NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

MSAI-6124
Neuro Evolution & Fuzzy Intelligence

Week 5 – Part 1
Clustering

Dr WANG Di
wangdi@ntu.edu.sg

Evaluating the Efficiency of Algorithms

- Algorithm complexity
  - Differentiates algorithms based on the scale of running time or memory space
  - Expresses it as a function to show how fast it grows along with the input(s), normally using the Big-O notation
- Time complexity estimates the running time of an algorithm
- Space complexity estimates the amount of memory space that an algorithm requires

Formal Definition of Big-O

- $f(n)$ is $O(g(n))$, if there exists a real constant $c > 0$ and an integer constant $n_0 \geq 1$, such that

  $$f(n) \leq c \cdot g(n), \forall n \geq n_0$$

  Note: $g(n)$ should be in the simplest form

  In such a case, we say $f(n)$ is in the order of $g(n)$
- Example
  - $f(n) = 3n + 5$
  - If we select $g(n) = n$, it is easy to find $c = 4$ and n_0 =5 that $f(n) = 3n + 5 \leq c \cdot g(n) = 4n$, for $n \geq 5$
  - Thus, this function $f(n)$ has a time complexity of $O(n)$

Time Complexity of Algorithms

```
def find_min(data):
mindata=data[0]
for value in data
    if value < mindata
      mindata = value
return mindata
```

The time complexity of find_min() is $O(n)$

*Similar functions often do not differ much in space complexity, but may significantly differ in time complexity

The time complexity of the most naïve sort function is $O(n^2)$

The time complexity of many advanced sort functions, e.g., quick_sort(), is $O(nlog(n))$

See more details here

Common Functions' Complexity

Function	Expression	Complexity
Constant	$f(n) = C$	$O(1)$
Logarithm	$f(n) = log(n)$	$O(\log(n))$
Linear	$f(n) = n$	$O(n)$
N-log-N	$f(n) = n\ log(n)$	$O(\mathrm{n}\log(n))$
Quadratic	$f(n) = n^2$	$O(n^2)$
Cubic	$f(n) = n^3$	$O(n^3)$
Polynomial	$f(n) = a_0 + a_1 n + a_2 n^2 + \ \ldots + a_d n^d$	$O(n^d)$
Exponential	$f(n) = 2^n$	$O(2^n)$

Comparative Analysis of Complexity

- Because $g(n)$ should be in the simplest form, if at the same level (e.g., in one loop), there are multiple operations having different complexity, take the highest:

$$f(n) = n(nlog(n) + n^2)$$

Its complexity is $O(n^3)$

- Always select/develop the algorithm with a lower complexity

6

What Is Clustering?

- Natural clusters exist in our real world
  - ➢ E.g., superstar, star, all-round player, defensive player, etc. in NBA; Size XS, S, M, L, XL, etc. for clothes
  - ➢ May or may not have clearly defined rules/criteria to categorize them precisely
    - ❑ Especially for the borderline objects
- In computer science, clustering is the task to group a set of objects, such that
  - ➢ Objects within each cluster are more similar to each other (homogeneous) than to those in other clusters (heterogeneous)

Unsupervised vs Supervised Learning

- Clustering algorithms aim to analyze entities by discovering their underlying structure and organize them into different categories without priori knowledge
  - ➢ Most well-known type of unsupervised learning method
- If we know the ground-truth labels of the objects
  - ➢ Normally apply supervised learning methods for classification
  - ➢ Could still apply clustering methods to differentiate the objects
    - ❑ Should not use the labels during learning
    - ❑ Nonetheless, the labels could be used as **external information** to evaluate the clustering results (see later slides)

Financial Applications of Clustering

- Customer profiling
  - ➢ Model the common behaviors and status of similar customers
  - ➢ Thus, specific strategies may be applied with group differences
- Fraud identification
  - ➢ Obtain the outlying clusters for further investigations
  - ➢ Thus, potential fraud cases may be identified from these outliers

Source: Internet

Desirable Properties

- Accurate, able to deal with noise and outliers
- Able to discover clusters with arbitrary shape and density
- Scalability (low complexity)
- Requiring minimal domain knowledge to set parameters
- Insensitive to the order of data inputs
- Robust in high dimensionality
- Incorporation of user-specific constraints
- Interpretability and usability

Cannot have it all!

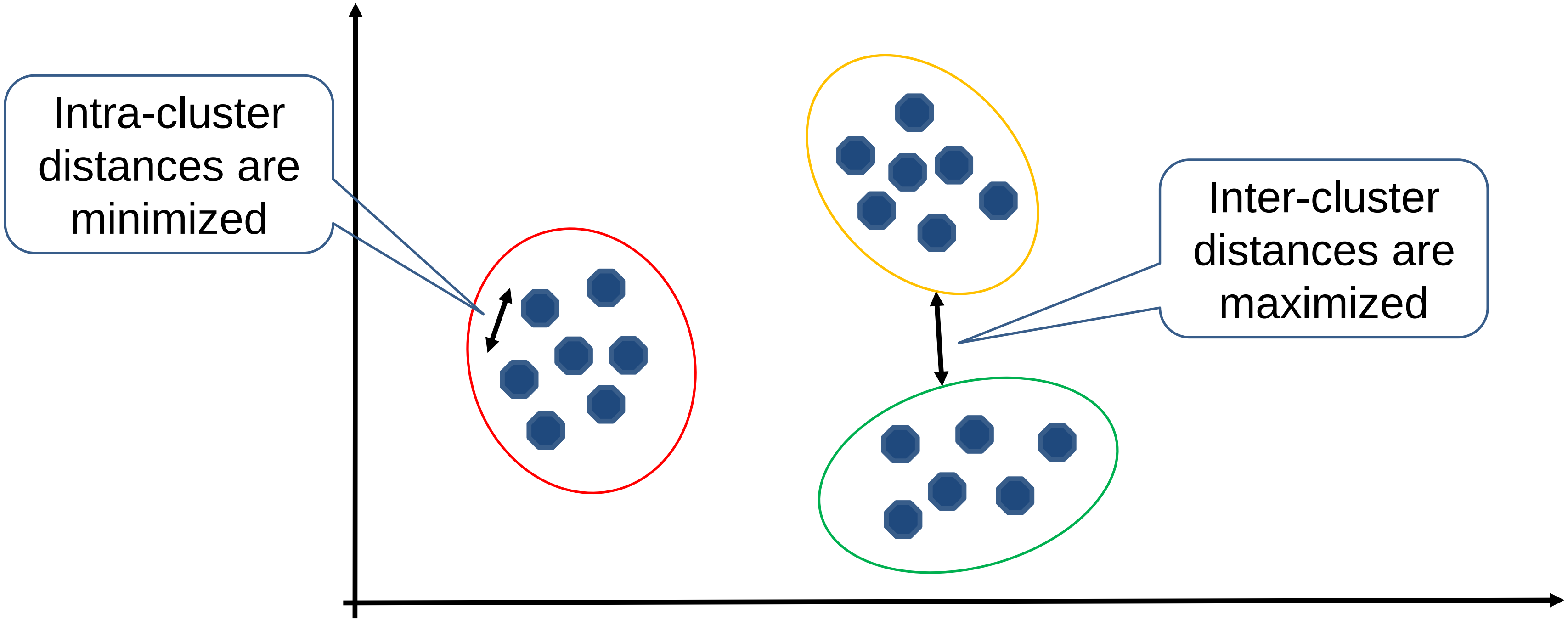
10

How Clustering Is Carried Out?

- A proximity measure is required to quantify the similarity/dissimilarity among objects
- Proximity should not be affected by the direction
- One of the widely adopted metrics is distance
  - Minkowski distance: $d(x_i, x_j) = \left(\sum_{n=1}^{N}\left(x_i^n - x_j^n\right)^p\right)^{\frac{1}{p}}$
  - When p=1: Manhattan distance (city block)
  - When p=2: Euclidean distance: $\sqrt{\sum_{n=1}^{N}(x_i^n - x_j^n)^2}$

N denotes the dimensionality

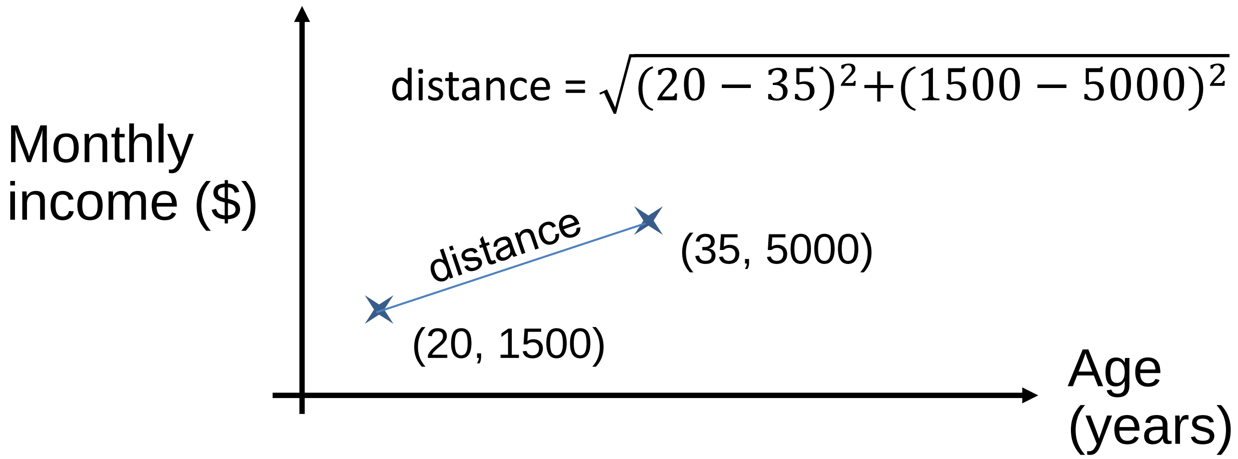
Forming Clusters



Cautious in Distance Measure

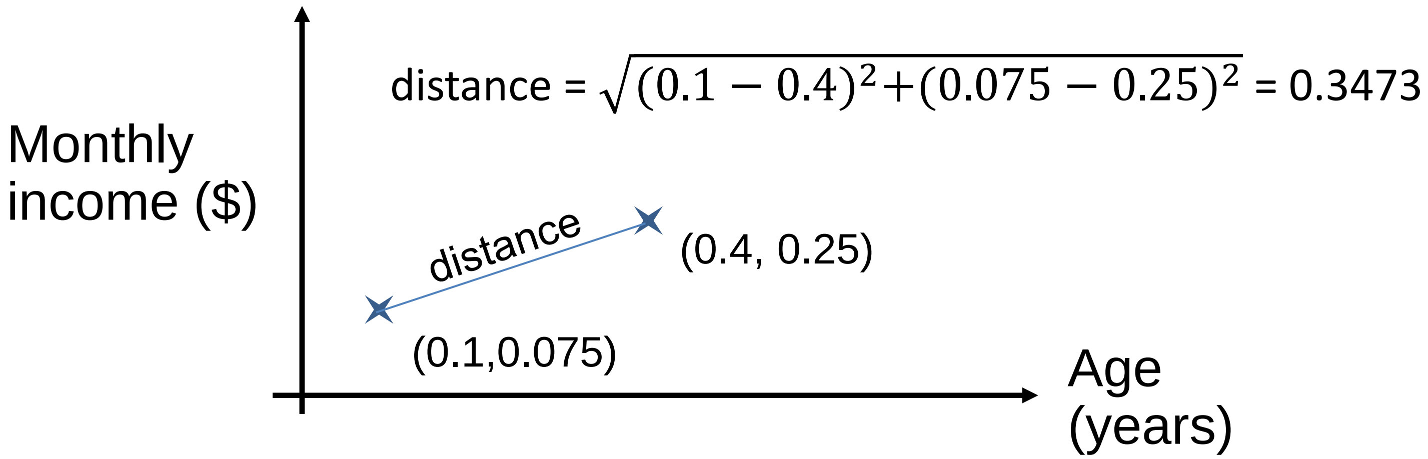
- Distance in the original data space may be dominated by large values

distance = $\sqrt{(20-35)^2+(1500-5000)^2}$ = 3500.03



Cautious in Distance Measure

- Solution: Normalization in each dimension
  - E.g., Age range: [15,65]→[0,1]; Income range: [0,20000]→[0,1]



14

What About Non-Numerical Values?

- Binary values: M vs F, absent vs present, etc.
- Derive a contingency table for binary vectors
  - ➢ Also known as a cross tabulation or crosstab

		object *j* 1	0
object *i*	1	a	b
	0	c	d

i = 110111

j = 101100

	1	0
1	a=2	b=3
0	c=1	d=0

a+b+c+d= length of the binary vector

Symmetric Binary Attributes

- A binary attribute is symmetric if both states (0, 1) have *equal importance*
  - ➢ E.g., male vs female

- Distance function: Simple matching coefficient
  - ➢ Proportion of mismatches of their values

$$d(i,j) = \frac{b + c}{a + b + c + d}$$

Asymmetric Binary Attributes

- A binary attribute is asymmetric if one state is *more important* or *more valuable* than the other
  - By conversion, state 1 is normally more important: Generally rare or infrequent as positive for some tests

- Distance function: Jaccard coefficient

$$d(i,j) = \frac{b+c}{a+b+c}$$

May also apply weights to represent relative importance

Distance vs Similarity

- Simple matching coefficient

$$d(i, j) = \frac{\text{number of non - common bit positions}}{\text{total number of bits}}$$

$$s(i,j) = 1 - d(i,j) = \frac{a + d}{a + b + c + d}$$

- Jaccard coefficient

$$d(i, j) = 1 - \frac{\text{number of 1's in } i \wedge j}{\text{number of 1's in } i \vee j}$$

$$s(i,j) = 1 - d(i,j) = \frac{a}{a + b + c}$$

Example of Jaccard Coefficient

Name	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	1	0	1	0	0	0
Ken	1	0	1	0	1	0
Bob	1	1	0	0	0	0

- All attributes are asymmetric binary
- 1 denotes presence or positive of test
- 0 denotes absence or negative of test

$$d(i,j) = \frac{b+c}{a+b+c}$$

$$d(Jack,Ken) = \frac{0+1}{2+0+1} = 0.33$$

$$d(Jack,Bob) = \frac{1+1}{1+1+1} = 0.67$$

$$d(Ken,Bob) = \frac{2+1}{1+2+1} = 0.75$$

Nominal Attributes

- Unlike binary attributes, a nominal attribute has *more than two states* or *values*
  - E.g., ethnic group, nationality, etc.
- May still apply the simple matching method
  - Let r denote the total number of attributes
  - Let q denote the number of values that match between the two given objects

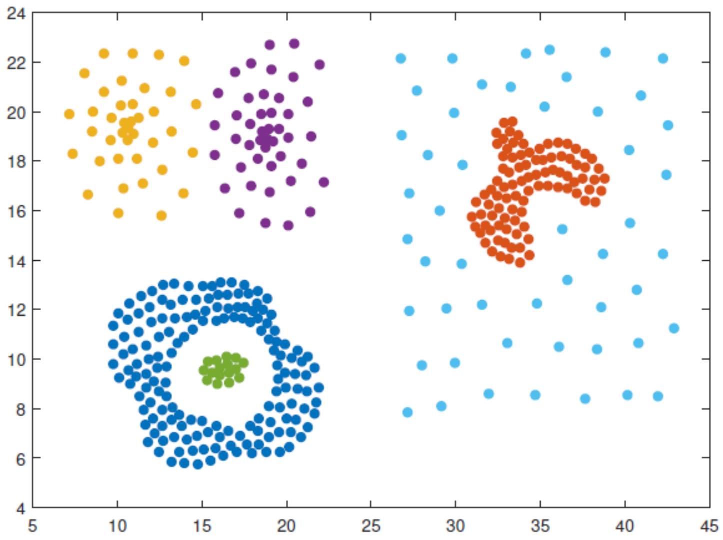
$$d(i,j) = \frac{r-q}{r}$$

Combining Different Types of Attributes

- Option 1:
  - ➢ Decide the dominate type
  - ➢ Convert the others to the dominate type
  - ➢ Calculate the distance
  - ➢ Although this is one commonly adopted approach, sometimes may not make much sense to perform the conversion
- Option 2:
  - ➢ Obtain the distance for each type individually
  - ➢ Combine the distance by applying a weighted sum
  - ➢ May not be straightforward to set the weights

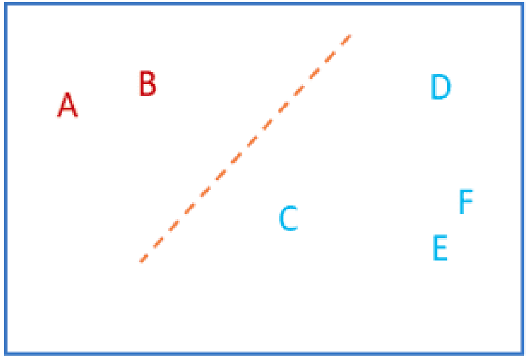
Types of Clustering Algorithms

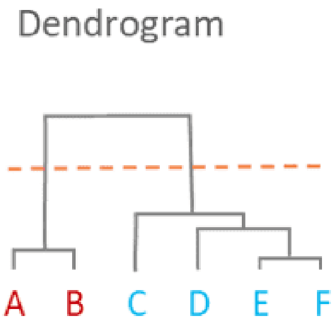
- Hierarchical vs Partitioning
- Online vs Offline
- Hard vs Soft (Fuzzy)
  - K-means vs [Fuzzy C-means]
- Density-based algorithms
- Model-based algorithms
  - E.g., SOM, [GMM]
- … …



Hierarchical Clustering

- Build a tree-based hierarchical taxonomy, known as **dendrogram**:
  - ➢ Clusters are obtained by cutting the dendrogram at a selected level: Each connected component forms a cluster





Source: [What is dendrogram]

23

Two Approaches to Get Dendrograms

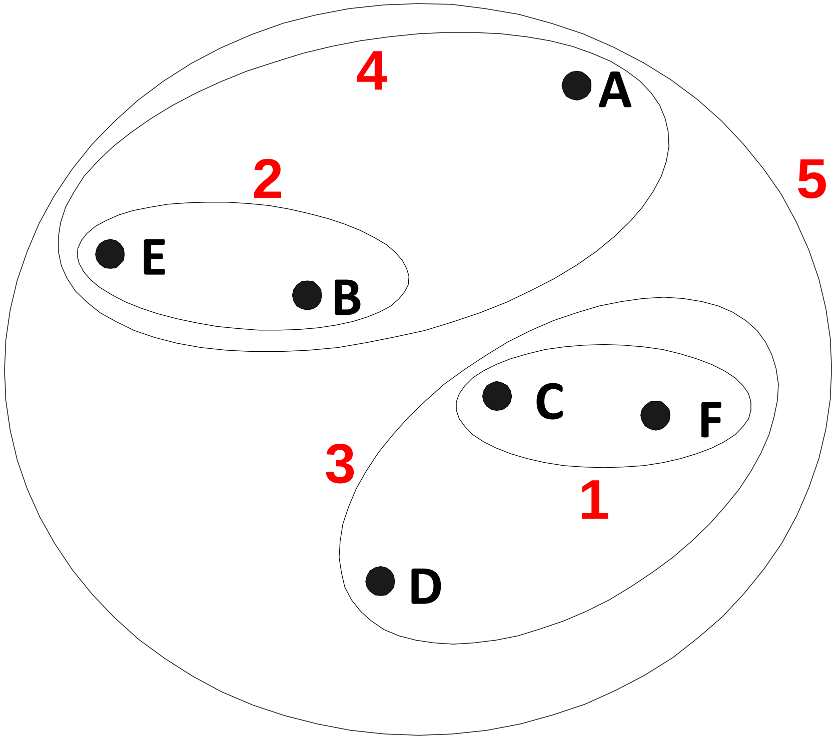
- Bottom-up approach: Agglomerative clustering
  - Each data element is treated as an initial cluster
  - Merge the most similar (or nearest) pair of clusters each time
  - Until all elements are merged into a single cluster
- Top-down approach: Divisive clustering
  - All data elements are grouped as one cluster called root
  - Recursively divide the cluster to obtain sub-clusters
  - Until only singleton clusters of individual elements remain or reach certain stopping criteria

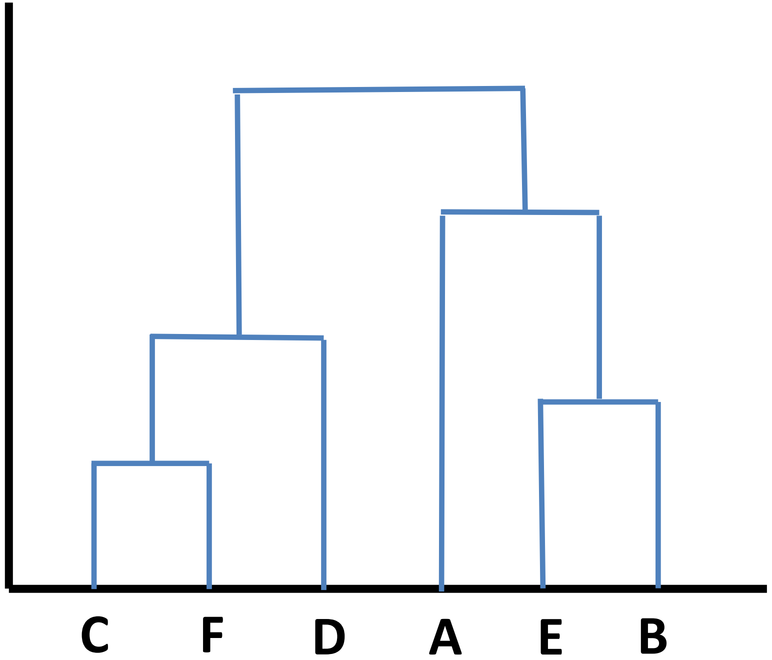
Agglomerative Clustering Algorithm

- Basic algorithm is straightforward:

1. Compute the proximity matrix
2. Let each data element be a cluster
3. **Repeat**
4. Merge two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

Agglomerative Clustering Algorithm



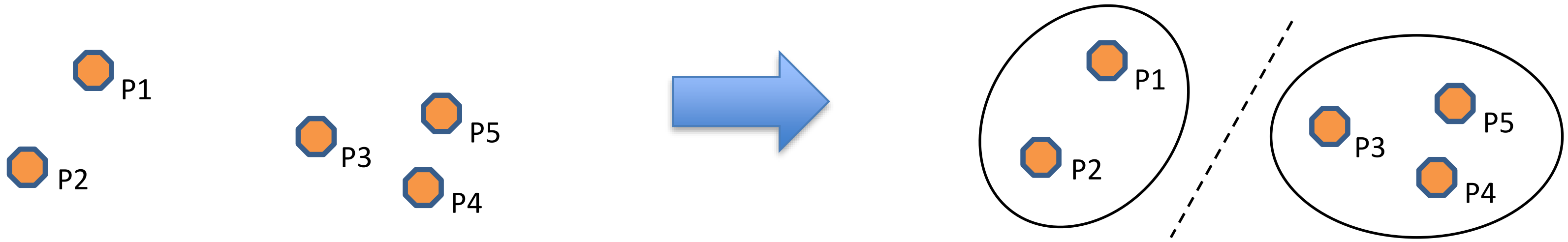


Complexity of Agglomerative Clustering

- Space complexity:
  - ➢ O(N^2) for the proximity matrix
  - ➢ N denotes the number of data elements
- Time complexity:
  - ➢ O(N^3) for many cases
    - ❑ N iterations
    - ❑ In each iteration, the proximity matrix needs to be searched and updated: N^2
  - ➢ O($N^2\log(N)$) for some approaches

Partitioning Clustering

- Construct partitions (usually random) of the given dataset, then refine them using certain criteria



- By far, the most well-known algorithm is K-means
  - ➢ K-medoids: Instead of mean, use a data element in the cluster to represent the cluster medoid (exemplar)

K-means Clustering Algorithm

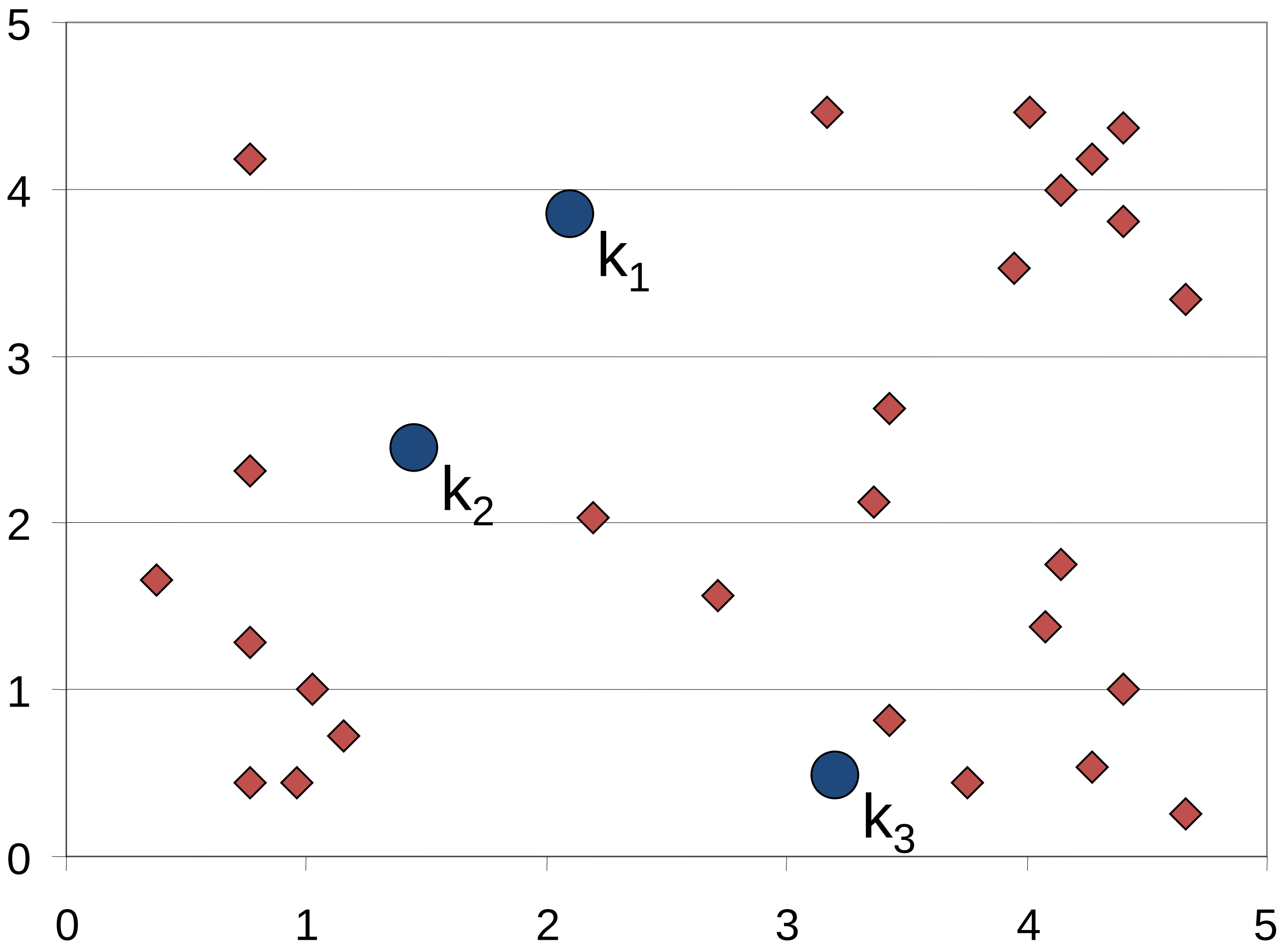
- Each cluster is associated with one and only one centroid (cluster mean)
- Each data element is assigned to the cluster with the closest centroid
- Need to predefine the number of clusters, i.e., K

```
Input: A set of data points D; Number of clusters K; Number of iterations G
Output: Cluster assignment of each data point
Randomly select K data points (may not from D) as initial cluster centroids
repeat
    Compute the distance between each data point and each centroid
    Associate each data point with its closest centroid
    All data points associated with the same centroid form a cluster
    Recompute the centroid of each cluster
until centroids are stabilized or number of iterations equals to G
```


29

Step 1:

Random initialization of centroids



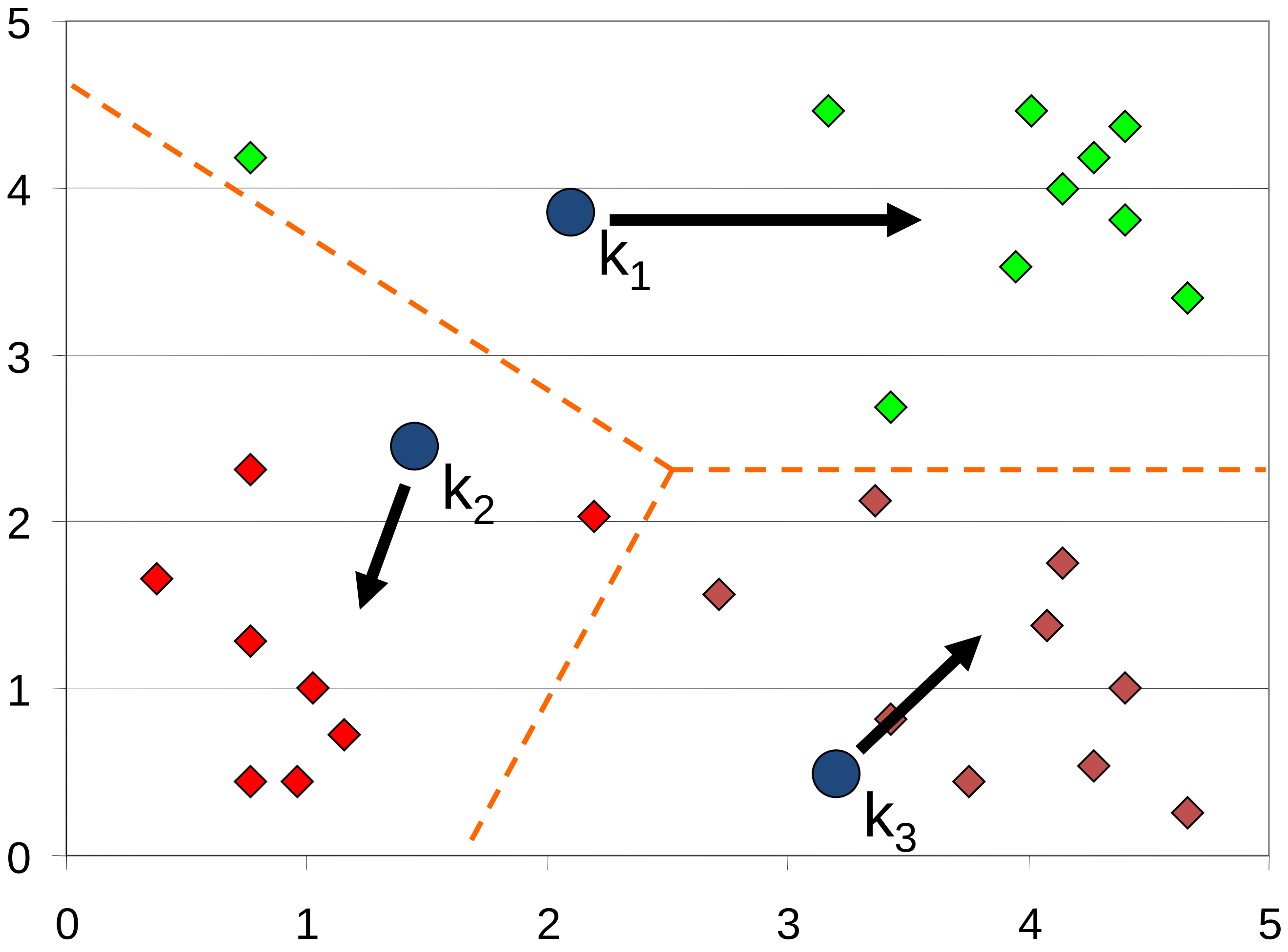
Source:
Lecture slides by
Carla Brodley,
Tufts University

Step 2:

Form clusters

Step 3:

Update the centroids

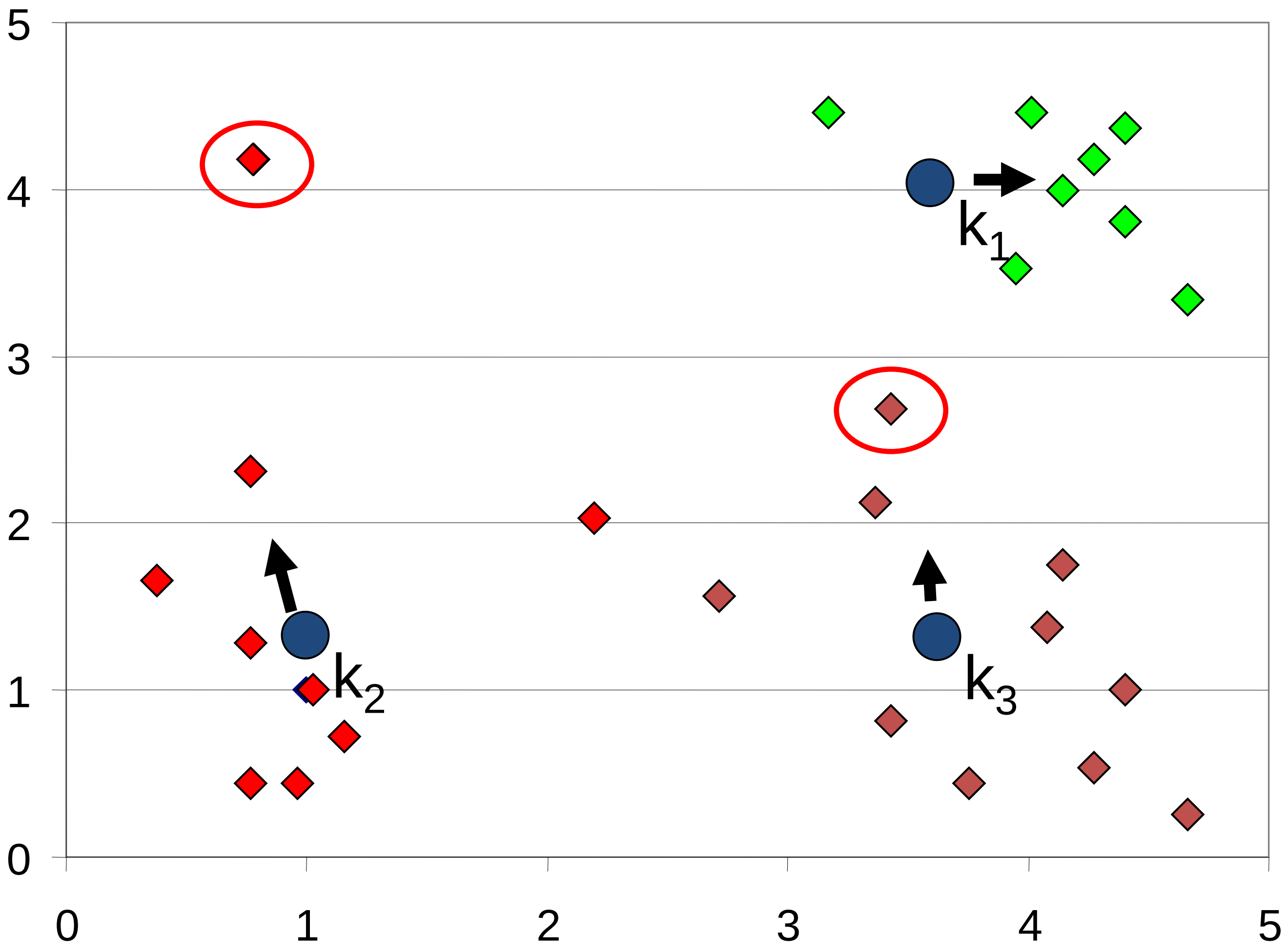


Step 4:

Update clusters

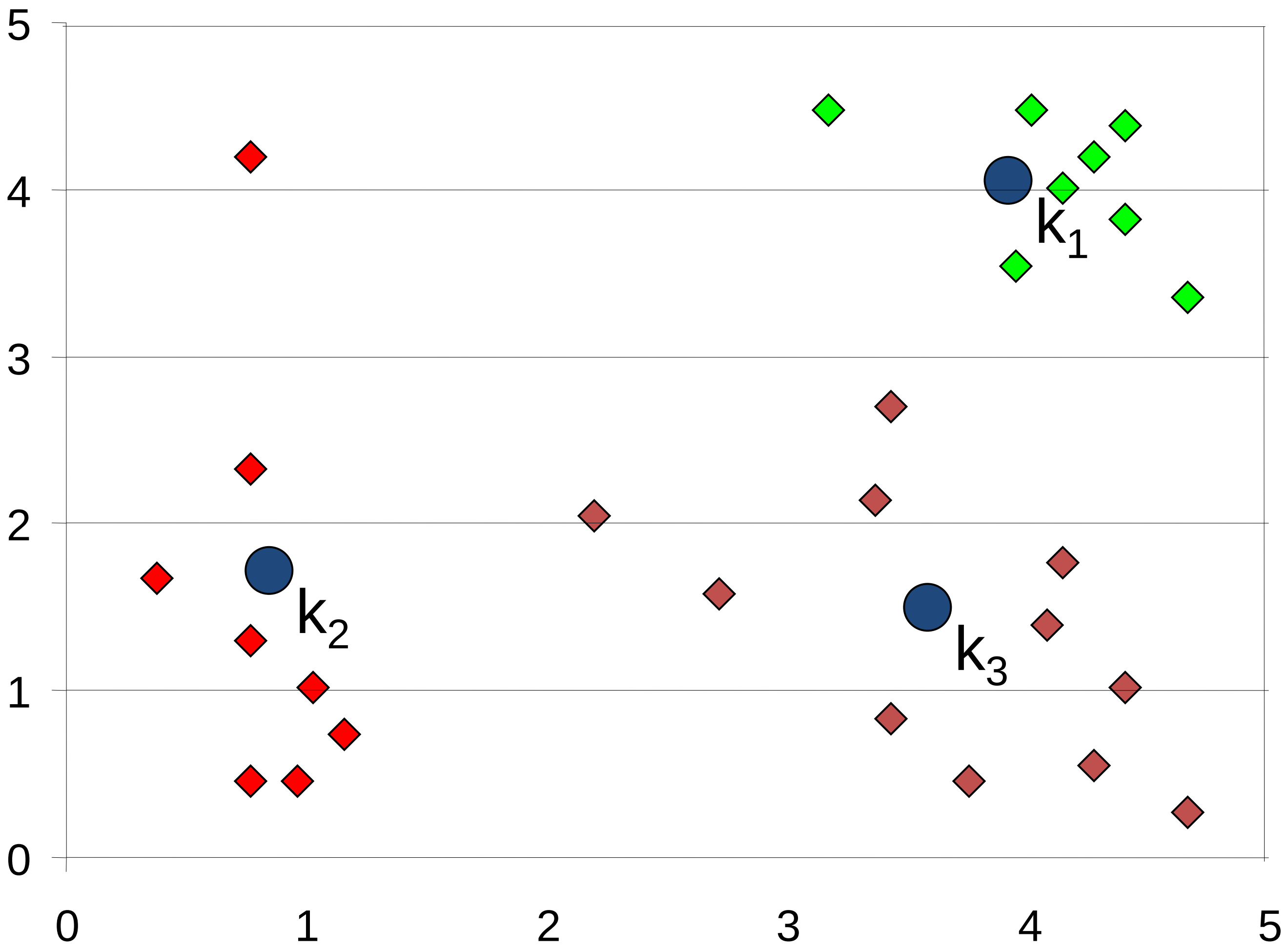
Step 5:

Update centroids again



Step 6:

Algorithm converges



Convergence of K-means

- Does K-means always converge?
  - Yes, because it is a special case of a general procedure known as the *Expectation Maximization (EM)* algorithm
  - EM is known to guarantee convergence
    - But the number of iterations needed to converge may be large
- For K-means:
  - Usually, the number of iterations needed to converge is not large
  - For complex datasets, no harm to set max_iteration

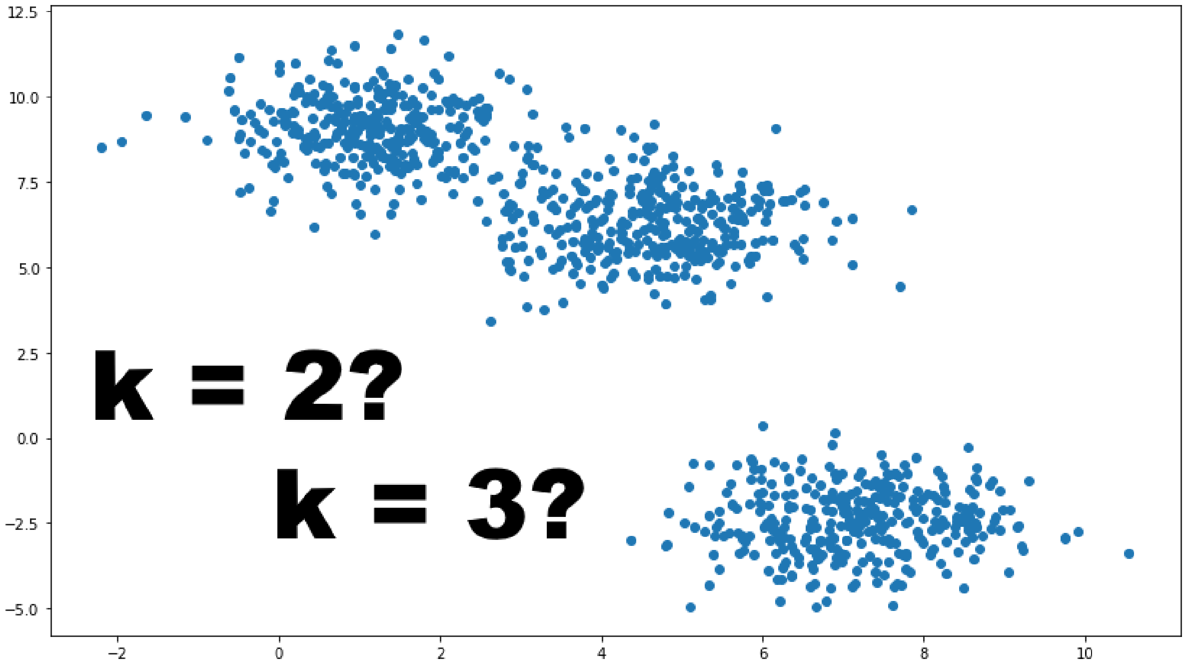
34

Complexity and Evaluation of K-means

- Time complexity of K-means is O(*I·K·N·F*)
  - *I*: #iterations; *K*: #clusters; *N*: # data elements; *F*: #features
  - Faster than hierarchical methods
- Commonly use *SSE* to evaluate the clusters
  - $SSE = \sum_{j=1}^{K} \sum_{x_i \in C_j} d(x_i, m_j)^2$
    - *C* denotes the clusters, x_i denotes each data element, and *m* denotes the cluster center
  - May use *SSE* to determine when to stop, and compare two sets of clustering results

Limitations of K-means (1)

- How to determine K?

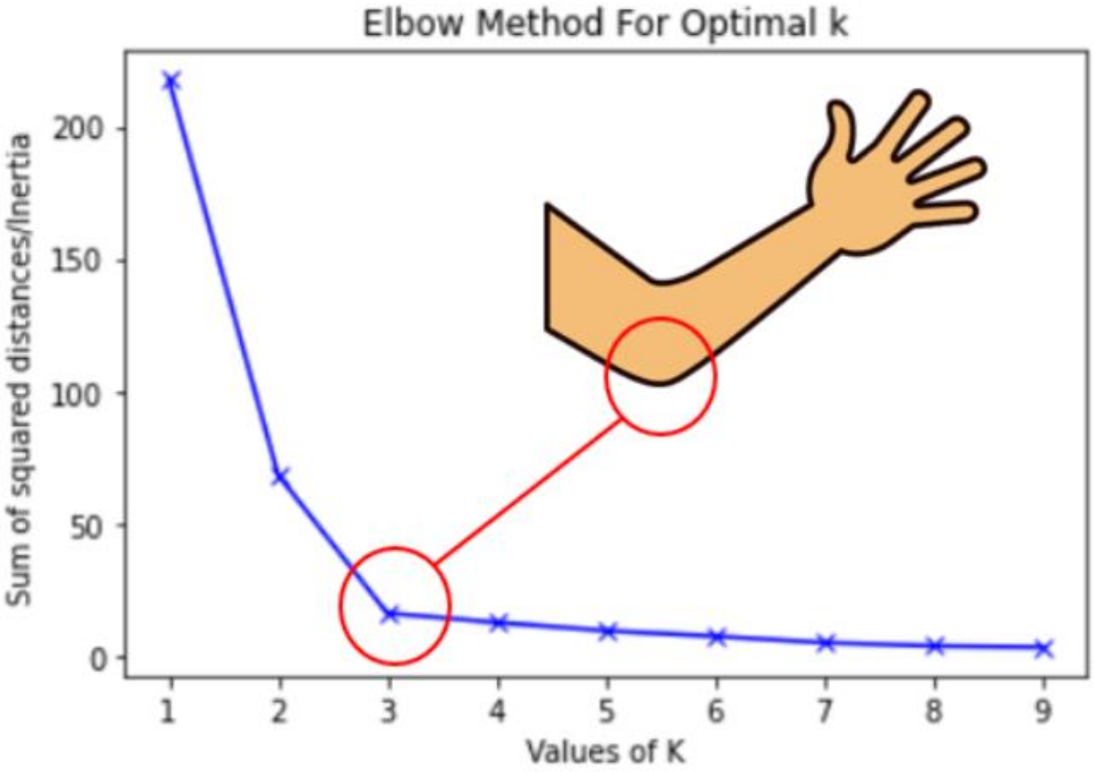


Source:
[How to Determine the Optimal K for K-Means?]

36

Determination of K

- Use the elbow method

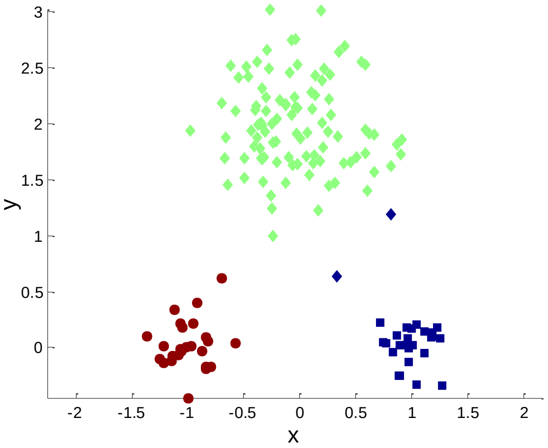


Source:
[K-Mean: Getting The Optimal Number Of Clusters]

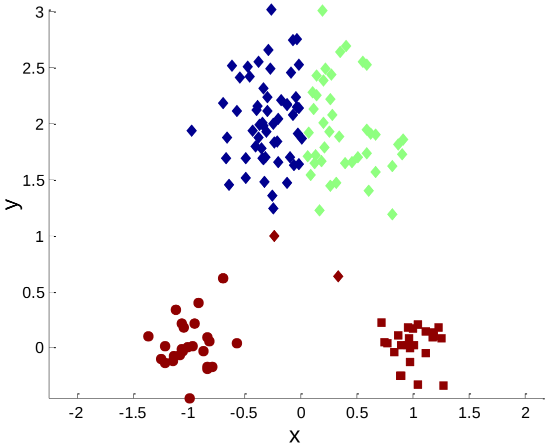
37

Limitations of K-means (2)

- Performance may vary drastically due to different initializations of cluster centroids
  - *Trap in local minima, hard to find global minima*



(Sub-)optimal result



Undesirable result

Source: Internet

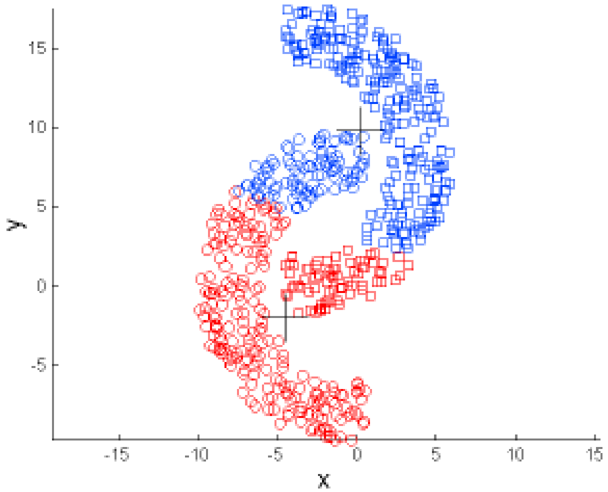
Initialization in K-means

- Try multiple times, then select the sensible cluster centroids as initializations
  - ➢ Computationally heavy, may not work for complex cases
- Pre-analyze the data distribution
  - ➢ Require additional computation and prior knowledge
- Select many centroids to start with, then select the well separated ones
  - ➢ Investigation required
  - ➢ May help to determine K at the same time

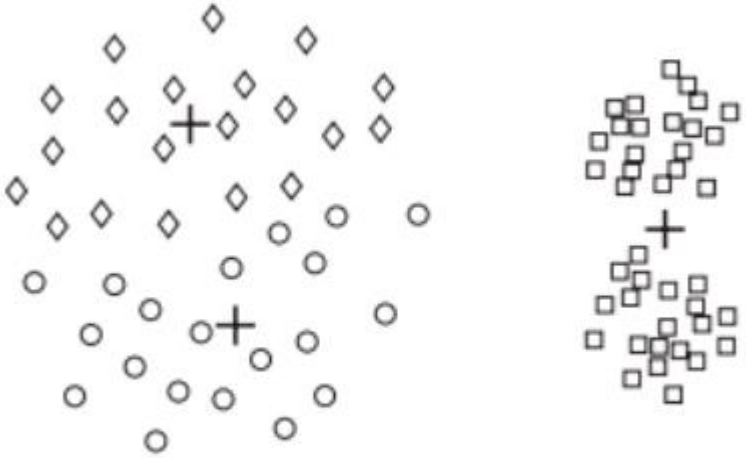
39

Limitations of K-means (3)

- Assume clusters are spherical



- Also, may not well handle different sizes and densities

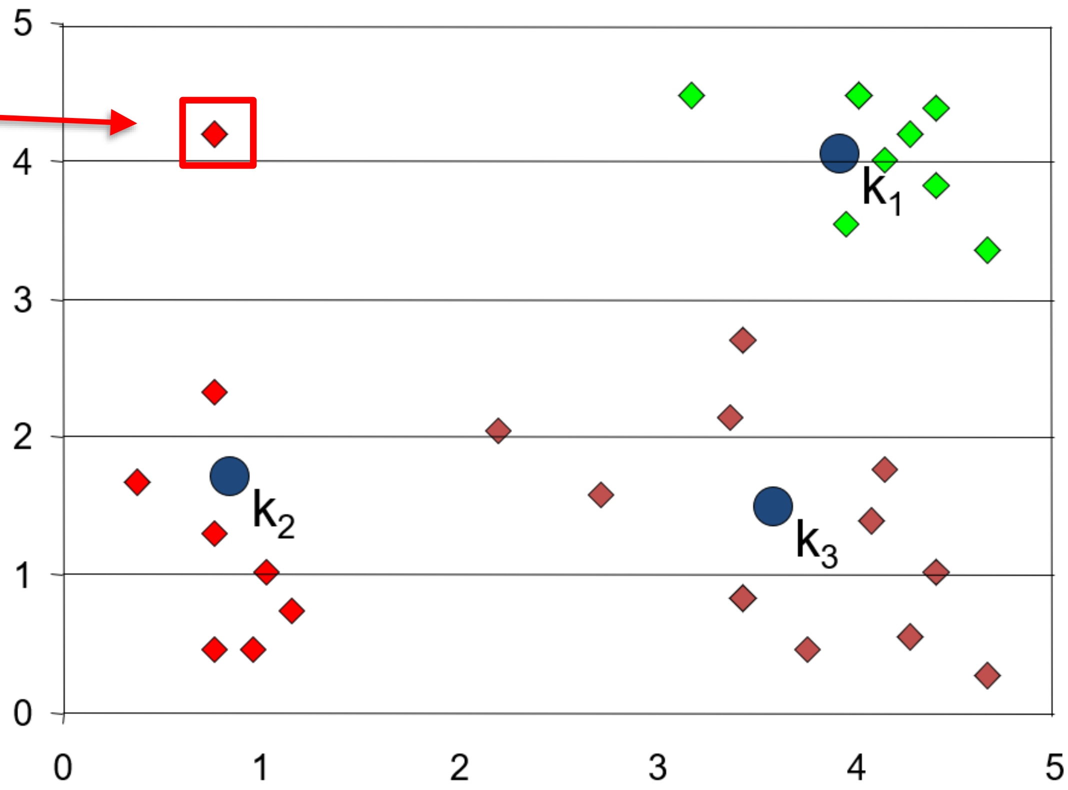


Source:
[Learn Clustering Method 101 in 5 minutes]

Limitations of K-means (4)

- Does not detect outliers and is sensitive to them

- Solution to limitations 3 & 4:
  - ➢ Use other clustering methods



41

Density-based Clustering Algorithms

- Clustering based on density (local cluster criterion), such as density-connected points
- Key features:
  - ➢ Able to discover clusters of arbitrary shape
  - ➢ Well handle noises / outliers
  - ➢ Need one or few scans only
  - ➢ Require density parameters to form clusters and know when to stop
- Representatives
  - ➢ DBSCAN (KDD'96), CLIQUE (SIGMOD'98), OPTICS (SIGMOD'99)
  - ➢ DPC (Science'14) and its variants: REDPC (Neurocomputing'19), McDPC (NCAA'20), VDPC (Information Sciences'23), etc.

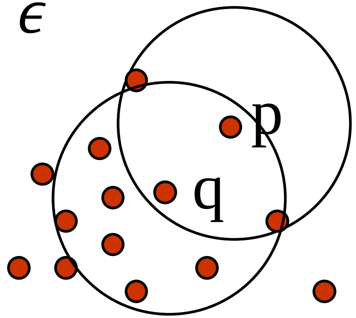
DBSCAN

- DBSCAN: **D**ensity-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise
- Two parameters:
  - ϵ : Radius of the neighborhood (*Eps*)
  - *MinPts* : A threshold to define density using the number of data points in the ϵ −neighborhood
- Neighborhood of data point *p*:
  - All data points within distance ϵ from *p*
  - $N_{Eps}(p) = \{q \mid d(p, q) \leq \epsilon\}$

43

Definitions in DBSCAN

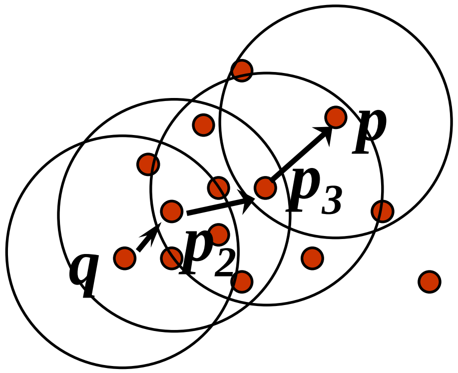
- If $|N_{Eps}(p)| \geq$ *MinPts*, then *p* is called a core point
- A data point *p* is directly density-reachable from another data point *q* if
  - $p \in N_{Eps}(q)$, i.e., $d(p, q) \leq \epsilon$
  - $|N_{Eps}(q)| \geq$ *MinPts*,
    i.e., *q* is a core point



MinPts = 5

Definitions in DBSCAN

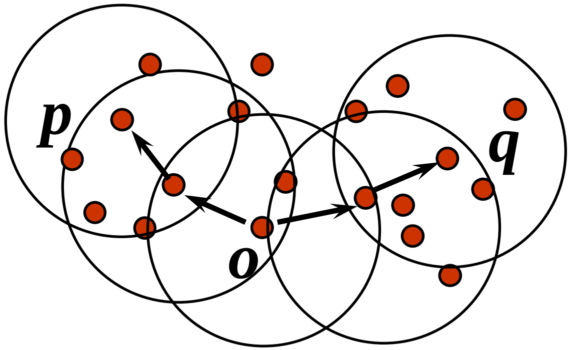
- A data point p is density-reachable from another data point q if
  - There is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



45

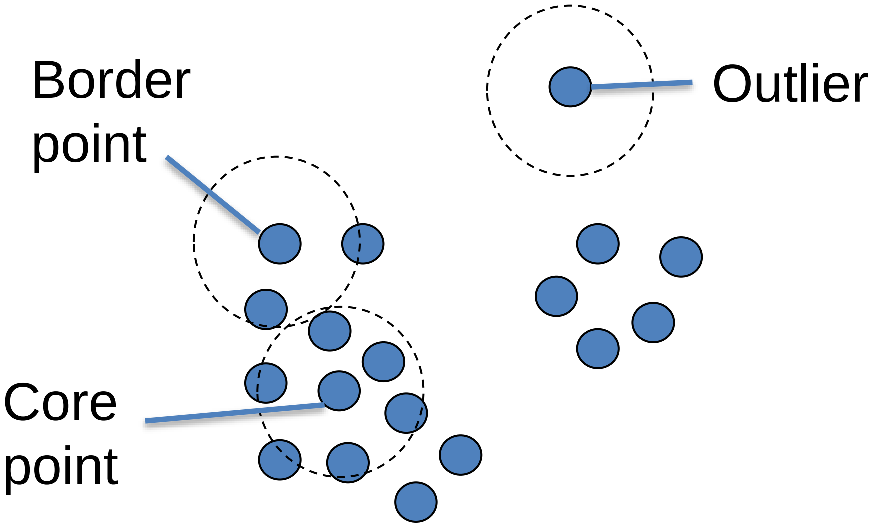
Definitions in DBSCAN

- A data point *p* is density-connected to another data point *q* if
  - ➢ There is a data point *o* such that both *p* and *q* are *density-reachable* from *o*



Cluster Formation in DBSCAN

- A cluster is defined as a maximal set of density-connected points
- If a data point is not included in any cluster:
  - ➢ It is labelled as an outlier



Algorithm of DBSCAN

Input: N objects to be clustered and global parameters *Eps*, *MinPts*.

Output: Clusters of objects.

Algorithm:

1) Arbitrary select a point *P*.
2) Retrieve all points density-reachable from *P* wrt ***Eps*** and ***MinPts***.
3) If *P* is a core point, a cluster is formed.
4) If *P* is a border point, no points are density-reachable from *P* and **DBSCAN** visits the next point of the database.
5) Continue the process until all of the points have been processed.

- Time complexity: O(*N log(N)*)
  - Worst case: O(N^2), when ϵ is ill selected (e.g., all distance $< \epsilon$)

 48

Limitations of DBSCAN

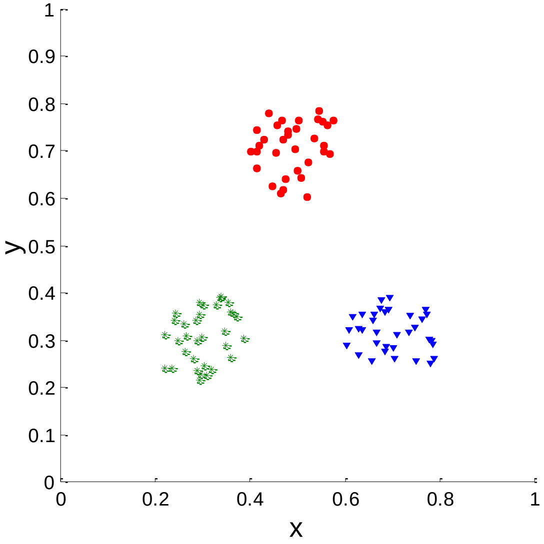
- Not entirely deterministic:
  - ➢ The assignment of borderline points on multiple borders
- May not work well if the dataset has large differences in densities
  - ➢ Because the combination of *MinPts* and *Eps* cannot be chosen properly for all clusters
- Need to predetermine two parameters
  - ➢ Although only two, not trivial, need to be carefully determined
- Hence, new algorithms such as DPC emerge

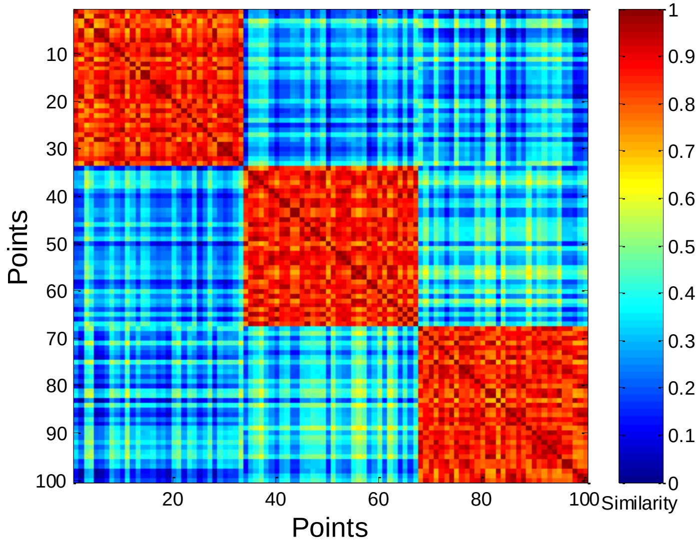
Validating Clusters

- In unsupervised learning, we need to define how to evaluate the goodness of the resulting clusters
  - Without ground-truth labels for measures like accuracy
- We need such evaluation criteria to:
  - Avoid finding poor representative patterns, especially when the dataset is noisy and/or tricky
  - Benchmark clustering algorithms
  - Compare different clustering results

Similarity Matrix for Cluster Validation

- Visualize the similarity matrix
  - Order the data points according to cluster labels



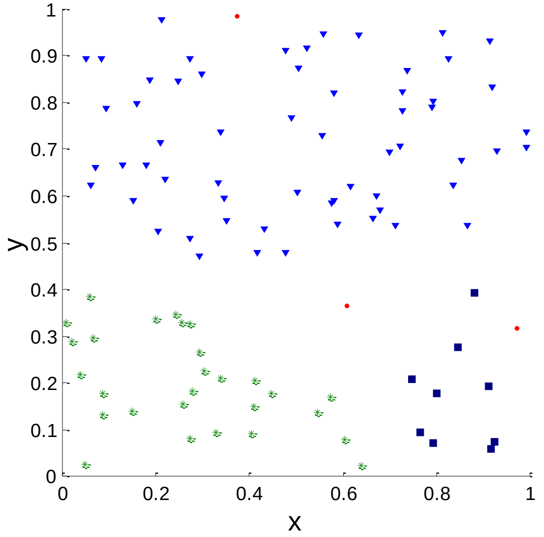


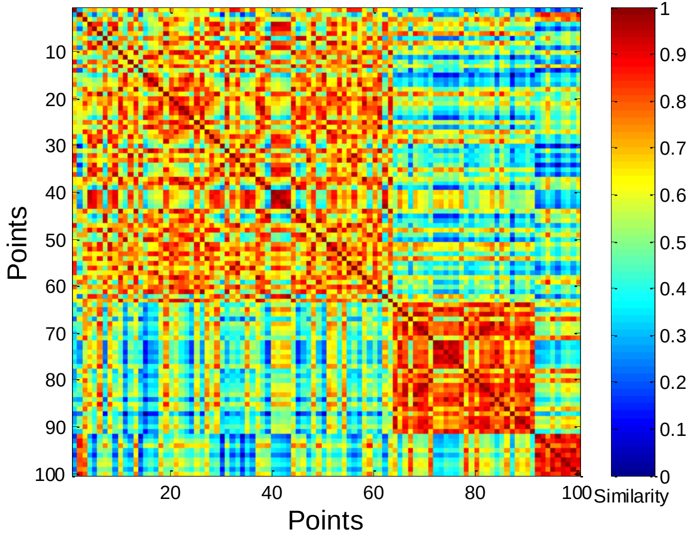
Source: Internet

51

Similarity Matrix for Cluster Validation

- Less well separated clusters





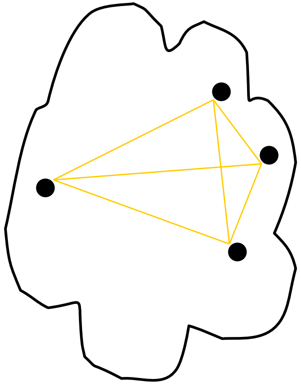
Source: Internet

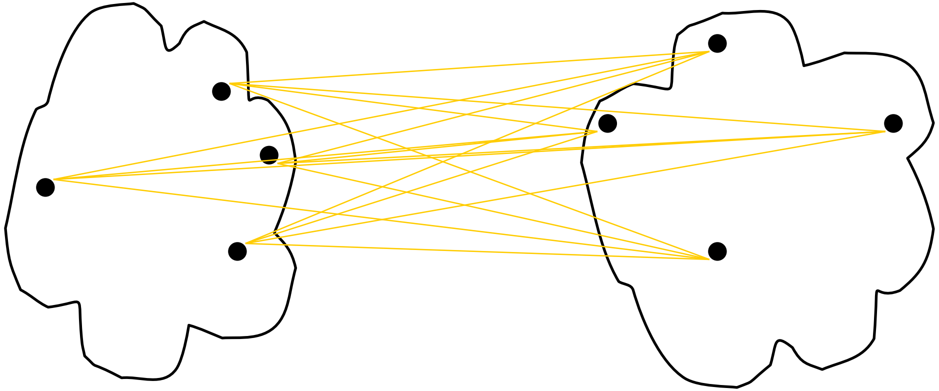
Cohesion vs Separation

- Cluster cohesion:
  - Measures how closely the objects are within the clusters
  - E.g., use within-cluster sum of squared errors (*SSE*):
    $WSS = \sum_{j=1}^{K} \sum_{x_i \in C_j} d\left(x_i, m_j\right)$
- Cluster separation:
  - Measures how distinct or well-separated the clusters are from the others
  - E.g., Use between-cluster sum of errors:
    $BSS = \sum_i |C_i| d(m, m_i)$, where m denotes the centroid of the whole dataset and $|\cdot|$ denotes the size of the cluster

Cohesion vs Separation

- Note that *WSS* + *BSS* = constant





- Cohesion: Sum of weights of all links within the cluster
- Separation: Sum of weights between data points in the cluster and those outside

54

Silhouette Coefficient (SC)

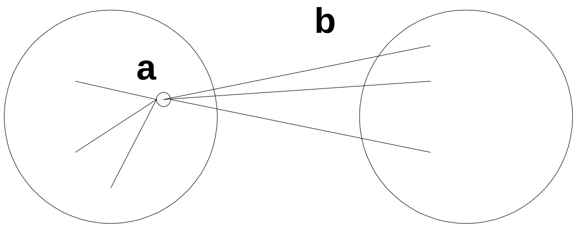
- It combines the ideas of both cohesion and separation
- For an individual data point $i \in C_I$:
  - Compute the mean distance between *i* and all other data points in the same cluster

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, j \neq i} d(i, j)$$

  - Compute the mean dissimilarity of data point *i* to another cluster C_J as the mean of the distance from *i* to all data points in C_J, and select the minimum value among all such distances

$$b(i) = \min \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j)$$

Silhouette Coefficient (SC)



- Then the silhouette coefficient for data point *i* is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, if\ |C_I| > 1$$

$$s(i) = 1, if\ |C_I| = 1$$

Silhouette Coefficient (SC)

- $-1 \leq s(i) \leq 1$
  - 1: Best matched to the cluster
  - 0: On the border between two clusters
  - -1: Better fit in the neighboring cluster

- Let SC = mean($s(i)$), obviously $-1 \leq SC \leq 1$
  - A larger value denotes an overall better clustering formation, usually

57

External Measures

- Evaluation of clustering results without ground-truth labels is called internal measure

- Evaluation with ground-truth labels (remember, not used during clustering) is called external measure
  - ➢ We can reuse the concept of confusion matrix
  - ➢ If the size of the dataset is N, TP+TN+FP+FN=C_N^2

	Same cluster in clustering	Different clusters in clustering
Same class in ground-truth	**TP**	**FN**
Different classes in ground-truth	**FP**	**TN**

58

External Measures

- Rand Index (RI):

$$RI = \frac{TP + TN}{TP + TN + FP + FN}$$

- Adjusted Rand Index (ARI)
  - Adjusted measure using contingency table
- Jaccard Index (JI)

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

Other Evaluation Measures

- A lot more measures proposed in literature to evaluate the clustering results
  - Internal measure (without ground-truth)
    - DBI (Davies–Bouldin Index), sensitive to #clusters
  - External measure (with ground-truth)
    - NMI (Normalized Mutual Information)

Infusing Clustering into FNN

- Clustering has been used in FNNs, to generate fuzzy membership functions (fmf)
  - Data samples are grouped in the clustering process
  - Representations of clusters are used to derive fmf
  - Thus, fmf are initialized rather than learned from scratch
  - Also, the structure of FNN is determined by the clustering results
- This leads to a family of self-organizing FNNs

61

Two Examples of Self-Organizing FNNs

- DENFIS
  - Dynamic Evolving Neural-Fuzzy Inference System
  - Kasabov & Song, 2002
  - Employs TS rules
  - Evolving Clustering Method (ECM)
  - Clusters are formed in the hyperspace
  - Each cluster is transformed into a fuzzy rule

- GenSoFNN
  - Generic Self-organizing Fuzzy Neural Network
  - Tung & Quek, 2002
  - Employs Mamdani rules
  - Discrete Incremental Clustering (DIC)
  - Clusters are formed on individual dimensions
  - Rules are determined based on the selected rule generation scheme

Evolving Clustering Method in DENFIS

- Both online and offline modes supported
- Its idea is generic:
  - If the new data point is close enough to some clusters
    - Merge it into the nearest cluster, update cluster characteristics if needed (based on distance criteria)
  - Else, create a new cluster
- Takes only one parameter
  - *Dthr*: A constant value as the threshold for distance constraint
- Use normalized Euclidean distance
  - $d(i, j) = \frac{\sqrt{\sum_q (x_i - x_j)^2}}{\sqrt{q}}$, where *q* denotes the number of dimensions

ECM Algorithm

create a cluster using the first data point

for each subsequent data point *i* **do**

  compute its distance to all cluster centers denoted as *C*

  find the nearest cluster *m*

  if $d(i,C_m) \leq R_m$, where *R* denotes the cluster radius, merge *i* into cluster *m*, no update needed

  else

    compute distance $s(i,C_j)$ to all cluster centers, where $s(i,C_j)=d(i,C_j)+R_j$

    find the nearest cluster *a*

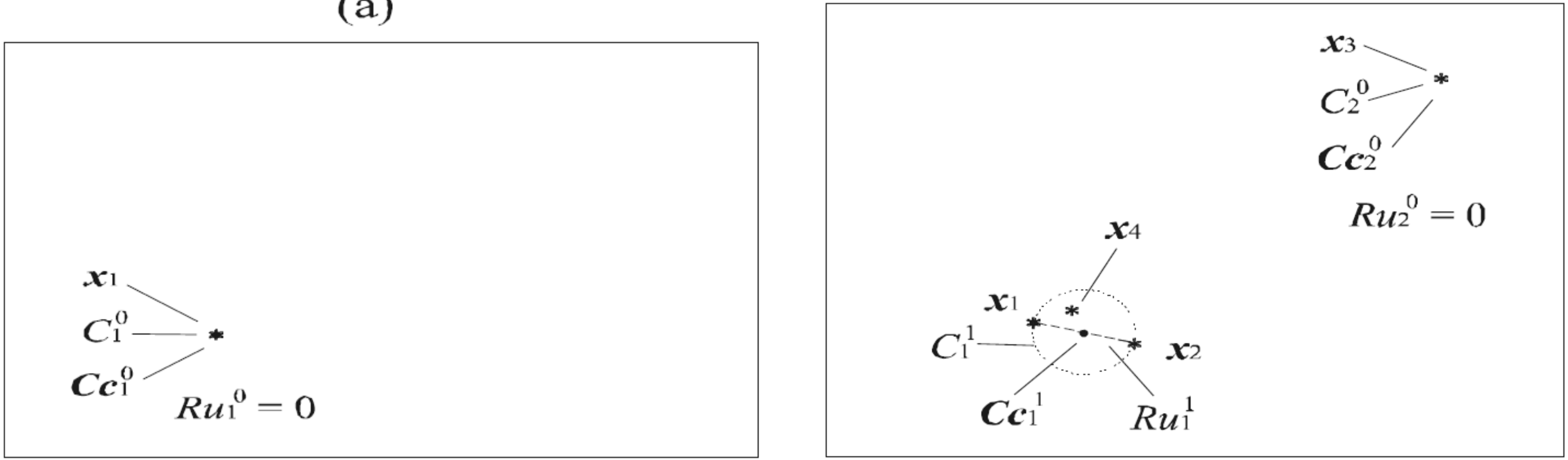
      if $s(i,C_a)>2Dthr$, create a new cluster

      else merge *i* into cluster *a* and if $R_a < Dthr$, update the center and radius

64

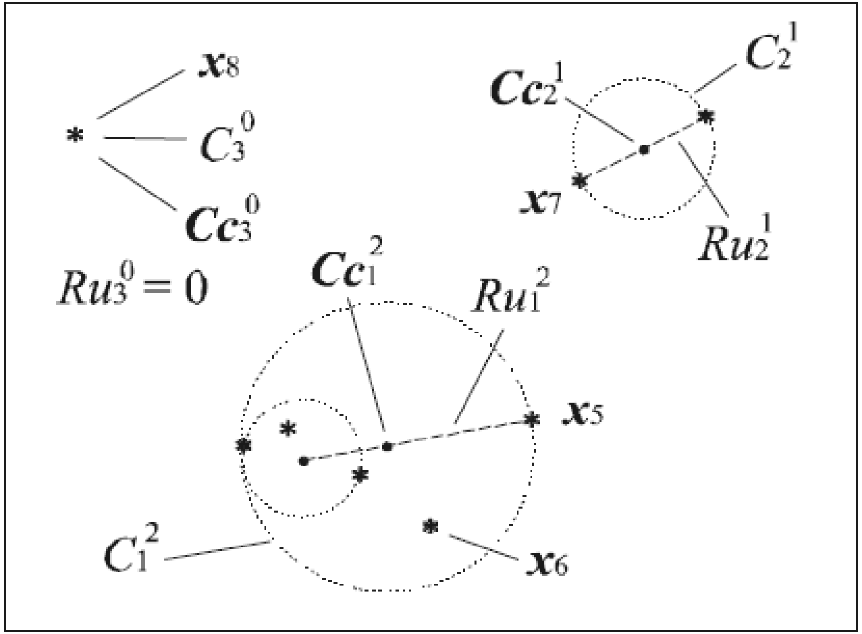
ECM in Action

(a) (b)

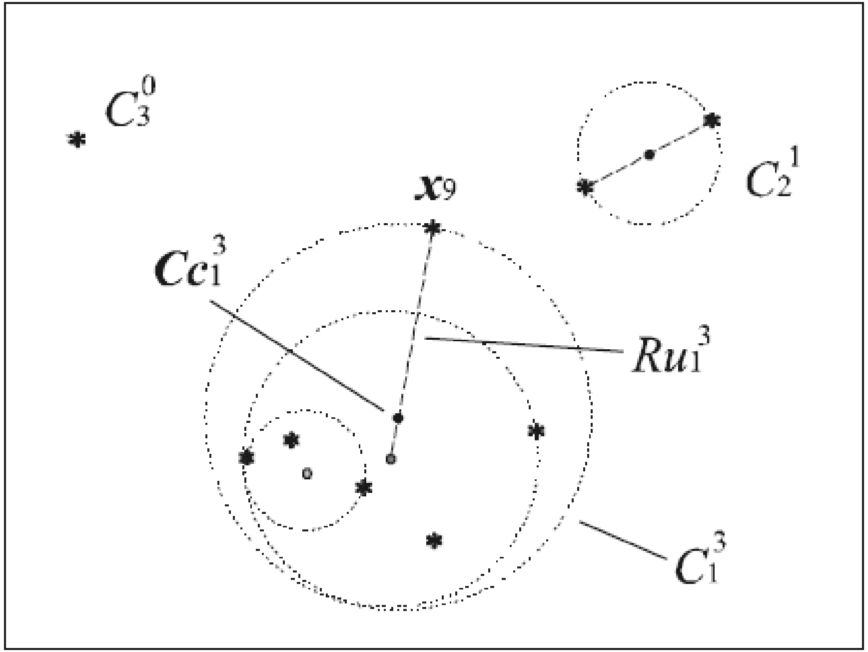


ECM in Action

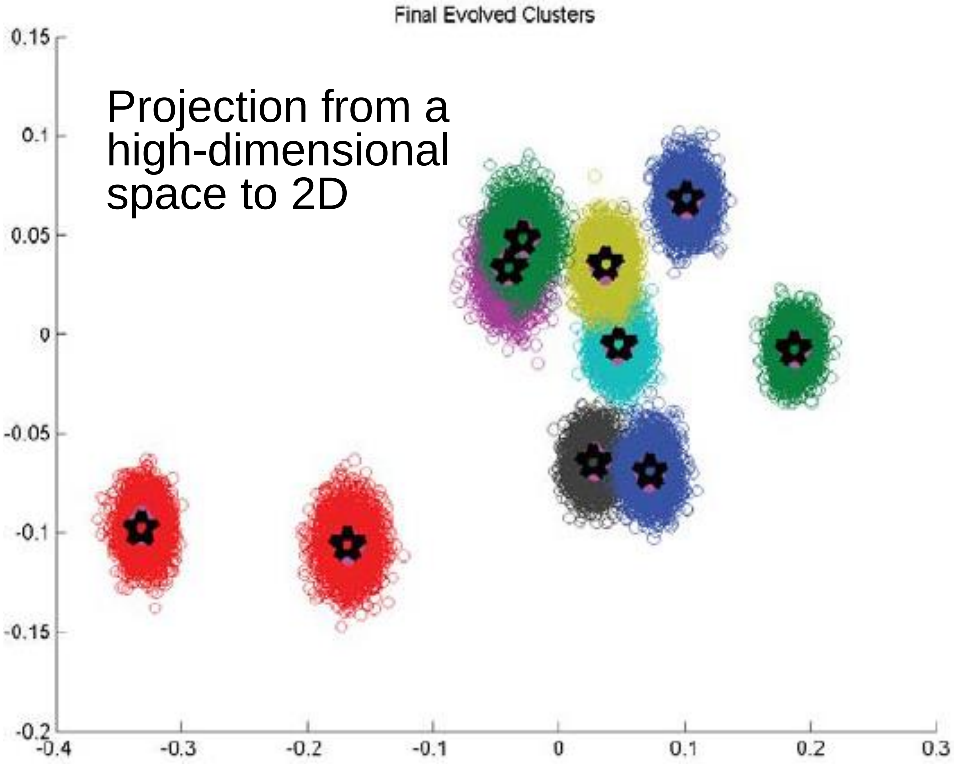
(c)



(d)



Clusters Obtained Using ECM

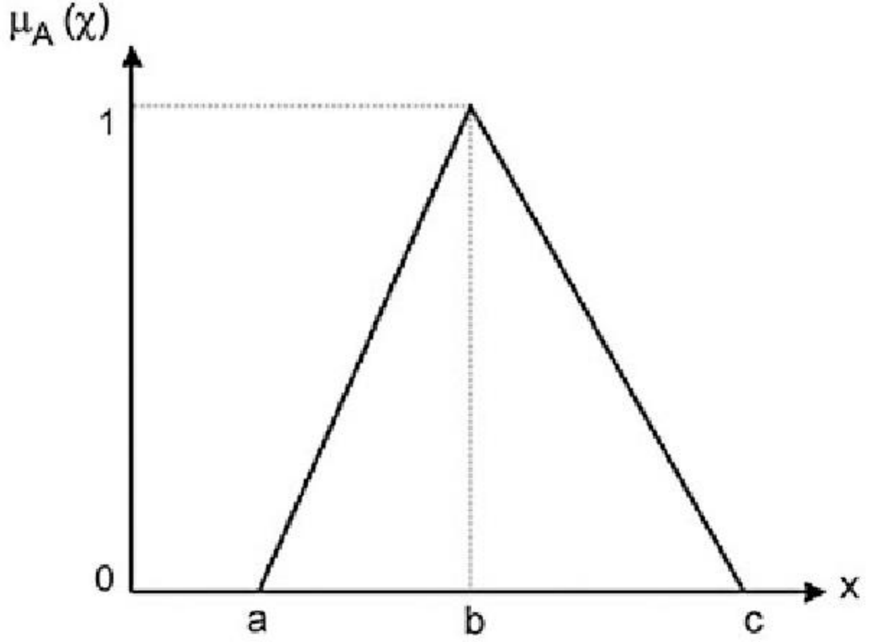


- Hyperspheres
- Each cluster is formulated as one fuzzy rule in DENFIS
  - Generates lots of rules (to cover the high-D space)
  - Curse of dimensionality

Source: [Karnowski's PhD Thesis]

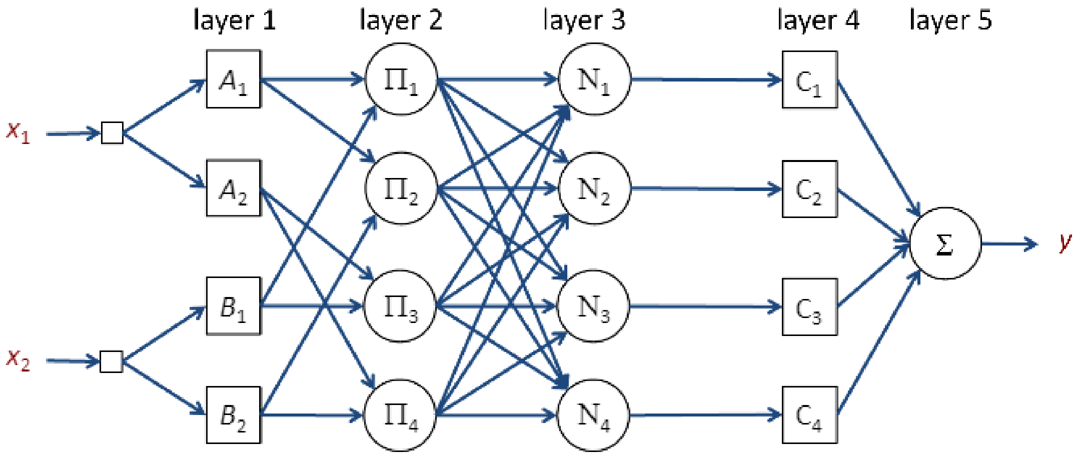
Transforming Clusters to FMF

- Triangular fmf
  - *b* takes the value of the cluster centre on the respective dimension
  - $a = b - \beta \cdot Dthr$
  - $c = b + \beta \cdot Dthr$
  - β is a predetermined parameter in the [1.2, 2] value range



DENFIS Architecture

- Same as ANFIS
  - ➢ #rules no longer p^N
- Same update mechanism for L4 parameters



- Different update mechanism for L1 parameters
  - ➢ Not learning from scratches means leads to shorter training time than ANFIS

Benchmark on MACKEY-GLASS

Methods	Neurons or Rules	Epochs	Training Time (s)	Training NDEI	Testing NDEI
MLP-BP	60	50	1779	0.083	0.090
MLP-BP	60	500	17928	0.021	0.022
ANFIS	81	50	23578	0.032	0.033
ANFIS	81	200	94210	0.028	0.029
DENFIS I	116	2	352	0.068	0.068
DENFIS I	883	2	1286	0.023	0.019
DENFIS II	58	100	351	0.017	0.016