

Data Mining - Clustering

Nhat-Quang Doan

University of Science and Technology of Hanoi

nq.doan@gmail.com

Overview

1 Introduction

2 Clustering methods

- Hierarchical Clustering
- Density-based Clustering
- Centroid-based Clustering

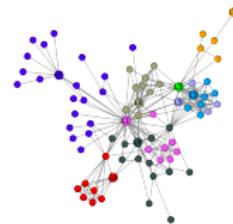
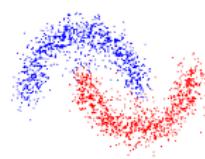
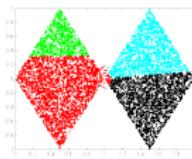
3 Clustering quality

Introduction

Objectives

- Clustering is the unsupervised learning task of data mining that retrieves information from data and determines the relationship between objects.
- Clustering consists in grouping a set of unlabeled data objects (instances) based on a similarity measure such that objects in the same *cluster* (group) are similar to each other and dissimilar to those in other clusters.

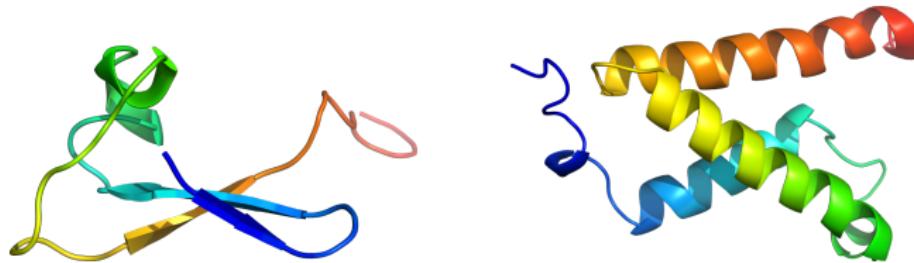
Examples



Introduction

Applications

- **Bio-informatics:** detecting the cancer patients or finding abnormal genes

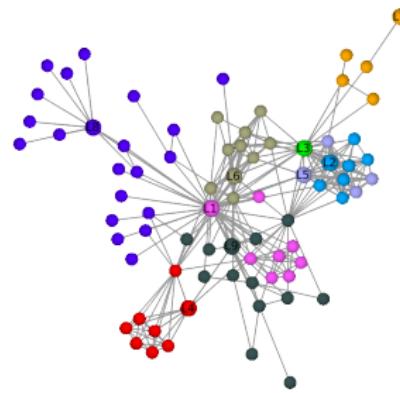
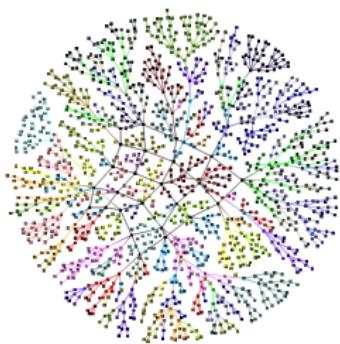


- **Marketing:** partitioning general customers' information.

Introduction

Applications

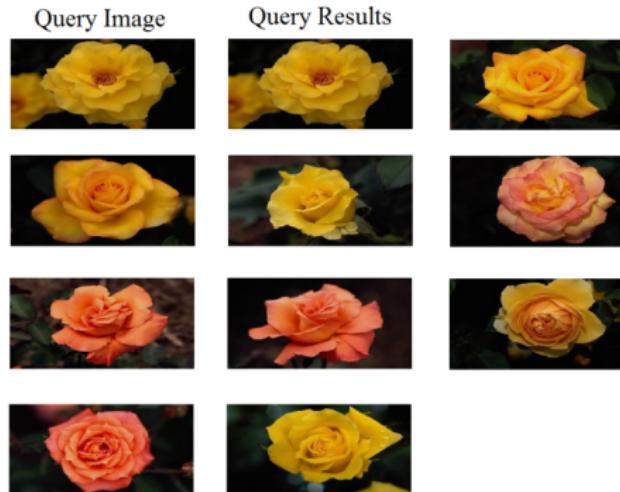
- **Image Segmentation:** locating the homogeneous zones in images.
- **Text Mining:** grouping texts, files, books.
- **Social Network:** detecting communities within large groups of people.



Introduction

Applications

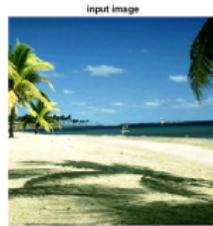
- Content-Based Information Retrieval: Wang dataset



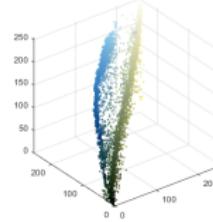
Introduction

Applications

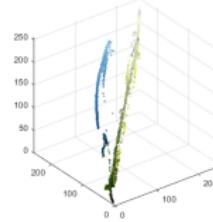
- Image Segmentation



Pixel Distribution Before Meanshift



Pixel Distribution After Meanshift



Introduction

Applications

- Recommendation System: Amazon dataset

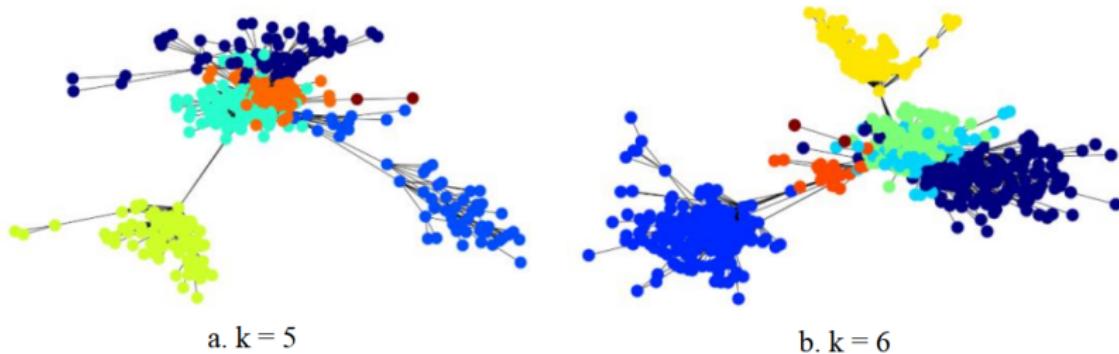


Fig. 2. Visualization of clustering result for the Cell Phones and Accessories dataset.

Problematics

Difficulties

- Data nature: binary, graph, vector, tree, etc...
- Definition of similarity (or dissimilarity) measure between data objects.
- Clustering algorithms
- Big data
- Evaluation of a clustering result
- Data visualization

Notions

Input set: $\mathcal{X} = \{x_i\}, \forall i = 1, \dots, n$: a set of n objects

Object: $x_i \in \mathbb{R}^d$ with $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$

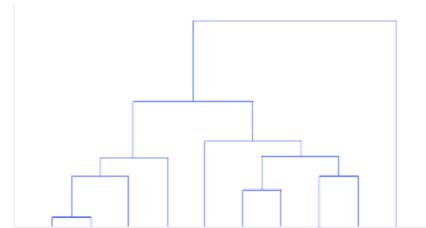
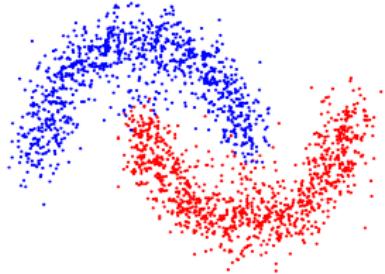
Cluster: $C_k \in \mathcal{C} = \{C_1, \dots, C_K\}, \forall k = 1, \dots, K$. The cardinality in each cluster C_k is denoted by n_k .

Problem: Each object $x_i \in \mathcal{X}$ is assigned to a cluster $C_k \in \mathcal{C}$ such that objects in C_k are similar.

Clustering algorithms

Clustering algorithm categories:

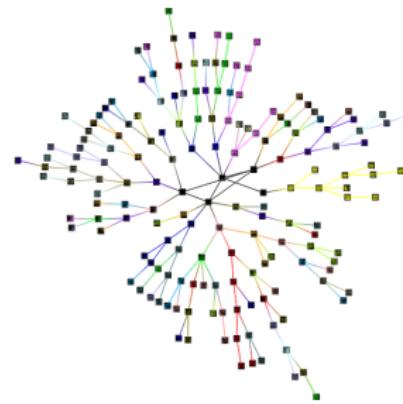
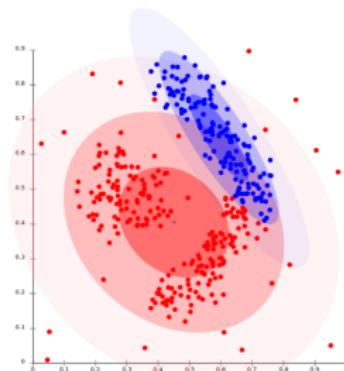
- Centroid-based clustering algorithms: K-means, Self-organizing maps (SOM), Neural Gas.
- Hierarchical clustering algorithms: Agglomerative Hierarchical Clustering (AHC), BIRCH, CURE. Clustering tree



Clustering algorithms

Clustering algorithm categories:

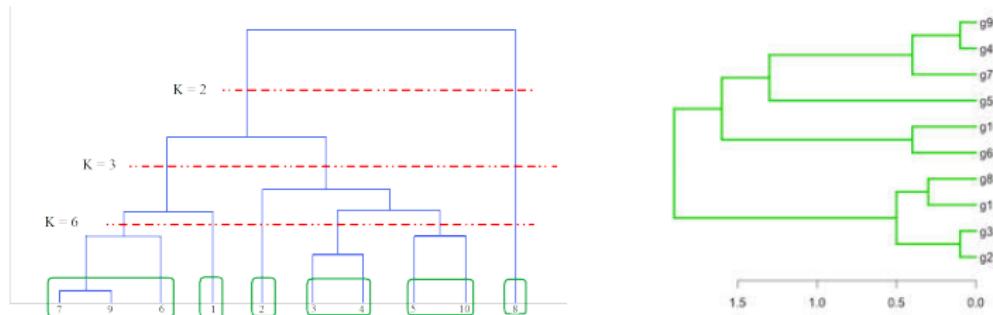
- Density-based clustering algorithms:
Expectation-Maximization.
- Hybrid clustering algorithms: AntTree, Self-organizing Trees (SoT).



Clustering algorithms: Hierarchical Clustering

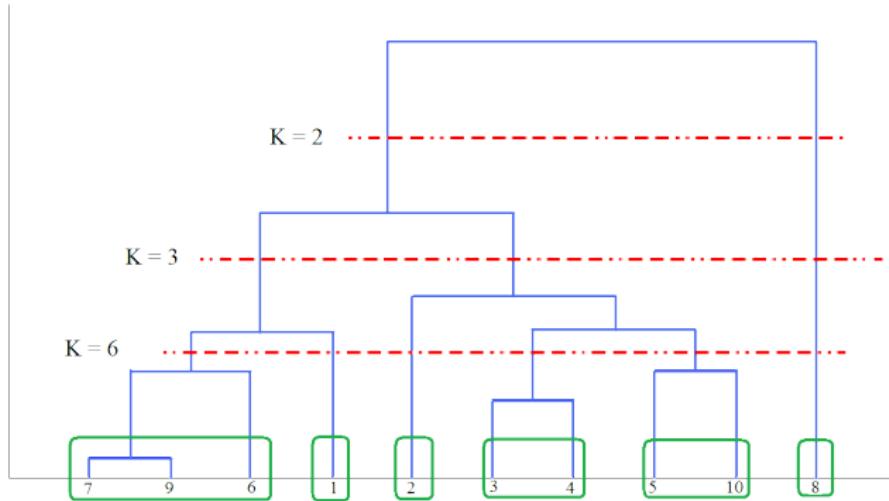
The principles are as following:

- A tree-based hierarchical taxonomy (dendrogram) is built from a set of objects.
- Each point or cluster is gradually "absorbed" by the nearest cluster.



Clustering algorithms: Hierarchical Clustering

Clustering obtained by cutting the dendrogram at a desired level:
each connected component forms a cluster.



Clustering algorithms: Hierarchical Clustering

- Agglomerative (bottom-up):
 - Start with each object being a single cluster.
 - Eventually all objects belong to the same cluster.
- Divisive (top-down):
 - Start with all objects belong to the same cluster.
 - Eventually each object forms a cluster.
- Cluster height in the dendrogram corresponds to the similarity (distance) between two clusters before the merge.

Clustering algorithms: AHC

Algorithm 1 AHC algorithm

- 1: Each objects $x_1, \dots, x_N \in \mathcal{X}$ is in its own cluster C_1, \dots, C_N
 - 2: **repeat**
 - 3: merge the nearest clusters involving C_i and C_j
 - 4: **until** only one cluster is left
-

Clustering algorithms: AHC

The nearest clusters can be variously defined as:

- Single-linkage: the link between two clusters is made by a single pair of objects that are closest to each other.

$$d(C_i, C_j) = \min_{x \in C_i, x' \in C_j} d(x, x')$$

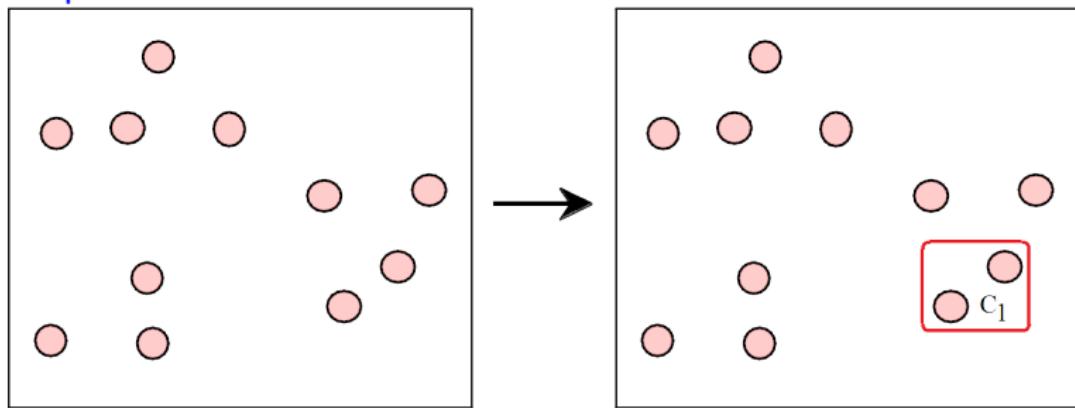
- Complete-linkage: the distance between clusters is equal to the distance between those two objects that are farthest away from each other.

$$d(C_i, C_j) = \max_{x \in C_i, x' \in C_j} d(x, x')$$

- And the other distances can be employed.

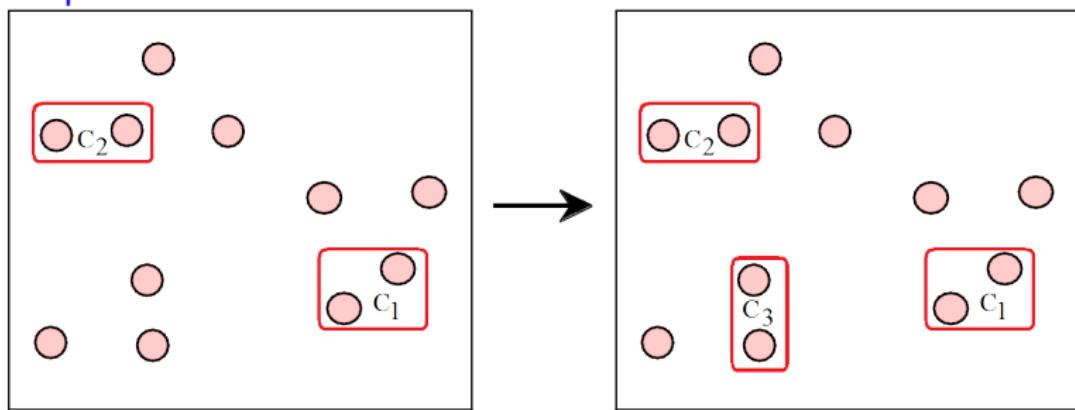
Clustering algorithms: AHC

Example



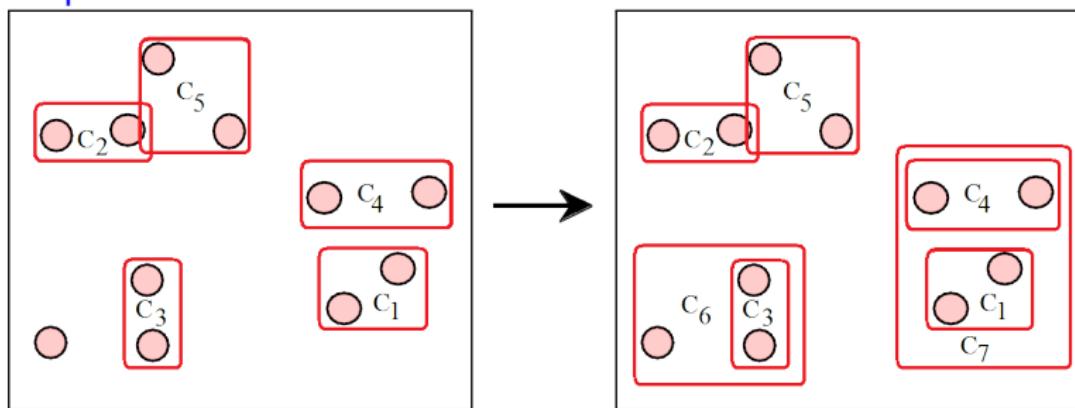
Clustering algorithms: AHC

Example



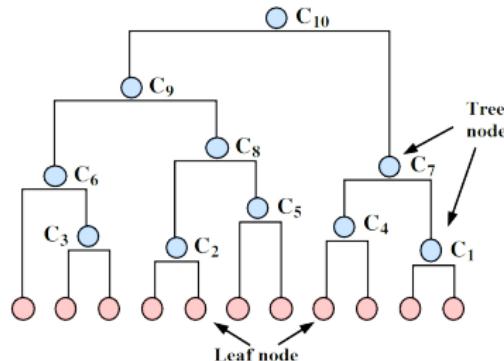
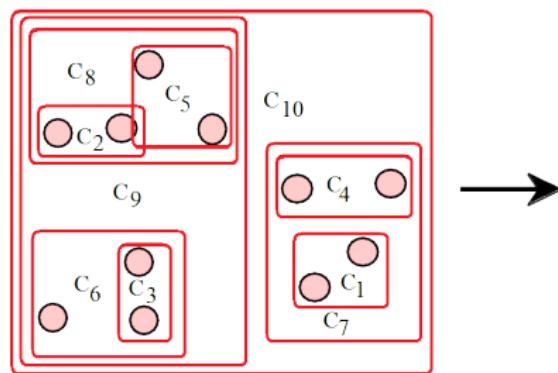
Clustering algorithms: AHC

Example



Clustering algorithms: AHC

Example



Clustering algorithms: AHC

Advantages

- does not require any input parameters in advance.
- simple visualization and easy comparison in similarity between objects.

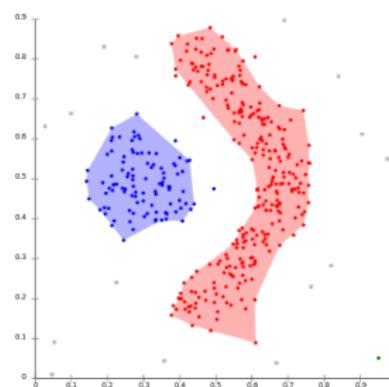
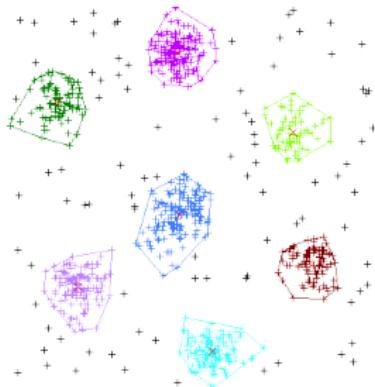
Drawbacks

- high complexity $O(n^2)$ or $O(n^3)$.
- sensible to noisy.
- does not scale well, check all the number of data before splitting.

Clustering algorithms: DBSCAN

Density-Based Spatial Clustering of Application with Noise groups points that are closely packed together (many nearby neighbors).

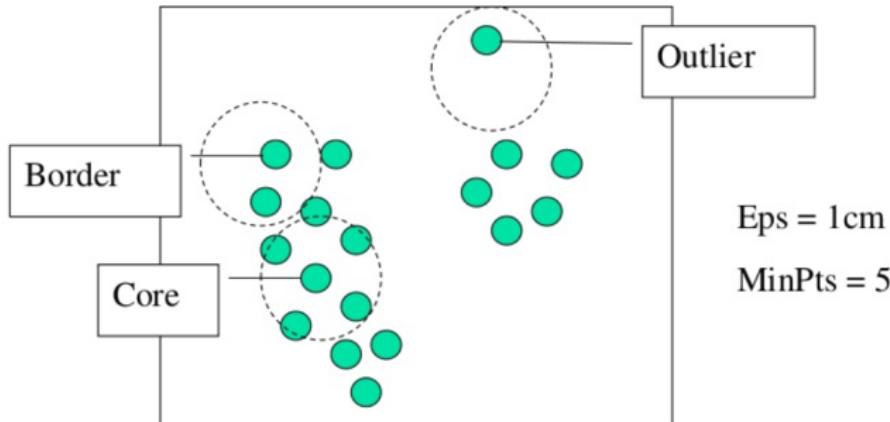
- If an object x_i is **density connect** to x_j , then x_i and x_j belong to the same cluster.
- If an object x_i is **not density connect** to any other object, x_i is considered as noise.



Clustering algorithms: DBSCAN

- **ϵ -neighborhood:** The ϵ -neighborhood of an object p is the set of objects within ϵ -distance of x_i ;
- **core object:** An object x_i is a core object if and only if there are **at least minPts** objects within the radius of ϵ .
- **noise object:** An object x_i is not a core object then it is a noise object.

Clustering algorithms: DBSCAN



Sample for different defined objects in DBSCAN

Clustering algorithms: DBSCAN

- ① Classify objects as noise or core
- ② Eliminate noise objects
- ③ Perform clustering on the core objects

```
1: current_Cluster_Label ← 1
2: for all core objects do
3:   if the core object  $x_i$  has no cluster label then
4:     current_Cluster_Label ← current_Cluster_Label + 1
5:   end if
6:   for all objects  $x_j$  in the radius of  $\epsilon$ , except  $x_i$  do
7:     if the core object  $x_j$  has no cluster label then
8:       Label the object with cluster label current_Cluster_Label
9:     end if
10:   end for
11: end for
```

Clustering algorithms: K -means

The principles are as following:

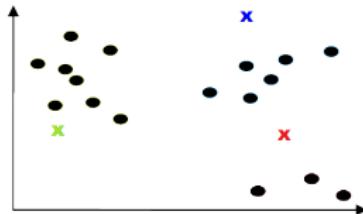
- Given that K centroids $\mathcal{W} = \{w_1, \dots, w_K\}$ are known, an object $x_i, \forall i = 1, \dots, n$ is assigned to the nearest centroid to minimize the quantization error.
- Once all the input objects have been assigned, for each cluster $C_k, \forall k = 1, \dots, K$, we estimate the new centroid w_k .
- These steps are repeated until the centroids are convergent.

Clustering algorithms: K -means

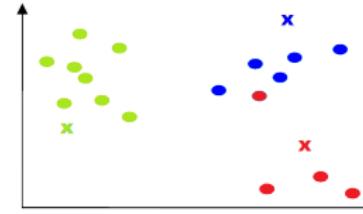
Algorithm 2 K-means algorithm

```
1: initialize randomly  $K$  prototypes / weight vectors
2: repeat
3:   for  $i = 1$  to  $n$  do
4:      $k = \arg \min_{k=1,\dots,K} \|x_i - w_k\|^2$ 
5:      $C_k = C_k \cup x_i$  {assign  $x_i$  to cluster  $C_k$ }
6:   end for
7:   for  $k = 1$  to  $K$  do
8:      $w_k = \frac{1}{n_k} \sum_{j=1}^{n_k} x_j$  {update prototype  $k$ , where  $n_k$  is the
       cardinality of cluster  $C_k$ }
9:   end for
10:  until stopping criterion has been fulfilled
```

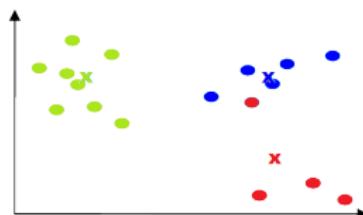
Clustering algorithms: *K*-means



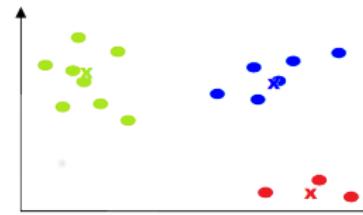
(a) Initial step



(b) Assignment step

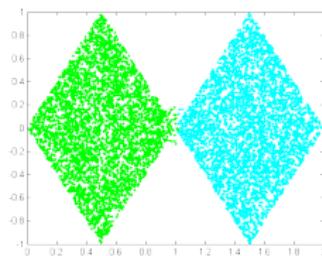


(c) Update step

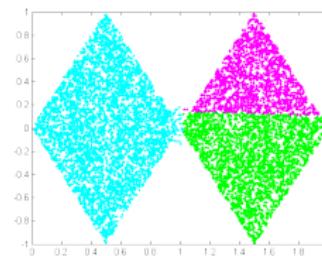


(d) Convergence.

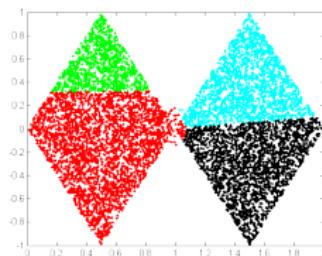
Clustering algorithms: K -means



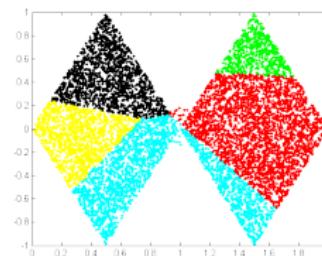
(a) $K = 2$



(b) $K = 3$



(c) $K = 4$



(d) $K = 5$

Clustering algorithms: K -means

1000 data and 5 clusters

Clustering algorithms: K -means

10000 data and 10 clusters

Clustering algorithms: K -means

Advantages:

- Simple algorithm and easy implementation.
- Low complexity $\theta(Knt)$ (t is the number of iterations).

Drawbacks:

- This algorithm depends greatly on the initialization:
 - K must be defined a priori. Which value is the best for a specific data set?
 - w_k is randomly generated in the input space, it converges to the local minimum. How to initialize centroids to obtain better result?

Clustering algorithms: K -means

To obtain a *stable* clustering result:

- Vary the values of K .
- Run the algorithm many times with random initialization of prototypes.
 - Group all the objects that are found together in the same cluster.

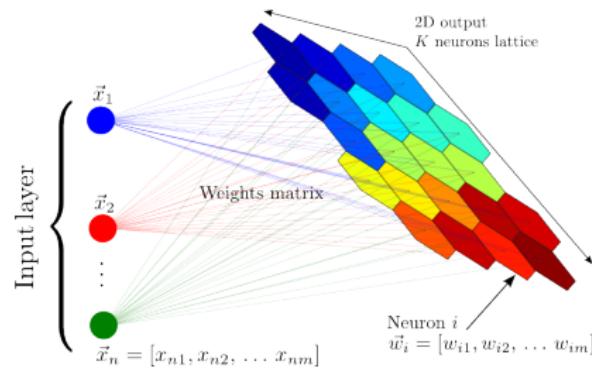
K -means variants:

- K-medians
- Self-organizing map
- Neural Network
- etc...

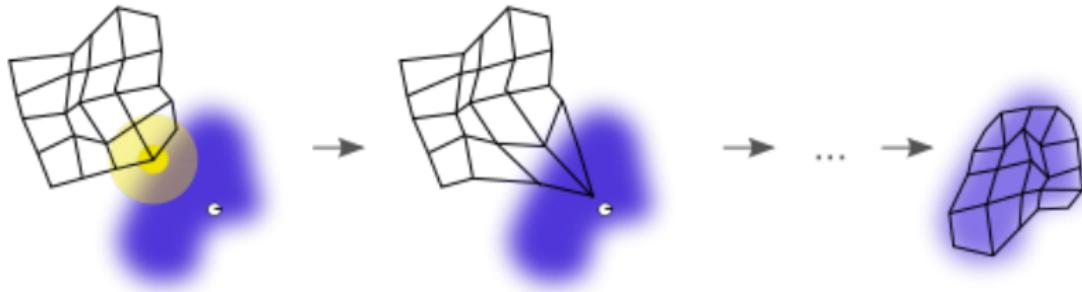
Clustering algorithms: Self-organizing Map

Introduction

- Introduce by Prof. Teuvo Kohonen in 1982
- Unsupervised neural network based on K-means
- Visualization tool for high-dimensional and complex data



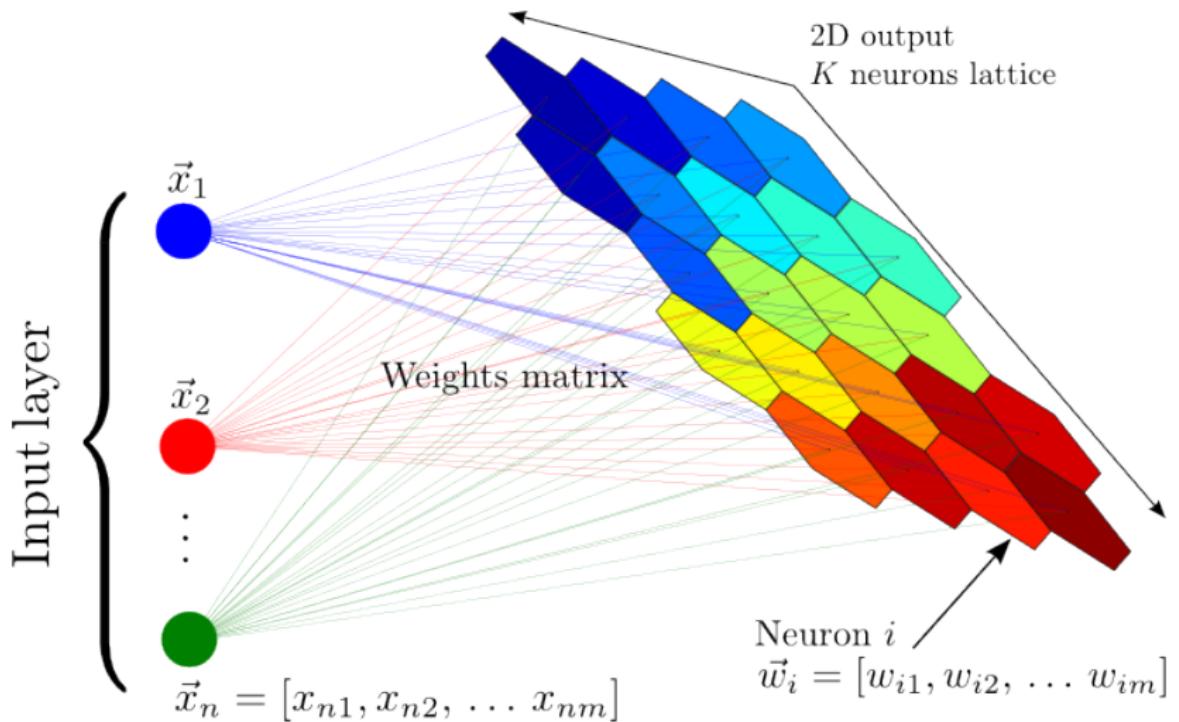
Clustering algorithms: Self-organizing Map



The idea:

- Cover the input data distribution by a topology map
- Similar data in a cluster is represented by map nodes (neurons)

Clustering algorithms: Self-organizing Map



Clustering algorithms: Self-organizing Map

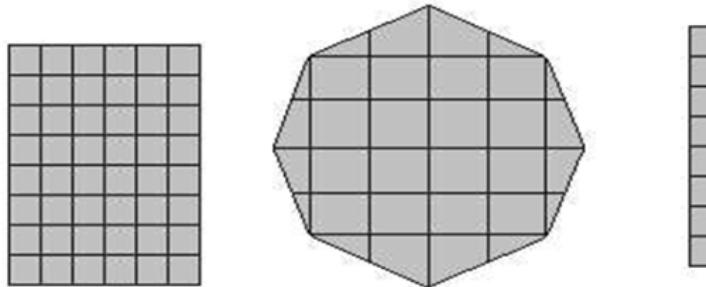
Four steps like K-means:

- Initialization: define the map topology, neurons etc...
- Assignment: assign data to the best match units (prototype)
- Update: recompute the new prototypes
- Convergence: repeat the above steps until the stopping criteria have been fulfilled

Clustering algorithms: Self-organizing Map

SOM initialization: definition of map structure

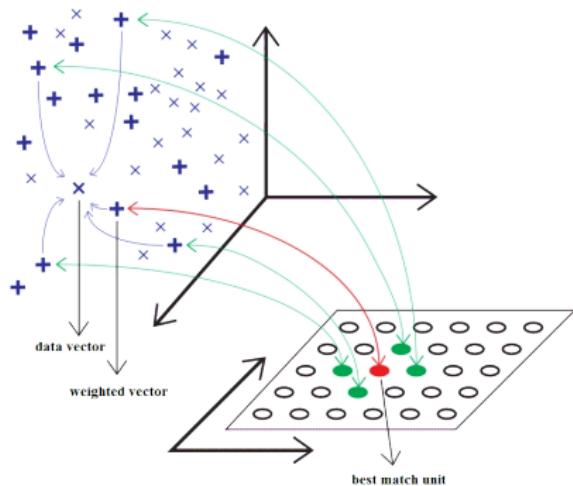
- 1-dimensional, 2-dimensional or 3-dimensional map,
- size of the map (number of neurons),
- map topology or shape: rectangle or hexa,
- a map node (neuron) associated with a weight vector (prototype),
- neighbor nodes are linked by topological links.



Clustering algorithms: Self-organizing Map

Assignment: find the nearest centroid (bmu)

$$k = \arg \min_{k=1,\dots,K} \|x_i - w_k\|^2$$



Clustering algorithms: Self-organizing Map

Update: "Winner-take-most" rule for adjusting the prototypes

- w_0 is the best match unit of the current x_i ; assigned to w_0
- $w_0 = w_0 + \alpha(x_i - w_0)$
- for all nodes w_r , the neighbors of w_0 , $w_r = w_r + \beta(x_i - w_0)$
where α, β are converging constants ($\alpha > \beta$) or neighborhood functions (kernel functions)

Clustering algorithms: Self-organizing Map

Advantages:

- Reduction of dimensionality to visualize data in a low dimensional space.
- Capable of clustering large, complex data sets.
- Neighborhood functions help the algorithm convergence.

Clustering algorithms: Self-organizing Map

Advantages:

- Reduction of dimensionality to visualize data in a low dimensional space.
- Capable of clustering large, complex data sets.
- Neighborhood functions help the algorithm convergence.

Variants:

- Neural Gas
- Growing Neural Gas
- etc...

Clustering evaluation

Each cluster C_k is associated with:

- a centroid (prototype): $w_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i$
- intra-class variance (quantization error):
$$error = \sum_{i=1}^{n_k} d(x_i, w_k)$$
- inter-class variance: $error = \sum_{k=1}^K \sum_{l \neq k; l=1}^K \sum_{j=1}^{n_l} d(x_j, w_k)$

Clustering evaluation

Internal validation:

- Davies-Bouldin index measures the correlation between two clusters:

$$DB = \frac{1}{N} \sum_{i=1}^K \max_{j, i \neq j} \frac{\mu_i + \mu_j}{d(w_i, w_j)}$$

where $\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} d(w_i, x_j)$

- Dunn index measures dense and well-separated clusters:

$$Dunn = \min_{i=1, \dots, K} \left(\min_{j=1, \dots, K; i \neq j} \left(\frac{d(w_i, w_j)}{\max_{l=1, \dots, K} \Delta(l)} \right) \right)$$

where $\Delta(l)$ measures the intra-cluster distance of cluster l

Clustering evaluation

Clustering methods group these objects into K clusters, thus two partitions to compare are defined: $\mathcal{C} = \{C_1, \dots, C_K\}$ is a random variable for data cluster assignments $\mathcal{X} = \{x_1, \dots, x_N\}$, and $Y = \{y_1, \dots, y_N\}$, where $y_l \in \mathcal{B} = \{B_1, \dots, B_L\}$ is a variable for the original labels.

$\mathcal{B} \setminus \mathcal{C}$	C_1	C_2	\cdots	C_k	\cdots	C_K	Sum
B_1	n_{11}	n_{12}	\cdots	n_{1k}	\cdots	n_{1K}	n_{B_1}
B_2	n_{21}	n_{22}	\cdots	n_{2k}	\cdots	n_{2K}	n_{B_2}
\vdots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots
B_I	n_{I1}	n_{I2}	\cdots	n_{Ik}	\cdots	n_{IK}	n_{B_I}
\vdots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots
B_L	n_{L1}	n_{L2}	\cdots	n_{Lk}	\cdots	n_{LK}	n_{B_L}
Sum	n_{C_1}	n_{C_2}		n_{C_k}		n_{C_K}	N

Clustering evaluation

External validation:

- Accuracy reflects the proportion of objects that were correctly assigned:

$$Acc = \frac{1}{N} \sum_{k=1}^K \max_{l=1,\dots,L} (n_{lk})$$

- Mutual Information measures how much information is shared between a clustering and a ground-truth classification:

$$MI = \sum_{l=1}^L \sum_{k=1}^K n_{lk} \log_2 \left(\frac{N n_{lk}}{n_{B_l} n_{C_k}} \right)$$

- Normalized Mutual Information:

$$NMI = \frac{MI}{\sqrt{\left(\sum_{l=1}^L n_{B_l} \log_2 \left(\frac{n_{B_l}}{N} \right) \right) \left(\sum_{k=1}^K n_{C_k} \log_2 \left(\frac{n_{C_k}}{N} \right) \right)}}$$

Clustering evaluation

External validation:

- Rand index computes how similar the obtained clusters are to the benchmark classifications: $\text{Rand} = \frac{N_{00} + N_{11}}{N_{00} + N_{01} + N_{10} + N_{11}}$
- Jaccard index is used to quantify the similarity between two partitions: $\text{Jaccard} = \frac{N_{11}}{N_{00} + N_{10} + N_{01}}$

where

N_{11} is the number of data pairs in the same cluster in both \mathcal{B} and \mathcal{C} ;

N_{10} is the number of data pairs in the same cluster in \mathcal{B} but not \mathcal{C} ;

N_{01} is the number of data pairs in the same cluster in \mathcal{C} but not \mathcal{B} ;

N_{00} is the number of data pairs in different clusters in both \mathcal{B} and \mathcal{C} .

Clustering evaluation

Example: 10 objects $\in \mathcal{X}$ having the original labels

$Y = \{1, 1, 1, 1, 2, 2, 2, 2, 2, 2\}$ are clustered into 3 clusters,

$Y_{new} = \{1, 1, 1, 2, 2, 2, 3, 3, 3, 3\}$. We have:

- $B_1 = \{x_1, x_2, x_3, x_4\}$ and $B_2 = \{x_5, x_6, x_7, x_8, x_9, x_{10}\}$
- $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6\}$ and $C_3 = \{x_7, x_8, x_9, x_{10}\}$

$\mathcal{B} \setminus \mathcal{C}$	C_1	C_2	C_3	Sum
B_1	3	1	0	4
B_2	0	2	4	6
Sum	3	3	4	10

Clustering evaluation

$\mathcal{B} \setminus \mathcal{C}$	C_1	C_2	C_3	Sum
B_1	3	1	0	4
B_2	0	2	4	6
Sum	3	3	4	10

- Accuracy: $Acc = \frac{1}{N} \sum_{k=1}^K \max_{l=1,\dots,L}(n_{lk}) = \frac{1}{10} \sum_{k=1}^3 \max((3, 0), (1, 2), (0, 4)) = \frac{3+2+4}{10} = 0.9$
- Mutual Information: $MI = \sum_{l=1}^L \sum_{k=1}^K n_{lk} \log_2 \left(\frac{N n_{lk}}{n_{B_l} n_{C_k}} \right) = 3 \log_2 \left(\frac{10*3}{3*4} \right) + 1 \log_2 \left(\frac{10*1}{3*4} \right) + 2 \log_2 \left(\frac{10*2}{3*6} \right) + 4 \log_2 \left(\frac{10*4}{4*6} \right) =$

Clustering evaluation

$\mathcal{B} \setminus \mathcal{C}$	C_1	C_2	C_3	Sum
B_1	3	1	0	4
B_2	0	2	4	6
Sum	3	3	4	10

- Accuracy: $Acc = \frac{1}{N} \sum_{k=1}^K \max_{l=1,\dots,L}(n_{lk}) = \frac{1}{10} \sum_{k=1}^3 \max((3, 0), (1, 2), (0, 4)) = \frac{3+2+4}{10} = 0.9$
- Mutual Information: $MI = \sum_{l=1}^L \sum_{k=1}^K n_{lk} \log_2 \left(\frac{N n_{lk}}{n_{B_l} n_{C_k}} \right) = 3 \log_2 \left(\frac{10*3}{3*4} \right) + 1 \log_2 \left(\frac{10*1}{3*4} \right) + 2 \log_2 \left(\frac{10*2}{3*6} \right) + 4 \log_2 \left(\frac{10*4}{4*6} \right) = 6.954$
- Normalized Mutual Information:
$$NMI = \frac{MI}{\sqrt{(\sum_{l=1}^L n_{B_l} \log_2 \left(\frac{n_{B_l}}{N} \right)) (\sum_{k=1}^K n_{C_k} \log_2 \left(\frac{n_{C_k}}{N} \right))}} = \frac{MI}{\sqrt{(3 \log_2 \left(\frac{3}{10} \right) + 3 \log_2 \left(\frac{3}{10} \right) + 4 \log_2 \left(\frac{4}{10} \right)) (4 \log_2 \left(\frac{4}{10} \right) + 6 \log_2 \left(\frac{6}{10} \right))}} =$$

Clustering evaluation

$\mathcal{B} \setminus \mathcal{C}$	C_1	C_2	C_3	Sum
B_1	3	1	0	4
B_2	0	2	4	6
Sum	3	3	4	10

- Accuracy: $Acc = \frac{1}{N} \sum_{k=1}^K \max_{l=1,\dots,L}(n_{lk}) = \frac{1}{10} \sum_{k=1}^3 \max((3, 0), (1, 2), (0, 4)) = \frac{3+2+4}{10} = 0.9$
- Mutual Information: $MI = \sum_{l=1}^L \sum_{k=1}^K n_{lk} \log_2 \left(\frac{N n_{lk}}{n_{B_l} n_{C_k}} \right) = 3 \log_2 \left(\frac{10*3}{3*4} \right) + 1 \log_2 \left(\frac{10*1}{3*4} \right) + 2 \log_2 \left(\frac{10*2}{3*6} \right) + 4 \log_2 \left(\frac{10*4}{4*6} \right) = 6.954$
- Normalized Mutual Information:
$$NMI = \frac{MI}{\sqrt{\left(\sum_{l=1}^L n_{B_l} \log_2 \left(\frac{n_{B_l}}{N} \right)\right) \left(\sum_{k=1}^K n_{C_k} \log_2 \left(\frac{n_{C_k}}{N} \right)\right)}} = \frac{MI}{\sqrt{(3 \log_2(\frac{3}{10}) + 3 \log_2(\frac{3}{10}) + 4 \log_2(\frac{4}{10})) (4 \log_2(\frac{4}{10}) + 6 \log_2(\frac{6}{10}))}} = \frac{6.954}{\sqrt{152.531}} = 0.563$$

Clustering evaluation

- $B_1 = \{x_1, x_2, x_3, x_4\}$ and $B_2 = \{x_5, x_6, x_7, x_8, x_9, x_{10}\}$
 - $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6\}$ and $C_3 = \{x_7, x_8, x_9, x_{10}\}$
- $N_{00} = 22$, $N_{11} = 10$, $N_{10} = 11$ and $N_{01} = 2$
- Rand index: $\text{Rand} = \frac{N_{00} + N_{11}}{\frac{N}{2}} = \frac{22+10}{22+10+11+2} = \frac{32}{45} = 0.711$
 - Jaccard index:
$$\text{Jaccard} = \frac{N_{11}}{N_{11} + N_{10} + N_{01}} = \frac{10}{10+11+2} = \frac{10}{23} = 0.434$$

Conclusion

- Clustering: unsupervised learning.
- Grouping the homogeneous objects into clusters (low intra-class variance and high inter-class variance).
- It exists many criteria to evaluate a clustering (internal and external validation).
- Algorithms are rich: AHC, K-means, EM et many more.