How to Evaluate Clustering Methods

105070004 郭家豪

105070032 潘美娟

107034037 李冠霆

106070020 何羿樺

指導教授: 謝文萍教授

Abstract:

Cluster analysis aims to group a set of object data with close similarity and separate else with different characteristics. It's a kind of exploratory data analysis used in many fields. However, cluster analysis isn't a specific algorithm. It can be achieved by various methods which might differ in their own understanding of cluster constitution and efficiency. Therefore, it's important for researchers to do cluster evaluation before conducting the analysis.

First, we are going to discuss the cluster tendency of dataset, see whether the dataset is suitable for clustering analysis. Next, we want to choose the optimal number of clusters and discuss relative methods. After we are sure about the dataset is able to do clustering and the number of clusters are optimal, we can determine different clustering method's quality and perform evaluation as our final goal is. We will introduce some different but popular clustering methods, then apply these methods on example datasets. Hence, we can do evaluation on clustering methods and have deeper understanding in practice.

1. Clustering tendency 3
1.1 Clustering introduction 3
1.2 Hopkins statistic 3
2. Numbers of Optimal Clusters, K 4
2.1 What is an "Optimal" number of K? 4
2.2 Methods 4
3. Clustering Quality 8
3.1 Internal measure 8
3.2 External measure 11
4. Clustering Methods 13
4.1 Hard vs Soft clustering 13
4.2 Seven methods 13
1. K-means、K-Medoids clustering 13
2. Hierarchical clustering 15
3. Clique clustering 19
4. DBSCAN clustering 20
5. GMM-EM clustering 23
6. Meanshift clustering 26
7. CLARANS clustering 30
5. Application and Examples 34
5.1 Dataset introduction 34
5.2 Cluster methods application 36
5.3 Cluster tendency and Numbers of k 43
5.4 Summary 49
6. Reference 50

1. Clustering tendency

1.1 Clustering Introduction

Clustering methods can return clusters in data even if the dataset isn't suitable for clustering(e.g. uniform random). If we don't evaluate the tendency of the target data before performing the analysis, algorithms might come out with a result that consists of no useful information. While conducting cluster evaluation, first we have to make sure they have similar and legal clustering tendencies.

1.2 Hopkins Statistic

Hopkins statistic reveals the probability of a certain dataset that is distributed uniformly and randomly. It tests the spatial randomness of the dataset. This statistic shows us the clustering tendency.

1.2.1 Definition

Hopkin's Statistic(**H**)=

$$\mathrm{H} = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} \omega_i^d}$$

i. **X** be the set of **n** data points with **d** dimensional.

ii. A random sample of **m** with members of $\mathbf{X_i}$ (**m**<<**n**)

iii. A random set **Y** of m uniformly randomly distributed data points.

iv. u_i, the distance of $\mathbf{y_i} \in \mathbf{Y}$ from its nearest neighbor in **X**.

ω_i,the distance of **m** number of randomly chosen $\mathbf{X_i}$, $\mathbf{X_i} \in \mathbf{X}$ from its nearest neighbor in **X.**

1.2.2 Null VS. Alternative hypothesis

H_0 : The data set D is uniformly distributed (i.e., no meaningful clusters)
H_1 : The data set D is not uniformly distributed (i.e., contains meaningful clusters)

1.2.3 Threshold

Usually, people take H=0.5 as a threshold to identify whether there are significant clusters or not. That is to say, if the value of H is smaller than this threshold, it's more unlikely to say there are significant clusters. If the value of H is much closer to 1, we can conclude that this dataset is more suitable for clustering.

2. Number of Optimal Clusters, k

2.1 What is an "Optimal" number of k?

When conducting certain clustering analysis, e.g, K-means or Expectation Maximization method(EM method), users are required to determine the "Optimal" number of the clusters. Increasing the number of clusters can reduce the total error which the model can't explain, but meanwhile it becomes harder for people to interpret the result. An ideal number of clusters can strike a balance between data compression in a single cluster and maximize the accuracy by assigning each data point to its own cluster.

Determining the number of **k** is not an easy task for human beings, unless the data is low-dimensional. The optimal number of **k** is often ambitious, but we can either make use of the domain knowledge of that field to determine the appropriate category numbers, or utilize the statistical methods below to gain the final result.

2.2 Methods

I. Direct Methods

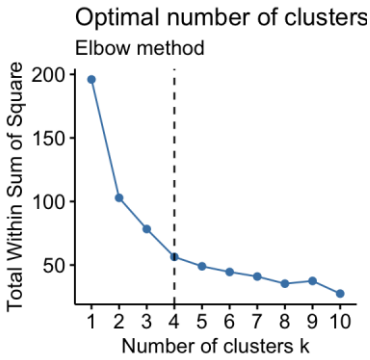
Direct method's target is to find out an optimal criterion for researchers, which refers to a relatively objective statistic or graph interpretation. There are two ways that we conduct direct method: Elbow method and Silhouette Method

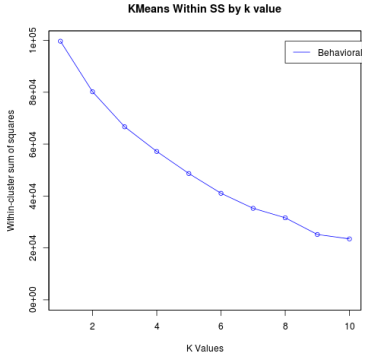
i. Elbow Method

$$D_k = \sum_{i=1}^{K} \sum dist(x, c_i)^2$$

When conducting cluster analysis, we want the total within-cluster sum of squares(WSS) to be as small as possible. A cluster with a large variation seldom gives useful information. Meanwhile, we don't want to see the number of **k** become too big and not able to explain, even though the clustering result can explain most of WSS.

Elbow method aims to find an optimal **k** so that adding another cluster won't give better modeling results of data. That certain point of k will show an angle in the scree plot, due to the drop of its marginal gain. The number of clusters is chosen at this point, hence the "elbow method" is named due to its shape. (See **Graph 2.2.1**)





(Graph 2.2.1) **(Graph 2.2.2)**

However, the "elbow" point isn't easy to determine in every case, it's relatively subjective and often ambiguous. If the slope is smooth and tidy, we can't really tell there is a marginal point on the graph, thus the elbow method won't function well in this case. (See **Graph 2.2.2**)

ii. Silhouette Method

Silhouette method tests the quality of the clustering. "Silhouette" refers to a method of interpretation and validation of consistency within clusters. It compares the similarity of a datapoint with its cluster and the separation of the same point with other clusters. If the number of clusters **k** is appropriate, the result may show that the object point is well matched with its cluster, but poorly matched with other neighbor clusters. Silhouette method requires a Silhouette coefficient to gain its result, we can get Silhouette coefficient by the following procedure:

(1)

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

For data point i, i belongs to its cluster $\mathbf{C_i}$.

d(i, j) is the distance between **i** and other points in the same cluster.

We can interpret **a(i)** as a measure of how well **i** is assigned to its cluster.

(2)

$$\mathrm{b}(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

For data point i, C_k are clusters which datapoint i doesn't belong to.

d(i, j) is the distance between i and other points in other clusters.

We can interpret **b(i)** to be the smallest mean distance between data i and other cluster data points.

(3)

We now define a *silhouette* (value) of one data **i**:

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

$\mathrm{s}(i) = 0$, if $|C_i|=1$

Therefore, we can get:

$$-1 \leq \mathrm{s}(i) \leq 1$$

(4)

Introduce Silhouette coefficient for the maximum value of the mean over all data of the entire dataset :

$$\mathrm{SC} = \min_k \tilde{s}(k)$$

Where $\tilde{s}(k)$ represents the mean $S(i)$ over all data of the entire dataset for a specific number of clusters **k**.

We can look through the Silhouette coefficient to find out among different numbers of **k**, which performs better clustering results. If the Silhouette value is close to 1, it's well clustered. On the contrary, if it's near -1, then the datapoint I seem better to cluster in different groups, hence this is a bad result for clustering.

II. Statistical testing Method

Statistical testing methods consists of comparing evidence against null hypothesis. The example for statistical testing method is: Gap statistic method.

i. Gap statistic Method

The Gap statistic compares the total within cluster variation (WSS) for different values of k with their expected values under null reference distribution of the data. The optimal clusters will have an estimated value that maximizes the gap statistic. This implies that the cluster model is far away from the random uniform distribution of points. We can get Gap statistic by the following procedure:

(1)

Cluster the observed data, varying the number of clusters from k = 1, …, *kmax*, and compute the corresponding total within cluster variation *Wk.*

(2)

Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, …, *kmax*, and compute the corresponding total within intra-cluster variation *Wkb*.

(3)

Compute the estimated gap statistic as the deviation of the observed *Wk* value from its expected value *Wkb* under the null hypothesis:

Gap(k)= $\frac{1}{B}\sum_{b=1}^{B} log\ (w_{kb}^{*}) - log(w_k)$

(4) Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1:

$Gap(k) \geq Gap(k+1) - sk+1.$

However, if two or more clusters are close together while others are far away from them, the gap statistic may underestimate its effect. In contrast, if all the clusters are close with each other, the gap statistic may overestimate its value. Since the gap statistic method won't always generate ideal results, it's better to run it more times and take average to make the evaluation more precise.

3. Clustering Quality

We already make sure that the dataset has a good clustering tendency and choose an optimal number of **k**. Next, we want to evaluate the performance for each clustering algorithm.

For supervised learning tasks, it's easier to evaluate the quality of the methods, since there are already labels for examples, too many ways of evaluation can be done by data comparison. Clustering methods deal with unlabeled data, being unsupervised learning content with no ground truth supported. However, there are still methods that can be used to evaluate the quality of clustering models. Those methods might give us the insight of how clusters change with different algorithms, and how well the cluster performs internally or externally with other groups.

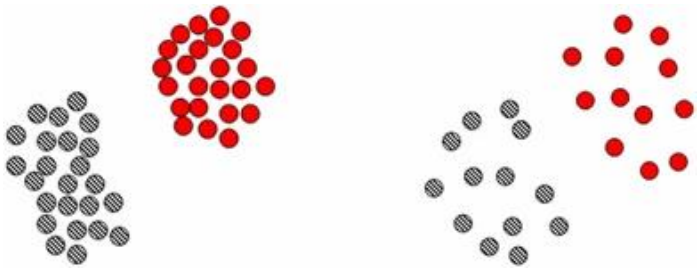
3.1 Internal Measure

When conducting clustering analysis, we split the original data into groups of objects. We want the objects in the same clusters to be as similar as possible, and the other object groups with itself should be highly distinct. That is to say, we want the average distance within clusters to be smaller as possible, while average distance between different clusters being larger and better.

Mostly, we utilize these two statistics to evaluate clustering quality:

I. Compactness

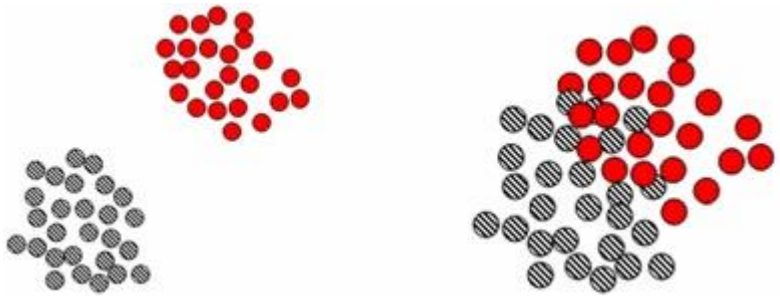
Compactness measures how closely data points are grouped in a cluster. Grouped points in the cluster are supposed to be related to each other, by sharing a common feature which reflects a meaningful pattern in practice. Compactness is normally based on distances between in-cluster points. The popular way of calculating the compactness is through variance, i.e., average distance to the mean, to estimate how objects are bonded together with its mean as its center. A small variance will indicate a high compactness. From **Graph 3.1**, we can see clusters on the left side have higher compactness, and clusters on the left side have a low compactness.



(Graph 3.1)

II. Separation

Separation measures how different the clusters are from each other. A distinct cluster that is far from the others corresponds to a unique pattern. Similar to compactness, the distances between objects are widely used to measure separation, e.g., pairwise distances between cluster centers, or pairwise minimum distances between objects in different clusters. Separation is an inter-cluster criterion representing the relation between clusters. From **Graph 3.2**, we can see the two clusters on the left side are far from each other, having a relatively high separation. But on the right side, we can see the two clusters already overlap with others, they have a low separation for sure.



(Graph 3.2)

Most of the indices used for internal clustering validation combines compactness and separation as below:

Index= (α * Separation) ÷ (β * Compactness)

Let's give some examples for indices that are good at evaluating cluster qualities:

1. Silhouette coefficient

$$S = \frac{1}{NC}\sum_{i}\left(\frac{1}{n_i}\sum_{x\in C_i}\frac{b(x)-a(x)}{\max[b(x),a(x)]}\right)$$

a(x), b(x), is already defined in 2.2 ii. (Silhouette). NC is the number of clusters.

As mentioned before in 2.2 ii., silhouette coefficient measures how well an observation is clustered, and it estimates the average distance between clusters. $a(x)$ relates to numerated compactness, and $b(x)$ measures the separation with the average distance of objects to other clusters. A well-clustered model will have its silhouette coefficient close to 1, when a small value of it means the observation lies between two or more clusters. If the silhouette coefficient of a model is negative, it probably means the data points are placed in wrong clusters.

2. Dunn index

$$\mathbf{D} = \frac{\min i \min j \left(\min_{x\in C_i, y\in C_j} d(x,y)\right)}{\max k \left(\max_{x,y\in C_k} d(x,y)\right)}$$

Dunn index (D) uses the minimum pairwise distance between points in different clusters as the inter-cluster separation and the maximum diameter among all clusters as the intra-cluster compactness.

If the dataset is well clustered, the model having small diameters of the clusters and large distance between clusters, shall have a maximized Dunn index.

3.2 External Measure

External measures can be used to select the suitable clustering algorithms for the given dataset. It compares cluster methods performance to an external reference. These references are: Rand index and Meila's VI.

1. Rand Index

Rand index is a measure of the similarity between two clustering groups, it is also said to be the accuracy of the clustering methods. Brief definition is as follow:

Given a set of **n** elements S and two partitions of S to compare:

$\mathbf{x} = \{X_1, X_2, \dots, X_r\}$ **with** r **groups.**

$\mathbf{Y} = \{Y_1, Y_2, \dots, Y_k\}$ **with** k **groups.**

Let

a= the number of pairs of elements in **S** that are in the **same** subset in $\mathbf{x}$ and in the **same** subset in $\mathbf{Y}$**.**

b= the number of pairs of elements in **S** that are in the **different** subset in $\mathbf{x}$ and in the **different** subset in $\mathbf{Y}$**.**

c= the number of pairs of elements in **S** that are in the **same** subset in $\mathbf{x}$ and in the **different** subset in $\mathbf{Y}$**.**

d= the number of pairs of elements in **S** that are in the **different** subset in $\mathbf{x}$ and in the **same** subset in $\mathbf{Y}$**.**

Then, the **Rand index (R)** is:

$$\mathrm{R} = \frac{a+b}{a+b+c+d} = \frac{a+b}{c_2^n}$$

It can also perform in this formula:

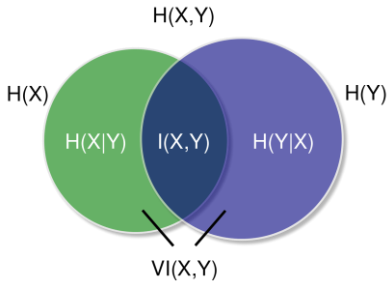
$$\mathrm{R} = \frac{TP+TN}{Tp+FP+FN+TN}$$

Since the denominator is the total number of pairs, the Rand index represents the frequency of occurrence of agreements over the total pairs, or the probability that X and Y will agree on a randomly chosen pair.

Rand index has a value between 0 and 1, with 0 indicating that the two data clustering do not agree on any pair of points and 1 indicating that the data clustering are exactly the same.

2. Meila's VI

Meila's variance index, or **shared information distance**, is a measure of the distance between two clusters. It is a simple linear expression involving mutual information.



Information diagram illustrating the relation between information entropies, mutual information and variation of information.

The definition of Meila's VI is:

Given a set of **n** elements A and two partitions of S to compare:

$\mathbf{X} = \{X_1, X_2, \dots, X_r\}$ **with** r **groups.**

$\mathbf{Y} = \{Y_1, Y_2, \dots, Y_k\}$ **with** k **groups.**

Let

$$\mathrm{n} = \sum_i |X_i| = \sum_j |Y_j| = |A|$$

$$p_i = \frac{|X_i|}{n}, q_i = \frac{|Y_j|}{n}$$

$$r_{ij} = |X_i \cap Y_j|/n$$

Then the variation of information between the two partitions is:

$$\mathrm{VI(X;Y)} = -\sum_{i,j} r_{ij}[\log\left(\frac{r_{ij}}{p_i}\right) + \log(\frac{r_{ij}}{q_j})]$$

4. Clustering Methods

4.1 Hard vs Soft Clustering

Hard clustering means that each sample point must be assigned to a cluster in an "either/or" manner. The counterpart of hard clustering is soft clustering. For each sample point, the soft clustering algorithm calculates the probability that the point belongs to different clusters.

4.2 Seven methods

Next, we will introduce seven common clustering methods.

1. K-means、K-Medoids Clustering

I. Introduction

K-means clustering is a hard clustering algorithm that assumes that there are N observations in an n-dimensional space. Mathematically, hard clustering means that K clusters divide the n-dimensional space into K mutually exclusive regions, and each observation belongs to one and only one of these K clusters. Let $\boldsymbol{S_k}$ represent cluster k, k belongs to {1, ..., K}, and the following relation is satisfied between different clusters $\boldsymbol{S_k}$.

$$\boldsymbol{S_k} \cap \boldsymbol{S_{k'}} = \emptyset, \forall \boldsymbol{k} \neq \boldsymbol{k'}$$

$$\boldsymbol{S} = \cup_{\boldsymbol{k=1}}^{\boldsymbol{K}} \boldsymbol{S_k} = \{\boldsymbol{1}, \dots, \boldsymbol{N}\}$$

The goal of the algorithm is to find the optimal part S = {$\boldsymbol{S_1}$, ..., $\boldsymbol{S_k}$} with respect to the sample space for a given number of clusters K, such that the within-cluster variation is minimized. The within-cluster variation is defined as the sum of the squares of the distances from each point in the cluster to the center of mass of the cluster, so the within-cluster variation is also called within-cluster sum of squares. Since the center of mass represents the mean value, this is also the meaning of the word mean in the name of K-means clusters. The method can be represented as:

$$\min \sum_{k=1}^{K} \sum_{X_i \in \boldsymbol{S_k}} \|X_i - \mu_k\|$$

II. Algorithm implementation steps：K-means vs K-medoids

K-means
1. randomly select the values of K prime 2. calculate the distance from each point to the prime 3. divide the class of points into the nearest prime, forming K clusters 4. according to the classified clusters, recalculate the prime in each cluster (the average value of each point) 5. repeat iteration 2-4 steps until the number of iterations is satisfied or the error is less than the specified value
K-medoids
1. randomly select the values of K prime (the prime must be the value of some sample points, not arbitrary values) 2. calculate the distance from each point to the prime value of K-medoids 3. divide the class of points into the nearest prime, forming K clusters 4. according to the classified clusters, recalculate the prime within each cluster: 4.1 Calculate the Manhattan distance and (absolute error) from all sample points in the cluster to one of the sample points 4.2 Select the sample point that minimizes the absolute error of the cluster as the center of mass

5. Repeat iterations 2-4 steps until the number of iterations is satisfied or the error is less than the specified value

The above shows the difference between the two: the center of mass of k-means is the average of the sample points, which may be a point that does not exist in the sample. The center of mass of k-medoids must be the value of a sample point.

	K-means	K-medoids
Center Point	Virtual Points	Actual Sample Points
New center point determination method	Intra-group sample average	Intra-cluster distance and minimum

III. Comparison of advantages and disadvantages

i. k-medoids runs slower and the time complexity of the step to calculate the center of mass is O(n^2) because he must calculate the distance between any two points. While k-means only needs to be averaged.

ii. k-medoids is more robust to noise. Example: When a cluster sample points are only a few, such as (1,1) (1,2) (2,1) (100,100). Where (100,100) is the noise. If we follow k-means the center of mass will be roughly in the middle of (1,1)(100,100), which is obviously not what we want. At this point k-medoids can avoid this situation, he will choose a sample point in (1,1) (1,2) (2,1) (100,100) so that the absolute error of the cluster is the smallest, the calculation can be seen in the first three points must be selected.

iii. Although k-medoids also has advantages, but it can only work on small samples. When the sample is large, the speed is too slow, and when there are many samples, a few noise on the k-means center of mass effect is not as heavy as imagined, so the application of k-means is obviously much more than k-medoids.

2. Hierarchical Clustering

I. Introduction

Hierarchical clustering is a hierarchical approach that splits or aggregates data repeatedly to produce a target set of clusters. There are two common hierarchical clustering methods:

i. **Agglomerative:** It is a "bottom-up" approach, which means that the basic components or solutions that may be needed to solve the problem are prepared first, and then these basic components are assembled to obtain the whole from small to large. Therefore, in the hierarchical grouping method, each data point is considered as an individual and is aggregated one by one.

ii. **Divisive:** It is a "top-down" method, in which we first have an overall concept of the problem, then gradually add details, and finally make the outline of the whole more and more clear. In this method, the whole data set is first considered as one, and then divided one by one.

II. Algorithm implementation steps

Step 1: Calculate the clustering of individual points between samples
Step 2: Combine the closest ones into a new group of sample points
Step 3: Repeat Step 1 and Step 2 until all samples become a group
Step 4: Cut according to the distance and determine the final number of clusters

III. Calculation of the distance between clusters

The distance between the sample points is well defined. Generally, we use the closest distance between two points, i.e., the Euclidean distance. Of course, we can also use other distance calculation methods such as Manhattan distance or Ming's distance. However, when the sample points are combined into a cluster, how to calculate the distance between clusters? When calculating the distance between groups, we can also define the distance between groups according to our needs. Here we provide the following methods:

A. Single linkage

Single linkage is selected from two clusters and the distance between the two points represents the distance between the two clusters.

$$d\left(Cluster_i, Cluster_j\right) = \min_{a \in Cluster_i, b \in Cluster_j} d(a, b)$$

B. Maximum (Complete) linkage

Maximum (Complete) linkage is selected from two clusters and the distance between the two points represents the distance between the two clusters.

$$d(Cluster_i, Cluster_j) = \max_{a \in Cluster_i, b \in Cluster_j} d(a, b)$$

C. Average linkage

Average linkage is the average distance between all sample points in two clusters.

$$d(Cluster_i, Cluster_j) = \sum_{a \in Cluster_i, b \in Cluster_j} \frac{d(a, b)}{|Cluster_i|| \ Cluster_j|}$$

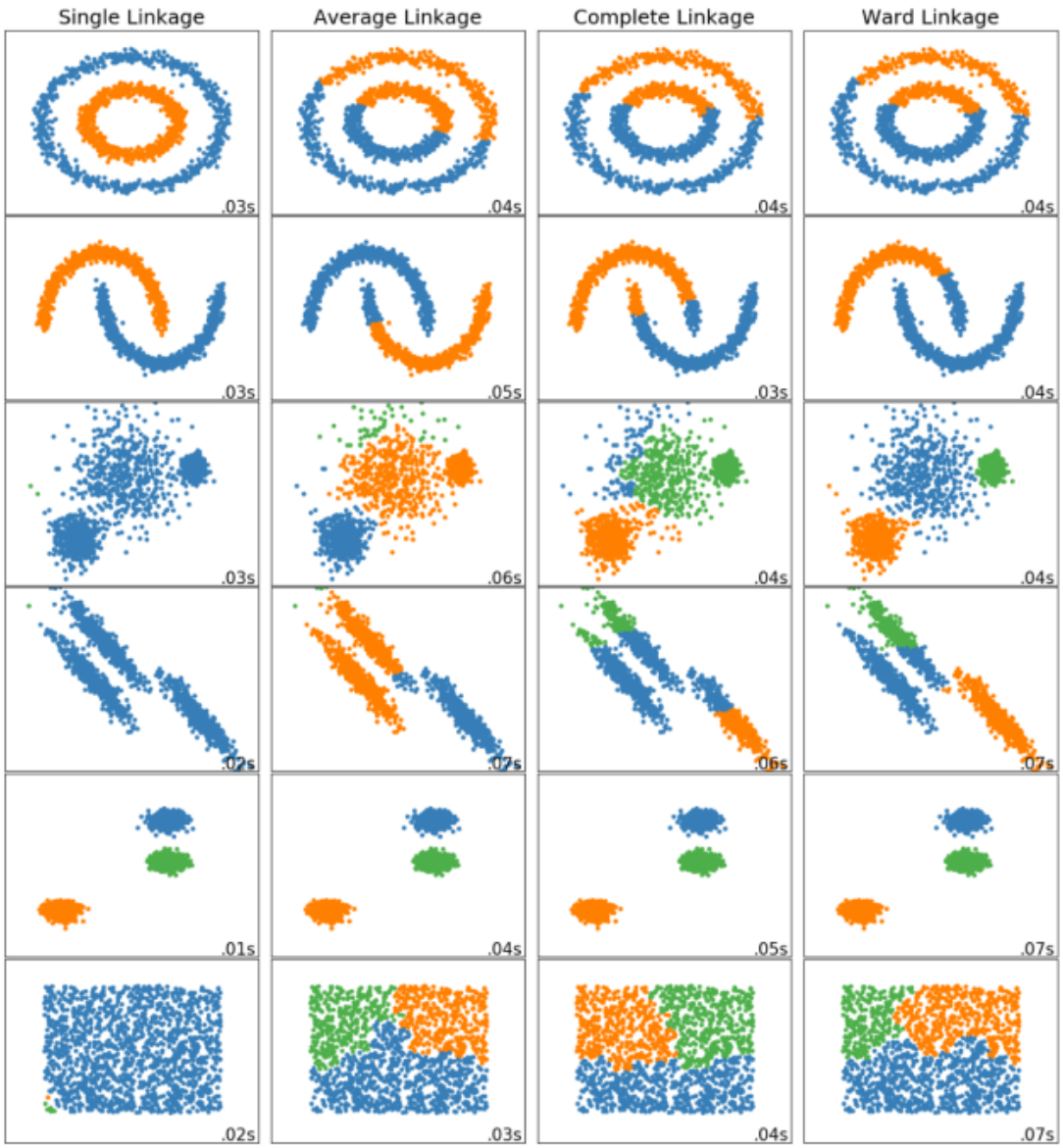
D. Ward's method (Least Square)

Main idea: The sum of squares of sample distances is smaller for intra-cluster samples and larger for inter-cluster samples.

Practice: After merging two groups, examine the sum of squared distances between each point and the center of the merged group. (in the process, the two groups with the smallest increase in the sum of squared distances will be selected for merging)

$$d(Cluster_i, Cluster_j) = \sum_{a \in Cluster_i, b \in Cluster_j} \|a - \mu Cluster_i \cup \ Cluster_j\|$$

The following figure **Figure 4.1** shows the results of classifying different types of sample data under different group and cluster distance calculation methods:



(**Figure 4.1**)

IV. Advantages and Disadvantages

The biggest advantage of hierarchical clustering is that it is quite easy to understand, and when we construct a complete tree classification, we can easily decide how many clusters we want to divide into. Moreover, in this cluster structure, we do not necessarily need to have the coordinates of the samples, even if we only have the distance between samples, we can still do the clustering very well, so it is very flexible.

However, from the above calculation process, we can find that the computation of this binning method is very large. Therefore, this method is only applicable to small samples of data.

3. Clique Clustering

I. Introduction

The CLIQUE algorithm is a grid-based spatial clustering algorithm, but it also combines very well with density-based clustering algorithms, so that it can both discover arbitrarily shaped clusters and handle larger multidimensional data as grid-based algorithms do.

The CLIQUE algorithm divides each dimension into non-overlapping communities, thus dividing the entire embedding space of data objects into cells, and it uses a density threshold to identify dense units, and a cell is dense if the objects mapped to it exceed the density threshold.

II. Algorithm implementation steps

The algorithm requires two parameters: one is the grid step size and the second is the density threshold. The grid step determines the division of the space, while the density threshold is used to define the dense grid.

(1) First, all grids are scanned. When the first dense grid is found, the expansion starts with that grid. The expansion principle is that if a grid is adjacent to a grid in a known dense region and is itself dense, the grid is added to that secret region until no more such grid is found. (Dense grid merging)

(2) The algorithm then continues to scan the grid and repeats the above process until all grids are traversed. to automatically discover the highest dimensional subspaces in which high-density clustering exists and is insensitive to the input order of the tuples, without assuming any canonical data distribution, it scales linearly with the size of the input data and has good scalability when the dimensionality of the data increases.

III. Advantages and Disadvantages

Advantages :

(1) Given the division of each attribute, a single data scan determines the grid cells and the count of grid cells for each object.

(2) Although the potential number of grid cells may be high, only a grid needs to be created for non-empty cells.

(3) The time complexity and space complexity of assigning each object to a cell and calculating the density of each cell is O(m), and the whole clustering process is very efficient.

Disadvantages :

(1) Like most density-based clustering algorithms, grid-based clustering is very dependent on the choice of a density threshold. (Too high and clusters may be lost. Too low and clusters that should be separated may be merged)

(2) If there are clusters of different densities and noise, it may not be possible to find values that fit all parts of the data space.

(3) As the dimensionality increases, the number of grid cells increases rapidly (exponential growth). That is, grid-based clustering tends to work poorly for high-dimensional data.

4. DBSCAN Clustering

I. Introduction

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm. DBSCAN defines clusters as the maximum set of densely connected points, which can divide regions with sufficient density into clusters and find clusters of arbitrary shapes in a spatial database of noise.

DBSCAN is based on a set of neighborhoods to describe the closeness of the sample set, and the parameter $(\epsilon, MinPts)$. Where ϵ describes the neighborhood distance threshold (radius) of a particular data point, $MinPts$ describes the minimum number of data points in the neighborhood with radius ϵ. The following are the definitions related to density clustering (assuming the sample set is $D = \{x_1, x_2, \dots, x_m\}$).

i. ϵ- Neighborhood:

For $x_j \in D$, the ϵ - Neighborhood contains the subsample set of the sample set$D$ whose distances from x_j are smaller than ϵ. That is, $N_\epsilon(x_j) = \{x_i \in D | distance(x_i, x_j) \leq \epsilon\}$, the number of this set is denoted as $|N_\epsilon(x_j)|$.

ii. Core object:

For any sample $x_j \in D$, if its ϵ - Neighborhood corresponds to $N_\epsilon(x_j)$ contains at least $MinPts$ samples, i.e. if$|N_\epsilon(x_j)| \geq MinPts$, then x_j is the core object.

iii. Density direct:

If x_i lies in the ϵ-neighborhood of x_j, and x_j is the core object, then x_i is said to be density direct from x_j. The opposite is not necessarily true, i.e., then x_j cannot be said to be density-directed by x_i, unless x_i is also a core object, i.e., density-directed does not satisfy the symmetry.

iv. Density reachable:

For x_i and x_j, x_j is said to be density reachable by x_i if there exists a sequence of sample samples $p_1, p_2, \ldots, p_T$ that satisfies $p_1 = x_i$, $p_T = x_j$, and p_{t+1} is directly reachable by p_t density. That is, the density reachable satisfies the transferability. At this point, the transmission samples $p_1, p_2, \ldots, p_{T-1}$ in the sequence are all core objects, because only the core objects can make the other samples density reachable directly. Density reachable also does not satisfy symmetry, and this can be derived from the asymmetry of density direct.

v. Density connected:

x_i and x_j are said to be density connected if there exists a core object sample x_k such that both x_i and x_j are reached by the density of x_k. The density connectivity relationship satisfies symmetry.

The definition of clustering in DBSCAN is simple: the set of samples connected by the maximum density derived from the density reachability relation is a class, or a cluster, for our final clustering.

So how can we find such a collection of cluster samples? It arbitrarily selects a core object without category as a seed, and then finds the set of samples for which all this core object can be densely reachable, i.e., a clustering cluster. Then it continues to select another core object without a category to find the set of samples that are densely reachable, thus obtaining another cluster. Keep running until all core objects have categories.

II. Algorithm implementation steps

i. **Parameter setting**: Determine the distance (radius ϵ) and the minimum number of points (threshold).

ii. Select any sample as the center point and draw a circle with the radius set in step 1. If the number of samples in the circle is greater than the threshold, this sample is the core point and the marker can reach any point in the circle.

iii. Repeat step 2 for each sample until all samples pass the center point.

iv. **Grouping**: The sample points that are connected (bi-directional reachable) are grouped into one group, while other outlier points can be grouped into different groups by examining whether they are single reachable or not.

III. Three additional issues to consider

1. **Anomaly problem**: Some anomalous sample points or a small number of sample points that stray outside the cluster, which are not in any of the core objects in the surrounding, in DBSCAN, we generally mark these sample points as noise points.

2. **Distance metric problem**: i.e., how to calculate the distance between a certain sample and the core object sample. In DBSCAN, the nearest neighbor idea is generally used, and a certain distance metric is used to measure the sample distance, such as Euclidean distance, Manhattan distance, etc.

3. **Data point priority assignment problem**: For example, some samples may have a distance to both core objects that is less thanϵ. In general, at this point, DBSCAN uses first come first served, and the category cluster that clusters first will mark this sample as its category. This means that the algorithm of DBSCAN is not a completely stable algorithm.

IV. Advantages and Disadvantages

The main benefits of DBSCAN are:

1. It is not affected by polarities. Therefore, the method is based on density classification, so the extremes can form their own group.

2. No need to choose the sample population in advance, it will be decided automatically in the model.

The main drawbacks of DBSCAN are:

1. Dimensional disaster. If the dimensionality is too high, a very large number of samples are required to achieve better prediction results.

2. If the data density varies greatly, the effect will be poor.

5. GMM-EM Clustering

I. Introduction

GMM and k-means are actually very similar, the difference is only that for GMM, we introduce probability. the process of GMM learning is to train several probability distributions, the so-called mixed Gaussian model means to estimate the probability density distribution of the sample, and the estimated model is the weighted sum of several Gaussian models (specifically several to be established before the model training). Each Gaussian model represents a class (a Cluster). By projecting the data in the sample on each of the Gaussian models, we obtain the probability on each class separately. Then we can select

the class with the highest probability as the judgment result.

From the point of view of the central limit theorem, it is reasonable to assume the mixture model as Gaussian, but of course, it can be defined as Mixture Model of any distribution according to the actual data, but the definition as Gaussian has some computational convenience, in addition, theoretically, it is possible to approximate any probability distribution by GMM by increasing the number of models

II. Calculation of GMM-EM

The hybrid Gaussian model is defined as:

$$p(x) = \sum_{k=1}^{K} \pi_k p(x|k)$$

Where K is the number of models, π k is the weight of the kth Gaussian, then is the probability density function of the kth Gaussian with mean μ k and variance σ k. Our estimate of this probability density is to require π k, μ k and σ k for each variable. When the expressions are derived, the result of each term of the summation equation represents the probability that sample x belongs to each class, respectively.

When doing parameter estimation, the method often used is maximum likelihood. The maximum likelihood method is to maximize the probability value of the sample points on the estimated probability density function. Since the probability values are generally small, the result of this concatenation is very small when N is large, which can easily cause floating point underflow. So we usually take log and rewrite the objective as

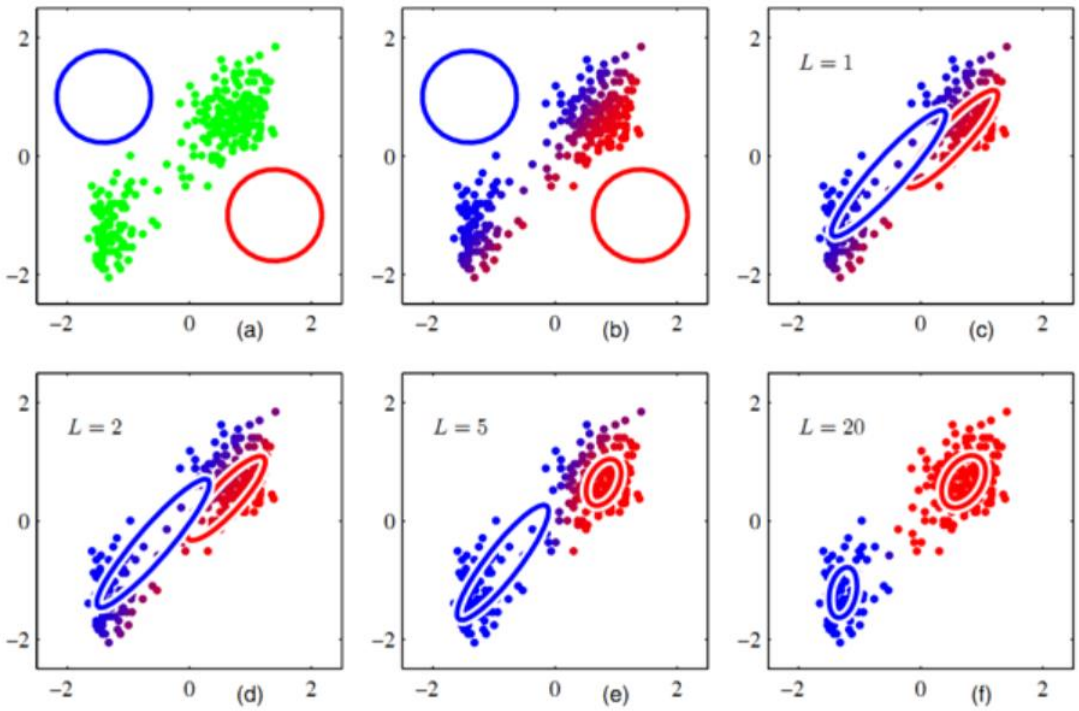
$$max \sum_{i=1}^{N} \log p(x_i)$$

That is, maximizing the log-likelyhood function, the full form is then:

$$max \sum_{i=1}^{K} \log(\sum_{k=1}^{K} \pi_k N(x_i|\, \mu_k, \sigma_k))$$

Generally when used to do parameter estimation, we find the extreme value by taking derivatives of the variables to be solved. In the above equation, the log function has a summation, and the system of equations will be very complicated if you want to calculate it by taking derivatives, so we are not good to consider

solving it by that method (there is no closed solution). A possible solution is the EM algorithm - the solution is divided into two steps: the first step is to estimate the weights of each Gaussian model, assuming we know the parameters of each Gaussian model (either by initializing one, or based on the results of the previous iteration); the second step is to go back and determine the parameters of the Gaussian model based on the estimated weights. These two steps are repeated until the fluctuations are small and the extremum is approximated (note that this is an extremum not an optimum, and the EM algorithm will fall into a local optimum).



III. Algorithm implementation steps

A. Initialize the K multivariate Gaussian distributions and their weights.

B. Estimate the posterior probability of each sample generated by each component according to Bayes' theorem (step E in the EM method)

C. Update the mean vector, see Eq., covariance matrix Eq., and weights according to the definition of mean, covariance, and the posterior probabilities derived in step 2 (step M in the EM method)

D. Repeat steps 2 to 3 until the increase in the likelihood function has been less than the convergence threshold, or the maximum number of iterations is

reached.

E. For each sample point, calculate its posterior probability of belonging to each cluster according to Bayes' theorem, and classify the sample into the cluster with the highest posterior probability.

IV. Advantages and Disadvantages

Pros: The advantage of GMM is that after projection the sample points are not given a definite classification label, but the probability of each class, which is an important piece of information. GMM can be used not only for clustering but also for probability density estimation.

Disadvantages: When there are not enough points in each mixture model, it becomes difficult to estimate the covariance, and the algorithm will scatter and find solutions with infinite likelihood function values unless the covariance is artificially regularized. The computational effort of each iteration of GMM is larger than k-means. The solution of GMM is based on the EM algorithm, so it is possible to fall into local extremes, which is very relevant to the selection of initial values. It is very relevant

6. Meanshift Clustering

I. Introduction

Mean Shift is a density-based nonparametric clustering algorithm, whose algorithm idea is to assume that the data sets of different cluster classes conform to different probability density distributions, find the fastest direction of density increase of any sample point (the meaning of the fastest direction is Mean Shift), the region with high sample density corresponds to the maximum value of this distribution, these sample points will eventually converge at the local density The points that converge to the same local maximum are considered to be members of the same cluster class.

II. Core density estimation

The Mean Shift algorithm estimates the density of a sample using a kernel function, and the most commonly used kernel function is the Gaussian kernel. It works by setting a kernel function at each sample point on the dataset, and then summing all the kernel functions to get the kernel density estimation of the dataset (kernel density estimation).

Suppose we have a d-dimensional dataset of size n $\{x_i\}$, and the bandwidth of the kernel function K is parameter h.

The kernel density estimation of the dataset is given by

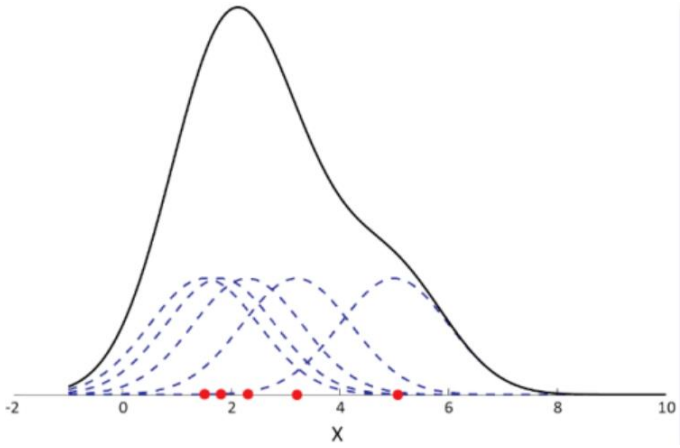
$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

where K(x) is a radially symmetric function (radially symmetric kernels), and define K(x) satisfying the kernel function condition as :

$$K(x) = c_{k,d} k(\|x\|^2)$$

where the coefficients $c_{k,d}$ are normalization constants such that the integral of K(x) is equal to 1.

From **Figure 4.2**, we estimate the density of a one-dimensional data set using a Gaussian kernel, and each sample point is set with a Gaussian distribution centered on that sample point, and accumulate all the Gaussian distributions to obtain the density of that data set. Where the dashed line indicates the Gaussian kernel for each sample point, and the solid line indicates the density of the dataset after accumulating all the sample Gaussian kernels. Thus, we get the density of the dataset by Gaussian kernel.



(**Figure 4.2**)

The density of the dataset can be known by introducing the Gaussian kernel, and the gradient is the direction of the fastest increasing function, so the direction of the gradient of the dataset density is the direction of the fastest increasing density. From the above, it can be seen that the density of the data set:

$$f(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

The gradient of the above equation is given by:

$$\nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n}(x - x_i)g(\left\|\frac{x - x_i}{h}\right\|^2)$$

$$= \frac{2c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n} g(\left\|\frac{x - x_i}{h}\right\|^2)\left[\frac{\sum_{i=1}^{n} x_i g(\left\|\frac{x - x_i}{h}\right\|^2)}{\sum_{i=1}^{n} g(\left\|\frac{x - x_i}{h}\right\|^2)} - x\right]$$

Where $g(s) = -k'(s)$

The first term of the above equation is a real value, so the vector direction of the second term is the same as the direction of the gradient, and the expression of the second term is

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g(\left\| \frac{x - x_i}{h} \right\|^2)}{\sum_{i=1}^{n} g(\left\| \frac{x - x_i}{h} \right\|^2)} - x$$

The meaning of the above equation is the subject of this article: mean drift. From the derivation of the above equation, it is clear that the direction pointed by the mean drift vector is the direction of the greatest increase in density.

III. Algorithm implementation steps

A. Select a random point among the unlabeled data points as the starting centroid center.

B. Find all data points appearing in the region with radius centered at center, and consider these points to belong to the same cluster C. Also record the number of occurrences of data points in this cluster plus one.

C. Using center as the center point, we calculate the vector of each element from center to the set M. We sum these vectors to get the vector shift.

D. center = center + shift. i.e. center moves in the direction of shift by ||shift||.

E. Repeat steps 2, 3, and 4 until shift is very small (that is, iterate to convergence), remembering center at this point. note that all points encountered during this iteration should be categorized into cluster C.

F. If the distance between the center of the current cluster C and the center of other already existing clusters C2 is less than the threshold at convergence, then C2 and C are merged, and the number of data point occurrences are merged accordingly. Otherwise, treat C as a new cluster.

G. Repeat 1, 2, 3, 4, 5 until all points are marked as visited.

H. Classification: According to each class, the one with the greatest visit frequency for each point is taken as the class to which the current point set belongs.

IV. Advantages and Disadvantages

Advantages:

1. No need to set the number of cluster classes.
2. Can handle cluster classes of arbitrary shape.
3. The algorithm only needs to set one parameter, bandwidth, which affects the kernel density estimation of the data set.
4. The algorithm results are stable and do not require sample initialization similar to K-means.

Disadvantages:

1. The clustering result depends on the bandwidth setting, if the bandwidth is set too small, the convergence is too slow and the number of cluster classes is too large; if the bandwidth is set too large, some cluster classes may be lost.
2. For a large feature space, the computation is very large.

7. CLARANS Clustering

I. Introduction

CLARANS (A Clustering Algorithm based on Randomized Search) is a large set of clustering algorithms based on randomized search in segmentation methods. The clustering process is to take numlocal subsamples from the actual data, and on each sample the corresponding clustering center is obtained using the K-medoids algorithm, and then the one with the smallest intra-class error sum is

selected as the final result among these, CLARANS algorithm is suitable for large data volume calculation.

II. Algorithm implementation steps

Step 1: Enter the parameters numlocal and maxneighbor.

Step 2: Randomly select k targets from the n targets to form the set of primes and make them current.

Step 3: Make j equal to 1.

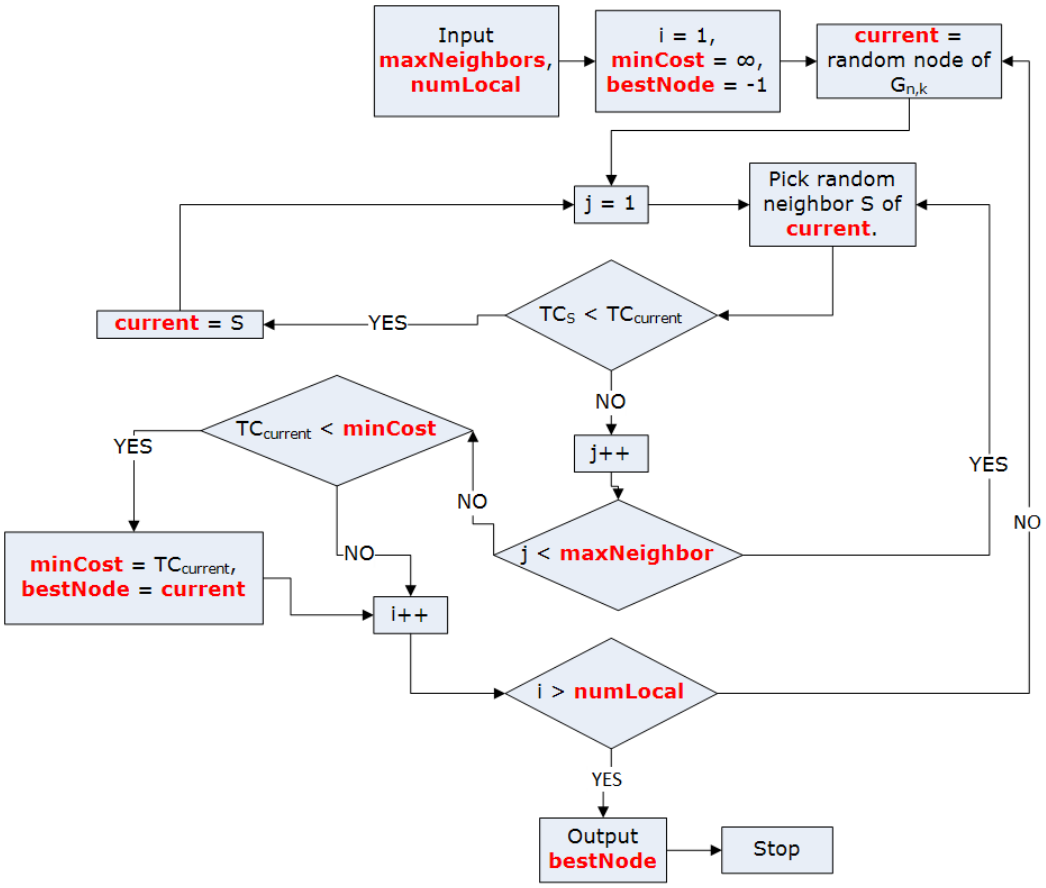
Step 4: randomly select a target from the remaining n-k targets in step 2, and replace it with a random prime in the set of primes to obtain a new set of primes, and calculate the difference in cost between the two sets of primes (this is similar to PAM, except that the replacement object and the replaced object are randomly selected).

Step 5: If the cost of the new set of prime is smaller then it is given to current, reset j=1, otherwise j+=1

Step 6: until j is greater than the maxneighbor, then current is the minimum cost of the set of plasmas at this time

Step 7: Repeat the above steps numlocal times, and take the set of prime with the smallest cost as the final set of prime

Step 8: divide and output the final set of prime according to the final set of prime



III. Comparison of Partitioning Methods

1. K-means:

The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. That is why K-means is sensitive to noise and outliers because a small number of such data can substantially influence the mean value.

2. K-medoids:

To overcome the problem of sensitivity to outliers, instead of taking the mean value as the centroid, we can take actual data point to represent the cluster, this is what K-medoids does. But the k-medoids methods is very expensive when the dataset and k value is large.

3. CLARA:

To scale up the K-medoids method, CLARA was introduced. CLARA does not take the whole dataset into consideration instead uses a random sample of the dataset, from which the best medoids are taken.But the effectiveness of CLARA depends on the sample size. CLARA cannot find a good clustering if any of the best sampled medoids is far from the best k-medoids.

4. CLARANS:

It presents a trade-off between the cost and the effectiveness of using samples to obtain clustering.

Parameters	K-means	K-mediods	CLARA	CLARANS
Complexity	O(ikn)	$O(k(n-k)^2)$	$O(ks^2 + k(n-k))$	$O(n^2)$
Implementation	Easy	Complicated	Complicated	Complicated
Sensitive to outliers?	Yes	No	No	No
No. of clusters parameter required?	Yes	Yes	Yes	Yes
Optimized for	Seperated Clusters	Seperated clusters, small dataset	Seperated clusters, large dataset	Seperated clusters, large dataset

IV. Advantages and Disadvantages

CLARA NS is based on the CLARA algorithm and its effectiveness depends on the size of the sample. Unlike CLARA, CLARANS is not limited to any sample at any given time. The time complexity of CLARANS is about O(n2). n is the number of objects. The advantage of this method is that on the one hand, it improves the quality of CLARA clustering. On the other hand, it extends the range of data processing and has a better clustering effect. However, its computational efficiency is low, and it is sensitive to the data input order, and can only cluster convex or spherical boundaries.

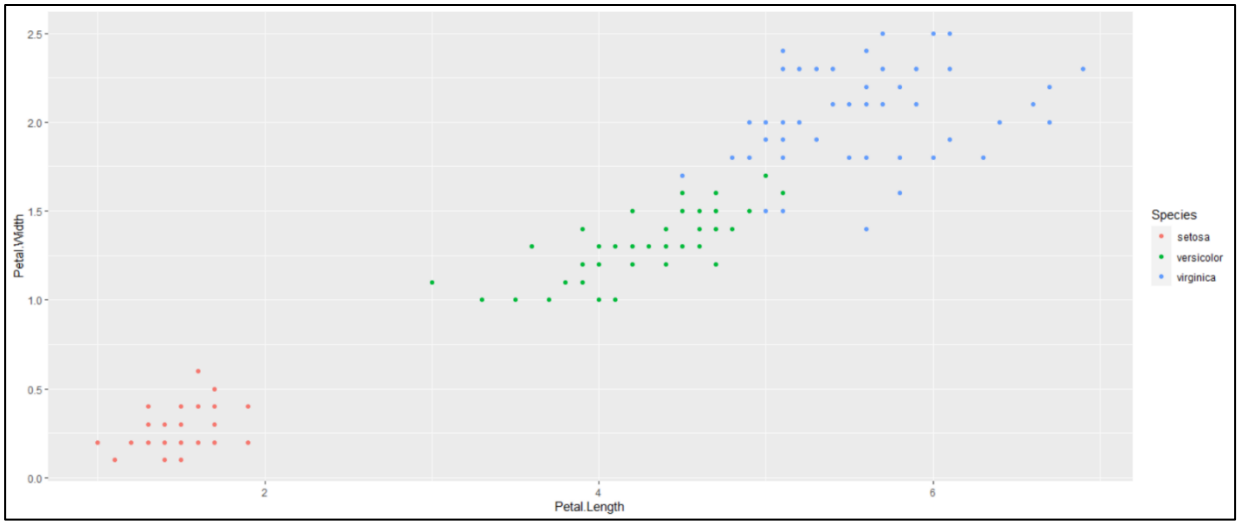
5. Application and Examples.

5.1 Dataset Introduction

In introducing the various tools for evaluating clustering methods, we will test the following five data (all two-dimensional and standardized). Through the different distribution of data, we want to find out which clustering methods are suitable for different data and which aspects of the data can be examined by different evaluation tools.

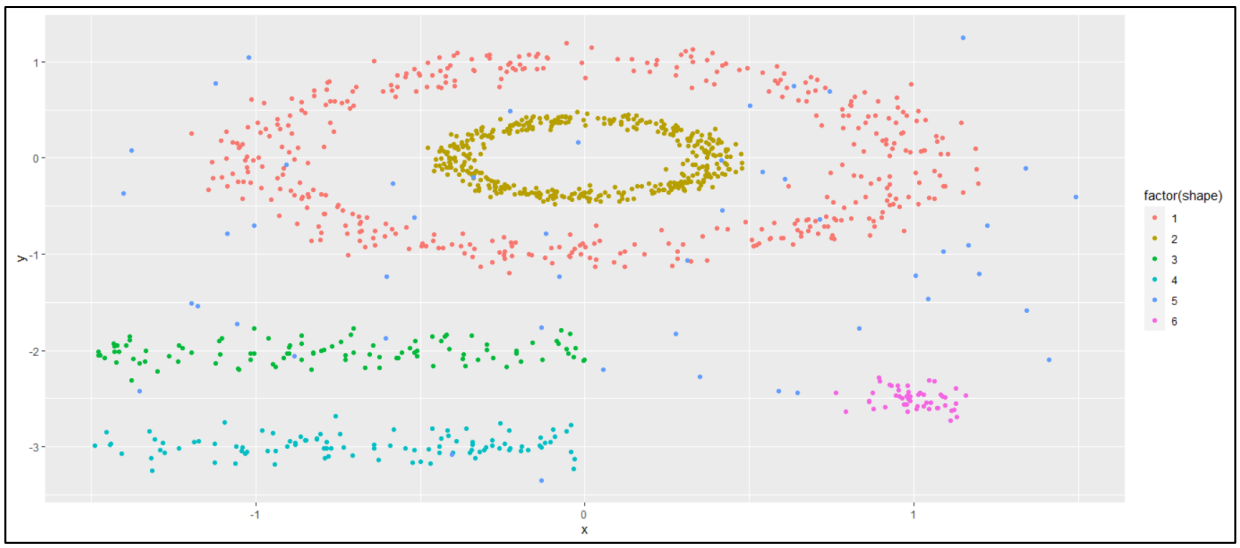
I. Iris

We used the variables of "Petal.Length" and "Petal.Width" in the Iris dataset for grouping, and this data features three groups, but two of them are quite close to each other.



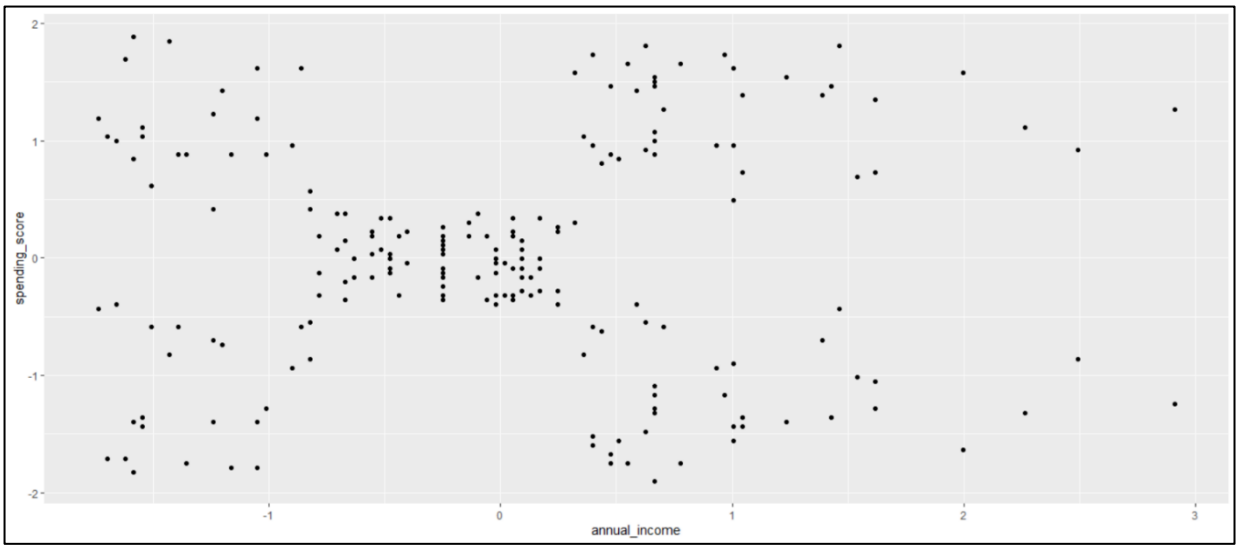
II. Multishapes

This data consists of two rings, two long bars, and a center-outward distribution of data, and it features a tangible shape.



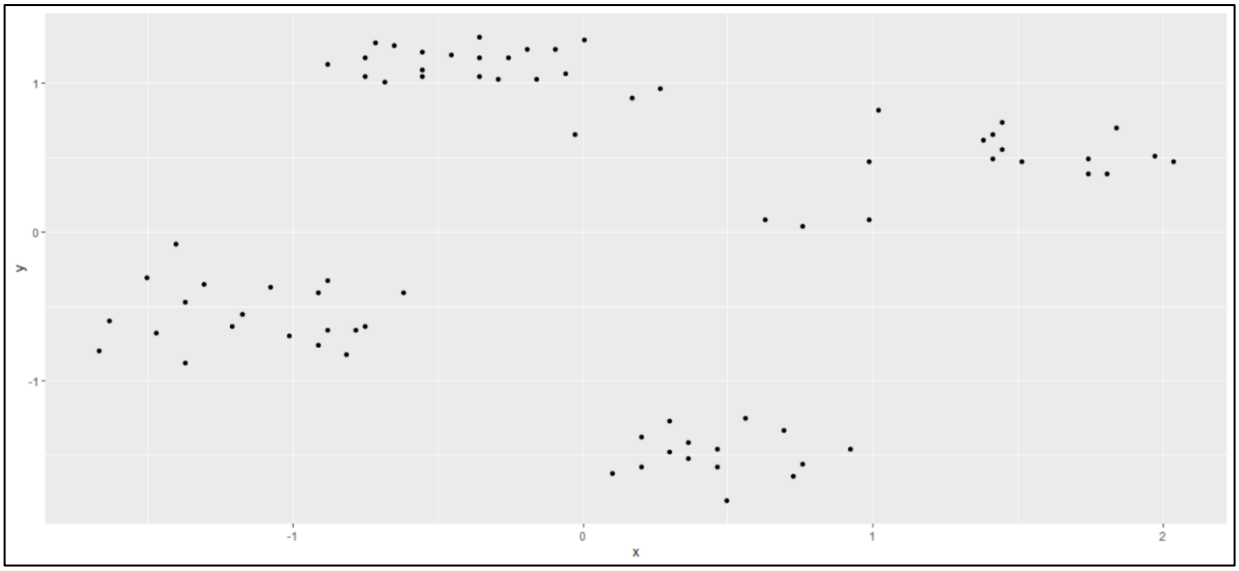
III. Mall Customer

This is the shopping center consumer data, using the variables "annual_income" and "spending_score", which is characterized by a grid-like distribution.



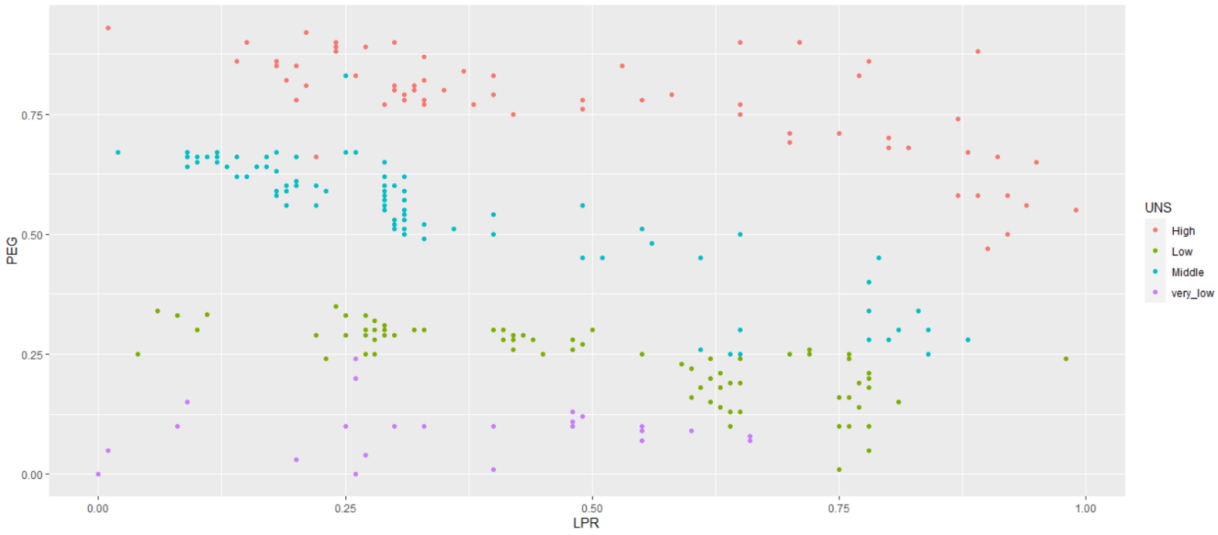
IV. Ruspini

This data consists of four clusters, which are clearly separated.



V. 知識掌握程度

This data was used to investigate the knowledge mastery of the test takers, and we used "PEG (performance in examinations of target subjects)" and "LPR (performance in examinations of related subjects)" as variables for the analysis. This data is characterized by a low negative correlation (r=-0.27), although there are no significant subgroups to identify.

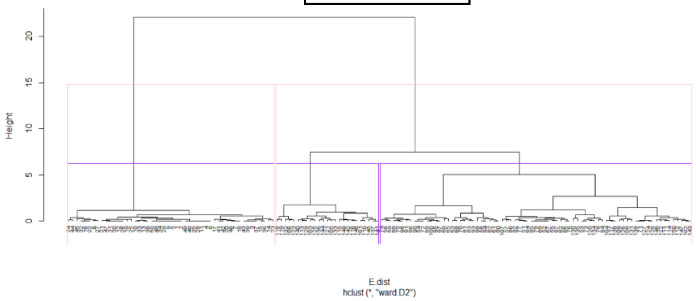


5.2 Clustering Methods Application

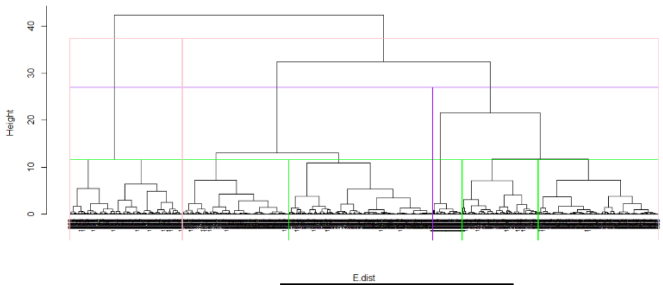
I. Hierarchical Clustering

The following is the hierarchical grouping calculated by applying Ward's Method. First of all, the results presented by the five types of data have obvious reference for grouping, for example, in the iris data, after splitting the group with "height" greater than 5, three groups are obtained.

Iris

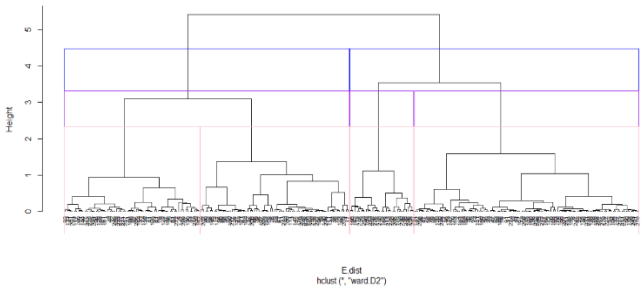


Multishapes

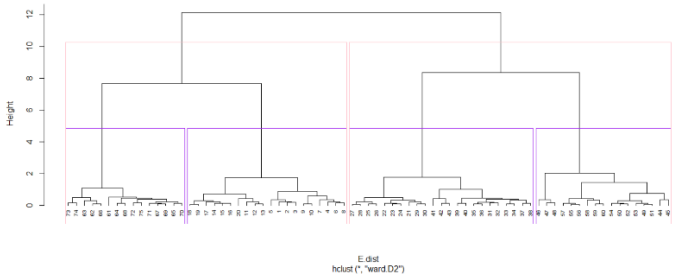


Mall Customer

知識掌握程度



Ruspini

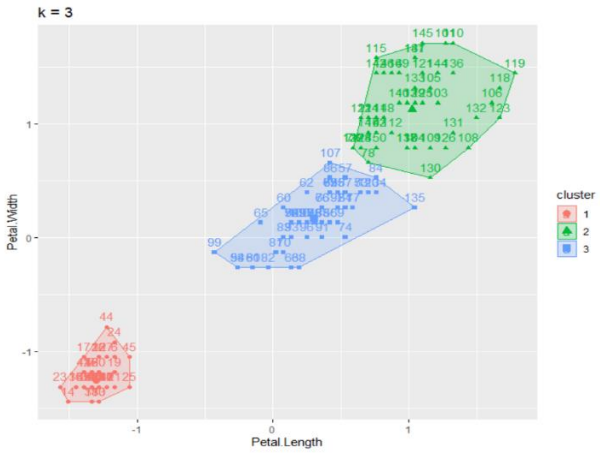


II. K-means

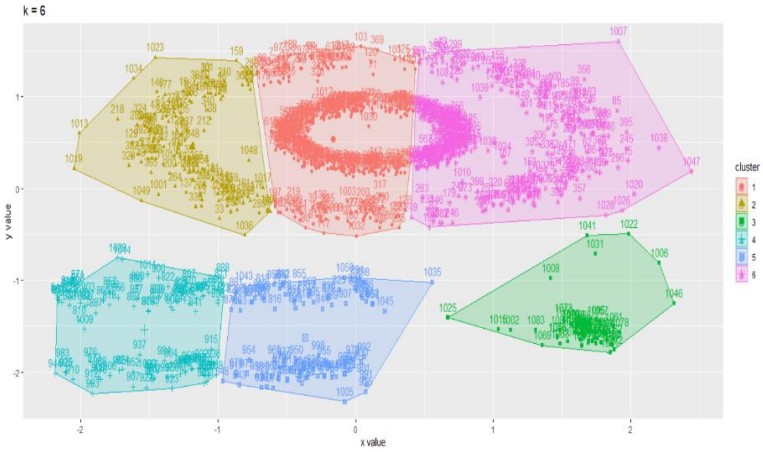
The following are the clustering results obtained by applying the K-means clustering method. It can be found that the data whose cluster distribution is centered outward (e.g., iris, ruspini) are more suitable for K-means as the clustering algorithm, while the results of multishapes with graphical clustering and mall customers with grid-like clustering are less suitable for the algorithm,

however, the data whose clustering status is less obvious have better clustering results than other algorithms.

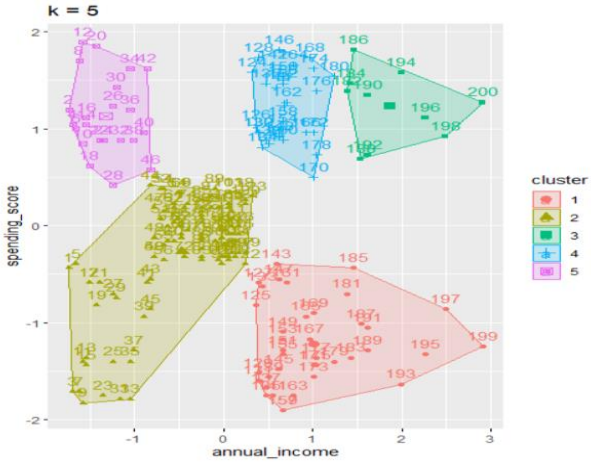
Iris



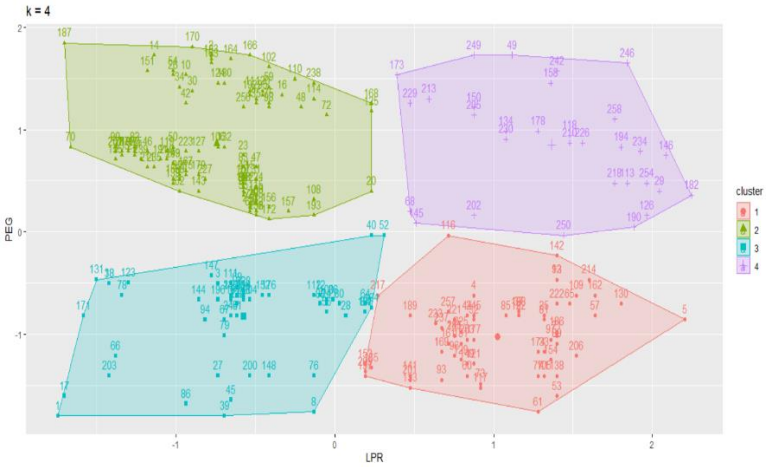
Multishapes



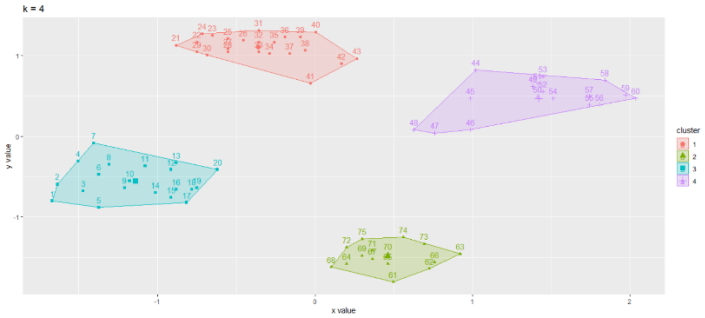
Mall Customer



知識掌握程度



Ruspini

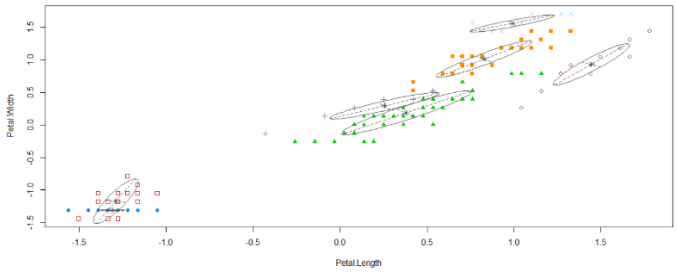


III. GMM-EM

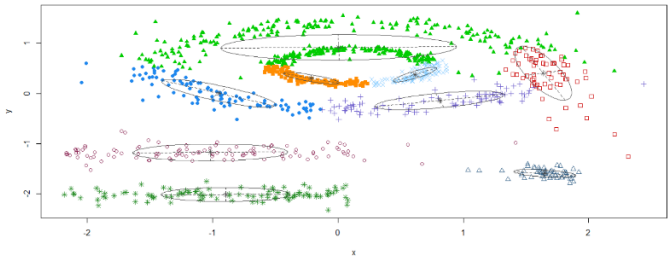
The following are the results of the GMM-EM algorithm, we can find that this algorithm is more sensitive to the sparse distribution of data, and there are more clusters than the original expected number. For example, the algorithm of mall customer has some outliers in the higher part of annual_income, so the algorithm determines the outliers as a group independently, and this phenomenon is found in other data such as iris and ruspini. However, this algorithm assumes that the clusters are normally distributed, so the multishape clustering results are less satisfactory.

In this algorithm, the clustering result of knowledge mastery is the most outstanding. Unlike other algorithms, the clustering result of knowledge mastery is a top-to-bottom stacking of clusters, which is more in line with the original data of "UNS (subject's knowledge level)" clustering, indicating that knowledge mastery is more suitable for this algorithm.

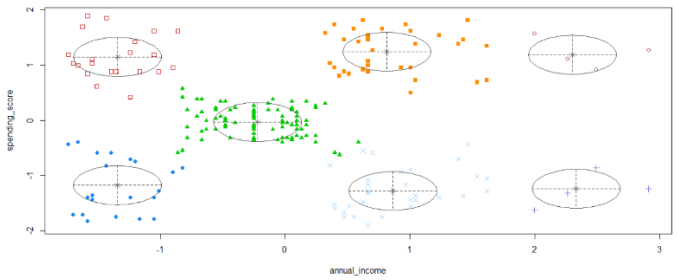
Iris



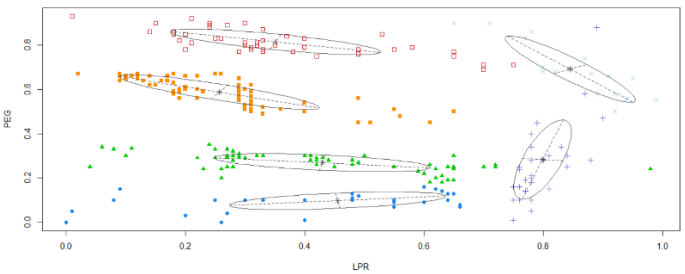
Multishapes



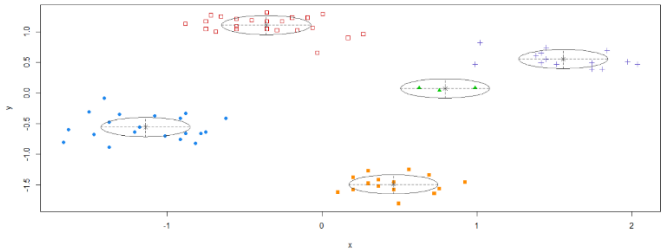
Mall Customer



知識掌握程度



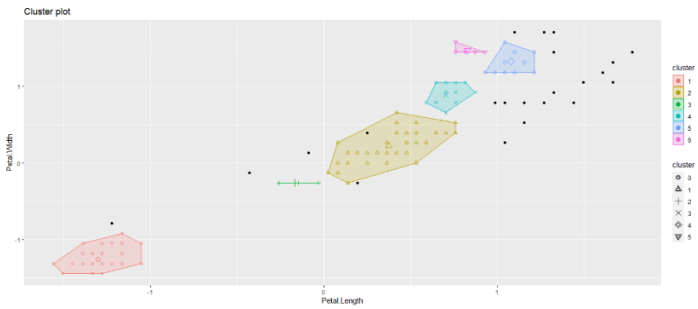
Ruspini



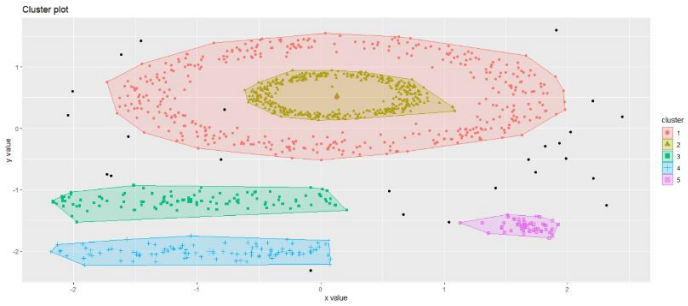
IV. DBSCAN

The following are the results of the clustering by the DBSCAN algorithm.First, we can observe that if the density of the clusters in the data level is not consistent, the clustering result is not satisfactory. For example, in the mall customer dataset, although the data are scattered in a grid pattern, the dispersion among the clusters is not consistent, which leads to the central cluster in the graph to include the data of other clusters, and the clustering result of knowledge mastery data is also similar. In addition, iris was also identified as three different clusters because the data were too dispersed in one group. The better performance of multishape and ruspini, both of which have the property that the data clusters are similar in density, and DBSCAN performs better when the data clusters are distributed in different graphs.

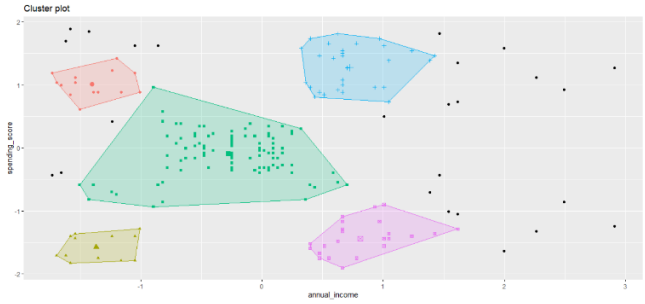
Iris



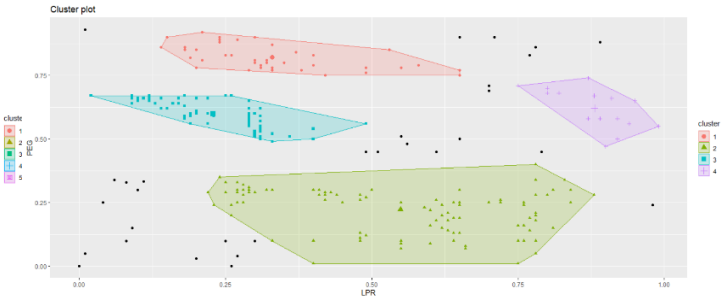
Multishapes



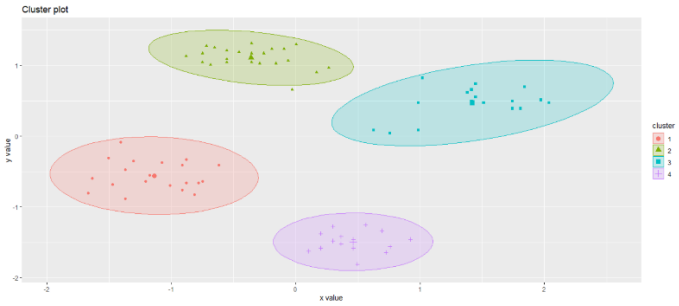
Mall Customer



知識掌握程度



Ruspini

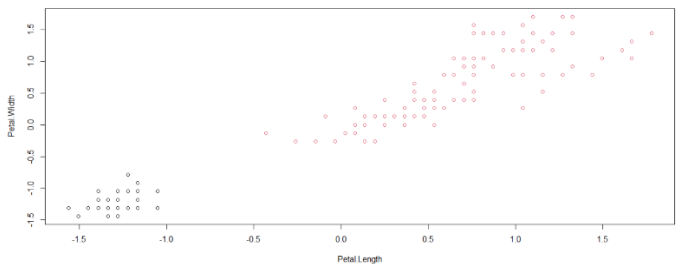


V. Mean Shift Method

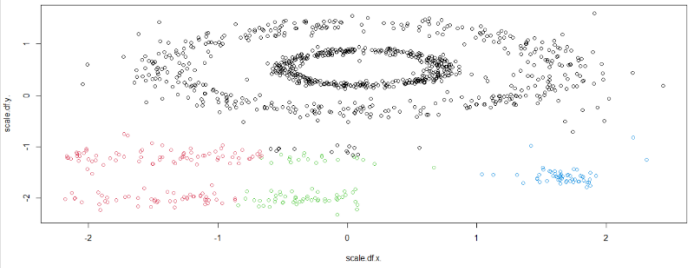
The following is the result of clustering by Mean Shift method. It can be found that the grouping method tends to treat groups that are closer together as the same group, while groups that are significantly farther apart will be treated as different groups.

For example, in the iris data, there are three groups, but two of them are considered as one group because they are closer to each other, and this is also the case in the knowledge mastery data and multishapes data. The results for mall customers and ruspini get the better results because the clusters are more clearly differentiated.

Iris

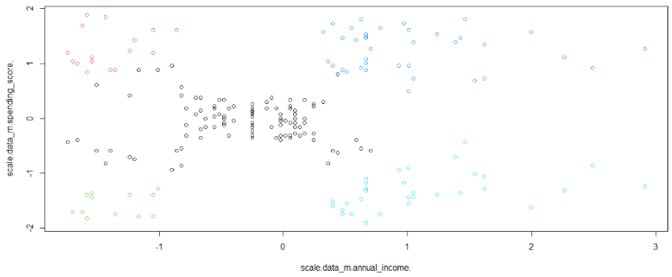


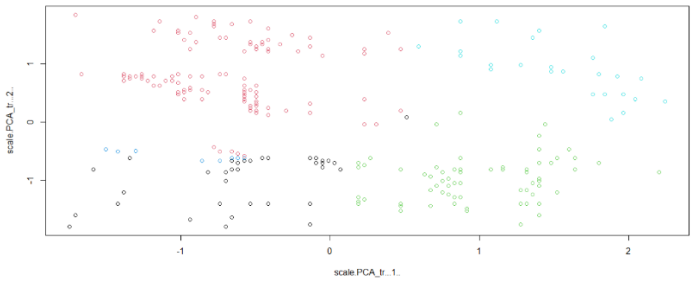
Multishapes

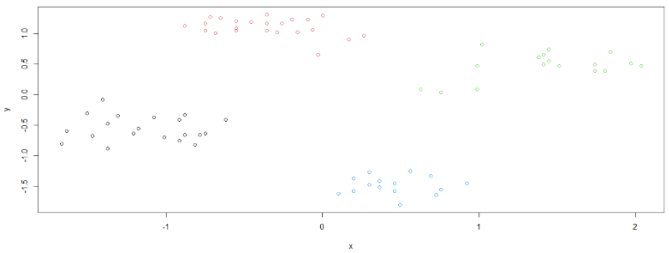


Mall Customer 知識掌握程度

Ruspini



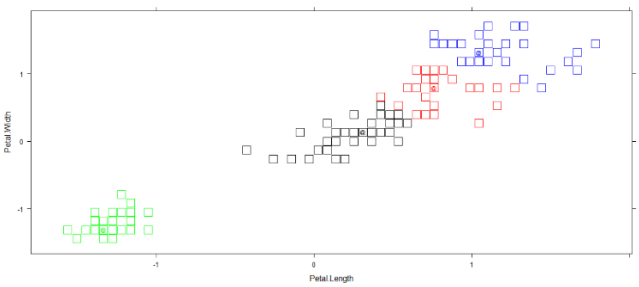


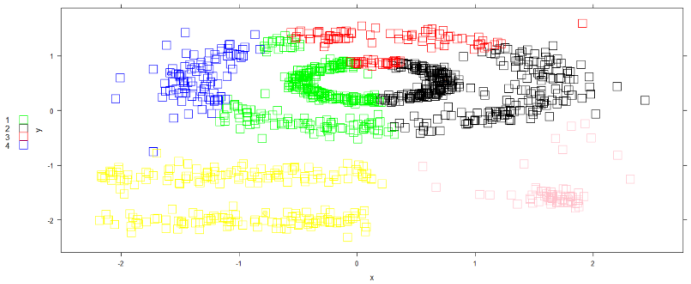


VI. CLARANS

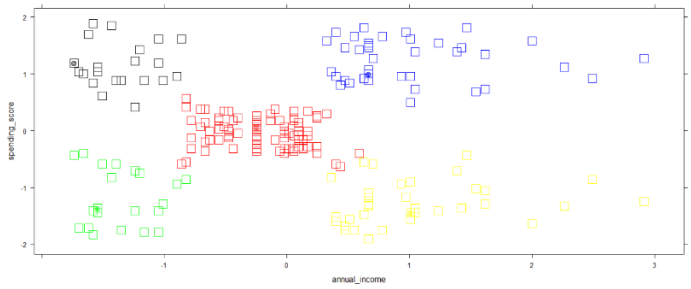
The following are the results of the CLARANS algorithm. This algorithm can separate the clusters with obvious distance as different clusters, but it can also choose a fixed number of clusters like the k-means algorithm. Therefore, the performance of Iris, mall customer, and ruspini is better because there are clearly separated clusters and a fixed number of clusters. However, the results for multishapes and knowledge mastery are less satisfactory because the separation between clusters of knowledge mastery is not obvious and CLARANS cannot handle clusters with multiple shapes like multishapes dataset.

Iris Multishapes

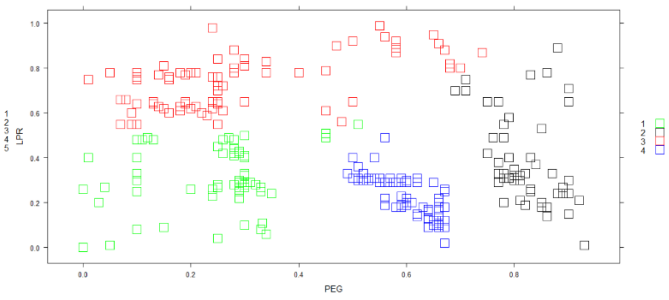




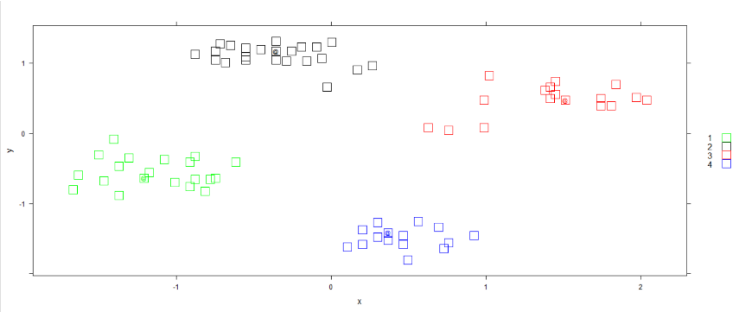
Mall Customer



知識掌握程度



Ruspini



5.3 Clustering tendency and Numbers of K

I. Hopkins Statistic

Hopkins Statistic can observe how the data tend to be clustered. The more concentrated the clusters are and the more intergroups are separated, the closer the Hopkins Statistic is to 1. Statistic of iris data is the highest, which means the clustering phenomenon of the data set is the most obvious, while the Hopkins Statistic of knowledge mastery data is the lowest, which means the clustering phenomenon of the data set is the least obvious, in line with the original judgment of using visualization chart

	Iris	Multishapes	Mall Customer	知識掌握程度	Ruspini
Hopkins Statistic	0.8675315	0.7226971	0.7057449	0.6485868	0.7816927

II. Dunn index

Dunn index is a measure of the degree of clustering of the results of the clustering algorithm, and the greater the concentration of the clustering results, the greater the dunn index. Therefore, dunn index does not compare the clustering results of different datasets, but the results of different clustering algorithms for the same dataset.

The following table shows the dunn index results of different datasets under different algorithms. In iris, since the binning results calculated by meanshift are two groups, those two groups are exactly the groups with a big difference in distance, so the dunn index is the highest, but it does not necessarily match the label of iris type.
Among Multishapes, DBSCAN has the highest dunn index. It is presumed that DBSCAN can effectively treat fixed density groups as the same group, while Multishapes has similar density in each group, so the method has the highest dunn index.

Mall Customer has the highest dunn index in the hierarchical clustering, but the difference between the dunn index and that of GMM-EM method is very small, and it can be observed that the result of GMM-EM considers the outliers and treats them as a group, so the dunn index is higher than other algorithms.

Since the results of ruspini are almost the same for different algorithms, the dunn index is also almost the same, and a high dunn index means that the data has a significant outlier phenomenon.

演算法 / 資料集	階層分群	K-means	GMM-EM	DBSCAN	Mean Shift	CLARANS
Iris	0.0832	0.0937	0.0321	0.1423	0.3794	0.1033
Multishapes	0.0055	0.0072	0.0039	0.0552	0.0064	0.0023
Mall Customer	0.0942	0.0542	0.0918	0.0620	0.0328	0.0612
知識掌握程度	0.0771	0.0416	0.0213	0.1582	0.0136	0.0359
Ruspini	0.5248	0.5248	0.3366	0.5248	0.5248	0.5248

III. Silhouette Widths

Silhouette Widths can observe the degree of data aggregation in different clustering methods, so if we want to obtain a non-central outward distribution of clustering results, this method will not be a reference, for example, the clusters of Multishapes are composed of different shapes, so the best performing DBSCAN does not perform well in Silhouette Widths.

In addition, other centroid-based algorithms such as Mean Shift, CLARANS, and K-means have obtained the highest Silhouette Widths in different datasets, especially Mean Shift has obtained the highest Silhouette Widths in two datasets. This means that Mean Shift can obtain better clustering results with high data aggregation.

Finally, the Silhouette Widths are almost the same for ruspini because the clustering results are almost the same for different algorithms.

演算法 / 資料集	階層分群	K-means	GMM-EM	DBSCAN	Mean Shift	CLARANS
Iris	0.6099	0.6741	0.1743	0.4270	0.7433	0.6007
Multishapes	0.3764	0.4074	0.2039	0.2074	0.4313	0.3854
Mall Customer	0.3842	0.4653	0.5409	0.4148	0.4967	0.5562
知識掌握程度	0.4964	0.5034	0.3057	0.3521	0.3522	0.4240
Ruspini	0.7368	0.7368	0.6887	0.7368	0.7368	0.7368

IV. Rand Index

The Rand Index can compare whether the clustering results match the original labels, and it can also compare the similarity between different clustering results. Here we have three data sets with labels for us to check whether different algorithms are similar to the labels, and if the Rand Index is closer to 1, it means that the clustering results match the classification of the label.

From the following table, we can find that the results of the Rand Index are in accordance with our original expectation through visualization, Iris has the highest rand index in K-means algorithm, Multishapes has the highest rand

index in DBSCAN algorithm, and Knowledge Mastery has the highest rand index in GMM-EM algorithm. The highest rand index was obtained by the GMM-EM algorithm, indicating that the combination of this dataset and the algorithm is most similar to the original labeling.

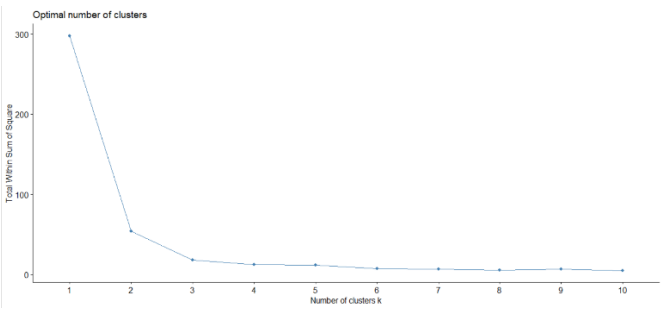
演算法 / 資料集	階層分群	K-means	GMM-EM	DBSCAN	Mean Shift	CLARANS
Iris	0.8322	0.9495	0.8027	0.8498	0.7763	0.9085
Multishapes	0.7275	0.7349	0.7580	0.9848	0.6539	0.7181
知識掌握程度	0.7388	0.7366	0.8643	0.8131	0.7164	0.7851

V. Elbow's method

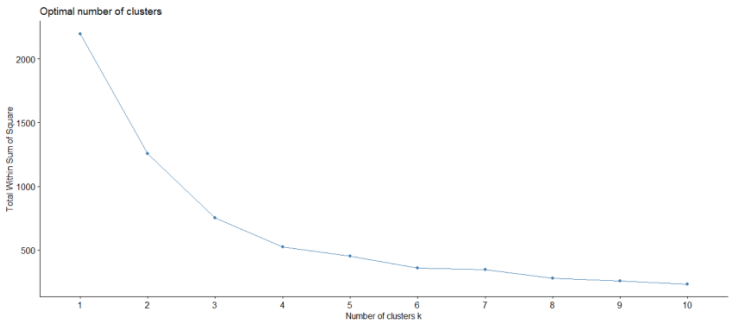
The following graphs show the intra-group variance of each dataset with k-means for different number of clusters, so that the more significant the clustering of the dataset, the less smooth the graphs will be and the more elbow shape will appear. For example, ruspini data has four distinct clusters, so after cluster number 4, the decline of WSS tends to be flat and elbow shape appears in the graph. iris data also has a distinct elbow shape at cluster number 2, which means there is a distinct cluster situation.

For Multishapes, Mall Customer, and Knowledge Mastery WSS, the rate of decline tends to level off as the number of clusters increases, but the elbow shape is less obvious, so the clustering of the data is less obvious, or the clusters are not spread outward from the center.

Iris

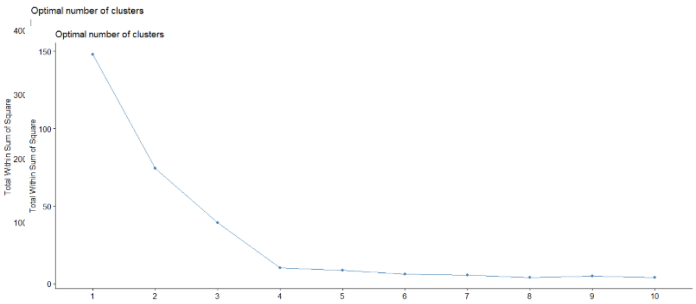


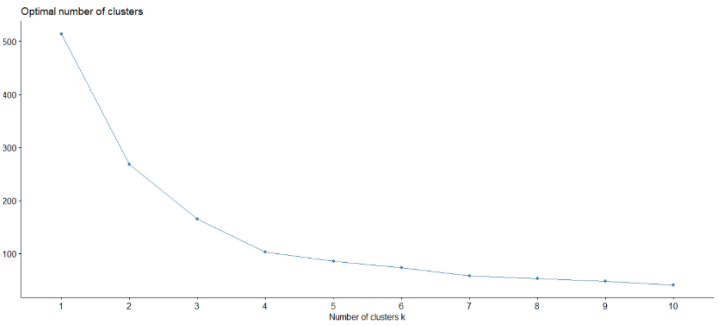
Multishapes



Mall Customer

知識掌握程度





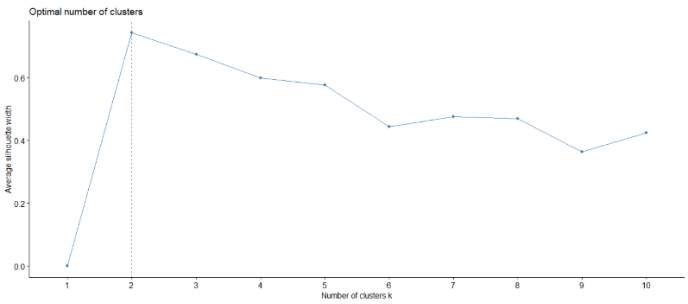
Ruspini

VI. Silhouette score

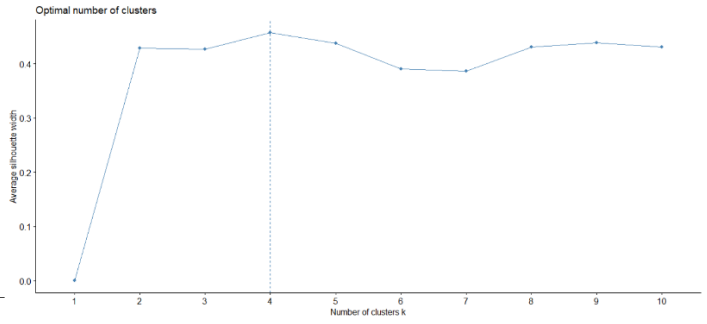
The following graph shows the Silhouette score of each dataset with k-means for different number of clusters, which means that the more significant the clustering of the dataset, the higher the Silhouette score will be when the right number of clusters is selected.

For example, iris has the highest Silhouette score for cluster number 2 and has the highest Silhouette score than other datasets because of its distinctive clustering. Conversely, because the clustering of Multishapes and knowledge acquisition is less obvious (or the clusters are not spread outward from the center), the Silhouette scores obtained at different cluster numbers are, on average, lower than those of the other datasets.

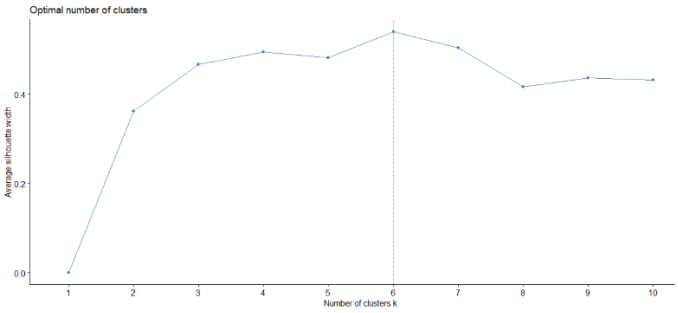
Iris



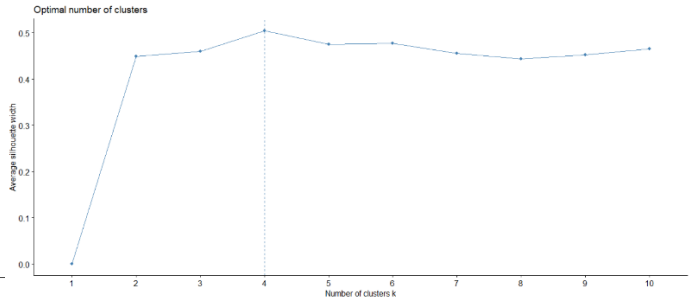
Multishapes



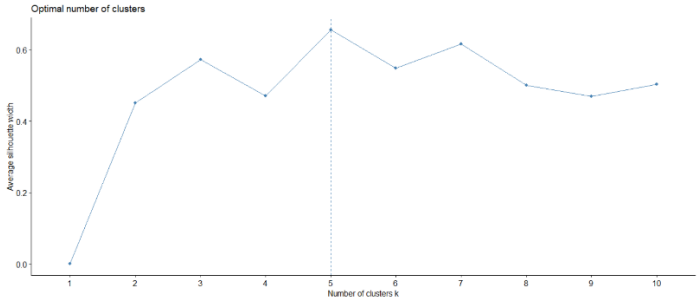
Mall Customer



知識掌握程度



Ruspini

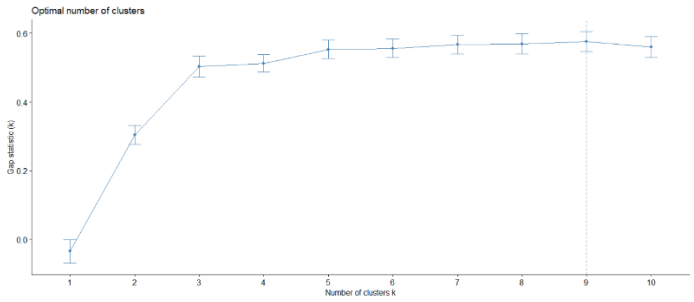


VII. Gap statistic

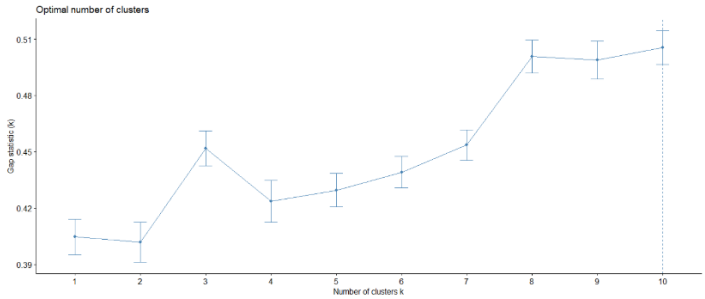
The following graph shows the Gap statistic for each dataset with different number of clusters using k-means. Therefore, it means that the Gap statistic will be higher when the dataset is more significantly clustered and the right number of clusters is selected. However, compared to elbow's method, when the number of clusters is higher, it does not mean that the Gap statistic is higher. For example, the Gap statistic of mall customer data is highest when the number of clusters is five, and then decreases smoothly, which means that when a cluster is split into more than two clusters, it does not make the Gap statistic higher. In addition, another important point to note is that if the clustering phenomenon of the data set is not obvious, it will cause the Gap statistic to be higher as the number of clusters increases because more clusters need to be set in order to get more concentrated clusters of data. For example, in the case of knowledge mastery data, since the clustering phenomenon is not

obvious, more clusters are needed to obtain more concentrated clusters in order to obtain a higher Gap statistic.

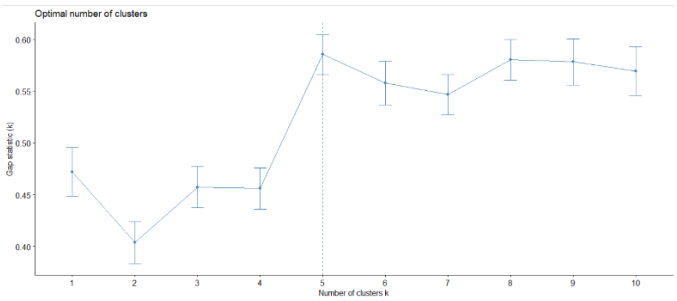
Iris



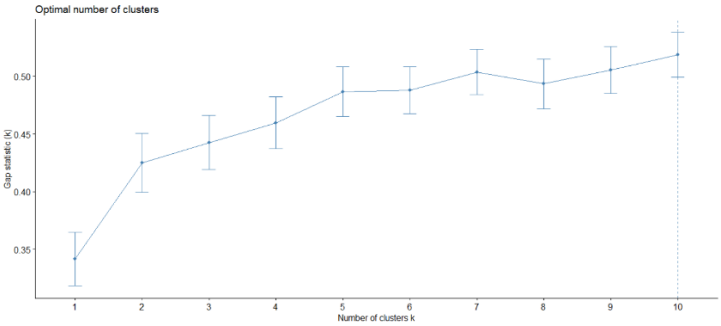
Multishapes



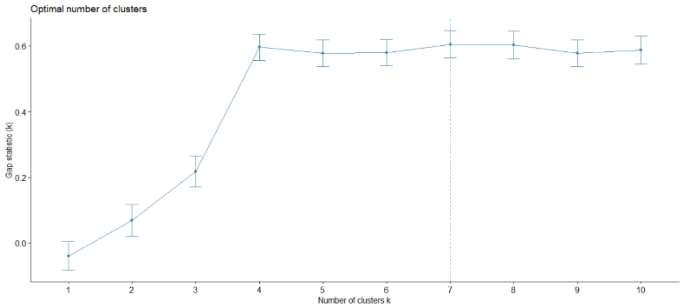
Mall Customer



知識掌握程度



Ruspini



5.4 Summary

The following table summarizes the characteristics of datasets suitable for each clustering algorithm through the above analysis process, so that we can use this table as a judgment to find suitable datasets when obtaining a dataset in the future.

不同的分群演算法所適合的資料集特性(○：適合 ×：不適合 ∆：對該演算法無主要影響)						
演算法 / 資料集	階層分群	K-means	GMM-EM	DBSCAN	Mean Shift	CLARANS
各群資料分布為不同形狀	×	×	×	○	×	×
各群資料分布為中心向外散佈	○	○	○	∆	○	○
各群資料分布為多元常態分佈	∆	○	○	∆	○	○
各群資料分布密度不同	○	○	×	×	×	○
不同群之間的距離較小	○	○	○	×	×	○
可選擇固定的群數	○	○	×	×	×	○

6. Reference

1. Mall customers on Kaggle
2. User Knowledge Modeling Data Set
3. R 筆記–(9)分群分析(Clustering)
4. GMM: Gaussian Mixture Model clustering
5. R DBSCAN 集群方法
6. clarans: K-medoids clustering of SNPs using randomized search
7. get_clust_tendency: Assessing Clustering Tendency
8. dunn: Dunn Index
9. rand.index: Rand Index and Adjusted Rand Index
10. Package ‘meanShiftR’

11. Cluster Validation Statistics: Must Know Methods

12. Evaluation of clustering - Stanford NLP Group

13. Assessing Clustering Tendency - Datanovia

14. Evaluation methods for a clustering techniques ?

15. Using internal evaluation measures to validate the quality of ...