Techniques for Evaluating Clustering Data in R. The **Clustering** Package

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

10/11/2020

Abstract

Clustering is a class of frequently studied and applied unsupervised learning methods whose purpose is a division of data into groups of similar objects. This technique is quite common among researchers as it allows to obtain knowledge quickly and easily. The use of this technique is suitable for automatic data classification in order to reveal concentrations of data. This paper presents the **Clustering** package which contains a set of clustering algorithms with two objectives: first, grouping data in a homogeneous way by establishing differences between clusters; and second, generating a ranking between algorithms and the attributes analyzed 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.

1 Introduction

Exploring the properties of information in order to generate groups is an unsupervised learning technique known as clustering [1] [2]. This technique is a concise data model where a set of data must be partitioned and introduced into groups or clusters. These clusters must meet two conditions: clusters must be as disparate as possible; and the elements that contain them as similar as possible. Throughout the literature related to clustering we can see that there are multiple fields where they can be applied, among which we highlight the following: Identifying tourists and analyzing their destination patterns from location-based social media data [3]; developing clustering algorithms that maximize performance on 5G heterogeneous networks [4]; the application of data mining techniques to agriculture data [5]; weighting characteristics based on the strength between categories and within categories for the analysis of feelings [6]; music classification, genres and taste patterns [7]; predicting the direction of fluctuation, maximum, minimum and closing prices of the daily exchange rate of bitcoins [8]; and `Clustering` people in a given social network based on textual similarity [9].

As a rule 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 consist of the definition of functions. In the literature we can group the data in different ways among which we highlight [10]: partitional, hierarchical or based on density. One of the best known algorithms that solves the clustering problem is the k-means [11].

A wide variety of frameworks have been presented in the literature based on clustering algorithms such as: Weka [12], ClustVis [13] and Keel [14], among others. Also within R there is a specific Cluster task view. Inside the Cluster task view we can see two well differentiated parts: on the one hand we have the most outstanding packages by functionality and in the other hand we observe the ordered set of packages that work with clusters. Among the set of packages we highlight the following: **ClusterR** [15] [16], **apcluster** [17] [18], **cluster** [19], **advclust** [20] as well as alternatives to the traditional implementation of k-means and agglomerative hierarchical clustering. Usually the task of comparing clustering algorithms is tedious, as it must be performed manually. This is quite time-consuming and in some cases there can be problems in transmitting the results. Similarly, when evaluating the distribution of data in clusters, it is necessary to indicate a categorical variable, so the selection of one variable or another from a data set can influence the results.

This paper presents the **Clustering** package. It is a package that allows us to compare multiple clustering algorithms simultaneously and assess the accuracy of the results. The purpose of this package is to allow the evaluation of a set of datasets in order to determine which attributes are most suitable for clustering. So, we can perform evaluations of the clusters created, how they have been distributed, whether the distributions are uniform and how they have been categorized from the data.

The structure of this contribution is as follows: Firstly, in section Clustering we present the concepts 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. Finally, Section The **Clustering** package describes the structure of the package and presents a complete example about the use of the package.

2 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 of the patterns is available to use in categorizing others and inferring the cluster structure of the whole data set [21]. 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, etc.) [22].”

This is a base definition of 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 have a significant impact on the problem definition.

2.1 Clustering types

There are a variety of clustering algorithms that can be classified into: hierarchical, partitioning, density-based, grid-based and probability distribution [23].

- Hierarchical clustering: Creates a hierarchical breakdown of data into a dendogram that recursively divides the data set into smaller and smaller groups. It can be created in two ways: bottom-up or top-down [24]. With the bottom-up method trees are 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 Figure ~1. Some hierarchical grouping algorithms that belong to this sorting mode are: CURE [25], CHAMELEON [26], and BIRCH [27].

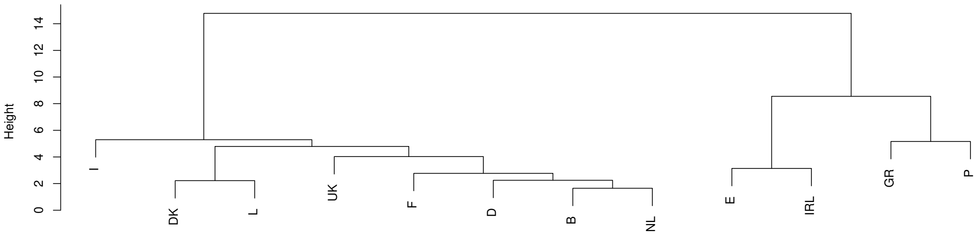


Figure 1: Hierarchical Clustering.

- Partitional clustering: Is considered to be the most popular of the clustering algorithms. Such an algorithm is 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 given 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 Figure ~2. Among the partitioning clustering algorithms we can find CLARANS, CLARA [28], K-prototype [29], K-mode [30] and K-means [31].

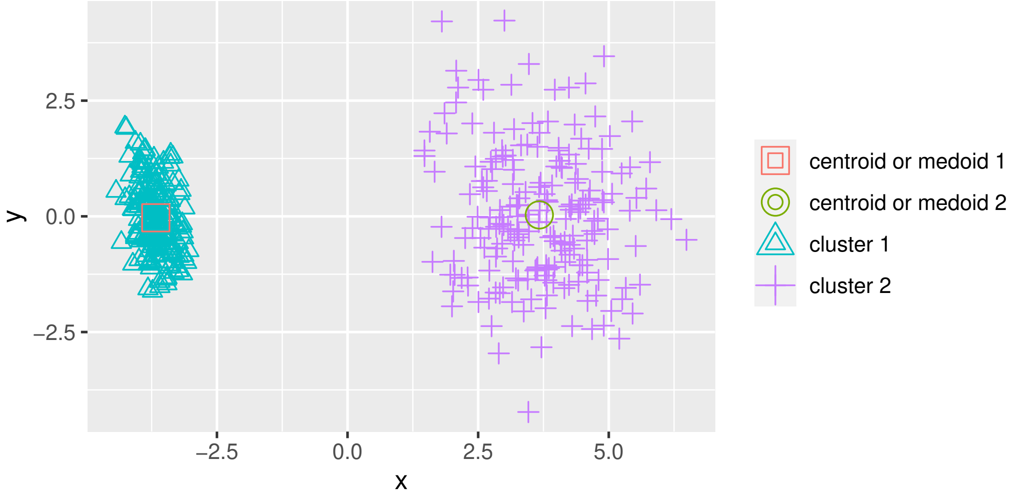


Figure 2: Partitional Clustering.

- 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 Figure ~3. Among the density-based algorithms, we highlight the following: Dbscan [32], and Denclue [33].

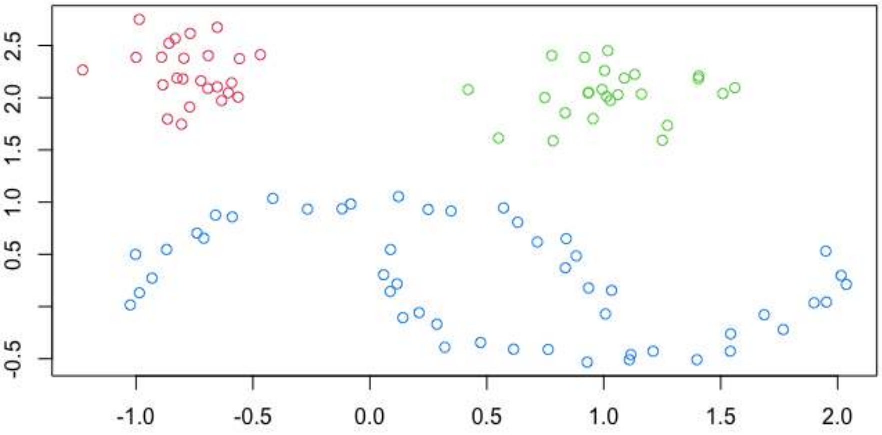


Figure 3: Density Clustering.

- Grid-based clustering: First quantizes the clustering space into a finite number of cells and then performs the required operations on the quantized space [34]. Cells that contain more than a certain number of points are treated as dense, and the dense cells are connected to form the clusters. Some of the best known grid-based clustering algorithms include: STING [35], Wave Cluster [36] and CLIQUE [37].
- Model-based methods: Are primarily based on a probability distribution. To be able to measure similarity it is based on the mean values that the algorithm tries to minimize with the square error function. The 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 [38]. Model-based clustering is shown in Figure ~4.

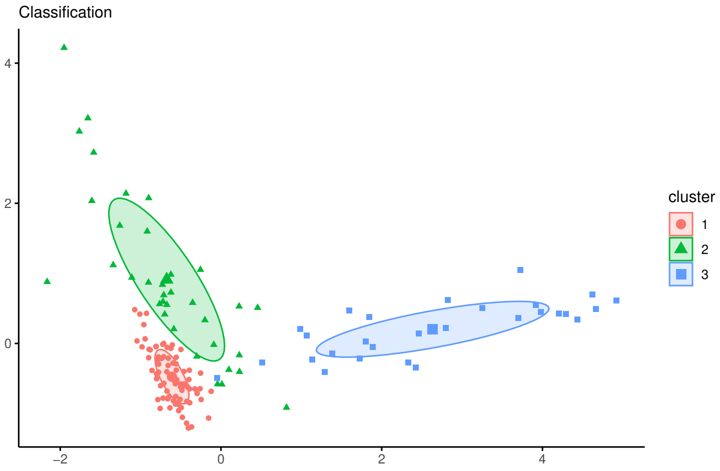


Figure 4: Model-Based Clustering.

2.2 Dissimilarity measures

Dissimilarity measurements are important in the creation of clusters with the closest neighbors and the detection of anomalies, and they are also used in a large number of data mining techniques. It is 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 a range other than [0,1], therefore it is recommended to normalize the values in the range [0,1].

Many distance measures have been proposed in the literature for data clustering. Choosing an appropriate similarity measure is crucial for cluster analysis, especially for a particular type of algorithm. For example, the density-based clustering algorithms, such as DBScan [32], 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 [39] [40].
As measures of dissimilarity in clustering we highlight the following:

- Minkowski: Is a metric in a normalized vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance [41].

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

  where m is a positive real number and x_i and y_i are two vectors in n-dimensional space.
- Euclidan distance: When in Minkowski metric the value of m is equal to 2 is calculated Euclidean distance. It is a measure of the true straight line distance between two points in a Euclidean space [24].

$$d_{euc} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

- Manhattan distance: The m parameter of the Minkowski Distance when is 1, represent Manhattan Distance. 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 [42].

$$d_{man} = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

- Mahalanobis distance: Is a data-driven measure in contrast to the Euclidean and Manhattan distances which are independent. It is tasked with measuring the distance in a multivariate space [42].

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

  where S is the covariance matrix of the dataset.
- Pearson correlation: Pearson correlation: A statistically based metric that measures the linear correlation between two variables, x and y [43].

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

  where μ_x and μ_y are the means for x and y respectively.
- Jaccard Index: Is a classical similarity measure performed on sets with several practical applications in information retrieval, data mining, machine learning, and many more [44] [45]. t 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 we can determine the degree of similarity between individuals who have had qualitative, quantitative characteristics (continuous and discrete) and binary characteristics measured.

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

  where S_{ij} the Gower similarity coefficient is [46].

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

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

3 Internal and External clustering validation measures

Clustering validation is a technique for finding a set of clusters that best fits natural partitions 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 the results of a clustering algorithm is evaluated in terms of the quantity involved in the vectors of the dataset itself (e.g. the proximity matrix). And there is a third criterion, known as the 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 compactness and separation [47] [48] as can be seen in Figure 7:

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

These criteria are graphically represented in the Figure 5.

Within external measures there are some measures to evaluate clustering results. Among these we highlight:

- Entropy: Entropy: Evaluates the distribution of categories in a cluster [50].

  $$Entropy(j) = \sum_{j=1}^{m} \frac{|C_j|}{n} E_j \tag{9}$$

  Where n_j the cluster size is 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.

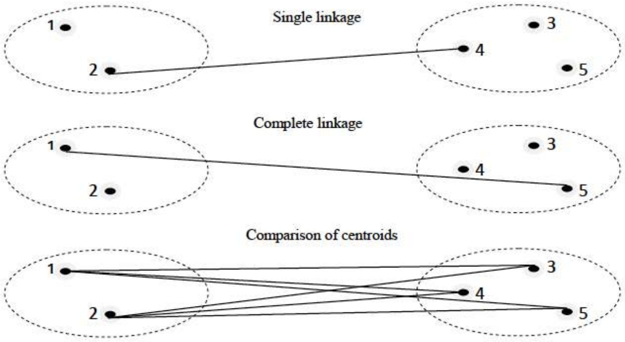


Figure 5: Intercluster distance [49].

- Recall: Indicates the proximity of the measurement results to the true value [51].

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

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

- Precision: Refers to the dispersion of the set of values obtained from repeated measurements of one magnitude [51].

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

n_j is the number of objects in cluster j.

- F-measure: Merges the concepts of accuracy and recall of the information retrieved. 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 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 [52]. The measure B_i is easily generalizable. 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 objects recovered and the total number of objects recovered).

$$Fowlkes(i,j) = \sqrt{Precision(i,j) * Recall(i,j)} \tag{14}$$

- Variation information: Variation in information or distance of shared information is a measure of the distance between two groups [52]. This measure is closely related to mutual information (mutual dependence between the two variables) [53].

$$VI(|C_i|, |C_j|) = 2H(C_i, C_j) - H(C_i) - H(C_j) \tag{15}$$

$H(C_i, C_j)$ is the joint `entropy` of two clusters, $H(C_i)$ is the `entropy` of C_i and $H(C_j)$ is the `entropy` of C_j.
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 neighbors in the data space, i.e. the degree of connection of the clusters should be minimal [54].

$$Connectivity = min_{1 \leq i \leq K} \left(min_{1 \leq j \leq K, i \neq j} \left(\frac{dist(C_i, C_j)}{max_{1 \leq k \leq K} \{diam(C_k)\}} \right) \right) \tag{16}$$

  Where $dist(C_i, C_j)$ is the distance between two clusters and $diam(C_k)$ is the diameter of a particular cluster [54].

- 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 [55].

$$Dunn = min_{1_{\leq} i \leq k} \left(min_{i+1 \leq j \leq k} \left(\frac{dist(C_i, C_j)}{max_{1_{\leq} l \leq k} diam(C_k)} \right) \right) \tag{17}$$

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

- Silhouette index: The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation) [56].

$$S = \frac{1}{N} \sum_{i=0}^{N} \frac{b_i - a_i}{max(a_i, b_i)} \tag{18}$$

  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

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

If we look at Figure ~6 we can group `Entropy, Recall, Precision, F-Measure, Fowlkes-Mallows Index and Variation information` into three families [57]:

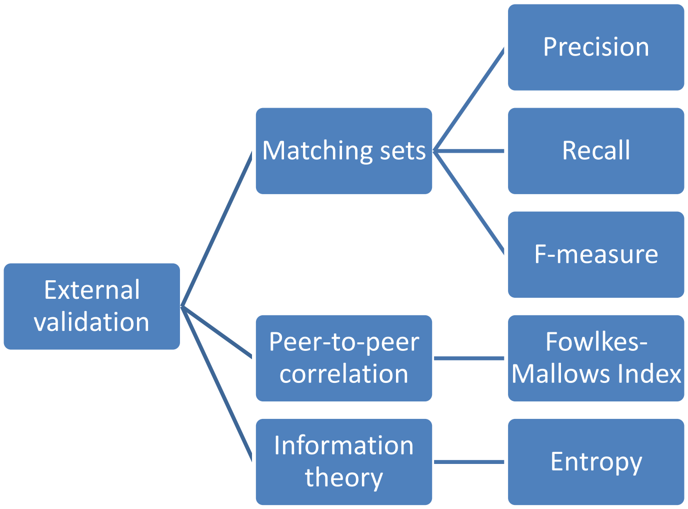


Figure 6: External validation methods.

- Matching Sets: Used to compare two partitions of data. Consists of those methods 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).
  Several measures can be defined for measuring 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: 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_{i} $ and $ C_{j} $. It is assumed that the examples that are in the same cluster in $ C_{i} $ should be in the same class in $ C_{j} $, and vice-versa. In our package we use the metric: `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 partition C. 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).

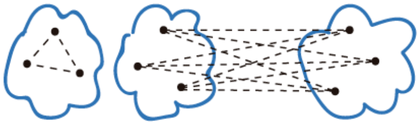


Figure 7: Representation of cohesion and separation in clustering [57].

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.

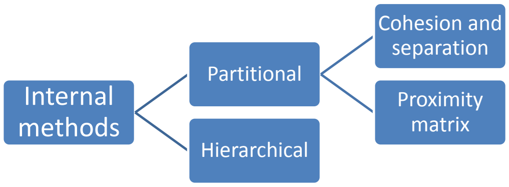


Figure 8: Internal validation methods [57].

4 The Clustering package

The **Clustering** package has been written entirely in R language. The package include other packages with hierarchical, partitional and agglomerative algorithms. The package has been provided with the ability to read data in different formats such as CSV, KEEL, ARFF (Weka) and `data.frame` objects. This package implements functionality not developed until now. In current implementations we cannot run several algorithms simultaneously. This option is very useful to compare the results of several algorithms. Another very useful option, not found until now is the ability to run all the dissimilarity measures implemented for an algorithm in the same run, e.g., running the K-means algorithm for the Euclidean and Manhattan dissimilarity measures simultaneously. This perspective is quite interesting when it comes to know what measure to use in the execution of the algorithm. In order to evaluate the quality of the clusters we find packages that perform external or internal evaluations, but there was no implementation that give us a joint evaluation of both. Finishing the review we find that when evaluating the quality of clusters it is necessary to indicate an attribute of the dataset used in the execution. Depending on the chosen attribute, results may vary, so it would be important to have methods that give us the result of running each of the quality measures by attribute, the measure of dissimilarity in the case of it existing, the number of clusters and algorithm. With this **Clustering** package we will be able to execute simultaneously several algorithms for each of the implemented dissimilarity 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. In addition, when running the algorithm all the attributes of the dataset will be executed for each of the quality metrics. The results of the executions are visually displayed so that you can draw conclusions quickly.

4.1 Package algorithms

These are the algorithms [1] 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`.

These are the algorithms included in the packages that have been mentioned most often in the literature.

4.2 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 see the execution time of these metrics, datasets, the correspondence between the evaluation metric calculation and the dataset attribute, the measures of similarity and the algorithms.
  - `has_internal_metrics.` It is a boolean operator that indicates whether internal evaluation measures have been used or not.
  - `has_external_metrics.` It is a boolean operator that indicates whether external evaluation measures have been used or not.
  - `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 algorithms indicated.

This class 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 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 ('[') in order to perform such operations in an easier way.

- External metrics: For external metrics we have a set of methods by which we can determine the behavior of the algorithms based on the best attribute of the data set, measures of dissimilarity and the number of clusters. The methods are as follows:
  - `best_ranked_external_metrics()`: With the execution of this method we obtain which attribute of the data set has better behavior by algorithm, measure of dissimilarity and number of clusters.
  - `evaluate_best_validation_external_by_metrics()`: This method should be used to group the data by algorithm and dissimilarity measure, instead of obtaining the best attribute from the data set.

[1] For more information about the advclust, amap, apcluster, cluster, Cluster, gmp, pvclust packages you can visit the following link https://cran.r-project.org/web/packages/available_packages_by_name.html

    - `evaluate_validation_external_by_metrics()`: Method for grouping the results of the execution by algorithms.
    - `result_external_algorithm_by_metric()`: It is used for obtaining the results of an algorithm indicated as a parameter grouped by number of clusters.
- Internal metrics: For internal metrics we have the same set of methods as mentioned above for external metrics.
    - `best_ranked_internal_metrics()`: With the execution of this method we obtain which attribute of the data set has better behavior by algorithm, measure of dissimilarity and number of clusters.
    - `evaluate_best_validation_internal_by_metrics()`: This method should be used to group the data by algorithm and dissimilarity measure, instead of obtaining the best attribute from the data set.
    - `evaluate_validation_internal_by_metrics()`: Method for grouping the results of the execution by algorithms.
    - `result_internal_algorithm_by_metric()`: It is used for obtaining the results of an algorithm indicated as a parameter grouped by number of clusters.
- `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 use this method to export the results of the internal metrics.

4.3 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 available on the github repository 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 used for implementing the clustering algorithms. We can 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.

4.4 Load and use of datasets

For the execution of the main method of the package we must provide data which 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. The code to read the files in ARFF format it has been extracted from the **mldr** package [58].

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.

4.5 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 aforementioned packages. When comparing we can do it by packages or simply by indicating the algorithms contained in them. To evaluate how the data have been distributed in the clusters, a set of evaluation measures are performed that return numeric values. One improvement built into the package is that in addition to returning the numerical value of the metrics, it can return the dataset attribute 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 file path. 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 the system 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 in order 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`.
- `metricsAttr`: Character vector with the metrics implemented in order to evaluate the distribution of the data in clusters. The nine metrics implemented are: `entropy, variation_information, precision,recall,f_measure,fowlkes_mallows_index,connectivity,dunn,silhouette`. This column is the same as the previous one, with the difference that it shows the dataset attribute.
- `attributes`: A boolean datum which indicates whether we want to show as a result the attributes of the datasets. The default value is F.

Once the definition of the input attributes of the `clustering` function has been completed, we will carry out a test. To do this we will use the `data.frame` Basketball (this dataset contains five attributes representing the statistics of a set of 96 basketball players). The attributes are: `assists per minute, height, time played, age and points per minute`. In the output data of the `clustering` function, these attributes are numbered from left to right in ascending order starting with one. The idea of doing it this way is to give it a homogeneity instead of using the names of the attributes from the dataset. As algorithms used in the execution we will use `gmm` and `fanny` (included in the **ClusterR** and **cluster** packages). We will indicate a range of partitions between 3 and 5 and evaluate `entropy` as an external evaluation measure and `dunn` as an internal one.

```r
result <- Clustering::clustering(df = basketball, min = 3, max = 5, algorithm = c('gmm',
                                 'fanny'), metrics = c('entropy','dunn'), attributes = T)
```

The attributes resulting from the execution of the method are:

- `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` `dataframe` appears by default, 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 taken 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 metrics implemented are: `entropy`, `recall`, `precision`, `f_measure`, `fowlkes_mallows_index`, `connectivity`, `dunn` and `silhouette`.
- `timeInternal:` Time taken to implement internal evaluation measures.
- `timeExternalAttr:` Time taken to implement external evaluation measures by attribute.
- `metricsAttr:` The same metrics that instead of showing numerical values show the attributes of the dataset.
- `timeInternalAttr:` Time taken to implement internal evaluation measures by attribute.

In Table ~1 we have the results of the execution of the `clustering()` method. According to the results, the algorithm that better behaves for the `entropy` metric is `gmm`, as highlighted in Table ~1.

Algorithm	Distance	Clusters	Dataset	tE	entropy	dunn	tI	tEAttr	enAttr	duAttr	tIAttr
gmm	gmm_euclidean	3	basketball	0.0042	0.2374	0.1096	0.0004	5	2	1	1
gmm	gmm_euclidean	3	basketball	0.0046	0.2120	0.1096	0.0004	1	4	2	2
gmm	gmm_euclidean	3	basketball	0.0079	0.0064	0.1096	0.0005	3	3	3	3
gmm	gmm_euclidean	3	basketball	0.0081	0.0032	0.1096	0.0005	2	5	4	4
gmm	gmm_euclidean	3	basketball	0.0089	0.0000	0.1096	0.0006	4	1	5	5
gmm	gmm_euclidean	4	basketball	0.0041	0.3734	0.1233	0.0004	5	2	1	4
gmm	gmm_euclidean	4	basketball	0.0041	0.2983	0.1233	0.0004	2	4	2	5
gmm	gmm_euclidean	4	basketball	0.0071	0.0064	0.1233	0.0004	4	3	3	1
gmm	gmm_euclidean	4	basketball	0.0071	0.0032	0.1233	0.0004	1	5	4	2
gmm	gmm_euclidean	4	basketball	0.0193	0.0000	0.1233	0.0005	3	1	5	3
gmm	gmm_euclidean	5	basketball	0.0044	0.4175	0.1619	0.0004	3	2	1	1
gmm	gmm_euclidean	5	basketball	0.0045	0.3857	0.1619	0.0004	1	4	2	4
gmm	gmm_euclidean	5	basketball	0.0079	0.0064	0.1619	0.0004	4	3	3	5
gmm	gmm_euclidean	5	basketball	0.0079	0.0032	0.1619	0.0005	2	5	4	3
gmm	gmm_euclidean	5	basketball	0.0091	0.0000	0.1619	0.0005	5	1	5	2
gmm	gmm_manhattan	3	basketball	0.0030	0.2498	0.1151	0.0004	3	2	1	4
gmm	gmm_manhattan	3	basketball	0.0036	0.2201	0.1151	0.0004	2	4	2	1
gmm	gmm_manhattan	3	basketball	0.0067	0.0064	0.1151	0.0004	5	3	3	5
gmm	gmm_manhattan	3	basketball	0.0073	0.0032	0.1151	0.0005	1	5	4	2
gmm	gmm_manhattan	3	basketball	0.0075	0.0000	0.1151	0.0005	4	1	5	3
gmm	gmm_manhattan	4	basketball	0.0034	0.3563	0.1179	0.0004	4	2	1	4
gmm	gmm_manhattan	4	basketball	0.0049	0.2919	0.1179	0.0004	2	4	2	1
gmm	gmm_manhattan	4	basketball	0.0068	0.0064	0.1179	0.0004	5	3	3	2
gmm	gmm_manhattan	4	basketball	0.0073	0.0032	0.1179	0.0005	1	5	4	5
gmm	gmm_manhattan	4	basketball	0.0076	0.0000	0.1179	0.0007	3	1	5	3
gmm	gmm_manhattan	5	basketball	0.0032	0.4290	0.1141	0.0004	4	2	1	1
gmm	gmm_manhattan	5	basketball	0.0036	0.3887	0.1141	0.0004	1	4	2	5
gmm	gmm_manhattan	5	basketball	0.0072	0.0064	0.1141	0.0005	3	3	3	4
gmm	gmm_manhattan	5	basketball	0.0079	0.0032	0.1141	0.0007	2	5	4	2
gmm	gmm_manhattan	5	basketball	0.0084	0.0000	0.1141	0.0011	5	1	5	3
fanny	fanny_euclidean	3	basketball	0.0107	0.2069	0.0000	0.0000	5	4	1	1
fanny	fanny_euclidean	3	basketball	0.0114	0.1675	0.0000	0.0000	2	2	2	2
fanny	fanny_euclidean	3	basketball	0.0140	0.0032	0.0000	0.0000	4	3	3	3
fanny	fanny_euclidean	3	basketball	0.0159	0.0032	0.0000	0.0000	1	5	4	4
fanny	fanny_euclidean	3	basketball	0.0161	0.0000	0.0000	0.0000	3	1	5	5
fanny	fanny_euclidean	4	basketball	0.0123	0.2069	0.0000	0.0000	3	4	1	1
fanny	fanny_euclidean	4	basketball	0.0128	0.1675	0.0000	0.0000	1	2	2	2
fanny	fanny_euclidean	4	basketball	0.0157	0.0032	0.0000	0.0000	4	3	3	3
fanny	fanny_euclidean	4	basketball	0.0157	0.0032	0.0000	0.0000	2	5	4	4
fanny	fanny_euclidean	4	basketball	0.0178	0.0000	0.0000	0.0000	5	1	5	5
fanny	fanny_euclidean	5	basketball	0.0167	0.2069	0.0000	0.0000	3	4	1	1
fanny	fanny_euclidean	5	basketball	0.0189	0.1675	0.0000	0.0000	2	2	2	2
fanny	fanny_euclidean	5	basketball	0.0219	0.0032	0.0000	0.0000	4	3	3	3
fanny	fanny_euclidean	5	basketball	0.0272	0.0032	0.0000	0.0000	1	5	4	4
fanny	fanny_euclidean	5	basketball	0.0286	0.0000	0.0000	0.0000	5	1	5	5
fanny	fanny_manhattan	3	basketball	0.0165	0.2143	0.0000	0.0000	3	4	1	1
fanny	fanny_manhattan	3	basketball	0.0184	0.1658	0.0000	0.0000	1	2	2	2
fanny	fanny_manhattan	3	basketball	0.0196	0.0032	0.0000	0.0000	4	3	3	3
fanny	fanny_manhattan	3	basketball	0.0247	0.0032	0.0000	0.0000	5	5	4	4
fanny	fanny_manhattan	3	basketball	0.0719	0.0000	0.0000	0.0000	2	1	5	5
fanny	fanny_manhattan	4	basketball	0.0152	0.2143	0.0000	0.0000	4	4	1	1
fanny	fanny_manhattan	4	basketball	0.0159	0.1658	0.0000	0.0000	1	2	2	2

fanny	fanny_manhattan	4	basketball	0.0183	0.0032	0.0000	0.0000	5	3	3	3
fanny	fanny_manhattan	4	basketball	0.0189	0.0032	0.0000	0.0000	2	5	4	4
fanny	fanny_manhattan	4	basketball	0.0193	0.0000	0.0000	0.0000	3	1	5	5
fanny	fanny_manhattan	5	basketball	0.0178	0.2143	0.0000	0.0000	4	4	1	1
fanny	fanny_manhattan	5	basketball	0.0183	0.1658	0.0000	0.0000	1	2	2	2
fanny	fanny_manhattan	5	basketball	0.0207	0.0032	0.0000	0.0000	3	3	3	3
fanny	fanny_manhattan	5	basketball	0.0211	0.0032	0.0000	0.0000	2	5	4	4
fanny	fanny_manhattan	5	basketball	0.0211	0.0000	0.0000	0.0000	5	1	5	5

Table 1: Results of running the main method of the package.

To make compression easier we have extracted the values of the `gmm` algorithm with five clusters the and `manhattan` measurement. The values are arranged in descending order. The values in the columns ending in Attr refer to the attributes of the dataset. As already mentioned, the attributes of the dataset are numbered from left to right in ascending order. Thus `assists per minute` which is the first attribute of the dataset found on the left is numbered with 1, `height` with 2, `time played` with 3, `age` with 4 and `points per minute` with 5.

Algorithm	Distance	Clusters	Dataset	tE	entropy	dunn	tI	tEAttr	enAttr	duAttr	tIAttr
gmm	gmm_manhattan	5	basketball	0.0032	0.4290	0.1141	0.0004	4	2	1	1
gmm	gmm_manhattan	5	basketball	0.0036	0.3887	0.1141	0.0004	1	4	2	5
gmm	gmm_manhattan	5	basketball	0.0072	0.0064	0.1141	0.0005	3	3	3	4
gmm	gmm_manhattan	5	basketball	0.0079	0.0032	0.1141	0.0007	2	5	4	2
gmm	gmm_manhattan	5	basketball	0.0084	0.0000	0.1141	0.0011	5	1	5	3

Table 2: Example of calculation of the attributes of a dataset by algorithm, means and cluster number.

The name of the column `tE` corresponds to `timeExternal`, `tI` to `timeInternal`, `tEAttr` to `timeExternalAttr`, `enAttr` to `entropyAttr`, `duAttr` to `dunnAttr` and finally `tIAttr` to `timeInternalAttr`. From the data in Table ~2 in row one we have the highest value of entropy for the basketball dataset. If we want to know the dataset attribute for the maximum value we look in the equivalent column of the metric, in our case as we are centered on `entropy` we must look in the column `enAttr`, which indicates that it is attribute two that corresponds to height.

In order to summarize the information this package provides a series of mechanisms that allow us to summarize the information from the result obtained by the clustering method in such a way that we can obtain the best results and attributes for one or several data sets. The mechanisms used are the following:

- `best_ranked_external_metrics()`: Method used for selecting from the data set attributes those that obtain the best result in the evaluation of the measure. We will now use this mechanism on the previous results.

```
Clustering::best_ranked_external_metrics(result)
```

Algorithm	Distance	Clusters	Dataset	timeExternal	entropy	timeExternalAttr	entropyAttr
gmm	gmm_euclidean	3	basketball	0.0104	0.2374	5	2
gmm	gmm_euclidean	4	basketball	0.0147	0.3734	2	2
gmm	gmm_euclidean	5	basketball	0.0058	0.4175	5	2
gmm	gmm_manhattan	3	basketball	0.0066	0.2498	5	2
gmm	gmm_manhattan	4	basketball	0.0083	0.3563	4	2
gmm	gmm_manhattan	5	basketball	0.0045	0.4290	5	2
fanny	fanny_euclidean	3	basketball	0.0177	0.2069	5	4
fanny	fanny_euclidean	4	basketball	0.0148	0.2069	5	4
fanny	fanny_euclidean	5	basketball	0.0199	0.2069	1	4
fanny	fanny_manhattan	3	basketball	0.0311	0.2143	3	4
fanny	fanny_manhattan	4	basketball	0.0210	0.2143	4	4
fanny	fanny_manhattan	5	basketball	0.0222	0.2143	2	4

Table 3: Results of the execution of the `best_ranked_external_metrics()`.

In the calculation of the `entropy` the results are in the interval [0,1]. For this example we have the best attributes in `entropy` grouped by algorithm, measure of dissimilarity and cluster number. When grouping the data, we choose the one with the highest value. We perform the same calculation for internal measurements.

```
Clustering::best_ranked_internal_metrics(result)
```

Algorithm	Distance	Clusters	Dataset	timeInternal	dunn	dunnAttr	timeInternalAttr
gmm	gmm_euclidean	3	basketball	0.0009	0.1096	1	5
gmm	gmm_euclidean	4	basketball	0.0008	0.1233	1	4
gmm	gmm_euclidean	5	basketball	0.0005	0.1619	1	2
gmm	gmm_manhattan	3	basketball	0.0006	0.1151	1	1
gmm	gmm_manhattan	4	basketball	0.0005	0.1179	1	2
gmm	gmm_manhattan	5	basketball	0.0006	0.1141	1	5
fanny	fanny_euclidean	3	basketball	0.0000	0.0000	1	1
fanny	fanny_euclidean	4	basketball	0.0000	0.0000	1	1
fanny	fanny_euclidean	5	basketball	0.0000	0.0000	1	1
fanny	fanny_manhattan	3	basketball	0.0000	0.0000	1	1
fanny	fanny_manhattan	4	basketball	0.0000	0.0000	1	1
fanny	fanny_manhattan	5	basketball	0.0000	0.0000	1	1

Table 4: Results of the execution of the `best_ranked_internal_metrics()`.

We already have the best attributes for each execution, and have available methods for grouping the measures of dissimilarity from the algorithms. When grouping the results by measures of dissimilarity and algorithm we do not use a specific grouping algorithm, but rather 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 dissimilarity 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.

- `evaluate_best_validation_external_by_metrics()`: With this method we intend to demonstrate whether the choice of measurement of similarity has an influence. The results obtained in Table ~5 are obtained by filtering the values of `best_ranked_external_metrics()` and aggregating them by algorithm and measure of dissimilarity.

```
Clustering::evaluate_best_validation_external_by_metrics(result)
```

Algorithm	Distance	timeExternal	entropy	timeExternalAttr	entropyAttr
fanny	fanny__euclidean	0.0199	0.2069	5	4
fanny	fanny__manhattan	0.0311	0.2143	4	4
gmm	gmm__euclidean	0.0147	0.4175	5	2
gmm	gmm__manhattan	0.0083	0.4290	5	2

Table 5: Results of execute `evaluate_best_validation_external_by_metrics()`.

If we observe Table ~3 we can see that rows three and six give us the highest value for the `gmm` algorithm and the measures `Euclidean` and `Manhattan`. In the case of `fanny` all the attributes return the same value, so we choose those whose execution time is lower.

- `evaluate_validation_external_by_metrics()`: If we want to go further and wish to determine the best algorithm from the attributes, we can do so in the following way.

```
Clustering::evaluate_validation_external_by_metrics(result)
```

Algorithm	timeExternal	entropy	timeExternalAttr	entropyAttr
fanny	0.0311	0.2143	5	4
gmm	0.0147	0.4290	5	2

Table 6: Results of `evaluate_validation_external_by_metrics()` method.

With `evaluate_validation_external_by_metrics` we can see that the best suited algorithm for the dataset is `gmm` algorithm, as the value of `entropy` is closer to 1.

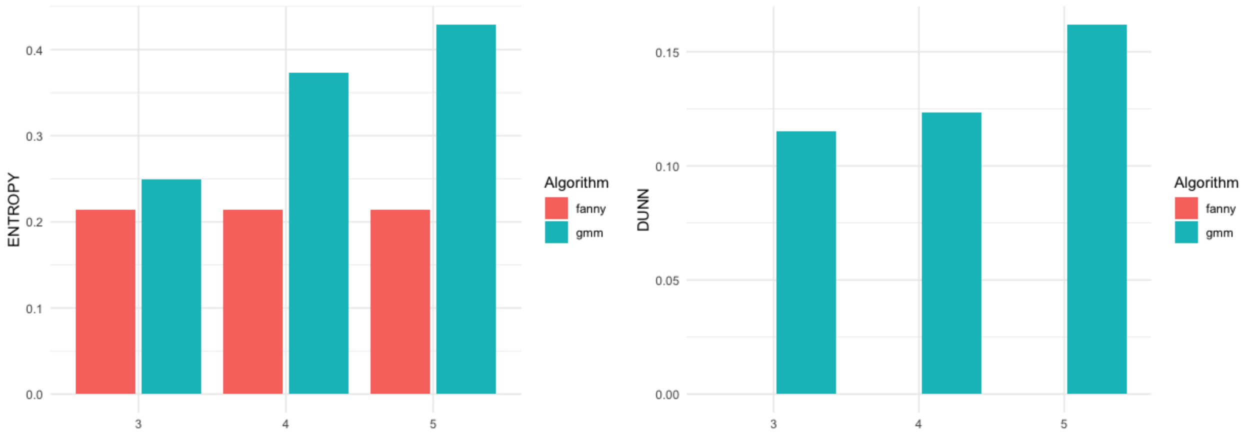
- `result_external_algorithm_by_metric()`: This method has been incorporated to filter the results of the clustering object from an algorithm in order to be able to choose a suitable cluster.

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

Algorithm	Clusters	timeExternal	entropy	timeExternalAttr	entropyAttr
gmm	3	0.0104	0.2498	5	2
gmm	4	0.0147	0.3734	4	2
gmm	5	0.0058	0.4290	5	2

Table 7: Results of the execution of the `result_external_algorithm_by_metric()`.

- `plot_clustering()`: Mechanism for representing the evaluation metrics based on the number of clusters, accelerating the process of choosing the best results, as shown below. In Figure we can see that, in the representation of dunn metric, no fanny algorithm appears. This is because the dunn values for fanny algorithm are zeros.



Graphic representation of internal and external evaluation measures grouped by algorithm and number of clusters

In conclusion, and based on the results, we can say that the `gmm` algorithm with five clusters and `Manhattan` similarity measure is the one with the best behavior for the `entropy` external metric. The attributes of the dataset that return the best value are the two that correspond to the column `heightInteger`. 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.

4.6 Data post-processing

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 results. To do this we will detail the methods used:

- `sort(clustering_object,column_name)`: Sometimes we need to sort the columns in order to find the maximum and minimum value. An example might be to sort the `entropy` column in ascending order to find the maximum and minimum value for the data set. The easiest way to perform the sorting by column is as follows:

```
result <- Clustering::clustering(df = Clustering::basketball, min = 3, max=3,
                 algorithm = c('gmm'), metrics = c('entropy','dunn'), attributes = T);

sort(result,T,'entropy')
```

Algorithm	Distance	Clusters	Dataset	tE	entropy	dunn	tI	tEAttr	enAttr	duAttr	tIAttr
gmm	gmm_manhattan	3	basketball	0.0026	0.2498	0.1151	0.0004	5	2	1	1
gmm	gmm_euclidean	3	basketball	0.0040	0.2374	0.1096	0.0004	5	2	1	1
gmm	gmm_manhattan	3	basketball	0.0033	0.2201	0.1151	0.0004	2	4	2	2
gmm	gmm_euclidean	3	basketball	0.0041	0.2120	0.1096	0.0004	1	4	2	2
gmm	gmm_euclidean	3	basketball	0.0074	0.0064	0.1096	0.0004	3	3	3	5
gmm	gmm_manhattan	3	basketball	0.0059	0.0064	0.1151	0.0004	3	3	3	3
gmm	gmm_euclidean	3	basketball	0.0074	0.0032	0.1096	0.0004	2	5	4	3
gmm	gmm_manhattan	3	basketball	0.0061	0.0032	0.1151	0.0004	1	5	4	4
gmm	gmm_euclidean	3	basketball	0.0083	0.0000	0.1096	0.0006	4	1	5	4
gmm	gmm_manhattan	3	basketball	0.0065	0.0000	0.1151	0.0004	4	1	5	5

Table 8: Sort results by `entropy` column.

- `"[.clustering"`: There are times when we need to apply filters on a series of columns for a set of values.

This process can be carried out using third party packages (`dplyr`), but due to its great usefulness we have incorporated this functionality. We must filter 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 & entropyAttr == 2]
```

Algorithm	Distance	Clusters	Dataset	tE	entropy	dunn	tI	tEAttr	enAttr	duAttr	tIAttr
gmm	gmm_manhattan	3	basketball	0.0067	0.2498	0.1151	0.0008	5	2	1	3

Table 9: Filtering information by different criteria.

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

5 Graphical User Interface of the Clustering package

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

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

The execution will open our default browser with the interface. As it can be observed in Figure ~9, we have a layout with header, side menu and main menu. In the header menu we can choose to see the results numerically 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 we can see the result of running the clustering algorithm.

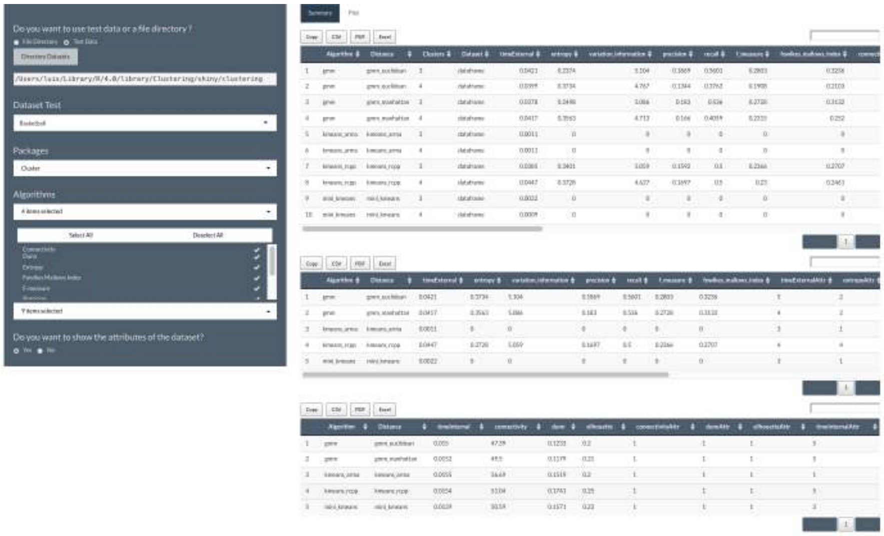
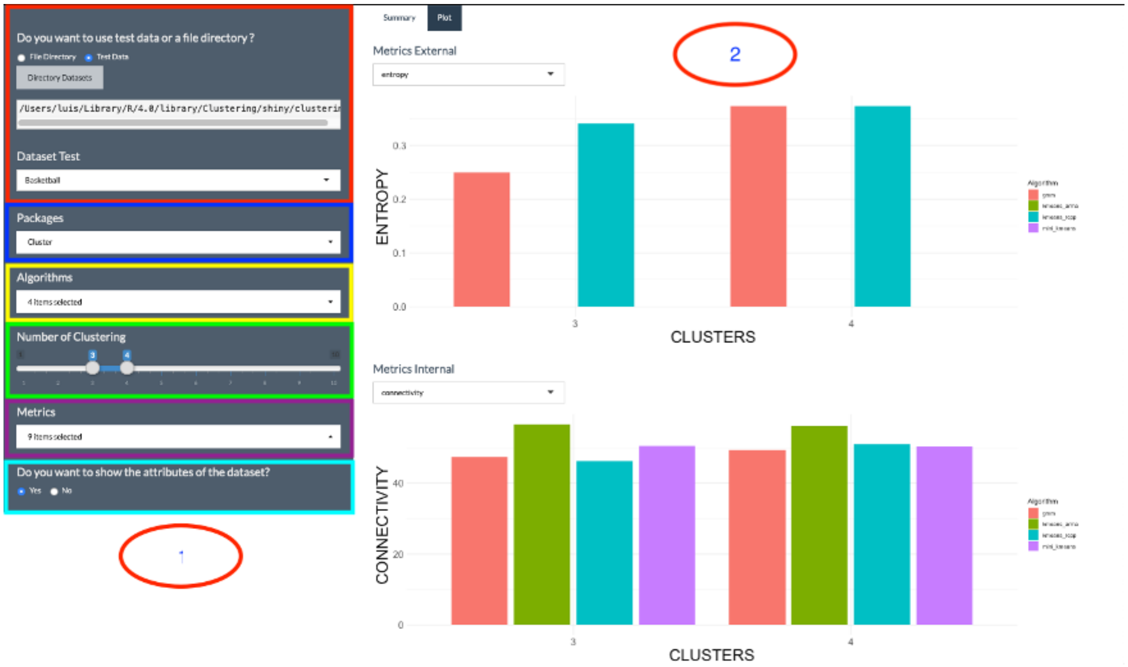


Figure 9: Clustering app user interface.

The operation of the application is very simple, as can be seen in Figure ~10 and we will proceed to explain it step by step. As can be seen in Figure ~10 we have two well differentiated parts:

- In this section we can find the different parameters used by the clustering function to filter the information.



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/clusterin

Dataset Test: Basketball

Packages: Cluster

Algorithms: 4 Items selected

Number of Clustering: 3 – 4

Metrics: 9 Items selected

Do you want to show the attributes of the dataset?

● Yes ○ No

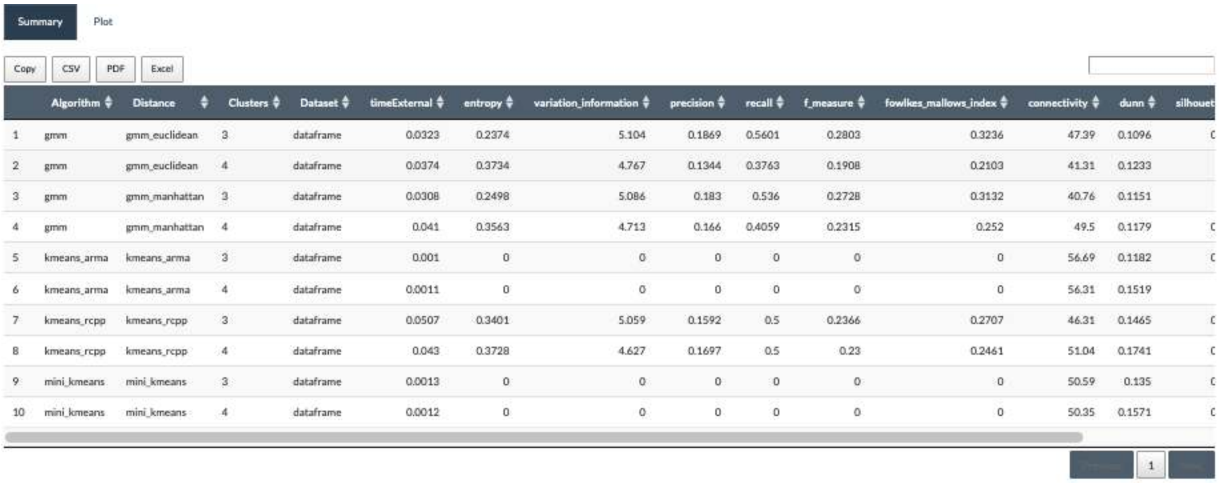
Metrics External: entropy

Metrics Internal: connectivity

Figure 10: Clustering app user interface.

- 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 packages or a subset of them. When a package is marked, all the algorithms implemented within the selected package are 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 maximum and minimum on the same value.
- In violet we indicate the evaluation metrics used when validating the clusters.
- Finally we show the attributes of the dataset that correspond to the values indicated in each of the metrics.

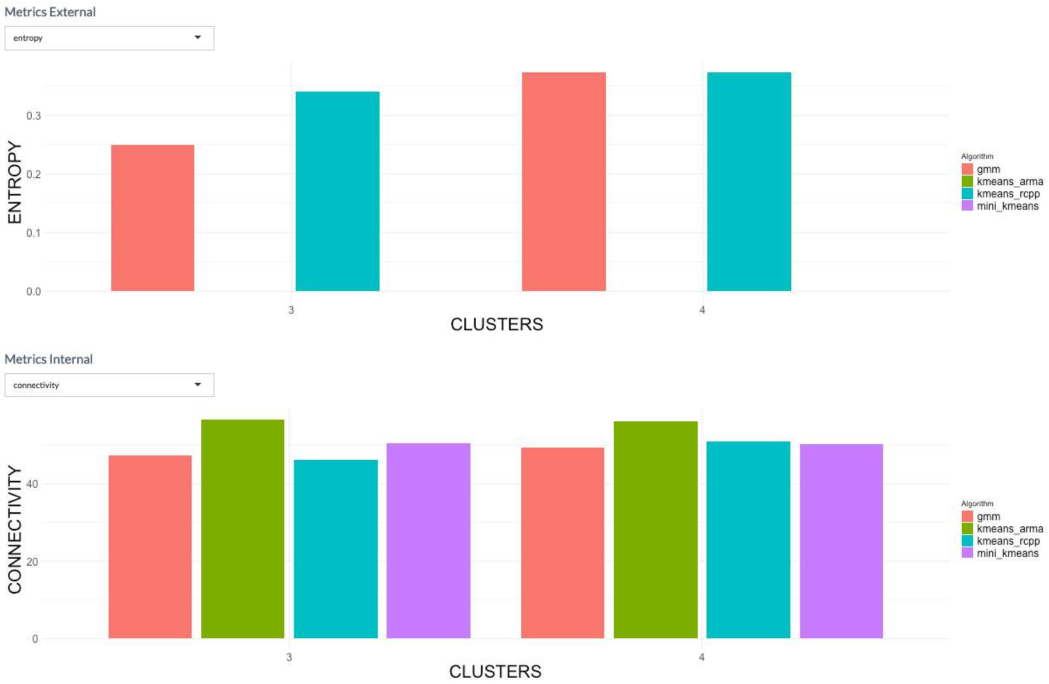
- In the main layout we have the options for representing the data.
    - If we click on the summary tab as shown in Figure ~11, we can see the data represented in tables. If we wish we can export the results in the following formats: CSV, PDF and XLS. We also have the option of copying the data.



	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 11: **Clustering** package execution summary tab..

- To view the data in graphical mode as shown in Figure ~12, 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.


Metrics External: entropy

Metrics Internal: connectivity

Figure 12: Tab with graphical representation of metrics.

6 Conclusion

In this paper we have introduced the **Clustering** package. The package has dependencies on other packages, as seen throughout the paper. It allows to read and load of datasets in KEEL, CSV or ARFF format. We 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. With the package we contribute to choosing which variable of a dataset is the one that obtains the best value when we evaluate the clusters, as well as applying relative criteria that allow us to compare the execution results of an algorithm with another one using different parameters. The development of the package will be continued with the inclusion of new algorithms, functionalities and improvement of the interface, and 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 to complement the package.

References

[1] A. Mann and N. Kaur, “Paper on clustering techniques by amandeep kaur mann & navneet,” 2013.

[2] M. Steinbach, G. Karypis, and V. Kumar, “A comparison of document clustering techniques,” *Proceedings of the International KDD Workshop on Text Mining*, 2000.

[3] 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*, vol. 96, pp. 38–54, 2018, doi: 10.1016/j.trc.2018.09.006.

[4] E. Balevi and R. Gitlin, “A clustering algorithm that maximizes throughput in 5G heterogeneous f-RAN networks,” *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2018, doi: 10.1109/ICC.2018.8422151.

[5] H. Patel and D. Patel, “A brief survey of data mining techniques applied to agricultural data,” *International Journal of Computer Applications*, vol. 95, no. 9, 2014, doi: 10.5120/16620-6472.

[6] Y. Wang and H. Y. Youn, “Feature weighting based on inter-category and intra-category strength for twitter sentiment analysis,” *Applied Sciences*, vol. 9, no. 1, p. 92, 2019, doi: 10.3390/app9010092.

[7] J. Vlegels and J. Lievens, “Music classification, genres, and taste patterns: A ground-up network analysis on the clustering of artist preferences,” *Poetics*, vol. 60, no. 1, pp. 76–89, 2017, doi: 10.1016/j.poetic.2016.08.004.

[8] D. C. A. Mallqui and R. Fernandes, “Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques,” *Appl. Soft Comput.*, vol. 75, pp. 596–606, 2019, doi: 10.1016/j.asoc.2018.11.038.

[9] K. Singh, H. K. Shakya, and B. Biswas, “Clustering of people in social network based on textual similarity,” *Perspectives in Science*, vol. 8, pp. 570–573, 2016, doi: 10.1016/j.pisc.2016.06.023.

[10] S. K. Popat and M. Emmanuel, “Review and comparative study of clustering techniques,” *International journal of computer science and information technologies*, vol. 5, no. 1, pp. 805–812, 2014.

[11] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” in *In 5-th berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.

[12] N. Sharma, A. Bajpai, and R. Litoriya, “Comparison the various clustering algorithms of weka tools,” 2012.

[13] T. Metsalu and J. Vilo, “ClustVis: A web tool for visualizing clustering of multivariate data using principal component analysis and heatmap,” *Nucleic acids research*, vol. 43, 2015, doi: 10.1093/nar/gkv468.

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

[15] D. Sculley, “Web-scale k-means clustering,” 2010, pp. 1177–1178, doi: 10.1145/1772690.1772862.

[16] L. Mouselimis, *ClusterR: Gaussian mixture models, k-means, mini-batch-kmeans, k-medoids and affinity propagation clustering*. 2020.

[17] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, pp. 972–977, 2007, doi: 10.1126/science.1136800.

[18] U. Bodenhofer, A. Kothmeier, and S. Hochreiter, “APCluster: An r package for affinity propagation clustering,” *Bioinformatics*, vol. 27, pp. 2463–2464, 2011, doi: 10.1093/bioinformatics/btr406.

[19] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik, *Cluster: Cluster analysis basics and extensions*. 2019.

[20] C. Correa, C. Valero, P. Barreiro, M.-P. Diago, and J. Tardaguila, “A comparison of fuzzy clustering algorithms applied to feature extraction on vineyard,” in *Inteligencia artificial revista iberoamericana de inteligencia artificial*, 2011, vol. 1, p. 778.

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

[22] M. Omran, A. Engelbrecht, and A. Salman, “An overview of clustering methods,” *Intell. Data Anal.*, vol. 11, pp. 583–605, 2007, doi: 10.3233/IDA-2007-11602.

[23] R. S. M. L. Patibandla and N. Veeranjaneyulu, “Survey on clustering algorithms for unstructured data,” 2018, pp. 421–429, doi: 10.1007/978-981-10-7566-7_41.

[24] A. K. Jain, M. Murty, and P. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, pp. 264–323, 1999.

[25] S. Guha, R. Rastogi, and K. Shim, “Cure: An efficient clustering algorithm for large databases,” *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437901000084.

[26] D. Guo, J. Zhao, and J. Liu, “Research and application of improved CHAMELEON algorithm based on condensed hierarchical clustering method,” 2019, pp. 14–18, doi: 10.1145/3375998.3376016.

[27] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An efficient data clustering method for very large databases,” 1996, pp. 103–114, doi: 10.1145/233269.233324.

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

[29] G. Nithya and K. A. Prabha, “A LION OPTIMIZATION BASED k-PROTOTYPE CLUSTERING ALGORITHM FOR MIXED DATA,” 2019.

[30] J. Huang, “A fast clustering algorithm to cluster very large categorical data sets in data mining,” 1997.

[31] M. Kushwaha, H. Yadav, and C. Agrawal, “A review on enhancement to standard k-means clustering,” 2020, pp. 313–326.

[32] S. Hu, “Indoor location method based on data mining,” 2019, pp. 11–15, doi: 10.1145/3377458.3377465.

[33] M. Khader and G. Al-Naymat, “An overview of various enhancements of DENCLUE algorithm,” 2019, pp. 1–7.

[34] S. Dang, “A REVIEW OF CLUSTERING TECHIQUES IN VARIOUS APPLICATIONS FOR EFFECTIVE DATA MINING,” *International Journal of Research in IT & Management 2231-4434*, vol. 1, pp. 50–66, 2011.

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

[36] L. Y. L. Xuecheng, "APPLYING WAVE CLUSTER ALGORITHM IN INTRUSION DETECTION [j]," *Computer Applications and Software*, vol. 6, 2010.

[37] S. Saini and P. Rani, "A survey on STING and CLIQUE grid based clustering methods," *International Journal of Advanced Research in Computer Science*, vol. 8, pp. 1510–1512, 2017.

[38] A. Thalamuthu, I. Mukhopadhyay, X. Zheng, and G. Tseng, "Evaluation and comparison of gene clustering methods in microarray analysis," *Bioinformatics (Oxford, England)*, vol. 22, pp. 2405–12, 2006, doi: 10.1093/bioinformatics/btl406.

[39] S. Pandit, S. Gupta, and others, "A comparative study on distance measuring approaches for clustering," *International Journal of Research in Computer Science*, vol. 2, no. 1, pp. 29–31, 2011.

[40] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Ying Wah, "A Comparison study on similarity and dissimilarity measures in clustering continuous data," *PLoS ONE*, vol. 10, no. 12, pp. 1–20, 2015.

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

[42] R. Xu and D. Wunsch, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, pp. 645–678, 2005, doi: 10.1109/TNN.2005.845141.

[43] T. O. Institute, "Learn About Pearson&#8217;s Correlation Coefficient in SPSS With Data From the Global Health Observatory Data (2012)," no. 2012, 2015, doi: https://dx.doi.org/10.4135/978147394 8167.

[44] S. Kosub, "A note on the triangle inequality for the jaccard distance," *arXiv.org*, vol. 120, 2016, doi: 10.1016/j.patrec.2018.12.007.

[45] J. Irani, N. Pise, and M. Phatak, "Clustering techniques and the similarity measures used in clustering a survey," *International Journal of Computer Applications*, vol. 134, pp. 9–14, 2016, doi: 10.5120/ijca2016907841.

[46] O. Niakšu, "Calculating distance measure for clustering in multi-relational settings," 2013.

[47] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2–3, pp. 107–145, 2001.

[48] G. S. Linoff and M. J. Berry, *Data mining techniques: For marketing, sales, and customer relationship management*. John Wiley & Sons, 2011.

[49] R. A. Johnson, D. W. Wichern, and others, *Applied multivariate statistical analysis*, vol. 5. Prentice hall Upper Saddle River, NJ, 2002.

[50] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, pp. 1495–1502, 2007, doi: 10.1093/bioinformatics/btm134.

[51] J. Kacprzyk and Y. Farhaoui, *Big data and smart digital environment*. 2019.

[52] S. Romano, N. the vinh, J. Bailey, and K. Verspoor, "Adjusting for chance clustering comparison measures," *Journal of Machine Learning Research*, vol. 17, pp. 1–32, 2016.

[53] Z. He, X. Xu, and S. Deng, "K-ANMI a mutual information based clustering algorithm for categorical data," *Information Fusion*, vol. 9, pp. 223–233, 2005, doi: 10.1016/j.inffus.2006.05.006.

[54] 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*, vol. 1, pp. 85–102, 2010, doi: 10.5121/ijcses.2010.1207.

[55] Z. Ansari, M. F. 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*, vol. 1, 2015.

[56] A. Starczewski and A. Krzyżak, "Performance evaluation of the silhouette index," *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 9120, pp. 49–58, 2015, doi: 10.1007/978-3-319-19369-4-5.

[57] J.-O. Palacio-Niño and F. Galiano, “Evaluation metrics for unsupervised learning algorithms,” *ArXiv*, vol. abs/1905.05667, 2019.

[58] F. Charte and D. Charte, “Working with multilabel datasets in R: The mldr package,” *The R Journal*, vol. 7, no. 2, pp. 149–162, 2015.