Techniques for Evaluating Clustering Data in R. The Clustering Package

by Luis Alfonso Pérez, Ángel Miguel García Vico, Pedro González and Cristóbal J. Carmona

Abstract Clustering is an unsupervised learning technique where the model is adjusted to the observations. This technique is quite common among researchers because they can obtain knowledge quickly and easily. The use of this technique is suitable for automatically classifying data to reveal concentrations of data. This paper presents the **Clustering** package which contains a sest of clustering algorithms with two objectives: first grouping data in an homogeneous way by establishing differences between clusters, and second generating a ranking between algorithms and the variables analysed in the dataset. This package contains references to other R packages without using external software. As a complement to the standard execution through the console, it incorporates a GUI through which we can execute the package without having to know the parameters.

Introduction

Exploring the properties of information in order to make groups is an unsupervised learning technique known as clustering (Mann and Kaur, 2013) (Karypis et al., 2000). This technique is a concise data model where a set of data must be partitioned and introduced in groups or clusters. These clusters must meet two conditions: clusters must be the most disparate possible among them, and the elements that contain them the most similar. If we review the literature related to clustering we can see that the fields where they can be applied are multiple, among which we highlight the following: Identify tourists and analyze their destination patterns from location-based social media data (Hasnat and Hasan, 2018), clustering algorithm that maximizes performance on 5G heterogeneous networks (Balevi and Gitlin, 2018), Application of data mining techniques to agriculture data (Ponperiasamy and Thenmozhi, 2017), Weighting of characteristics based on strength between categories and within categories for the analysis of feelings (Wang and Youn, 2018), Music classification, genres and taste patterns (Vlegels and Lievens, 2017), Predict the direction, maximum, minimum and closing prices of the daily exchange rate of bitcoins (Mallqui and Fernandes, 2018) and Clustering of people in a social network based on textual similarity (Singh et al., 2016).

As a rule, the clustering algorithms are based on the optimization of an objective function, which is usually the weighted sum of the distance to the centers, although these functions may vary and in some cases consists of the definition of functions. In the literature we can group the data in different ways among which we highlight (Popat and Emmanuel, 2014): partitional, hierarchical or based on density. One of the best known algorithms that solves the clustering problem is the k-means (Macqueen, 1967)

Throughout the literature we have located a wide variety of frameworks that work with clustering algorithm implementations among which we can cite the following: Weka (Litoriya, 2012), ClustVis (Metsalu T, 2015) and Keel (Fernández et al., 2009) among others. Also within R there is a specific Cluster task view. Inside this section we see two well differentiated parts: on one hand we have the most outstanding packages by functionality and in second place we observe the set of packages that work with cluster ordered. From the set of packages we highlight the following: **ClusterR** (Sculley, 2010), **apcluster** (Frey and Dueck, 2007), **cluster** (Mächler et al., 2017), **advclust** (Farias et al., 2011) as well as alternatives to the traditional implementation of k-means and agglomerative hierarchical clustering.

This contribution presents the **Clustering** package. It is a package that allows you to compare multiple clustering algorithms simultaneously and assess the accuracy of the results. The purpose of this package is to evaluate a set of datasets to determine which variables are most suitable for clustering. So we can make evaluations of the clusters created, how they have been distributed, if the distributions are uniform or how they have been categorized from the data.

The distribution of the content of this contribution is as follows: Firstly, in section Clustering we have the presentation of clustering, types of clustering and similarity measures. Section Internal and External clustering validation measures presents the definition of the evaluation measures in order to value the distribution of the data in the clusters and finally Section The **Clustering** package describes the structure of the package and it presents a complete example about the use of the package.

Clustering

Cluster analysis is an unsupervised learning method that constitutes a cornerstone of an intelligent data analysis process. It is used for the exploration of inter-relationships among a collection of patterns, by organizing them into homogeneous clusters. It is called unsupervised learning because unlike classification (known as supervised learning), no a priori labeling of some patterns is available to use in categorizing others and inferring the cluster structure of the whole data (Kotsiantis and Pintelas, 2004). The basic concept of clustering should be expressed as follows:

> Clustering is the process of identifying natural clusters or clusters within multidimensional data based on some measure of similarity (Euclidean, Manhattan) (Omran et al., 2007).

This is a base definition of the clustering so variations in the problem definition can be significant, depending mostly on the model specified. For example, a generative model should define similarity based on a probabilistic generative mechanism, while a distance-based approach will use a traditional distance function to quantify it. In addition, the types of data specified also have a significant impact on the problem definition.

Clustering types

There are a variety of clustering algorithms that can be classified into: hierarchical, partitioning, density-based, grid-based and probability distribution. The most commonly used groupings are: hierarchical, part-based and density-based.

- Hierarchical clustering algorithms: create a hierarchical breakdown of data into a dendogram that recursively divides the data set into smaller and smaller data. It can be created in two ways: bottom-up or top-down (Jain et al., 1999a). In bottom-up trees also known as agglomerative, as the objects are successively combined according to the measurements, until they are all joined into one or meet a completion condition. In the case of top-down, it is known as divisive, where all the objects are in the same group, and as we iterate they are divided into smaller subsets, until each object is in an individual group or fulfills a condition of completion. An example of this type of clustering can be found in the Figure 1. Some hierarchical grouping algorithms that belong to this sorting mode are CURE (Guha et al., 1998), CHAMELEON (Guo et al., 2019), and BIRCH (Zhang et al., 1996).

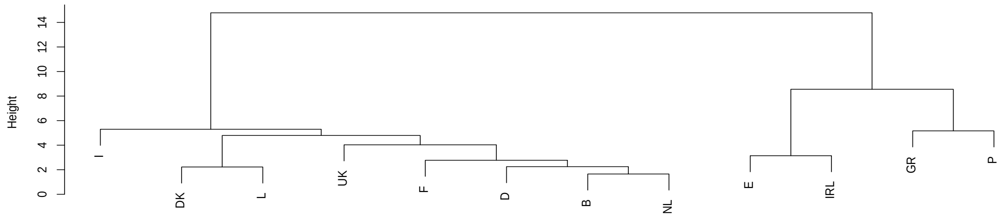


Figure 1: Hierarchical Clustering.

- Partial clustering: is considered to be the most popular of the clustering algorithms (Saxena et al., 2017). Such an algorithm is also known as an iterative relocation algorithm. This algorithm minimizes a given clustering criterion by iteratively relocating data points between clusters until an optimal partition is reached. This type of algorithm divides the data points into a partition k, where each partition represents a cluster. Partial clustering organizes the objects within k clusters so that the total deviation of each object from the center of its cluster or from a cluster distribution is minimal. The deviation of a point can be evaluated differently according to the algorithm, and is generally known as a similarity function. If we want to observe graphically how this type of clustering works we can see it in the Figure 2. Among the partitioning clustering algorithms we can find CLARANS, CLARA (Ramprasanth and Devi, 2019), K-prototype (Nithya and Prabha, 2019), K-mode (Huang, 1997) and K-means (Kushwaha et al., 2020).

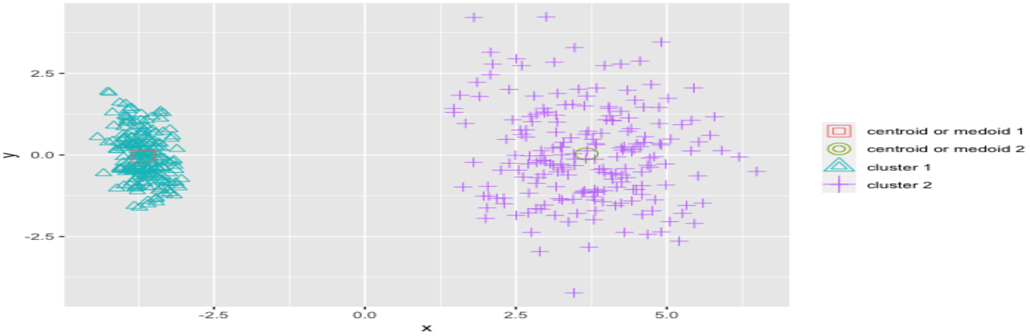


Figure 2: Partitional Clustering.

- Density-based algorithms: obtain clusters based on dense regions of objects in the data space that are separated by low-density regions (these isolated elements represent noise). These regions are represented in the Figure 3. Among the density-based algorithms, we highlight the following: Dbscan (Hu, 2019), and Denclue (Khader and Al-Naymat, 2019).

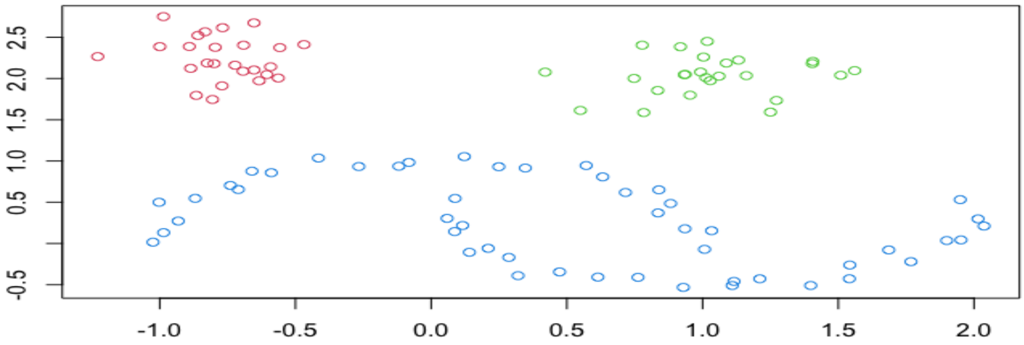


Figure 3: Density Clustering.

- Grid-based clustering algorithms: first quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space (Dang, 2011). Cells that contain more than certain number of points are treated as dense and the dense cells are connected to form the clusters. Some of the grid-based clustering algorithms are: STING (MR and MOHAN, 2010), Wave Cluster (Xuecheng, 2010) and CLIQUE (Saini and Rani, 2017).
- Model-based methods: are primarily based on a probability distribution. To be able to measure similarity it is based on the mean values and the algorithm tries to minimize the square error function. Auto Class algorithm uses the Bayesian approach, starting with a random initialization of parameters that is gradually adjusted in order to find the maximum probability estimates. Among the model-based algorithms we highlight SOM (Thalamuthu et al., 2006). Model-based clustering is shown in the Figure 4.

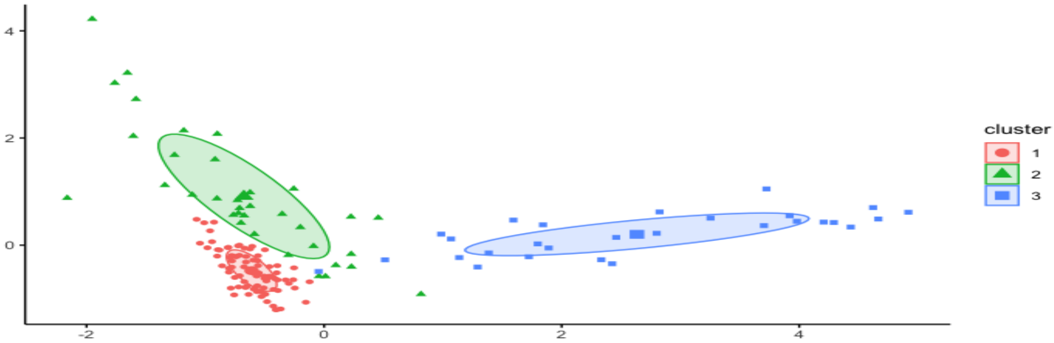


Figure 4: Model-Based Clustering.

Dissimilarity measures

Dissimilarity measurements are important because they allow us the creation of clusters with the closest neighbours and the detection of anomalies, and they are also used in a large number of data mining techniques. It is also a measure that determines the degree to which objects are different. We often use the term distance as a synonym for dissimilarity. The values of dissimilarity should be in the range [0,1], but it is common to find in some cases the range 0 to ∞.
Many distance measures have been proposed in literature for data clustering. The choosing an appropriate similarity measure is also crucial for cluster analysis, especially for a particular type of algorithms. For example, the densit -based clustering algorithms, such as DBScan (Hu, 2019), rely heavily on the similarity computation. Density-based clustering finds clusters as dense areas in the data set, and the density of a given point is in turn estimated as the closeness of the corresponding data object to its neighboring objects (Pandit et al., 2011) (Shirkhorshidi et al., 2015).
As measures of dissimilarity in clustering we highlight the following:

- Minkowski: Is a metric in a normed vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance (Gan et al., 2007).

$$d_{min} = (\sum_{i=1}^{n} |x_i - y_i|^m)^{\frac{1}{m}}, m \geq 1 \tag{1}$$

  where m is a positive real number and x_i and y_i are two vectors in n-dimensional space.

- Euclidan distance: Is a special case of Minkowski distance. Is a measure of the true straight line distance between two points in Euclidean space. (Jain et al., 1999b).

$$d_{xy} = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \tag{2}$$

- Manhattan distance: Also known as the geometry cab driver is defined as the sum of the lengths of the projections of the line segment between the points onto the coordinate axes (Xu and Wunsch, 2005).

$$d_{xy} = \sum_{i=1}^{n} |X_i - Y_i| \tag{3}$$

- Mahalanobis distance: Is a data-driven measure in contrast to Euclidean and Manhattan distances that are independent. It is in charge of measuring the distance in a multivariate space (Xu and Wunsch, 2005).

$$d_{mah} = \sqrt{(x-y)S^{-}1(x-y)^T} \tag{4}$$

  where S is the covariance matrix of the dataset

- Pearson correlation: It's a statistically based metric that measures linear correlation between two variables x and y.

$$Pearson(x,y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}} \tag{5}$$

  where μ_x and μ_y are the means for x and y respectively.

- Jaccard Index: Is a classical similarity measure on sets with a lot of practical applications in information retrieval, data mining, machine learning, and many more (Kosub, 2019) (Irani et al., 2016). It measures the similarity of the two data elements as the intersection divided by the union of the data elements as shown below:

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{6}$$

- Gower distance: It is a measure of similarity that allows the simultaneous use of quantitative, qualitative and dichotomous variables. By applying this similarity coefficient can be to determine the degree of similarity between individuals; who have been measured qualitative, quantitative characteristics (continuous and discrete) and binary.

$$d_{ij} = \sqrt{(1 - S_{ij})} \tag{7}$$

where S_{ij} is Gower similarity coefficient.

$$S_{ij} = \frac{\sum_k^n w_{ijk} S_{ijk}}{\sum_n^k w_{ijk}} \tag{8}$$

S_{ijk} denotes the contribution provided by the k-th variable, and w_{ijk} is usually 1 or 0 depending if the comparison is valid for the k-th variable.

Internal and External clustering validation measures

Clustering validation is a technique to find a set of clusters that best fits natural partitions (number of clusters) without any class information.The results of a clustering algorithm are known as cluster validity. The following criteria must therefore be taken into account when investigating the validity of clusters. The first criterion is based on external measures, which involves evaluating the results of a base algorithm in a pre-specified structure which is imposed on a data set and reflects our intuition about the structure of clustering of the data set. The second criterion is based on internal measures where it evaluates the results of a clustering algorithm in terms of the quantity involved in the vectors of the dataset itself (e.g. the proximity matrix). And as a third criterion known as relative criterion whose purpose is to compare the results of execution of an algorithm with another using different parameters. When we talk about criteria based on internal measures we must take into account the criteria of compaction and separation (Halkidi et al., 2001) (Berry and Linoff, 2004) as you can see in the Figure 7:

1. Compactness, the members of each cluster should be as close to each other as possible. A common measure of compactness is the variance, which should be minimized.
2. Separation, the clusters themselves should be widely spaced. There are three common approaches measuring the distance between two different clusters:
   - Single linkage: It measures the distance between the closest members of the clusters.
   - Complete linkage: It measures the distance between the most distant members.
   - Comparison of centroids: It measures the distance between the centers of the clusters.

These criteria are graphically represented in the image 5

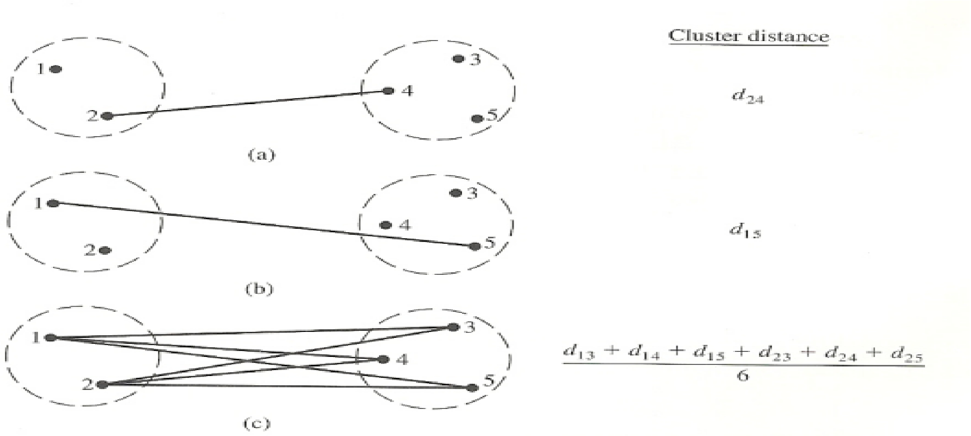

Cluster distance
1
2
3
4
5
(a)
d_{24}
(b)
d_{15}
(c)
$\frac{d_{13} + d_{14} + d_{15} + d_{23} + d_{24} + d_{25}}{6}$

Figure 5: Intercluster distance (Johnson and Wichern, 1998).

Inside external tests exits some measures to evalute clustering results. Among which we highlight:

- Entropy: It evaluates the distribution of categories in a cluster (Kim and Park, 2007).

$$Entropy = \sum_{j=1}^{m} \frac{n_j}{n} E_j \tag{9}$$

Where n_j is the cluster size j, n is the number of clusters, and m is the total number of data points. To calculate the entropy of a data set, we need to calculate the class distribution of the objects in each group as follows:

$$E_j = \sum_i p_{ij} log(p_{ij}) \tag{10}$$

Where p_{ij} is the probability of a point in the cluster i of being classified as class j.

- Recall: It indicates the proximity of the measurement results to the true value (Kacprzyk and Farhaoui, 2019).

$$Recall(i,j) = \frac{n_{ij}}{n_i} \tag{11}$$

  n_{ij} is the number of objects of class i that are in cluster j, n_j is the number of objects in cluster j and n_i is the number of objects in cluster i.

- Precision: It refers to the dispersion of the set of values obtained from repeated measurements of one magnitude (Kacprzyk and Farhaoui, 2019).

$$Precision(i,j) = \frac{n_{ij}}{n_j} \tag{12}$$

  n_{ij} is the number of objects in class i that are in cluster j, n_j is the number of objects in cluster j and n_i is the number of objects in class i.

- F-measure: It merges the concepts of accuracy and recall of the retrieved information (Rendón et al., 2011). Therefore, we calculate the cluster accuracy and recall for each class as:

$$F - measure(i,j) = \frac{2 * (Precision(i,j) * Recall(i,j))}{(Precision(i,j) + Recall(i,j))} \tag{13}$$

- Fowlkes-Mallows Index: It is a measure of comparison of hierarchical clustering, however it can also be used in flat clustering since it consists of the calculation of an index B_i for each level i = 2 ,..., n-1 of the hierarchy (Romano et al., 2016). The measure B_i is easily generalizable to a measure for clustering of different clusters. It can therefore be said that Fowlkes is a measure that can be interpreted as the geometric mean of accuracy (ratio between the number of relevant documents recovered and the total number of documents recovered).

$$Fowlkes = \sqrt{\frac{TP}{TP+FP} * \frac{TP}{TP+FN}} \tag{14}$$

$$Fowlkes = \sqrt{Precision * Recall} \tag{15}$$

- Variation information: Variation in information or distance of shared information is a measure of distance between two groups (Romano et al., 2016). This measure is closely related to mutual information (mutual dependence between the two variables). However, in contrast to mutual information, variation of information is a true metric, in the sense that it is due to the inequality of triangles (for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side).

$$VI = -\sum_{i}^{j} r_{ij} \left[\log\left(\frac{r_{ij}}{p_i}\right) + \left(\frac{r_{ij}}{q_j}\right) \right] \tag{16}$$

As with the external measures, we will now list the most relevant internal measures:

- Connectivity: This measure reflects the extent to which items placed in the same group are considered their closest neighbours in the data space, i.e. the degree of connection of the clusters should be minimal (Deborah et al., 2010).

$$connectivity = min_{1 \le i \le K}\left(min_{1 \le j \le K, i \ne j}\left(\frac{dist(C_i, C_j)}{max_{1 \le k \le K}\{diam(C_k)\}}\right)\right) \tag{17}$$

  Where $dist(C_i, C_j)$ is the distance between two clusters and $diam(C_k)$ is diameter of a particular cluster.

- Dunn: It represents the relationship of the smallest distance between observations that are not in the same cluster and the largest distance within the same cluster (Ansari et al., 2015).

$$dunn = min_{1_{\le} i \le k}\left(min_{i+1 \le j \le k}\left(\frac{dist(C_i, C_j)}{max_{1_{\le} l \le k} diam(C_l)}\right)\right) \tag{18}$$

  Where $dist(C_i, C_j)$ is distance between clusters C_i and C_j and $diam(C_l)$ is diameter of cluster C_l.

- Silhouette index: The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation) (Starczewski and Krzyzak, 2015).

$$S = \frac{1}{N}\sum_{i=0}^{N}\frac{b_i - a_i}{max(a_i, b_i)} \tag{19}$$

where

$$a_i = \frac{1}{|C_j| - 1}\sum_{y \in C_j, y \neq x_i} \|y - x_i\|$$

and

$$b_i = \min_{l \in H, l \neq j} \frac{1}{|C_l|}\sum_{y \in C_l} \|y - x_i\|$$

with

$$x_i \in C_j, H = \{h : 1 \leq h \leq K\}$$

The above evaluation measures can be grouped into families in order to evaluate the quality of the clusters. If we look at the Figure 6 we can group Entropy, Recall, Precision, F-Measure, Fowlkes-Mallows Index and Variation information into three families (Palacio-Niño and Berzal, 2019):

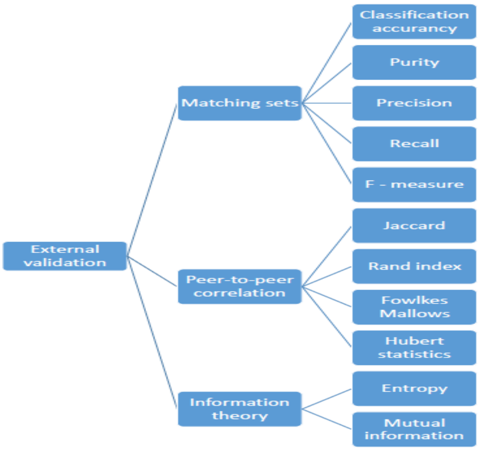


Figure 6: External validation methods (Palacio-Niño and Berzal, 2019).

- Matching Sets: used to compare two partitions of data consists of those method that identify the relationship between each cluster detected in C and its natural correspondence to the classes in the reference result defined by P (clustering result prediction) (Palacio-Niño and Berzal, 2019). Several measures can be defined to measure the similarity between the clusters in C, obtained by the clustering algorithm, and the clusters in P, corresponding to our prior (external) knowledge. The metrics included in this method are: Precision, Recall and F-measure.
- Peer-to-peer Correlation: are based on the correlation between pairs, i.e., they seek to measure the similarity between two partitions under equal conditions, such as the result of a grouping process for the same set, but by means of two different methods C and P (Palacio-Niño and Berzal, 2019). It is assumed that the examples that are in the same cluster in C should be in the same class in P, and viceversa. We highlight the following metrics: Fowlkes-Mallows Index.
- Measures Based on Information Theory: A third family is based on Information Theory concepts, such as the existing uncertainty in the prediction of the natural classes provided by the partition P (Palacio-Niño and Berzal, 2019). This family includes basic measures such as entropy and variation information.

Internal evaluation metrics (see Figure 7) do not require external information, so they are focused on measuring cohesion (how close the elements are to each other) and separation (they quantify the level of separation between clusters).

According to the figure 8, the internal Dunn, Silhouette and Connectivity metrics are based on the concepts mentioned above so we can group them as partitioning methods.

The Clustering package

Clustering package is a package that has been written entirely in R language. The package contains other **Clustering** packages that run hierarchical, partitional and agglomerative hierarchical algorithms.

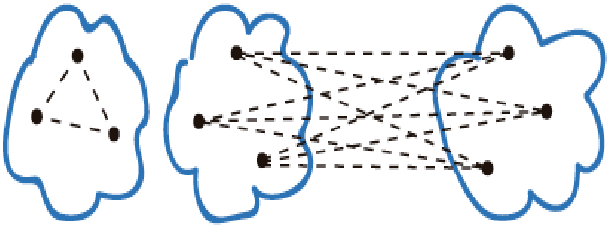


Figure 7: Representation of cohesion and separation in clustering (Palacio-Niño and Berzal, 2019).

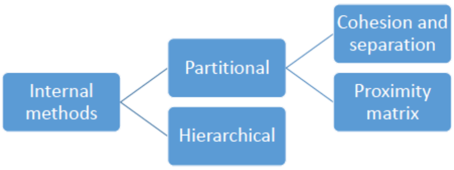


Figure 8: Internal validation methods (Palacio-Niño and Berzal, 2019).

As an addition to the package it has been provided with the ability to read data in different formats such as CSV, KEEL, ARFF (Weka) and `data.frame` objects. Most of the methods have been provided with a set of default parameters, so we can easily run our algorithm without knowing the parameters. Reviewing the related implementations in Clustering so far, if we needed to run many algorithms simultaneously to be able to compare the results we couldn't do it, since we had one by one. In addition, if an algorithm has implemented several measures of similarity, it is necessary to perform as many executions as measures have, so it requires a lot of time and in some cases can lead to confusion in the executions. Besides, when we evaluate the clustering results it is necessary to indicate a variable of the data set, so the execution times increase more and not only that, but also depending on the chosen variable it can vary in the results so with this package we will try to solve this problem. With the Clustering package we will be able to execute simultaneously several algorithms for each of the implemented similarity measures. In addition, when evaluating the results of the executions, we have a set of measures that are executed together, which until now was done one execution per measure. It is possible to incorporate new measures quickly in the future. Another problem we are solving is whether the choice of one or another dataset variable influences the quality of the results. Therefore, this problem is solved by carrying out executions for each of the variables. The package shows the results visually, so we can draw conclusions more quickly.

Algorithms of the package

These are the algorithms available within the package, which we will classify as follows:

- Hierarchical Clustering: `agnes, clara, daisy, diana, fanny, fuzzy_cm, fuzzy_gk, fuzzy_gg, hcluster, mona, pam, pvpick and pvclust.`
- Partitioning Clustering: `gama, gmm, kmeans`
- Aglomerative Clustering: `aggExCluster, apclusterK`

Package Architecture

The main class of the package is the **Clustering** object.

- `clustering`: This object stores the results of the **Clustering** package execution and contains the following properties:
  - `result.` It represents the `data.frame` with the results. In each column we have represented the evaluation metrics used to evaluate the clusters. We can also see the execution time of these metrics, datasets, the calculated variables, the measures of similarity and the algorithms.

    - `has_internal_metrics`. It is a boolean operator that indicates if we have used internal evaluation measures in the calculation. It serves to indicate if we have classified the data correctly.
    - `has_external_metrics`. It is a boolean operator that indicates if we have used external evaluation measures in the calculation.
    - `algorithms_executed`. It represents a character vector with the algorithms executed independently of the package.
    - `measures_executed`. It represents a vector of characters with the measures of similarity used by the indicated algorithms.

  This class also exports the well-known S3 methods print() and summary() that show the data structure without codification and a summary with basic information about the dataset respectively. We can also perform sorting and filtering operations for further processing of the results. In any case if we need to perform filtering operations we can overload the operator (’[’) to perform such operations in an easier way.

- `best_ranked_external_metrics()`, `evaluate_validation_external_by_metrics()`, `evaluate_best_validation_external_by_metrics()`, `result_external_algorithm_by_metric()`: These are methods for working with the results of external metrics. The methods indicated allow us to determine the behaviour of the algorithms based on the best variable, on the measures of similarity and on the number of clusters. This translates as follows. If we have a dataset with five variables, using as similarity measures Euclidean and Manhattan and with k partitions the method best_ranked_external_metrics() will return us those variables that better result return for the indicated algorithms, similarity measures and partitions. If what we want is to group the results by algorithm and similarity measure and obtain the results based on these properties the method we must use is evaluate_best_validation_external_by_metrics(). Another method that allows us to obtain the results of the evaluation metrics by algorithm is evaluate_validation_external_by_metrics(). If we simply want to group the results of the algorithms by number of clusters we must use the result_external_algorithm_by_metric() method. With this set of methods we manage to group the data obtained and see their behaviour based on the number of partitions, metrics or similarity measures. These methods are used in external evaluation measures.
- `best_ranked_internal_metrics()`, `evaluate_validation_internal_by_metrics()` , `evaluate_best_validation_internal_by_metrics()`, `result_internal_algorithm_by_metric()`: Just as we have indicated that there are methods for working with external metrics, we also have them for internal ones. The methods indicated allow us to determine the behaviour of the algorithms based on the best variable, on the measures of similarity and on the number of clusters. The operation is similar to those indicated above for external evaluation measures.
- `plot_clustering`: Method that represents the results of clustering in a bar chart. The graph represents the distribution of the algorithms based on the number of partitions and the evaluation metrics which can be internal or external.
- `export_external_file`: The results of external metrics can be exported in Latex format, for integration into documents with that format.
- `export_internal_file`: As indicated above, we have this method to export the results of the internal metrics.

Use of Clustering package

The fastest way to download the **Clustering** package and use it is to use the install instruction.

```
install.packages("Clustering")
```

A development version is also available on the github `https://github.com/laperez/Clustering`. To use the development version you must install the **devtools** package and use the `install_github method`.

```
devtools::install_github('laperez/Clustering')
```

The main dependencies of the **Clustering** package are: **advclust**, **amap**, **apcluster**, **cluster**, **ClusterR**, **gmp** and **pvclust**. These are the packages in charge of implementing the clustering algorithms. We can also find dependencies for data processing and GUI, such as **shiny** and **DT** among others. Once the package is installed it is necessary to load it in the following way:

```
library("Clustering")
```

Once the installation and loading process has been completed, we proceed with the processing of the data and its execution.

Use and load of datasets

For the execution of the main method of the package we must provide you with data that can be in different formats. The file formats accepted by the package are: KEEL, ARFF and CSV. The data can be loaded in two ways, firstly we can indicate a directory with files in the formats indicated above and load all the available files and secondly we can provide a `data.frame` with the necessary data for execution. To read the files in ARFF format it has been extracted from the **mldr** package (Charte et al., 2016).
If we need to work with test data, we have pre-loaded data. The loaded datasets have been obtained from the KEEL repository url `https://sci2s.ugr.es/keel/category.php?cat=uns` in csv format. Note that the extension is used to determine the type of file format.

Analysis of clustering methods using the Clustering package

Once the way to provide the data has been defined the next step is to be able to execute the main method of the application which is `clustering`. With this method we can compare the clustering algorithms included in the mentioned packages. When comparing we can do it by packages or simply by indicating the algorithms contained in them. In partitional clustering it is necessary to indicate the number of partitions. Apart from indicating the number of partitions, we can evaluate the algorithms by means of a range of partitions, indicating the maximum and minimum of partitions. To evaluate how the data have been distributed in the clusters, it is done through a set of evaluation measures that return numeric values. An improvement incorporated in the package is that instead of returning the numerical value, it can return the dataset variable corresponding to that value. In addition, the algorithms are executed for all measures of similarity implemented. All this functionality is incorporated into the main method. Therefore the parameters of the `clustering` method are the following:

- `path`: The path of file. It is only allowed to use path or df but not both at the same time. Only files in .dat, .csv or arff format are allowed.
- `df`: data matrix or data frame, or similarity matrix.
- `packages`: character vector with the packets running the algorithm. The seven packages implemented are: cluster, ClusterR, advclust, amap, apcluster, gama, pvclust. By default runs all packages.
- `algorithm`: is an array with the list of the algorithms implemented by the packages. The algorithms are: `fuzzy_cm,fuzzy_gg,fuzzy_gk,hclust,apclusterK,agnes,clara, daisy,diana,fanny,mona,pam,gmm,kmeans_arma,kmeans_rcpp,mini_kmeans,gama,pvclust`.
- `min`: An integer with the minimum number of clusters This data is necessary to indicate the minimum number of clusters when grouping the data. The default value is 3.
- `max`: An integer with the maximum number of clusters. This data is necessary to indicate the maximum number of clusters when grouping the data. The default value is 4.
- `metrics`: Character vector with the metrics implemented to evaluate the distribution of the data in clusters. The night metrics implemented are: `entropy, variation_information, precision,recall,f_measure,fowlkes_mallows_index,connectivity,dunn,silhouette`.
- `variables`: an boolean which indicates that if we want to show as a result the variables of the datasets or the numerical value of the calculation of the metrics. The default value is F.

Once the definition of the attributes of the main method has been completed, we will carry out a test. To do this we will use the `data.frame` Basketball included in the package, and the gmm and fanny algorithms (included in the **ClusterR** and **cluster** packages). We will also indicate a range of partitions between [3.5] and evaluate entropy as an external evaluation measure and dunn as an internal one.

```
> result <- Clustering::clustering(df = Clustering::basketball, min = 3, max = 5,

            algorithm = c('gmm', 'fanny'), metrics = c('entropy','dunn'))

Algorithm    Distance     Clusters    Dataset    timeExternal  entropy   dunn   timeInternal
     gmm   gmm_euclidean    3       dataframe    0.0054      0.2374  0.1096    0.0004
```

```
 gmm   gmm_euclidean    3       dataframe    0.0091      0.2120  0.1096    0.0005
 gmm   gmm_euclidean    3       dataframe    0.0093      0.0064  0.1096    0.0005
 gmm   gmm_euclidean    3       dataframe    0.0107      0.0032  0.1096    0.0007
 gmm   gmm_euclidean    3       dataframe    0.0185      0.0000  0.1096    0.0013
 gmm   gmm_euclidean    4       dataframe    0.0053      0.3734  0.1233    0.0004
 gmm   gmm_euclidean    4       dataframe    0.0111      0.2983  0.1233    0.0005
 gmm   gmm_euclidean    4       dataframe    0.0119      0.0064  0.1233    0.0005
 gmm   gmm_euclidean    4       dataframe    0.0122      0.0032  0.1233    0.0006
 gmm   gmm_euclidean    4       dataframe    0.0649      0.0000  0.1233    0.0007
 gmm   gmm_euclidean    5       dataframe    0.0050      0.4175  0.1619    0.0004
 gmm   gmm_euclidean    5       dataframe    0.0051      0.3857  0.1619    0.0004
 gmm   gmm_euclidean    5       dataframe    0.0086      0.0064  0.1619    0.0004
 gmm   gmm_euclidean    5       dataframe    0.0088      0.0032  0.1619    0.0005
 gmm   gmm_euclidean    5       dataframe    0.0100      0.0000  0.1619    0.0005
 gmm   gmm_manhattan    3       dataframe    0.0032      0.2498  0.1151    0.0004
 gmm   gmm_manhattan    3       dataframe    0.0038      0.2201  0.1151    0.0004
 gmm   gmm_manhattan    3       dataframe    0.0070      0.0064  0.1151    0.0005
 gmm   gmm_manhattan    3       dataframe    0.0109      0.0032  0.1151    0.0005
 gmm   gmm_manhattan    3       dataframe    0.1737      0.0000  0.1151    0.0008
 gmm   gmm_manhattan    4       dataframe    0.0031      0.3563  0.1179    0.0004
 gmm   gmm_manhattan    4       dataframe    0.0055      0.2919  0.1179    0.0004
 gmm   gmm_manhattan    4       dataframe    0.0063      0.0064  0.1179    0.0004
 gmm   gmm_manhattan    4       dataframe    0.0071      0.0032  0.1179    0.0006
 gmm   gmm_manhattan    4       dataframe    0.0096      0.0000  0.1179    0.0007
 gmm   gmm_manhattan    5       dataframe    0.0036      0.4290  0.1141    0.0004
 gmm   gmm_manhattan    5       dataframe    0.0040      0.3887  0.1141    0.0004
 gmm   gmm_manhattan    5       dataframe    0.0067      0.0064  0.1141    0.0004
 gmm   gmm_manhattan    5       dataframe    0.0076      0.0032  0.1141    0.0004
 gmm   gmm_manhattan    5       dataframe    0.0083      0.0000  0.1141    0.0005
fanny fanny_euclidean   3       dataframe    0.0121      0.2069  0.0000    0.0000
fanny fanny_euclidean   3       dataframe    0.0125      0.1675  0.0000    0.0000
fanny fanny_euclidean   3       dataframe    0.0152      0.0032  0.0000    0.0000
fanny fanny_euclidean   3       dataframe    0.0178      0.0032  0.0000    0.0000
fanny fanny_euclidean   3       dataframe    0.0218      0.0000  0.0000    0.0000
fanny fanny_euclidean   4       dataframe    0.0161      0.2069  0.0000    0.0000
fanny fanny_euclidean   4       dataframe    0.0190      0.1675  0.0000    0.0000
fanny fanny_euclidean   4       dataframe    0.0205      0.0032  0.0000    0.0000
fanny fanny_euclidean   4       dataframe    0.0208      0.0032  0.0000    0.0000
fanny fanny_euclidean   4       dataframe    0.0265      0.0000  0.0000    0.0000
fanny fanny_euclidean   5       dataframe    0.0165      0.2069  0.0000    0.0000
fanny fanny_euclidean   5       dataframe    0.0171      0.1675  0.0000    0.0000
fanny fanny_euclidean   5       dataframe    0.0221      0.0032  0.0000    0.0000
fanny fanny_euclidean   5       dataframe    0.0226      0.0032  0.0000    0.0000
fanny fanny_euclidean   5       dataframe    0.0250      0.0000  0.0000    0.0000
fanny fanny_manhattan   3       dataframe    0.0156      0.2143  0.0000    0.0000
fanny fanny_manhattan   3       dataframe    0.0180      0.1658  0.0000    0.0000
fanny fanny_manhattan   3       dataframe    0.0201      0.0032  0.0000    0.0000
fanny fanny_manhattan   3       dataframe    0.0238      0.0032  0.0000    0.0000
fanny fanny_manhattan   3       dataframe    0.0278      0.0000  0.0000    0.0000
fanny fanny_manhattan   4       dataframe    0.0171      0.2143  0.0000    0.0000
fanny fanny_manhattan   4       dataframe    0.0181      0.1658  0.0000    0.0000
fanny fanny_manhattan   4       dataframe    0.0225      0.0032  0.0000    0.0000
fanny fanny_manhattan   4       dataframe    0.0266      0.0032  0.0000    0.0000
fanny fanny_manhattan   4       dataframe    0.0279      0.0000  0.0000    0.0000
fanny fanny_manhattan   5       dataframe    0.0235      0.2143  0.0000    0.0000
fanny fanny_manhattan   5       dataframe    0.0242      0.1658  0.0000    0.0000
fanny fanny_manhattan   5       dataframe    0.0259      0.0032  0.0000    0.0000
fanny fanny_manhattan   5       dataframe    0.0262      0.0032  0.0000    0.0000
fanny fanny_manhattan   5       dataframe    0.0271      0.0000  0.0000    0.0000
```

These are the results of running the `clustering` method. The meaning and functionality of each column is as follows:

- `Algorithm`: indicates the clustering algorithm used in the data processing.
- `Distance`: is the measure of similarity used by the algorithm to calculate the similarity between

the data.

- `Clusters`: is the number of clusters used by the algorithm. Used in Partitional Clustering.
- `Dataset`: is the name of the `data.frame`. By default appears `dataframe` but if instead of using the `df` parameter in the clustering method we use path (directory with files with extension dat), in the column must appear the names of the processed datasets.
- `timeExternal`: time to implement external evaluation measures.
- `metrics`: each metric indicated in the execution is presented in individual columns. In this case we have both external (entropy) and internal (dunn) metrics. Note: in the metric field we indicate all the measurements we wish to evaluate. The implemented metrics are: `entropy`, `recall`, `precision`, `f_measure`,`fowlkes_mallows_index`,`connectivity`,`dunn` and `silhouette`.
- `timeInternal`: time taken to implement internal evaluation measures.

The basketball data employed in this example contains five variables. The idea is to demonstrate one of the main functionalities of the package, which is to see if the choice of variables influences the results. To do this we have executed the same algorithm with the same measure of similarity and the same number of clusters for each variable. The results obtained in the execution can already be appreciated. And not only that, but we also observe that the choice of the measure of similarity also has an influence.

```
> Clustering::best_ranked_external_metrics(result)

Result:
Algorithm  Distance        Clusters   Dataset  timeExternal  entropy
  gmm        gmm_euclidean     3      dataframe    0.0047    0.2374
  gmm        gmm_euclidean     4      dataframe    0.0050    0.3734
  gmm        gmm_euclidean     5      dataframe    0.0059    0.4175
  gmm        gmm_manhattan     3      dataframe    0.0030    0.2498
  gmm        gmm_manhattan     4      dataframe    0.0032    0.3563
  gmm        gmm_manhattan     5      dataframe    0.0041    0.4290
fanny      fanny_euclidean     3      dataframe    0.0117    0.2069
fanny      fanny_euclidean     4      dataframe    0.0148    0.2069
fanny      fanny_euclidean     5      dataframe    0.0187    0.2069
fanny      fanny_manhattan     3      dataframe    0.0159    0.2143
fanny      fanny_manhattan     4      dataframe    0.0189    0.2143
fanny      fanny_manhattan     5      dataframe    0.0202    0.2143
```

`Clustering::best_ranked_external_metrics` method is in charge of selecting from the data set variables, those that obtain the best result in the evaluation of the measure. In the calculation of the `entropy` the results are in the interval [0.1]. Being the next to 1 very good. For this particular example, select the variables of the data set whose value in `entropy` is close to 1. We perform the same calculation for internal measurements.

```
> Clustering::best_ranked_internal_metrics(result)

Result:
Algorithm      Distance     Clusters   Dataset  timeInternal   dunn
   gmm       gmm_euclidean     3      dataframe   0.0005      0.1096
   gmm       gmm_euclidean     4      dataframe   0.0004      0.1233
   gmm       gmm_euclidean     5      dataframe   0.0004      0.1619
   gmm       gmm_manhattan     3      dataframe   0.0004      0.1151
   gmm       gmm_manhattan     4      dataframe   0.0004      0.1179
   gmm       gmm_manhattan     5      dataframe   0.0004      0.1141
 fanny     fanny_euclidean     3      dataframe   0.0000      0.0000
 fanny     fanny_euclidean     4      dataframe   0.0000      0.0000
 fanny     fanny_euclidean     5      dataframe   0.0000      0.0000
 fanny     fanny_manhattan     3      dataframe   0.0000      0.0000
 fanny     fanny_manhattan     4      dataframe   0.0000      0.0000
 fanny     fanny_manhattan     5      dataframe   0.0000      0.0000
```

We already have the best variables for each execution, we also have methods to group the measures of similarity from the algorithms. When grouping the results by measures of dissimilarity and algorithm, we do not use a specific grouping algorithm, but we keep those values whose value is the maximum depending on the type of metric. In this case we see that for the fanny algorithm with similarity measure euclidean and taking into account the number of clusters, the value closest in entropy to 1 is 0.2090. For the rest of the algorithms the same process is followed.

```
> Clustering::evaluate_best_validation_external_by_metrics(result)

Result:
Algorithm     Distance      timeExternal entropy
 fanny     fanny_euclidean      0.0163   0.2069
 fanny     fanny_manhattan      0.0209   0.2143
  gmm       gmm_euclidean       0.0076   0.4175
  gmm       gmm_manhattan       0.0039   0.429
```

With this method we intend to demonstrate whether the choice of measurement of similarity also has an influence. In this case we have that the best result in entropy is achieved with 5 clusters. If we look at the results for the gmm algorithm we see that those of the manhattan measurement are superior to euclidean.
If we want to go further and we want to determine the best algorithm from the variables, we can do it in the following way.

```
> Clustering::evaluate_validation_external_by_metrics(result)

Result:
  Algorithm timeExternal entropy
    fanny     0.0209     0.2143
    gmm       0.0076     0.4290
```

With Clustering::evaluate_validation_external_by_metrics we can see that the most correct algorithm for the dataset is gmm algorithm, as the value of entropy is closer to 1.

As an addition the Clustering::result_external_algorithm_by_metric method has been incorporated to filter the results of the clustering object from an algorithm to be able to choose a suitable cluster.

```
> Clustering::result_external_algorithm_by_metric(result,'gmm')

Result:
Algorithm Clusters timeExternal entropy
    gmm      3       0.0045      0.2498
    gmm      4       0.0056      0.3734
    gmm      5       0.0047      0.429
```

On the basis of the executions carried out we can state that the gmm algorithm with five clusters is the best distributed data for the measurement of manhattan similarity.

All these operations that we have carried out to evaluate the external measures can be extrapolated to the internal ones and obtain the necessary information for the appropriate choice of the algorithm as well as the number of clusters. Another feature incorporated in the package is the possibility of being able to represent the evaluation metrics according to the number of clusters, so that in some cases you can be quite quick in choosing the best results. Figure 9 shows this representation. The method that represents the data graphically is Clustering::plot_clustering which receives as parameter the metric.

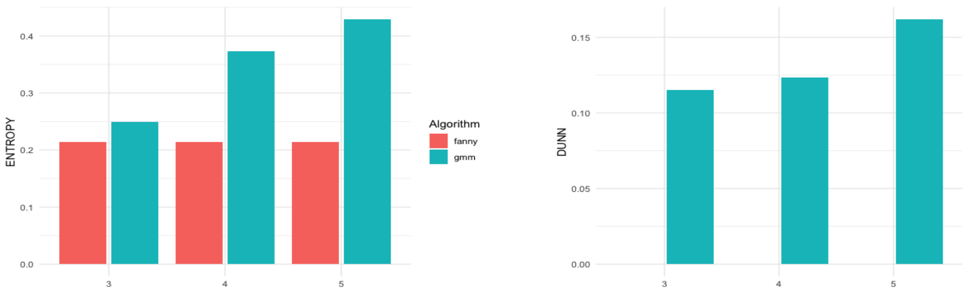


Figure 9: Graphical representation of evaluation measures.

Post-processing of data

To conclude the definition of the methods of the package it is always necessary to have a functionality that allows us to sort, filter the information or export the data set. To do this we will detail the methods used:

- `sort(clustering_object,column_name)`: Sometimes it is necessary to determine what the maximum or minimum value of a column is and the easiest way to do this is by sorting the column. Here is an example of how to sort by column.

```
> result <- Clustering::clustering(df = Clustering::basketball, min = 3, max=3,

           algorithm = c('gmm'), metrics = c('entropy','dunn'))

>  sort(result,F,'entropy')

Result:
 Algorithm    Distance    Clusters   Dataset   entropy  dunn
   gmm    gmm_euclidean   3       dataframe  0.0000  0.1096
   gmm    gmm_manhattan   3       dataframe  0.0000  0.1151
   gmm    gmm_euclidean   3       dataframe  0.0032  0.1096
   gmm    gmm_manhattan   3       dataframe  0.0032  0.1151
   gmm    gmm_euclidean   3       dataframe  0.0064  0.1096
   gmm    gmm_manhattan   3       dataframe  0.0064  0.1151
   gmm    gmm_euclidean   3       dataframe  0.2120  0.1096
   gmm    gmm_manhattan   3       dataframe  0.2201  0.1151
   gmm    gmm_euclidean   3       dataframe  0.2374  0.1096
   gmm    gmm_manhattan   3       dataframe  0.2498  0.1151
```

- `"[.clustering"`: There are times when we need to apply filters on a series of columns for a set of values. This process can be done using third party packages, but due to its great usefulness we have incorporated this functionality. To filter we must do it in the following way: clustering_object [column_1 operator value_1 conditional_1 .... column_n operator value_n]. Example of filtering:

```
> result[entropy > 0.11 & dunn > 0.11 ]

Result:
Algorithm   Distance      Clusters   Dataset    entropy dunn
   gmm     gmm_manhattan      3     dataframe   0.2498 0.1151
   gmm     gmm_manhattan      3     dataframe   0.2201 0.1151
```

- `Clustering::export_file_external()`: exports the results of the clustering object to latex format This method is very useful when working with documents in latex format.
- `Clustering::export_file_internal()`: this method is similar to the previous one, but only exports the internal metrics.

Graphical User Interface of the Clustering package

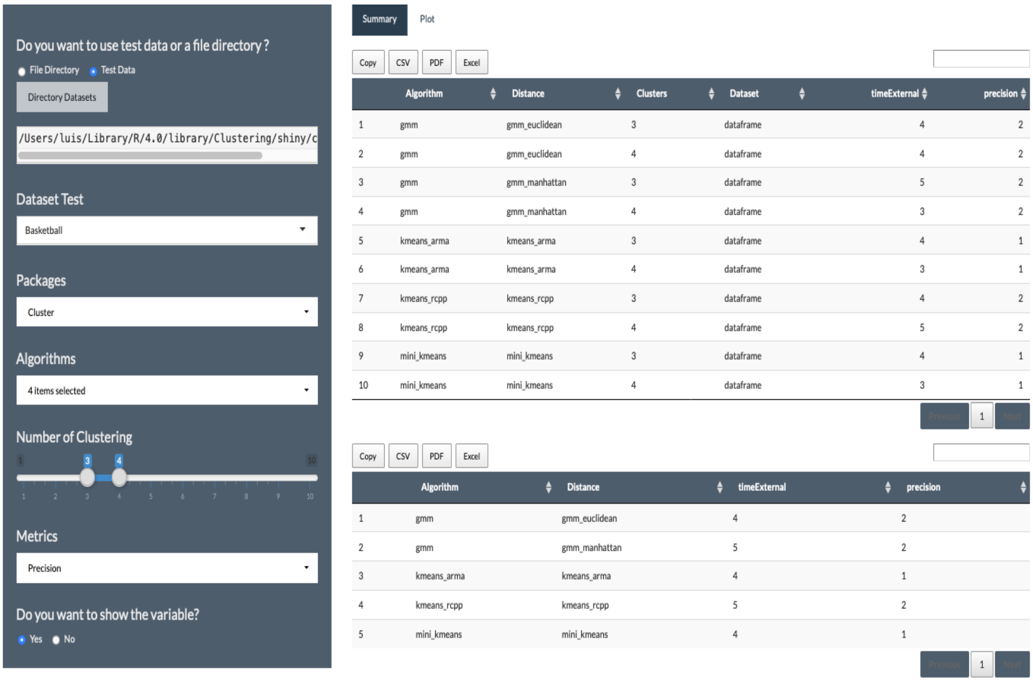
As mentioned throughout this paper, the **Clustering** package provides a GUI to work with clustering algorithms and to be able to evaluate and run the results more comfortable. The way to run the user interface is to execute the following instruction:

```
> Clustering::appClustering()
```

The execution will open our default browser with the interface. As it can be observed in the Figure 10, we have a layout with header, side menu and main. In the header menu we can choose to see the numerical results or in graphical mode. In the left menu we can see the different parameters with which we can run our algorithm and finally in the central menu you can see the result of running the clustering algorithm.

The operation of the application is very simple as can be seen in the Figure 11 and we will proceed to explain it step by step.

As you can see in the Figure 11 we have two well differentiated parts:



Do you want to use test data or a file directory ?

○ File Directory ● Test Data

Directory Datasets

/Users/luis/Library/R/4.0/library/Clustering/shiny/c

Dataset Test: Basketball

Packages: Cluster

Algorithms: 4 items selected

Number of Clustering: 3 – 4

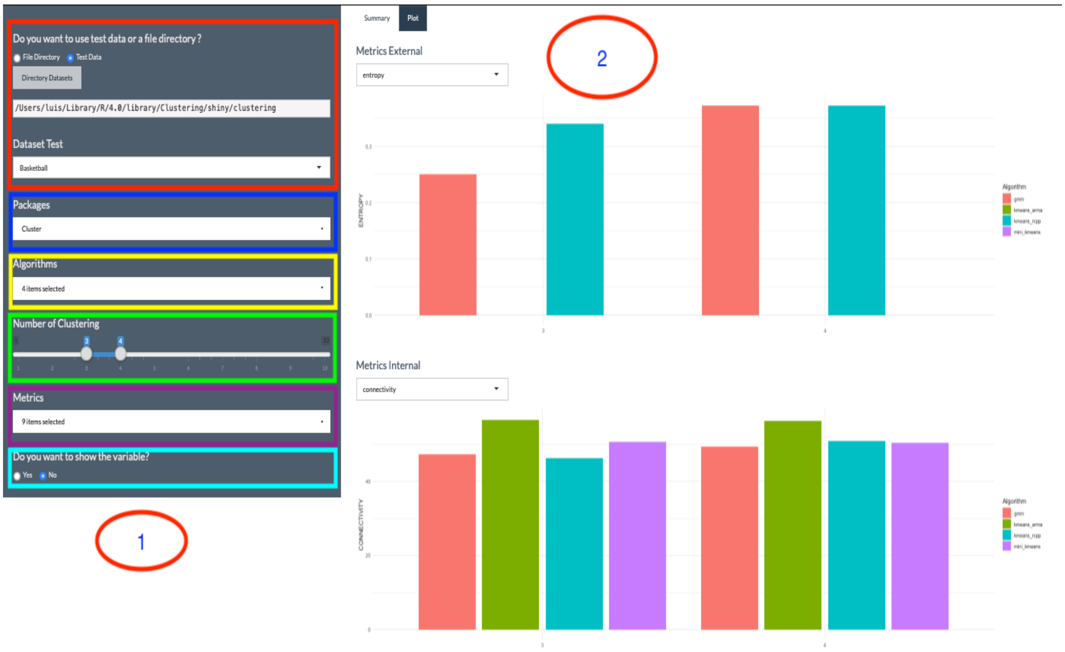
Metrics: Precision

Do you want to show the variable? ● Yes ○ No

	Algorithm	Distance	Clusters	Dataset	timeExternal	precision
1	gmm	gmm_euclidean	3	dataframe	4	2
2	gmm	gmm_euclidean	4	dataframe	4	2
3	gmm	gmm_manhattan	3	dataframe	5	2
4	gmm	gmm_manhattan	4	dataframe	3	2
5	kmeans_arma	kmeans_arma	3	dataframe	4	1
6	kmeans_arma	kmeans_arma	4	dataframe	3	1
7	kmeans_rcpp	kmeans_rcpp	3	dataframe	4	2
8	kmeans_rcpp	kmeans_rcpp	4	dataframe	5	2
9	mini_kmeans	mini_kmeans	3	dataframe	4	1
10	mini_kmeans	mini_kmeans	4	dataframe	3	1

	Algorithm	Distance	timeExternal	precision
1	gmm	gmm_euclidean	4	2
2	gmm	gmm_manhattan	5	2
3	kmeans_arma	kmeans_arma	4	1
4	kmeans_rcpp	kmeans_rcpp	5	2
5	mini_kmeans	mini_kmeans	4	1

Figure 10: Clustering app user interface.



Do you want to use test data or a file directory ?

○ File Directory ● Test Data

Directory Datasets

/Users/luis/Library/R/4.0/library/Clustering/shiny/clustering

Dataset Test: Basketball

Packages: Cluster

Algorithms: 4 items selected

Number of Clustering: 3 – 4

Metrics: 9 items selected

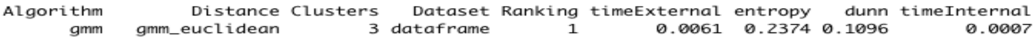
Do you want to show the variable? ○ Yes ● No

Metrics External: entropy

Metrics Internal: connectivity

Figure 11: Clustering app user interface.

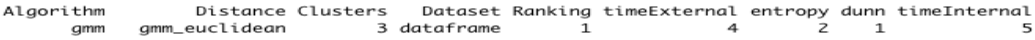
1. In this section we can find the different parameters used by the clustering function to filter the information.
    - Marked in red, we can indicate if we want to work with test datasets or indicate a directory of dataset files to be processed.
    - In blue we have the packages that implement the clustering algorithms mentioned throughout the paper. We can mark all the packages or individually. When a package is marked, all the algorithms implemented within the selected package are also marked.
    - In yellow we have the algorithms implemented by the packages. If we mark an algorithm it will automatically mark its corresponding package in the package combo.
    - In green we have the number of clusters. We can indicate ranges or select only one cluster by positioning the max and minimum on the same value.
    - In violet we indicate the evaluation metrics used when validating the clusters.
    - Finally we have a check through which instead of showing the metric values we can show the the dataset variable. In the image we have an example of execution with default variables.



```
Algorithm         Distance Clusters   Dataset Ranking timeExternal entropy   dunn timeInternal
      gmm    gmm_euclidean        3 dataframe       1       0.0061  0.2374 0.1096       0.0007
```

Figure 12: Execution of the default `clustering` method.

In the Figure 13 it is the execution of the same method with variable to true.



```
Algorithm         Distance Clusters   Dataset Ranking timeExternal entropy dunn timeInternal
      gmm    gmm_euclidean        3 dataframe       1            4       2    1            5
```

Figure 13: Execution of the `clustering` method with variable to true.

Note: The variables in the data set can be translated into whole numbers. As we have worked with the basketball dataset throughout the paper, the dataset contains the following variables: `assists_per_minute, height, time_played, age, points_per_minute`. These variables are translated into whole numbers from left to right, i.e. `assists_per_minute` corresponds to 1, `height` to 2 and so on. This is ideal, when instead of showing the numerical values of the metrics we need to see the variables of the set.

2. In main layout we have the options to represent the data.
    - To view the data in graphical mode as shown in the Figure 14, we mark the Plot tab. In the figure we can see represented the internal and external evaluation metrics and depending on the type of evaluation we can filter individually by metrics to see the data represented graphically.

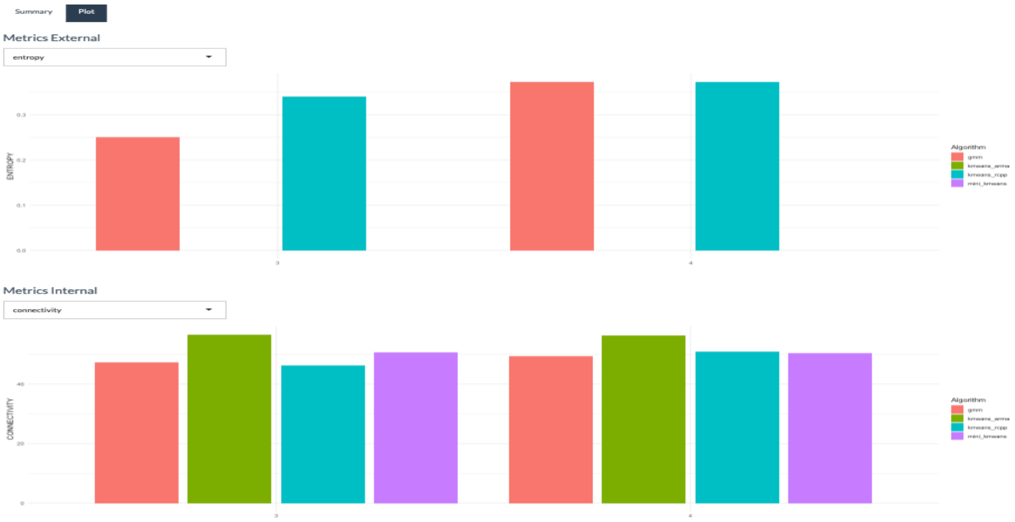
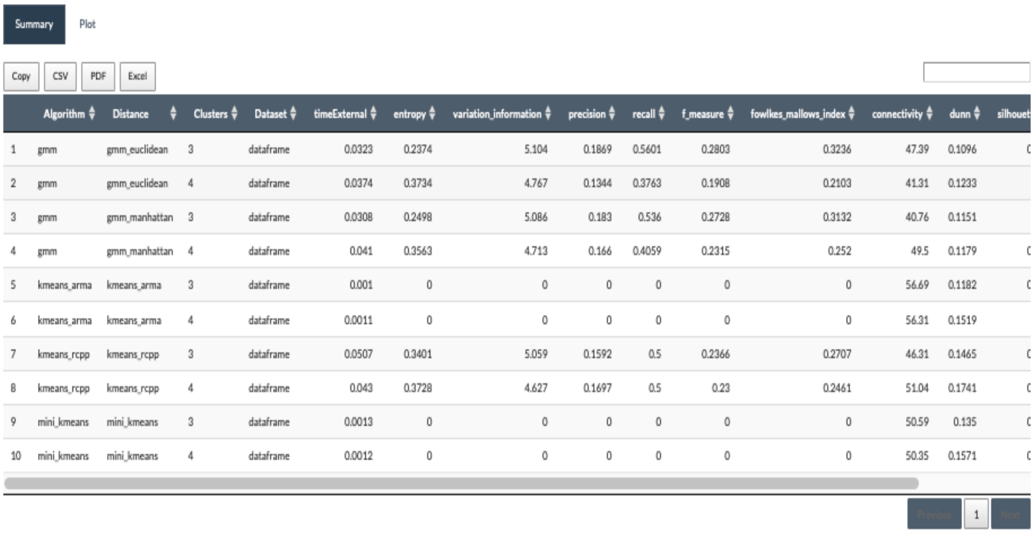


Figure 14: Tab with graphical representation of metrics.

- If we click on the summary tab as shown in the Figure 15, we can see the data represented in tables. If you wish you can export the results in the following formats: csv,pdf and xls. We also have the option of copying the data.



	Algorithm	Distance	Clusters	Dataset	timeExternal	entropy	variation_information	precision	recall	f_measure	fowlkes_mallows_index	connectivity	dunn	silhouet
1	gmm	gmm_euclidean	3	dataframe	0.0323	0.2374	5.104	0.1869	0.5601	0.2803	0.3236	47.39	0.1096	C
2	gmm	gmm_euclidean	4	dataframe	0.0374	0.3734	4.767	0.1344	0.3763	0.1908	0.2103	41.31	0.1233	
3	gmm	gmm_manhattan	3	dataframe	0.0308	0.2498	5.086	0.183	0.536	0.2728	0.3132	40.76	0.1151	
4	gmm	gmm_manhattan	4	dataframe	0.041	0.3563	4.713	0.166	0.4059	0.2315	0.252	49.5	0.1179	C
5	kmeans_arma	kmeans_arma	3	dataframe	0.001	0	0	0	0	0	0	56.69	0.1182	C
6	kmeans_arma	kmeans_arma	4	dataframe	0.0011	0	0	0	0	0	0	56.31	0.1519	
7	kmeans_rcpp	kmeans_rcpp	3	dataframe	0.0507	0.3401	5.059	0.1592	0.5	0.2366	0.2707	46.31	0.1465	C
8	kmeans_rcpp	kmeans_rcpp	4	dataframe	0.043	0.3728	4.627	0.1697	0.5	0.23	0.2461	51.04	0.1741	C
9	mini_kmeans	mini_kmeans	3	dataframe	0.0013	0	0	0	0	0	0	50.59	0.135	C
10	mini_kmeans	mini_kmeans	4	dataframe	0.0012	0	0	0	0	0	0	50.35	0.1571	C

Figure 15: Clustering package execution summary tab.

Conclusion

In this paper we have made an introduction to the **Clustering** package. The package has dependencies with other packages as seen throughout the paper. It allows the reading and loading of datasets in KEEL, CSV or ARFF format. We also offer the functionality of loading a `data.frame` in memory or using test datasets. As a complement the package has been enhanced with the inclusion of a graphical interface that allows the user to run the package in a simple way without the need to know the parameters. The development of the package will be continued with the inclusion of new algorithms, functionalities and improvement of the interface, therefore we encourage developers to contribute to the improvement of the package with the inclusion of new algorithms or functionalities or the inclusion of new proposals that complement the package.

Bibliography

Z. Ansari, M. Azeem, W. Ahmed, and A. Babu. Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. *World of Computer Science and Information Technology Journal*, 1, 07 2015. [p6]

E. Balevi and R. D. Gitlin. A Clustering Algorithm That Maximizes Throughput in 5G Heterogeneous F-RAN Networks. Technical report, 2018. URL `http://iwinlab.eng.usf.edu/papers/ICC{_}Clustering{_}Fog{_}Final.pdf`. [p1]

M. J. Berry and G. S. Linoff. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004. [p5]

F. Charte, A. M. Vico, P. González, C. J. Carmona, and M. J. Del Jesus. Subgroup discovery with evolutionary fuzzy systems in r: The sdefsr package. *The R Journal*, 8:307–323, 01 2016. [p10]

S. Dang. A review of clustering techiques in various applications for effective data mining. *International Journal of Research in IT and Management 2231-4434*, 1:50–66, 01 2011. [p3]

L. J. Deborah, R. Baskaran, and A. Kannan. A survey on internal validity measure for cluster validation. *International Journal of Computer Science and Engineering Survey*, 1:85–102, 2010. [p6]

C. C. Farias, C. V. Ubierna, P. B. Elorza, M. P. D. Santamaria, and J. T. Laso. A comparison of fuzzy clustering algorithms applied to feature extraction on vineyard. Noviembre 2011. URL `http://oa.upm.es/9246/`. LPF-TAGRALIA. [p1]

A. Fernández, J. Luengo, J. Derrac, J. Alcala-Fdez, and F. Herrera. Implementation and integration of algorithms into the keel data-mining software tool. 5788:562–569, 09 2009. doi: 10.1007/978-3-642-04394-9_68. [p1]

B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814): 972–976, 2007. ISSN 0036-8075. doi: 10.1126/science.1136800. URL https://science.sciencemag.org/content/315/5814/972. [p1]

G. Gan, C. Ma, and J. Wu. Data clustering - theory, algorithms, and applications. 2007. [p4]

S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. *SIGMOD Rec.*, 27(2):73–84, June 1998. ISSN 0163-5808. [p2]

D. Guo, J. Zhao, and J. Liu. Research and application of improved chameleon algorithm based on condensed hierarchical clustering method. page 14–18, 2019. doi: 10.1145/3375998.3376016. URL https://doi.org/10.1145/3375998.3376016. [p2]

M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17, 10 2001. doi: 10.1023/A:1012801612483. [p5]

M. M. Hasnat and S. Hasan. Identifying tourists and analyzing spatial patterns of their destinations from location-based social media data. *Transportation Research Part C: Emerging Technologies*, 96 (September):38–54, 2018. ISSN 0968090X. doi: 10.1016/j.trc.2018.09.006. URL https://doi.org/10.1016/j.trc.2018.09.006. [p1]

S. Hu. Indoor location method based on data mining. In *Proceedings of the 2019 5th International Conference on Systems, Control and Communications*, ICSCC 2019, page 11–15, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450372640. [p3, 4]

J. Z. Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *DMKD*, 1997. [p2]

J. A. Irani, N. Pise, and M. Phatak. Clustering techniques and the similarity measures used in clustering: A survey. *International Journal of Computer Applications*, 134:9–14, 2016. [p4]

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, Sept. 1999a. ISSN 0360-0300. doi: 10.1145/331499.331504. [p2]

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. 31(3), 1999b. ISSN 0360-0300. [p4]

R. A. Johnson and D. W. Wichern. Clustering, Distance Methods and Ordination. *Applied Multivariate Statistical Analysis*, pages 729–799, 1998. [p5]

J. Kacprzyk and Y. Farhaoui. *Big Data and Smart Digital Environment*. 01 2019. ISBN 2197-6503. doi: 10.1007/978-3-030-12048-1-2. [p6]

M. S. G. Karypis, V. Kumar, and M. Steinbach. A comparison of document clustering techniques. In *TextMining Workshop at KDD2000 (May 2000)*, 2000. [p1]

M. Khader and G. Al-Naymat. An overview of various enhancements of denclue algorithm. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, DATA 19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450372848. [p3]

H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 05 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm134. URL https://doi.org/10.1093/bioinformatics/btm134. [p5]

S. Kosub. A note on the triangle inequality for the jaccard distance. *Pattern Recognition Letters*, 120: 36 – 38, 2019. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2018.12.007. URL http://www.sciencedirect.com/science/article/pii/S0167865518309188. [p4]

S. Kotsiantis and P. Pintelas. Recent advances in clustering: A brief survey. *WSEAS Transactions on Information Science and Applications*, 1:73–81, 01 2004. [p2]

Kushwaha, Mohit, Y. Himanshu, Agrawal, and Chetan. A review on enhancement to standard k-means clustering. In Shukla, R. Kumar, J. Agrawal, Sharma, Sanjeev, Chaudhari, N. S., and K. K. Shukla, editors, *Social Networking and Computational Intelligence*, pages 313–326, Singapore, 2020. Springer Singapore. ISBN 978-981-15-2071-6. [p2]

R. Litoriya. Comparison of the various clustering algorithms of weka tools. 2:73–80, 05 2012. [p1]

J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. [p1]

D. C. A. Mallqui and R. A. S. Fernandes. Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. *Applied Soft Computing Journal*, 75:596–606, 2018. ISSN 1568-4946. doi: 10.1016/j.asoc.2018.11.038. URL https://doi.org/10.1016/j.asoc.2018.11.038. [p1]

A. K. Mann and N. Kaur. Paper on clustering techniques. 2013. [p1]

V. J. Metsalu T. Clustvis: a web tool for visualizing clustering of multivariate data using principal component analysis and heatmap-. 2015. [p1]

I. MR and D. MOHAN. A survey of grid based clustering algorithms. *International Journal of Engineering Science and Technology*, 2, 08 2010. [p3]

M. Mächler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, M. Studer, P. Roudier, and J. Gonzalez. Cluster: "finding groups in data": Cluster analysis extended rousseeuw et al. 03 2017. [p1]

G. Nithya and K. A. Prabha. A lion optimization based k-prototype clustering algorithm for mixed data. 2019. [p2]

M. G. Omran, A. P. Engelbrecht, and A. Salman. An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605, 2007. [p2]

J.-O. Palacio-Niño and F. Berzal. Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv:1905.05667*, 2019. [p7, 8]

S. Pandit, S. Gupta, et al. A comparative study on distance measuring approaches for clustering. *International Journal of Research in Computer Science*, 2(1):29–31, 2011. [p4]

A. R. Ponperiasamy and E. Thenmozhi. A Brief survey of Data Mining Techniques Applied to Agricultural Data. 2017. ISSN 2347-2693. URL www.ijcseonline.org. [p1]

S. K. Popat and M. Emmanuel. Review and comparative study of clustering techniques. *International journal of computer science and information technologies*, 5(1):805–812, 2014. [p1]

H. Ramprasanth and A. Devi. Outlier analysis of medical dataset using clustering algorithms. *Journal of Analysis and Computation ISSN:(0973-2861)*, pages 1–9, 2019. [p2]

E. Rendón, I. Abundez, A. Arizmendi, and E. Quiroz. Internal versus external cluster validation indexes. *International Journal of Computers and Communications*, 5:27–34, 01 2011. [p6]

S. Romano, N. X. Vinh, J. Bailey, and K. Verspoor. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17:1–32, 2016. ISSN 15337928. [p6]

S. Saini and P. Rani. A survey on sting and clique grid based clustering methods. *International Journal of Advanced Research in Computer Science*, 8(5), 2017. [p3]

A. Saxena, M. Prasad, A. Gupta, N. Bharill, o. Patel, A. Tiwari, M. Er, W. Ding, and C.-T. Lin. A review of clustering techniques and developments. *Neurocomputing*, 267, 07 2017. [p2]

D. Sculley. Web-scale k-means clustering. page 1177–1178, 2010. doi: 10.1145/1772690.1772862. URL https://doi.org/10.1145/1772690.1772862. [p1]

A. S. Shirkhorshidi, S. Aghabozorgi, and T. Ying Wah. A Comparison study on similarity and dissimilarity measures in clustering continuous data. *PLoS ONE*, 10(12):1–20, 2015. ISSN 19326203. doi: 10.1371/journal.pone.0144059. [p4]

K. Singh, H. K. Shakya, and B. Biswas. Clustering of people in social network based on textual similarity. *Perspectives in Science*, 8:570–573, 2016. ISSN 22130209. doi: 10.1016/j.pisc.2016.06.023. URL http://linkinghub.elsevier.com/retrieve/pii/S2213020916301628. [p1]

A. Starczewski and A. Krzyzak. Performance evaluation of the silhouette index. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, pages 49–58, Cham, 2015. Springer International Publishing. [p7]

A. Thalamuthu, I. Mukhopadhyay, X. Zheng, and G. C. Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412, 07 2006. [p3]

J. Vlegels and J. Lievens. Music classification, genres, and taste patterns: A ground-up network analysis on the clustering of artist preferences. *Poetics*, 60:76–89, 2017. ISSN 0304422X. [p1]

Y. Wang and H. Y. Youn. Feature weighting based on inter-category and intra-category strength for twitter sentiment analysis. *Applied Sciences*, 9(1):92, 2018. ISSN 2076-3417. [p1]

R. Xu and D. Wunsch. Survey of Clustering Algorithms. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 16(3), 2005. doi: 10.1109/TNN.2005.845141. URL http://axon.cs.byu.edu/Dan/678/papers/Cluster/Xu.pdf. [p4]

L. Y. L. Xuecheng. Applying wave cluster algorithm in intrusion detection [j]. *Computer Applications and Software*, 6, 2010. [p3]

T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. page 103–114, 1996. doi: 10.1145/233269.233324. URL https://doi.org/10.1145/233269.233324. [p2]

Luis Alfonso Pérez Martos
Computer Department
University of Jaén
Spain
(ORCiD if desired)
lapm0001@gmail.com

Ángel Miguel García Vico
Computer Department
University of Jaén
Spain
(ORCiD if desired)
agvico@ujaen.es

Pedro González
Computer Department
University of Jaén
Spain
(ORCiD if desired)
pglez@ujaen.es

Cristóbal J. Carmona
Computer Department
University of Jaén
Spain
(ORCiD if desired)
ccarmona@ujaen.es