) = 0$$

- $W \geq 0$: provides better interpretations
- $\|W^T W\|_{1,1}$: more efficient than SSC
- Block diagonal property:

Desired property for affinity matrix!

$$\bar{W}^* = \Gamma^{-1}\bar{W}\Gamma = \begin{pmatrix} \bar{W}^*_1 & & & 0 \\ & \bar{W}^*_2 & & \\ & & \ddots & \\ 0 & & & \bar{W}^*_K \end{pmatrix}_{N \times N}$$

where Γ is a permutation matrix, submatrix $\bar{W}^*_k \in \mathbb{R}^{N_k \times N_k}$

Wang et al. Efficient Subspace Segmentation via Quadratic Programming. AAAI 2011.

Supervised learning (DL)

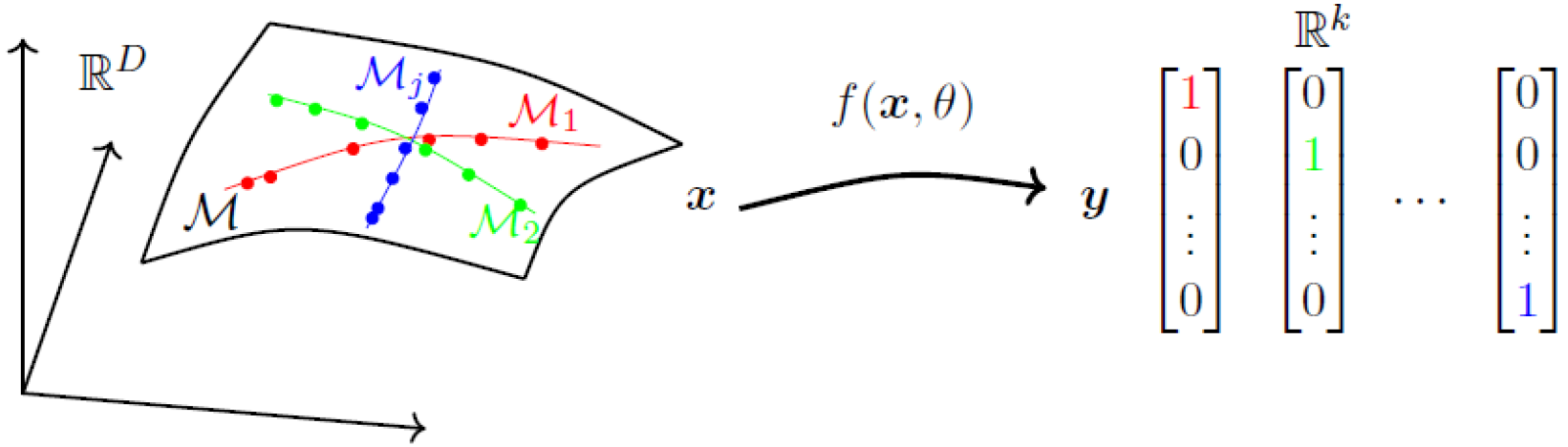


Figure: **Black Box Classification:** $\boldsymbol{y}$ is the class label of $\boldsymbol{x}$ represented as a "one-hot" vector in $\mathbb{R}^k$. To learn a nonlinear mapping $f(\cdot, \theta) : \boldsymbol{x} \mapsto \boldsymbol{y}$, say modeled by a deep network.

In a supervised setting, using cross-entropy (CE) loss:

$$\min_{\theta \in \Theta} \mathrm{CE}(\theta, \boldsymbol{x}, \boldsymbol{y}) \doteq -\mathbb{E}[\langle \boldsymbol{y}, \log[f(\boldsymbol{x}, \theta)] \rangle] \approx -\frac{1}{m} \sum_{i=1}^{m} \langle \boldsymbol{y}_i, \log[f(\boldsymbol{x}_i, \theta)] \rangle$$

See more…

- 马毅（加利福尼亚大学伯克利分校）：
  - 基于第一原理的深度（卷积）神经网络
- https://mp.weixin.qq.com/s/JopCQqoFq5CDl6nikrQ7yw

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis, such as constraint-based clustering

有监督的分类？

（一）背景知识

两种蠓虫 Af 和 Apf 已由生物学家罗纳（w.L.Grogna）和维尔恩（W.W.Wirth）于 1981 年根据它们的触角长（mm）和翅长（mm）加以区分，6 只 Apf 和 9 只 Af 蠓虫的触长,翅长数据如下：

Apf：(1.14,1.78)，(1.18,1.96)，(1.20,1.86)，(1.26,2.00)，(1.28,2.00)，(1.30,1.96)；

Af：(1.24,1.72)，(1.36,1.74)，(1.38,1.64)，(1.38,1.82)，(1.38,1.90)，(1.40,1.70)，(1.48,1.82),(1.54,1.82)，(1.56,2.08)。

在生物学中，根据触角长和翅长来识别一只蠓虫标本是 Af 还是 Apf 是很重要的。

（二）要解决的问题

1、根据给定的数据，制定一种方法，正确区分两类蠓虫；

2、用我们的方法对触长、翅长分别为(1.24,1.80)、(1.28,1.84)、(1. 40,2.04)的三个样本进行识别；

3、假设 Af 是宝贵的传粉益虫，Apf 是某种疾病的载体，在这种情况下我们是否应该修改所用的分类方法，且如何修改。

中国科学技术大学

University of Science and Technology of China

谢　谢！