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 of data. 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 have the best behavior for a series of clustering algorithms. 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 2 we have the presentation of clustering, types of clustering and similarity measures is performed. Section 2 presents the definition of the evaluation measures in order to value the distribution of the data in the clusters and finally Section 2.1 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 (Figure 1), partitioning (Figure 2), density-based (Figure 3), grid-based and probability distribution (Figure 4). The most commonly used groupings are: hierarchical, part-based and density-based.

- Hierarchical clustering algorithms Jain et al. (1999) create a hierarchical breakdown of data into a dendogram that recursively divides the data set into smaller and smaller data. The tree can be created in two ways: top-down or bottom-up. In bottom-up trees we can also call it 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. 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).

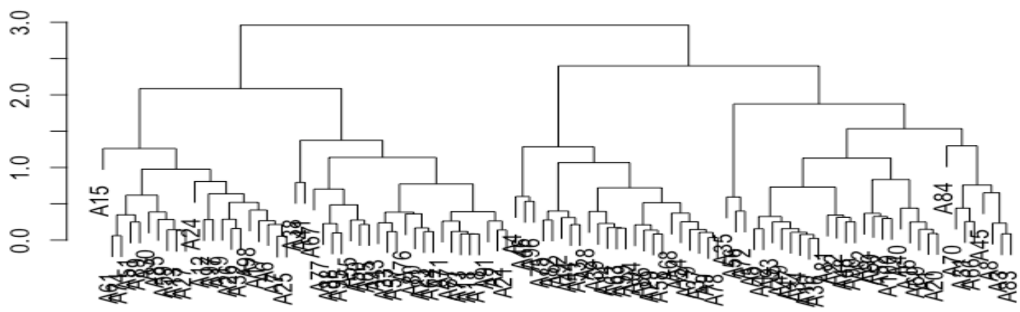

Figure 1: Hierarchical Clustering

- 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. 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).
- 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). Among the

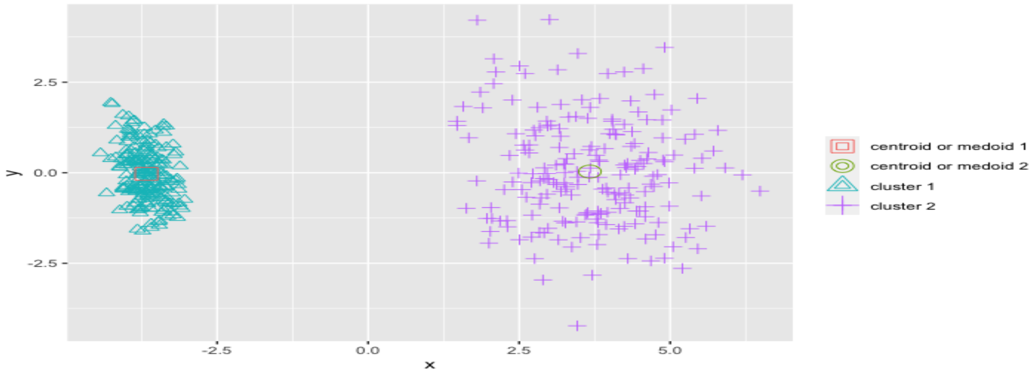

Figure 2: Partitional Clustering

density-based algorithms, we highlight the following: Dbscan Hu (2019), and Denclue Khader and Al-Naymat (2019).

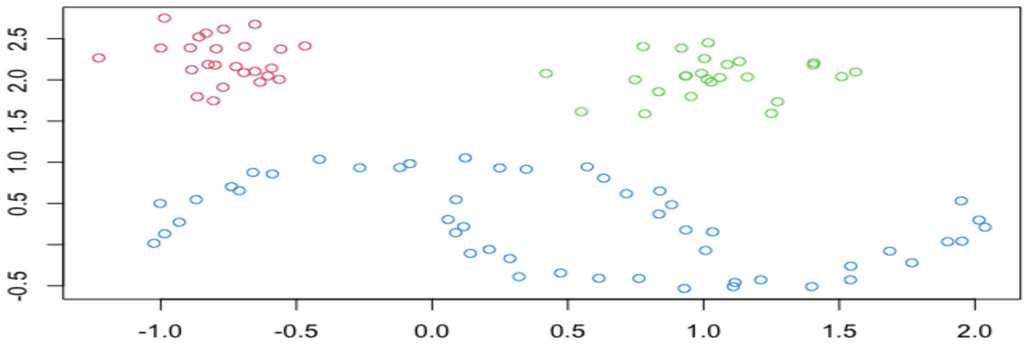

Figure 3: Density Clustering

- Grid-based clustering algorithms Dang (2011) first quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space. 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 SOM Thalamuthu et al. (2006)

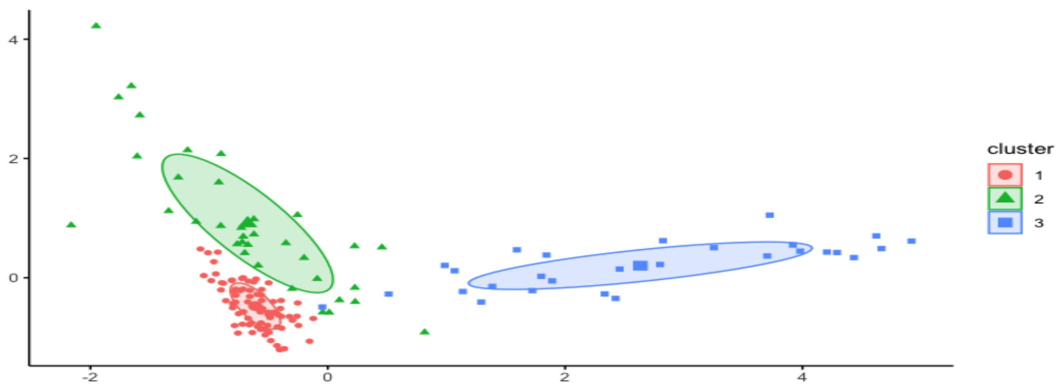

Figure 4: Model-Based Clustering

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 clustering 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: The Minkowski family includes Euclidean distance and Manhattan distance, which are particular cases of the Minkowski distance. The Minkowski distance performs well when the dataset clusters are isolated or compacted; if the dataset does not fulfil this condition, then the large-scale attributes would dominate the others. Another problem with Minkowski metrics is that the largest-scale feature dominates the rest.

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

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

- Euclidan distance: Is a special case of Minkowski distance. It works very well when deployed to datasets that include compact or isolated clusters. Although Euclidean distance is very common in clustering, it has a drawback: if two data vectors have no attribute values in common, they may have a smaller distance than the other pair of data vectors containing the same attribute values. Another problem with Euclidean distance as a family of the Minkowski metric is that the largest-scaled feature would dominate the others. Normalization is the solution to the problem.

$$d_{ij} = \sqrt{\sum_{c=1}^{p} (X_{ic} - X_{jc})^2} \tag{2}$$

- Manhattan distance: Also known as the geometry cab driver is sensitive to outliers. When this distance measure is used in clustering algorithms, the shape of clusters is hyper-rectangular. This metric was created by Hermann Minkowski in the 19th century and its name refers to the grid pattern of most of the streets on Manhattan Island.

$$d_{ij} = \sum_{c=1}^{p} |X_{ic} - X_{jc}| \tag{3}$$

- Mahalanobis distance: Is a data-driven measure in contrast to Euclidean and Manhattan distances that are independent of the related dataset to which two data points belong. Also can be used for extracting hyperellipsoidal clusters.

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

  where S is the covariance matrix of the dataset

- Pearson correlation: It's a statistically based metric, widely used in clustering gene expression data. This similarity measure calculates the similarity between the shapes of two gene expression patterns.

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

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

- Jaccard Index: Kosub (2019) Is a classical similarity measure on sets with a lot of practical applications in information retrieval, data mining, machine learning, and many more. Measuring the relative size of the overlap of two finite sets A and B, the Jaccard index J is formally defined as:

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

- 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.

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

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. There are two proposed criteria for the evaluation and selection of an optimal clustering Halkidi et al. (2001) Berry and Linoff (2004):

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.

The two first approaches are based on statistical tests and their major drawback is their high computational cost. Moreover, the indices related to these approaches aim at measuring the degree to which a data set confirms an a-priori specified scheme. On the other hand, the third approach aims at finding the best clustering scheme that a clustering algorithm can be defined under certain assumptions and parameters.

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

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

$$Entropy = \sum_{j=1}^{m} \frac{n_j}{n} E_j \tag{8}$$

  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:

$$E_j = \sum_i p_{ij} log(p_{ij}) \tag{9}$$

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

$$Recall(i,j) = \frac{n_{ij}}{n_i} \tag{10}$$

  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: Kacprzyk and Farhaoui (2019) It refers to the dispersion of the set of values obtained from repeated measurements of one magnitude. The lower the dispersion, the higher the accuracy.

$$Precision(i,j) = \frac{n_{ij}}{n_j} \tag{11}$$

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

$$F - measure(i,j) = \frac{2 * (Precision(i,j) * Recall(i,j))}{(Precision(i,j) + Recall(i,j))} \tag{12}$$

- Fowlkes-Mallows Index: Romano et al. (2016) 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. The measure B_i is easily generalizable to a measure for clustering of different clusters.

$$Fowlkes = \frac{n_{11}}{\sqrt{(n_{11} + n_{10})(n_{11} + n_{01})}} \tag{13}$$

  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).

- Variation information: Romano et al. (2016) Variation in information or distance of shared information is a measure of distance between two groups. This measure is closely related to mutual information. 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.

$$VI = -\sum_{i}^{j} r_{ij} \left[\log\left(\frac{r_{ij}}{p_i}\right) + \left(\frac{r_{ij}}{q_j}\right) \right] \tag{14}$$

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).

$$connectivity = [1 \leq i \leq K]\, min\{1 \leq j \leq K, i \neq j\} min \left\{ \frac{dist(C_i, C_j)}{max_1 \leq k \leq K\{diam(C_k)\}} \right\} \tag{15}$$

- Dunn: Ansari et al. (2015) 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.

$$dunn = min_{1 \leq i \leq K} min \left\{ \frac{d(c_i, c_j)}{max_{1 \leq k \leq c(d(X_k))}} \right\} \tag{16}$$

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

$$S = \frac{1}{N} \sum_{i=0}^{N} \frac{b_i - a_i}{max(a_i, b_i)} \tag{17}$$

  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 5 we can group Entropy, Recall, Precision, F-Measure, Fowlkes-Mallows Index and Variation information into three families Palacio-Niño and Berzal (2019):

- Matching Sets: Palacio-Niño and Berzal (2019) 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.
  Several measures can be defined to measure the similarity between the clusters in C, obtained by the clustering algorithm, and the clusters if P, corresponding to our prior (external) knowledge. The metrics included in this method are: Precision, Recall and F-measure

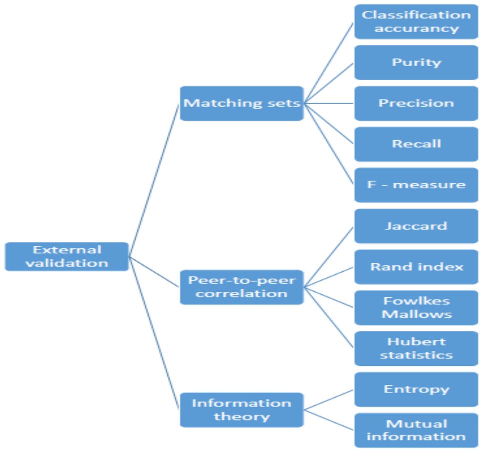

Figure 5: External validation methods Palacio-Niño and Berzal (2019)

- Peer-to-peer Correlation: Palacio-Niño and Berzal (2019) 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. 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: Palacio-Niño and Berzal (2019) 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. This family includes basic measures such as entropy and variation information.

Internal evaluation metrics (see Figure 6) 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).

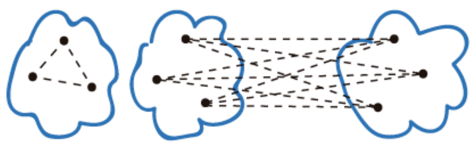

Figure 6: Representation of cohesion and separation in clustering Palacio-Niño and Berzal (2019)

According to the figure 7, the internal Dunn, Silhouette and Connectivity metrics are based on the concepts mentioned above so we can group them as partitioning methods.

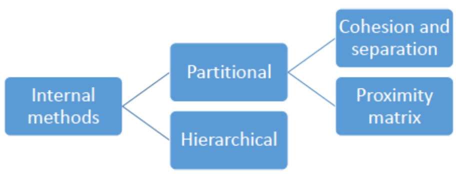

Figure 7: Internal validation methods Palacio-Niño and Berzal (2019)

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. 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. Later we will talk about the GUI to make the executions more attractive.

Algorithms of the package

These are the algorithms available within the package: `aggExCluster, agnes, apclusterK, clara, daisy, diana, fanny, fuzzy_cm, fuzzy_gk, fuzzy_gg, gama, hcluster, gmm, kmeans, mona, pam, pvpick and pvclust.`

Package Architecture

The main advantage of this package is that it allows to compare the most used clustering algorithms in the literature and to be able to compare them to determine which variable is the best behavior for the set of algorithms. With this package we will be able to compare the results based on the best variable and evaluate the results by means of a series of metrics that will indicate how the data has been distributed within our clusters.
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 dissimilarity 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_execute.` It represents a character vector with the algorithms executed independently of the package.
  - `measures_execute.` It represents a vector of characters with the measures of dissimilarity 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 dissimilarity and on the number of clusters.
- `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 dissimilarity and on the number of clusters.

Finally we have the plot_clustering methods to graphically represent the evaluation measurements by clusters as well as to export the results of both internal and external measurements in latex format with the export_external_file and export_internal_file methods

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 Vico 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 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. For reading files in csv format the default values have been used, except for the stringsAsFactors parameter which indicates how to treat the character vector.

Analysis of clustering methods using the Clustering package

Once the way to provide the data to the clustering method has been defined, we can execute it in two ways as we have indicated above. For this example we are going to use a dataset of those provided by the package. The parameters are as follows:

- df: is a `data.frame` type field with the data we are going to use to evaluate.
- min: This is a numeric field to indicate the minimum number of clusters.
- max: This is a numeric field to indicate the maximum number of clusters.
- 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`.

We go there with the test, we use the basketball dataset, with a range of clusters [3,5], for the gmm algorithm using as external metric (entropy) and as internal metric (dunn).

```
> 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 dissimilarity used by the algorithm to calculate the similarity between the data.
- Clusters: is the number of clusters used by the algorithm. When working with clustering it is necessary to indicate the number of clusters.
- Dataset: is the name of the `data.frame`. By default appears `data.frame` 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.
- entropy: as mentioned in the section 2, is the measure in charge of measuring the uncertainty in the prediction of natural classes provided by the P partition.

- dunn: is an internal evaluation measure, it is focused on measuring cohesion (how close the elements are to each other) and separation (they quantify the level of separation between clusters).
- timeInternal: time taken to implement internal evaluation measures.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`.

The basketball dataset containing 5 variables. The operation is to execute the same algorithm with the measurement of dissimilarity and the number of clusters for each variable.With this we intend to see if the behaviour is the same for all the variables, or if the choice of a variable influences the quality of the results. From the measures indicated we can start to evaluate the quality of the clusters. Depending on the selected metrics we can evaluate some characteristics or others.

From the data we can see that the value of the variables influences the quality of the results, therefore the next step is to select those variables that provide better quality.

```
> 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
```

The Clustering::best_ranked_external_metrics method is responsible for obtaining the best external variables for each execution, i.e. the best results by algorithm, measure of dissimilarity and number of clusters. 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 dissimilarity from the algorithms.

```
> 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
```

If we compare the results of Clustering::best_ranked_external_metrics with Clustering::evaluate_best_validation_external_by_metrics the functionality it gives us is that of being able to classify the quality of the data based on dissimilarity.This same method is also implemented for internal measures.
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 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 dissimilarity.

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 8 shows this representation

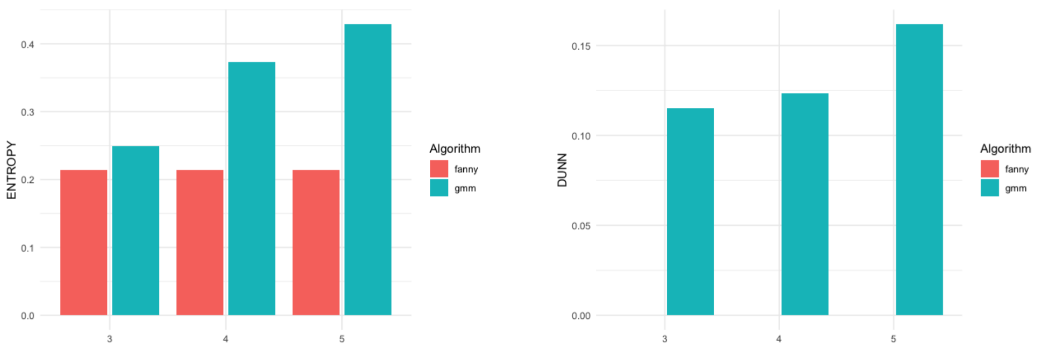

Figure 8: Graphical representation of evaluation measures

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:

```
> appClustering()
```

The execution will open our default browser with the interface. As it can be observed in the Figure 9, 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

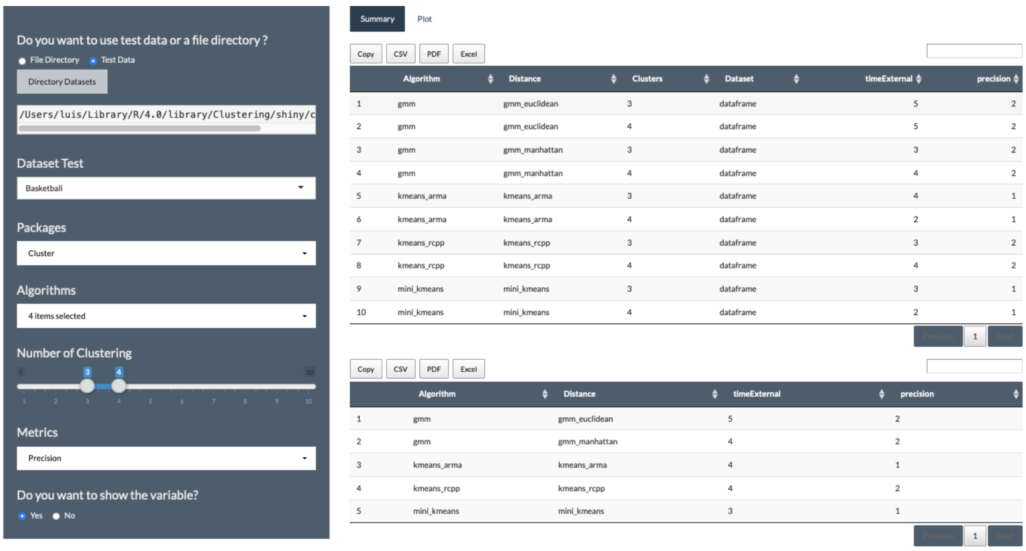

Figure 9: Clustering app user interface

which we can run our algorithm and finally in the central menu you can see the result of running the clustering algorithm.

The operation is very simple, we can choose to work with test data or file directory, work a range of clusters, specify algorithms individually or select the algorithms contained in a package, indicate what type of measures we want to evaluate and finally we can indicate if we want to see the results calculated or translate it to the variables of the dataset. We will see this operation step by step.

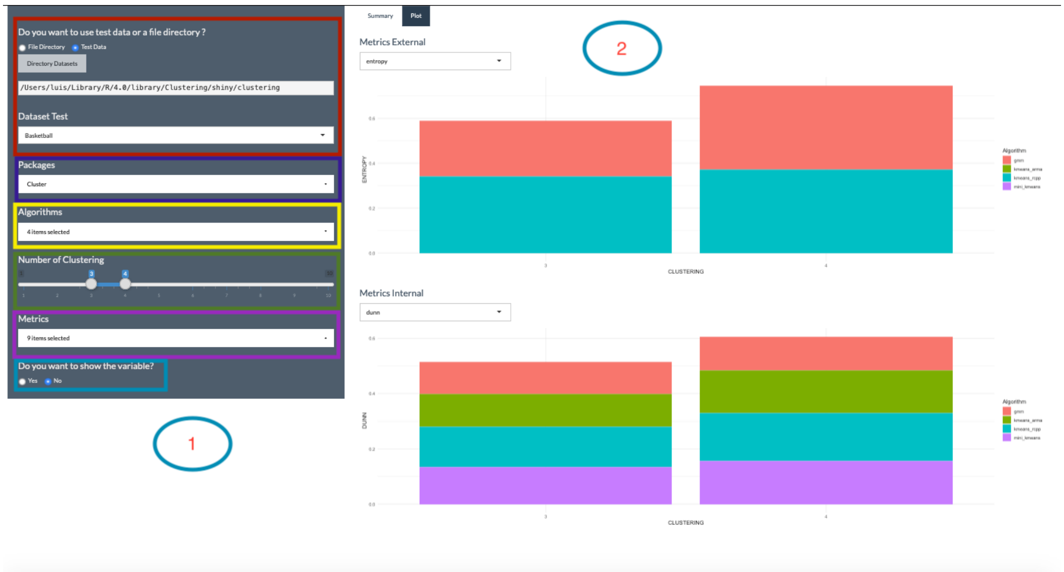

Figure 10: Clustering app user interface

As you can see in the Figure 10 we have two well differentiated parts:

1. In this section we can find the different parameters used by the clustering algorithm 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 variable number. 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 11: Execution of the default `clustering` method

In the image 12 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 12: Execution of the `clustering` method with variable to true

Note: the variables represent the dataset variables.

2. In main layout we have the options to represent the data.
   - To view the data in graphical mode as shown in the Figure, 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.
   - If we click on the summary tab we can see the data represented in tables. If you wish you can export the results in the following formats: csv,pdf and xls. If you wish you can copy the results.

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]

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]

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]

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

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. [p5]

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. [p6, 7]

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. [p6]

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]

A. M. Vico, F. Charte, 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. [p9]

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]

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