Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute InformingScience.org

IJIKM.org

Volume 15, 2020

A MULTICLUSTER APPROACH TO SELECTING INITIAL SETS FOR CLUSTERING OF CATEGORICAL DATA

Carlos Santos-Mangudo*	Complutense University of Madrid, Madrid, Spain	casant01@ucm.es
Antonio J. Heras	Complutense University of Madrid, Madrid, Spain	aheras@ccee.ucm.es

* Corresponding author

ABSTRACT

Aim/Purpose	This article proposes a methodology for selecting the initial sets for clustering categorical data. The main idea is to combine all the different values of every single criterion or attribute, to form the first proposal of the so-called multiclusters, obtaining in this way the maximum number of clusters for the whole dataset. The multiclusters thus obtained, are themselves clustered in a second step, according to the desired final number of clusters.
Background	Popular cluster methods for categorical data, such as the well-known K-Modes, usually select the initial sets by means of some random process. This fact introduces some randomness in the final results of the algorithms. We explore a different application of the clustering methodology for categorical data that overcomes the instability problems and ultimately provides a greater clustering efficiency.
Methodology	For assessing the performance of the proposed algorithm and its comparison with K-Modes, we apply both of them to categorical databases where the response variable is known but not used in the analysis. In our examples, that response variable can be identified to the real clusters or classes to which the observations belong. With every data set, we perform a two-step analysis. In the first step we perform the clustering analysis on data where the response variable (the real clusters) has been omitted, and in the second step we use that omitted information to check the efficiency of the clustering algorithm (by comparing the real clusters to those given by the algorithm).
Contribution	Simplicity, efficiency and stability are the main advantages of the multicluster method.

Accepting Editor Christine Nya-Ling TAN | Received: July 22, 2020 | Revised: September 20, 2020 | Accepted: September 28, 2020. Cite as: Santos-Mangudo, C., & Heras, A. J. (2020). A multicluster approach to selecting initial sets for clustering of categorical data. *Interdisciplinary Journal of Information, Knowledge, and Management*, *15,* 227-246. https://doi.org/10.28945/4643

Findings	The experimental results attained with real databases show that the multicluster algorithm has greater precision and a better grouping effect than the classical K-modes algorithm.
Recommendations for Practitioners	The method can be useful for those researchers working with small and medium size datasets, allowing them to detect the underlying structure of the data in an intuitive and reasonable way.
Recommendations for Researchers	The proposed algorithm is slower than K-Modes, since it devotes a lot of time to the calculation of the initial combinations of attributes. The reduction of the computing time is therefore an important research topic.
Future Research	We are concerned with the scalability of the algorithm to large and complex data sets, as well as the application to mixed data sets with both quantitative and qualitative attributes.
Keywords	clustering, categorical data, K-Modes

INTRODUCTION

The term *Cluster Analysis* encompasses a wide variety of techniques and methods, all of them aimed to a single purpose: to classify the items belonging to a given set, and to cluster them into a finite number of subsets or clusters. It is therefore a multivariate statistical procedure that takes a given dataset – a collection of items – as the starting point and classifies them into homogeneous groups in such a way as to maximize the similarity of individuals within the same cluster, and making at the same time the differences between different groups as large as possible.

K-Means and K-Modes are two popular algorithms for clustering numerical and categorical data, respectively. Both are based in the same methodology: they select k entities as initial representative points, i.e. the centers or centroids of the initial clusters; then they assign every object to its closest representative point, and these points and clusters are recalculated, repeating the process until no more changes are observed. K-Means and K-Modes choose as representative points the means and modes of the clusters, respectively.

K-Modes usually chooses k random entities as the initial modes, i.e. the centroids of the initial clusters. Nevertheless, as recognized in Huang (1998), this random selection of the initial seeds often leads to very different final cluster aggregations. In other words, the algorithm is instable because several executions over the same dataset can give different final clusters. K-Means is also affected by this problem, because the initial cluster centroids are not fixed and we have randomness in the following computation steps. Since this paper is focused on categorical data, we will only address K-Modes, as it is the standard and most popular method for clustering this kind of data. We think, however, that our methodology could be generalized for working with numerical or mixed data.

In this article, we want to explore an alternative application of the clustering methodology for categorical data that overcomes the instability problems and ultimately provides a greater clustering efficiency. Our method is based on the calculation of all possible combinations of the values of the attributes or criteria that characterize the different objects in the data set and always obtains the same initial groups as well as their centroids, assuming that the desired number of final clusters, k, is known in advance. Those combinations having a greater number of objects will be selected as starting points of the iterative process.

To carry out this process, an analysis of the different attributes is needed in the first place. In this first step, the algorithm calculates the number of clusters for every single criterion automatically, according to the different values of these criteria. In a second step, the algorithm calculates the first proposal of the so-called "multiclusters", which are built by forming all the non-empty combinations

of the single-criterion clusters obtained in the previous step. Finally, in the last step, the multiclusters obtained in the previous step (may be too many), are themselves clustered according to the desired final number of clusters, taking as starting point those multiclusters containing the highest number of objects.

The paper is organized as follows: after the Introduction, the first section (Literature Review) provides an overview of the main clustering algorithms and the previous attempts to solve the problem of instability of K-Modes. The second and third sections (Conceptual Explanation and Methods) explain the main features of the proposed clustering algorithm. In the fourth section (Experimental Results), the methodology is applied to several well-known real databases, showing an increase of the accuracy and clustering efficiency when compared with other popular algorithms. The two last sections (Discussion and Conclusions) conclude the paper.

LITERATURE REVIEW

Jain and Dubes (1988), in their book "Algorithms for Clustering Data", characterize Cluster Analysis as a tool for data exploration, complemented with visualization techniques. The objective of the Cluster Analysis is therefore to find the most natural way of grouping a set of individuals, objects, patterns, observations, etc., depending on the degree of similarity of their characteristics.

OVERVIEW OF CLUSTERING

Several types of methods have been developed, following different induction principles (as shown in Figure 1). Fraley and Raftery (1998) suggest classifying the cluster methodologies into two groups: *hierarchical* and *partitioned* methods. Han et al. (2011) suggest three groups: *density-based*, *model-based* and *grid-based* methods.

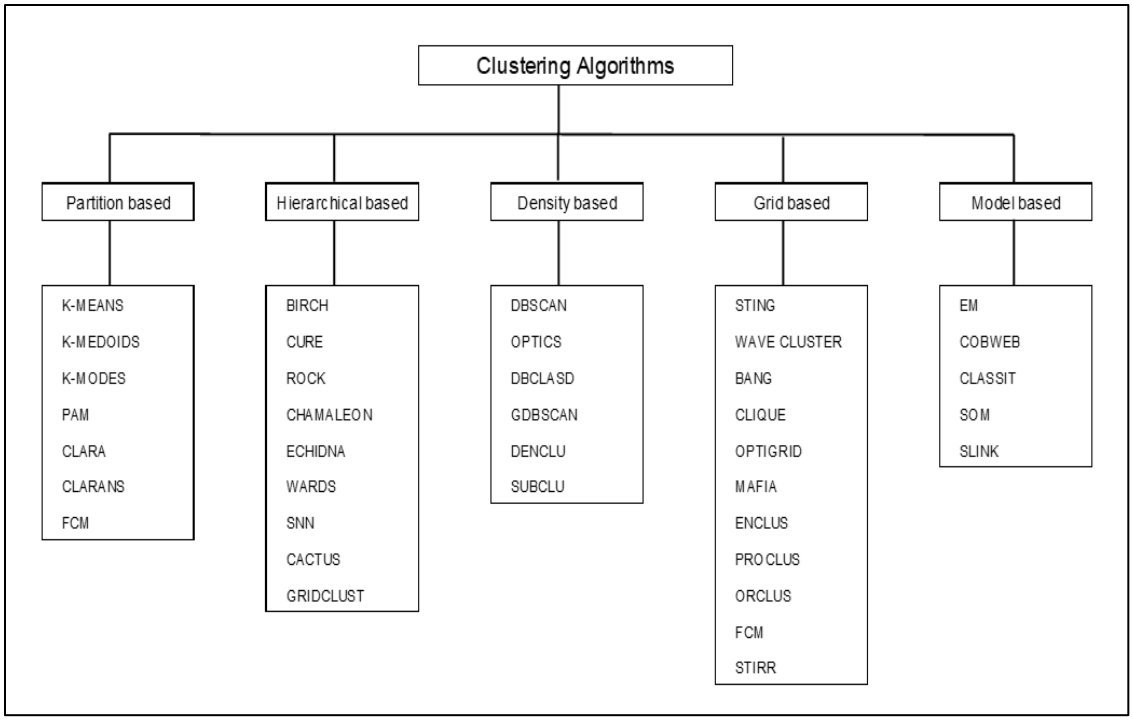

Figure 1: Taxonomy of cluster methods (Prakash et al., 2016)

Clustering methodologies can also be classified according to the different similarity measures that are used in the analysis: we find, among others, Aldenderfer and Blashfield (1984), Duda et al. (1973), Guha et al. (2000a, 2000b), Jain et al. (1999), Selosse et al. (2020), and Yuan et al. (2020). Many books

have also been published on this subject, such as Agresti (2018), Anderberg (1973), Bagirov et al. (2020), Bailey (1975), King (2015), Sibson (1976), Sneath and Sokal (1973), Upton (2017), and Wierzchon and Klopotek (2018).

There is a lot of literature devoted to the different cluster methodologies, which we briefly summarize in Table 1.

Table 1: Cluster methodologies

Methodology	Typical Algorithm	Authors
Grid	STING	Makhabel (2015)
Density	DBSCAN	Pietrzykowski (2017); Zhu et al. (2013)
Density	IDCUP	Altaf et al. (2020)
Partitioning	K-MEANS	MacQueen (1967)
Partitioning	K-MODES	Dorman and Maitra (2020); Huang (1997a, 1997b, 1998, 2009)
Partitioning	PAM, CLARA, CLARANS	Kaufman and Rousseeuw (1990)
Hierarchical	BIRCH	Chiu et al. (2001); Zhang et al. (1996, 1997)
Hybrid	CURE	Guha et al. (2000b)
Hierarchical	ROCK	Guha et al. (2000a)
Hierarchical	DIANA, AGNES	Kaufman and Rousseeuw (1990)
Mixed	K-PROTOTYPES	Huang et al. (2005); Ji et al. (2020); Kim (2017); Szepannek (2018)
Mixed	ClicoT	Behzadi et al. (2020)

POPULAR CLUSTER ALGORITHMS

Actually, *K-Means* (Forgy, 1965; McQueen, 1967) is one of the most popular cluster algorithms. This algorithm represents each cluster by its center of gravity – its mean value - and assigns the objects to their nearest clusters (using the Euclidean distance). After every object has been assigned to a cluster, the algorithm recalculates all the centers of gravity. The process is repeated until no change is observed in the formed clusters.

K-Means only works with numerical data. However, in many problems, we find categorical data, with nominal, ordinal, interval or binary variables, and in these cases, different types of clustering algorithms are needed. The well-known *K-Modes* algorithm, according to Huang (1997a, 1997b, 1998) can be considered as an adaptation of K-Means to categorical data, since both are inspired in similar ideas: K-Modes works in a similar way to K-Means, considering the modes of the clusters instead of their means, and using dissimilarities instead of numerical distances. A mode is a vector of elements that minimizes the dissimilarities between the vector itself and each object of the cluster.

To calculate the distance (or *dissimilarity*) between two objects X and Y described by m categorical attributes, the distance function in K-modes is defined as:

$$d(X, Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \quad (1)$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$

Here, xj and yj are the values of attribute j in X and Y, and $d(X, Y)$ gives equal importance to each category of an attribute. This function is often referred to as *simple matching dissimilarity measure* or *Hamming distance*. The larger the number of mismatches of categorical values between X and Y is, the more dissimilar the two objects.

Let N be a set of n categorical data observations described by m categorical attributes. When the distance function defined in Eq. (1) is used as the dissimilarity measure for categorical data observations, the cost function becomes:

$$\sum_{i=1}^{n} d(N_i, C_i) \quad (2)$$

where N_i is the ith element and C_i is the nearest cluster centroid to N_i. K-Modes minimizes the cost function defined in (2).

The K-modes algorithm assumes the number k of clusters as predetermined, and consists of the following steps (Huang, 1997a):

1. Select the k initial cluster centroids.
2. Assign every data observation to the cluster with the nearest centroid, according to Eq. (2).
3. Update the k clusters after the reallocation of step 2, and compute their modes, which will be the new centroids.
4. Repeat steps 2 and 3 until no more changes are observed.

LIMITATIONS OF EXISTING ALGORITHMS FOR CLUSTERING

A key issue for the performance of K-Modes is the selection of the seeds or initial centroids. This is usually done by means of some random procedure, but this random selection of the initial seeds often leads to very different final cluster aggregations. In other words, the algorithm is instable because several executions over the same dataset can give different final clusters, as recognized in Huang (1998), and Khan and Ahmad (2013), among others.

In order to overcome this problem, some solutions have been suggested in the literature. The performance of the K-Modes algorithm has been improved using the tabu search technique (Ng & Wong, 2002) and genetic algorithms (Gan et al., 2005). Outlier detection techniques have been applied to the initialization of K-Modes (Jiang et al., 2016; Knor & Ng, 1998), based on the idea that outliers should not be selected as initial centers of the clusters. Also, Bradley's and Fayyad's (1998) iterative initial-point refinement algorithm has improved the accuracy and repetitiveness of the clustering results.

Density-based multi-scale data condensation has also been used together with Hamming distance to extract the initial cluster centers from the datasets; see Khan and Ahmad (2013, 2015), and Mitra et al. (2002). Cao et al. (2009) compute the density of each data cluster and propose as initial clusters those with maximum average densities.

Wu et al. (2007) develop a density-based method to compute the initial cluster centers and to reduce the algorithmic complexity, however, there is some randomness in the final results and repeatability of the clustering results may not be achieved. Bai et al. (2012) propose a method to compute the initial cluster centers based on a density function and a distance function, and Dinh and Huynh (2020) propose a k-Pbc algorithm to improve cluster center initialization for categorical data clustering.

Khan and Ahmad (2013) propose a seed selection methodology with three attribute selection methods, based on the significance of attributes. The first method is the vanilla approach, where all the attributes are considered significant. The second method is the prominent attribute method, where an attribute is significant if the number of unique values of the attributes is lower than or equal to the required number of clusters; see also Khan and Ahmad (2012). The third method is to identify the most significant attributes by measuring the co-occurrence of their values with the values of the other attributes (Ahmad & Dey, 2007a, 2007b). The initial seed selection algorithm is applied to the attributes obtained by means of these three attribute selection methods, and K–Modes clustering algorithm is then executed (Sajidha et al., 2018).

In general, these methods are difficult to implement and some of them do not completely wipe out randomness. In the next section we will propose a simple clustering algorithm for categorical data

that uses the same distance function as K-Modes but overcomes its instability problems and also provides a greater clustering efficiency. As we commented in the Introduction, the key idea is to form the so-called "multiclusters", which are non-empty combinations of the different values of the attributes or criteria. Those multiclusters containing the highest number of objects will be taken as seeds of the clustering process. We will see that an algorithm based on this simple idea outperforms K-Modes both in terms of stability and clustering efficiency. In other terms, simplicity, stability and efficiency are the main advantages of the proposed algorithm.

CONCEPTUAL EXPLANATION

This section explains the main ideas of the proposed "K-multicluster" algorithm for categorical data.

The algorithm works as follows:

I. First, the clusters for each single criterion are easily calculated, since they coincide with the different categorical values of the criteria.

II. Second, once the clusters have been obtained for every single criterion or attribute, their values are combined to form the first proposal of the multi-clusters, obtaining in this way the maximum global number of clusters for the whole dataset.

   This way, we obtain the clusters that should appear when we consider each attribute as an independent entity of the other attributes existing in the database. Each one of these multi-clusters is based on the exact coincidence, for the objects belonging to the cluster, of all the values of their attributes. In other words, we obtain clusters in which all the objects have a 100% coincidence in their attributes. In order to visualize these coincidences, it is useful to build the so-called *Coincidence Matrix*, showing the number of coincidences between the attributes of every couple of clusters.

III. Third and last step, it is clear that the number of multi-clusters obtained before may be too large in many applications. For this reason, this set of clusters shall, in turn, be clustered, in order to obtain the predetermined number of clusters as the final output of the algorithm.

   To achieve this goal, we will start from the biggest clusters, that is, those clusters containing the highest number of objects, and we will try to link to them each of the other smaller clusters. In order to break the tie in those cases of equal similarities, we will use the Fleiss' Kappa coefficient (Fleiss et al., 1969, 2003; Fleiss, 1971), a well-known statistical measure for assessing the degree of coincidence or agreement between items with categorical features.

Figure 2 summarizes the structure of the "K-multicluster" algorithm:

Library definitions
Input: M = Categorical Dataset, N = data objects or data observations, R = Set of attributes in the data
Output: S = File of MultiCluster results of Dataset and MC = Multicluster Coincidence matrix
 # ------ Step 1.
for every attribute R ∈ M **do**
    Group by attribute "R" summarizing all their different possibilities
    Barplot to show graphically the values of categorical attribute.
    Compute cluster distribution C[R] for attribute (∀ R ∈ M) using a partitioning method like CLARA (Kaufman & Rousseeuw, 1990)
end for
 # ------ Step 2.
Combining all clusters obtained in step 1 for each attribute, a multicluster will be obtained for each observation of the dataset.
Compute *The Multicluster Matrix*
    Collecting all observations of the dataset belonging to the same multiclusters, we obtain the Multicluster Matrix showing the frequency of observations for each multicluster. We get in this way the maximum global number of clusters for the whole dataset, which coincides with the total number of multiclusters obtained.
for every row of the Multicluster Matix obtained before **do**
    Build a Coincidence Matrix, showing the coincidences between the attributes of every couple of clusters
end for
 # ------ Step 3.
Select K number of final clusters ∈ (2 ≤ K < Maximun global)
while not achieve K selected **do**
    for every Transmitter or row of the Coincidence Matrix **do**
        for every Receiver or column of the Coincidence Matrix **do**
            Select multiclusters containing the highest number of objects and compute the proximity between Transmitter and every possible Receiver using Fleiss' Kappa coefficient (Fleiss et al., 2003)
        end for
        Change Multicluster Transmitter to that Multicluster Receiver with the greather value of Fleiss' Kappa coefficient calculated before
    end for
end while

Figure 2: Proposed "K-multicluster" algorithm

METHODS

The following example illustrates the methodology. We use Unsupervised Breast Cancer data from an Institute of Oncology in Ljubljana, Slovenia, obtained from the *UCI Machine Learning Repository* (http: //archive.ics .uci.edu / ml /) (Dua & Graff, 2019), with 286 observations and 8 different attributes. We show part of this information in the matrix of Table 2, taking a random sample of 20 observations and 4 different attributes, where the rows represent the observations and the columns represent the attributes and their categorical values. Column OBSERVATION reports the codes of the patients, AGE shows their ages (at the time of diagnosis), NODE-CAPS indicates whether or not the cancer metastasizes to a lymph node, MENOPAUSE reports if the patients are pre- or post-menopausal at the time of diagnosis and BREAST indicates the breast side where the cancer appears.

Table 2: Data Matrix

OBSERVATION	AGE	NODE-CAPS	MENOPAUSE	BREAST
RE219	60-69	no	ge40	right
RE247	30-39	yes	premeno	left
NRE41	50-59	no	ge40	right
NRE127	30-39	yes	premeno	right
NRE130	40-49	yes	premeno	right
NRE180	40-49	no	premeno	right
RE237	40-49	no	premeno	right
NRE21	50-59	no	ge40	left
NRE86	50-59	no	ge40	left
NRE122	50-59	no	ge40	right
RE256	40-49	yes	premeno	right
RE242	40-49	yes	premeno	left
NRE168	40-49	yes	ge40	right
NRE105	40-49	no	premeno	right
NRE97	60-69	no	ge40	left
RE251	40-49	no	premeno	left
RE233	30-39	no	premeno	right
NRE99	40-49	no	premeno	left
NRE5	40-49	no	premeno	right
NRE162	40-49	yes	premeno	right

As a first step in the implementation of the algorithm, we calculate the clusters for the attributes, based on their different values (see Figure 3).

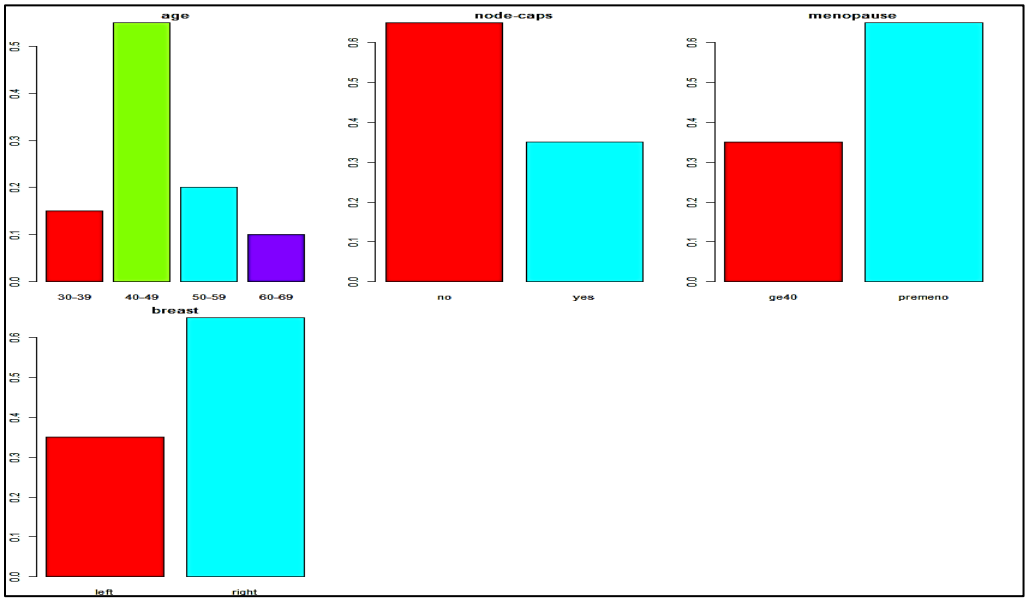

Figure 3: Natural Cluster distribution for each attribute

In our example, there are 4 clusters for the age intervals, 2 clusters for the node-caps, 2 clusters for the menopause attribute and 2 clusters for the breast attribute (see Table 3).

Table 3: Cluster division per attribute

AGE	CLUSTER
60-69	1
30-39	2
50-59	3
40-49	4

NODE-CAPS	CLUSTER
no	1
yes	2

MENOPAUSE	CLUSTER
ge40	1
premeno	2

BREAST	CLUSTER
right	1
left	2

Combining all the different possibilities, the maximum number of possible clusters that could be obtained is $\prod_{i=1}^{n} C_{v(i)}$ where "v(i)" denote the i-th attribute, "n" is the total number of attributes included in our data base and "C" represent the number of clusters that has been calculated by the algorithm for each attribute.

In our case, the total number of possible clusters will be (as shown in Table 4):

$$\prod_{i=1}^{4} C_{v(i)} = 4 * 2 * 2 * 2 = 32$$

Table 4: MultiCluster attribute Matrix

Observation	Multicluster	age	node-caps	menopause	breast
RE219	1111	1	1	1	1
RE247	2222	2	2	2	2
NRE41	3111	3	1	1	1
NRE127	2221	2	2	2	1
NRE130	4221	4	2	2	1
NRE180	4121	4	1	2	1
RE237	4121	4	1	2	1
NRE21	3112	3	1	1	2
NRE86	3112	3	1	1	2
NRE122	3111	3	1	1	1
RE256	4221	4	2	2	1
RE242	4222	4	2	2	2
NRE168	4211	4	2	1	1
NRE105	4121	4	1	2	1
NRE97	1112	1	1	1	2
RE251	4122	4	1	2	2
RE233	2121	2	1	2	1
NRE99	4122	4	1	2	2
NRE5	4121	4	1	2	1
NRE162	4221	4	2	2	1

However, most of them are empty. In fact, we only find 12 nonempty multiclusters, which can be represented in a MultiCluster Data Table (Table 5).

Table 5: Multicluster Data Table

OBSERVATION	K-MULTICLUSTER	AGE	NODE-CAPS	MENOPAUSE	BREAST
RE219	**1111**	60-69	no	ge40	right
NRE97	**1112**	60-69	no	ge40	left
RE233	**2121**	30-39	no	premeno	right
NRE127	**2221**	30-39	yes	premeno	right
RE247	**2222**	30-39	yes	premeno	left
NRE41	**3111**	50-59	no	ge40	right
NRE122		50-59	no	ge40	right
NRE21	**3112**	50-59	no	ge40	left
NRE86		50-59	no	ge40	left
NRE180	**4121**	40-49	no	premeno	right
RE237		40-49	no	premeno	right
NRE105		40-49	no	premeno	right
NRE5		40-49	no	premeno	right
RE251	**4122**	40-49	no	premeno	left
NRE99		40-49	no	premeno	left
NRE168	**4211**	40-49	yes	ge40	right
NRE130	**4221**	40-49	yes	premeno	right
RE256		40-49	yes	premeno	right
NRE162		40-49	yes	premeno	right
RE242	**4222**	40-49	yes	premeno	left

According to the information in Table 5, the maximum number of multiclusters is 12, and based on the Coincidence Matrix (Table 6), the minimum number of multiclusters is one (corresponding to the biggest cluster 4121). Of course, the decision maker can choose any desired number of clusters, strictly between 1 and 12.

Table 6: Matrix of the coincidences between the multiclusters

Multicluster	# obs	1111	1112	2121	2221	2222	4211	4222	3111	3112	4122	4221	4121
1111	1	0	3	2	1	0	2	0	3	2	1	1	2
1112	1			1	0	1	1	1	2	3	2	0	1
2121	1				3	2	1	1	2	1	2	2	3
2221	1					3	2	2	1	0	1	3	2
2222	1						1	3	0	1	2	2	1
4211	1							2	2	1	1	3	2
4222	1									1	3	3	2
3111	2									3	1	1	2
3112	2										2	0	1
4122	2											2	3
4221	3												3
4121	4												

EXPERIMENTAL PROCESS

Let us suppose that, based on the information of Tables 4 and 5, the decision maker chooses to calculate only two final clusters. Looking at their sizes, it is clear that these clusters must be based on the last two rows of both Tables, clusters 4121 and 4221.

As commented before, we follow two procedures to attach clusters. The first procedure requires to tie together those clusters sharing the highest number of features. We will associate similar clusters by taking into account the number of attributes' values that they share: for example, the cluster 3111 would be associated to cluster 3112 and not to cluster 4121, because in the first case the two clusters share 3 values, and in the second case only 2 (see Figure 4)

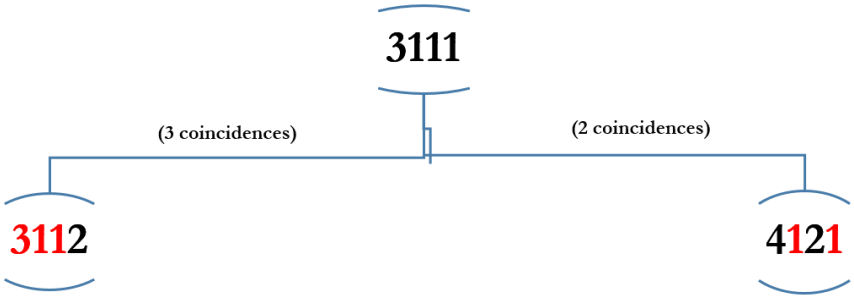

Figure 4: Multicluster association diagram

In order to break the tie in those cases of equal similarities, we use a second procedure: the Fleiss' Kappa coefficient (Fleiss et al., 1969, 2003; Fleiss, 1971): the cluster 4222, for instance, shares three attributes with clusters 4122 and 4221, and a smaller number of attributes with the other clusters, which are therefore discarded; since cluster 4122 gets the best Kappa concordance value, we conclude that clusters 4222 and 4122 will be tied together.

The diagram in Figure 5 shows the whole process of the clusters' associations.

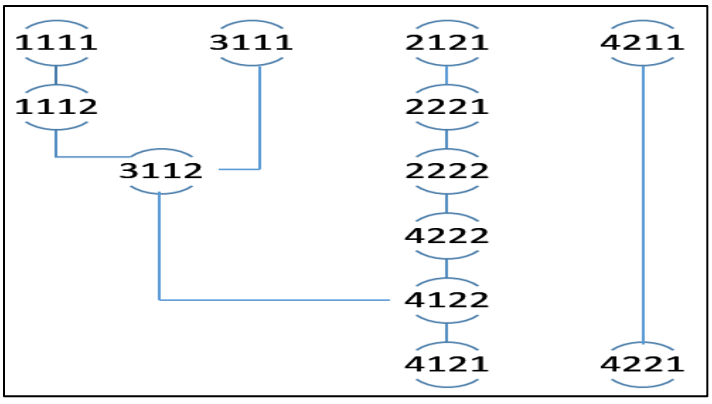

Figure 5: Multicluster association diagram

Table 7 shows the final clustering for each element in the database of Table 2. Also, in the last three columns of Table 7 we show the results of running three times the K-Modes algorithm on this database. We see how in this case K-Modes leads to different final results, i.e. the same observations are not always in the same clusters, may be due to the random search of the initial centroids. The "top-down" multicluster methodology presented in this paper does not face this problem, since it always leads to the same final clusters.

Table 7: Example of final clustering after three different executions of the algorithms

Observation	age	node-caps	menopause	breast	Multicluster exe1	Multicluster exe2	Multicluster exe3	K-modes exe1	K-modes exe2	K-modes exe3
RE219	60-69	no	ge40	right	4121	4121	4121	1	1	1
RE247	30-39	yes	premeno	left	4121	4121	4121	2	2	2
NRE41	50-59	no	ge40	right	4121	4121	4121	1	1	1
NRE127	30-39	yes	premeno	right	4121	4121	4121	2	1	2
NRE130	40-49	yes	premeno	right	4221	4221	4221	2	1	2
NRE180	40-49	no	premeno	right	4121	4121	4121	2	1	2
RE237	40-49	no	premeno	right	4121	4121	4121	2	1	2
NRE21	50-59	no	ge40	left	4121	4121	4121	1	2	1
NRE86	50-59	no	ge40	left	4121	4121	4121	1	2	1
NRE122	50-59	no	ge40	right	4121	4121	4121	1	1	1
RE256	40-49	yes	premeno	right	4221	4221	4221	2	1	2
RE242	40-49	yes	premeno	left	4121	4121	4121	2	2	2
NRE168	40-49	yes	ge40	right	4221	4221	4221	2	1	1
NRE105	40-49	no	premeno	right	4121	4121	4121	2	1	2
NRE97	60-69	no	ge40	left	4121	4121	4121	1	2	1
RE251	40-49	no	premeno	left	4121	4121	4121	1	2	2
RE233	30-39	no	premeno	right	4121	4121	4121	2	1	2
NRE99	40-49	no	premeno	left	4121	4121	4121	1	2	2
NRE5	40-49	no	premeno	right	4121	4121	4121	2	1	2
NRE162	40-49	yes	premeno	right	4221	4221	4221	2	1	2

EXPERIMENTAL RESULTS

DATASETS USED FOR EVALUATION

For assessing the performance of the proposed algorithm and its comparison with other clustering algorithms, we apply them to categorical databases (see Table 8) where the response variable is known but not used in the analysis. In our examples, that response variable can be identified with the real clusters or classes to which the observations belong. With every data set, we perform a two-step analysis. In the first step we perform the clustering analysis on data where the response variable (the real clusters) has been omitted, and in the second step we use that omitted information to check the efficiency of the clustering algorithm (by comparing the real clusters to those given by the algorithm). Actually, this is a procedure commonly used in the clustering literature: see, among others, Yu et al. (2018), and Zhu and Ma (2018).

Table 8: The datasets used in the experimental analysis

Dataset	Nbr. Observations	Nbr. attributes	Nbr. Maximum clusters	Nbr. Final clusters
Ballons	20	4	16	2
Bank Marketing	4521	11	3144	2
German credit data	1000	17	996	2
House congressional voting	435	15	325	2
Mushroom	8416	22	8076	2
Tic Tac Toe	958	9	958	2

UCI Machine Learning Repository

The following databases can be found in the *UCI Machine Learning Repository* (http: //archive.ics .uci.edu / ml /) (Dua & Graff, 2019):

(a) The *"Ballons" dataset* contains 20 observations and a total of 4 categorical attributes representing different conditions used in cognitive psychology experiments, allowing to classify them into 2 different clusters.

(b) The "*House Congressional Voting*" dataset contains 435 observations and a total of 15 categorical attributes, from the 1984 U.S. Congressional Voting Records, which are finally classified into 2 different clusters.

(c) The *"Mushroom" dataset* contains 8416 observations and a total of 22 categorical attributes describing samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota families, which are finally classified into 2 different clusters.

(d) The *"Tic Tac Toe" dataset* contains 958 observations encoding the complete set of possible board configurations at the end of tic-tac-toe games, with a total of 9 categorical attributes, which are finally classified into 2 different clusters.

MLD Machine Learning Data Repository

From the *Machine Learning Data Repository* (https://www.mldata.io/datasets/):

(e) The *"Bank Marketing" dataset* contains 4521 observations and a total of 11 categorical attributes (five numerical attributes have been removed), about the subscription of clients, aiming to distinguish 2 clusters.

(f) The *"German Credit" dataset* contains 1000 observations and a total of 17 categorical attributes, (three numerical attributes have been removed), about customers' credit ratings, aiming to distinguish 2 different clusters.

Table 8 shows a brief summary of the databases that have been considered for our experimental analysis, where the columns are interpreted as follows:

1. Dataset: name of the dataset
2. Nbr. Observations: number of observations in the dataset.
3. Nbr. Attributes: number of attributes in the dataset.
4. Nbr. Maximum clusters: maximum number of clusters that could be obtained.
5. Nbr. Final clusters: desired number (k) of final clusters.

Performance Evaluation Metric

For the comparisons between different cluster algorithms, we will use here the well-known *Confusion Matrix*, which is probably the most popular tool for assessing the precision and accuracy of the clustering algorithms, see Tharwat (2018), Townsend (1971), and Visa et al. (2011).

From the confusion matrix shown in Table 9, the measures of the Accuracy, F1-Score, Recall/TPR, Precision/PPV and NPV can be calculated. These measures are also commonly used to evaluate the accuracy of the data clustering systems (Powers, 2007; Shung, 2018; Swets, 1988; Tharwat, 2018; Trevethan, 2017; Van Rijsbergen, 1979).

Table 9: Confusion Matrix

	Positive	**Negative**
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Accuracy (AC) measure, one of the most commonly used measures of clustering performance, is defined as the ratio between the correctly classified samples and the total number of samples (Eq. 3)

$$AC = Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

F1-Score (Eq. 6), which can be considered as a harmonic mean of Accuracy (Eq. 3) and Recall (Eq. 5). It is a good precision measure when the data are unbalanced between the clusters. In the case of balanced data, it is more common to use the Accuracy (Eq. 4)

$$F1 = F1\text{-}Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

Recall (RE) or True positive rate (TPR), is defined as the ratio between the positive correctly classified samples and the total number of positive samples (Eq. 5)

$$RE = Recall = TPR\ (True\ Positive\ Rate) = \frac{TP}{TP+FN} \quad (5)$$

Precision (PR) or Positive Prediction Value (PPV), is defined as the ratio between the positive samples that were correctly classified and the total number of positive predicted samples (Eq. 6)

$$PR = Precision = PPV\ (Positive\ Prediction\ Value) = \frac{TP}{TP+FP} \quad (6)$$

Negative Predictive Value (NPV) or inverse precision, is defined as the ratio between the negative samples that were correctly classified and the total number of negative predicted samples (Eq. 7)

$$NPV\ (Negative\ Prediction\ Value) = \frac{TN}{TN+FN} \quad (7)$$

According to Kohavi and Provost (1998), Visa et al. (2011), and Yevseyeva et al. (2013), in all these expressions, TP, TN, FP and FN stand for the True Positives, True Negatives, False Positives and False Negatives, respectively, given by the clustering method.

Table 10 shows the main results of the comparison between K-Multicluster and K-Modes algorithms, using the Accuracy, F1-score, Recall/TPR, Precision/PPV and NPV measures to assess the accuracy of the clustering effect.

Table 10: Comparison between K-multicluster and K-modes

	Accuracy		F1/Score		Recall / TPR		Precision / PPV		NPV	
Dataset	K-Modes	K-Multicluster	K-Modes	K-Multicluster	K-Modes	K-Multicluster	K-Modes	K-Multicluster	K-Modes	K-Multicluster
Ballons	0,550	**0,700**	0,308	**0,400**	0,250	0,250	0,400	**1,000**	0,600	**0,667**
Bank Marketing	0,481	**0,611**	0,631	**0,746**	0,501	**0,644**	0,851	**0,886**	0,078	**0,117**
German credit data	0,508	**0,676**	0,639	**0,799**	0,624	**0,919**	0,656	**0,707**	0,367	0,367
House congressional voting	0,845	**0,878**	0,864	**0,902**	0,797	**0,918**	**0,942**	0,888	0,742	**0,862**
Mushroom	0,306	**0,534**	0,038	**0,696**	0,025	**1,000**	0,072	**0,533**	0,360	**1,000**
Tic Tac Toe	0,504	**0,621**	0,598	**0,758**	0,564	**0,906**	0,636	**0,651**	0,322	**0,322**

Figure 6 shows a graphical representation of the comparison, where positive and negative bars are associated to a better efficiency of K-multicluster and K-modes methods, respectively.

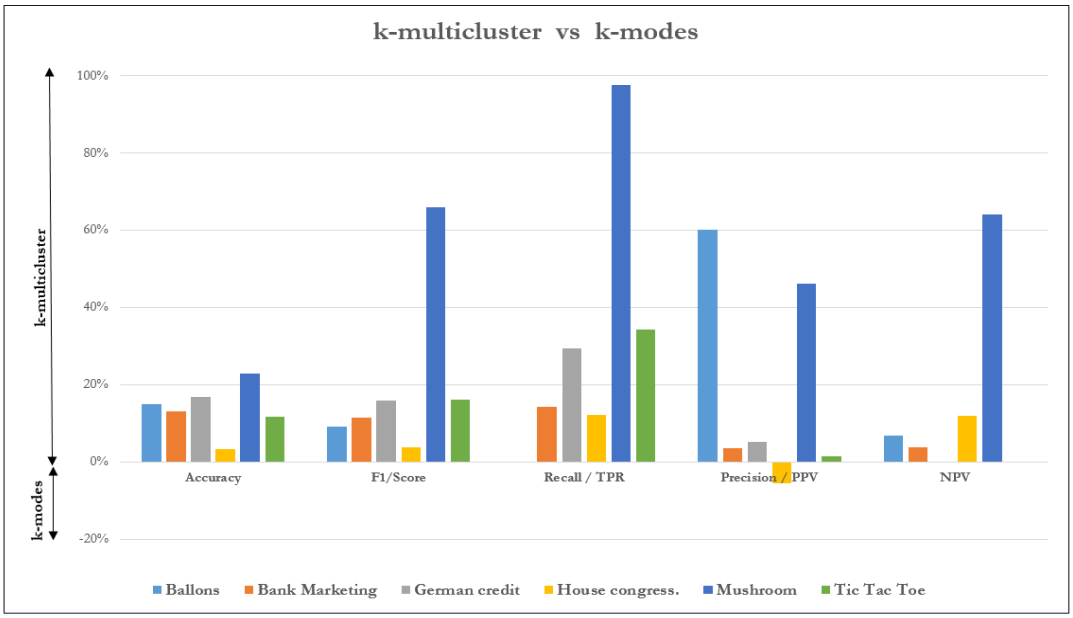

Figure 6: Multicluster association diagram

From the numerical results shown in Table 10 and the graphical representations in Figures 6, we conclude that the proposed K-multicluster algorithm outperforms K-Modes in all the accuracy measures.

DISCUSSION

The main idea behind the K-Multicluster methodology is very easy to explain in intuitive terms: since the data with qualitative attributes form natural clusters according to the different combinations of these attributes, it makes sense to classify the data starting from the biggest clusters, i.e. those having the biggest number of observations from the beginning, and then aggregating to them the rest of the clusters according to their degree of similarity. It is remarkable that such a simple algorithm outperforms the popular K-Modes both in terms of clustering efficiency and repeatability.

It is not surprising the repeatability of the results of the K-Multicluster method, since there is no randomness at all in the selection of the initial clusters. On the contrary, these clusters are chosen according to a natural criterion, since they are defined from the most frequent combinations of the attributes. It is more striking the great clustering efficiency of the methodology. As we have seen in the databases examples, in many cases K-Multicluster outperforms K-Modes when we consider the well-known Accuracy, F1-score, Recall, Precision and Negative Precision Value efficiency measures.

It is important to remark that the Multicluster algorithm is slower than K-Modes, since it devotes a lot of time to the initial calculation of the clusters. Actually, the algorithm works well with small or medium size databases, since in these cases it is affordable to calculate clusters based on the combinations of attributes. However, we have empirically checked that even in large datasets most of these possible clusters are empty, and this is a good thing from the perspective of computational efficiency, not affecting the quality of the final cluster distribution.

Simplicity, efficiency and stability are, therefore, the main advantages of the K-Multicluster method.

CONCLUSION AND FUTURE WORK

In this paper, a K-Multicluster algorithm is proposed for clustering categorical datasets in subgroups or clusters. The algorithm follows a "top-down" methodology, forming in the first step the so-called "multiclusters" or combinations of all the different values of the attributes, and then reducing their number until obtaining the desired number of clusters.

This methodology overcomes some of the drawbacks of the well-known K-Modes algorithm, perhaps the most popular algorithm for cluster analysis for categorical datasets. Unlike K-modes, the K-multicluster algorithm always leads to the same final results, since it takes as starting point the biggest multiclusters.

Besides, we have empirically compared the clustering efficiency of both algorithms in six categorical databases, using five well-known accuracy measures (Accuracy, F1-score, Recall, Precision and Negative Precision Value), obtaining a better performing and a more stable clustering in each execution than K-Modes algorithm. We conclude that the multicluster algorithm can be considered as a powerful tool for cluster analysis.

We think that this method can be useful for those researchers working with small and medium size datasets, allowing them to detect the underlying structure of the data in an intuitive and reasonable way. Regarding the future developments of this research, we are concerned with the reduction of the computing time, as well as the extension of the methodology to clustering more complex and larger data sets, including those with mixed – qualitative and quantitative - types of attributes.

REFERENCES

Agresti, A. (2018). *An introduction to categorical data analysis* (3rd ed.). Wiley. https://dl.uswr.ac.ir/bitstream/Hannan/130987/1/Alan_Agresti_An_Introduction_to_Categorical_Data_Analysis_Wi.pdf

Ahmad, A., & Dey, L. (2007a). A k-mean clustering algorithm for mixed numeric and categorical data. *Data and Knowledge Engineering, 63*(2), 503–527. https://doi.org/10.1016/j.datak.2007.03.016

Ahmad, A., & Dey, L. (2007b). A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters*, *28*(1), 110–118. https://doi.org/10.1016/j.patrec.2006.06.006

Aldenderfer, M. S., & Blashfield, R. K. (1984). *Cluster analysis.* Series: Quantitative applications in the social sciences. Sage Publications. https://doi.org/10.4135/9781412983648

Altaf, S., Waseem, M. W., & Kazmi, L. (2020). IDCUP Algorithm to classifying arbitrary shapes and densities for center-based clustering performance analysis. *Interdisciplinary Journal of Information, Knowledge, and Management*, *15*, 91–108. https://doi.org/10.28945/4541

Anderberg, M. R. (1973). *Cluster analysis for applications*. Probability and Mathematical Statistics: A Series of Monographs and Textbooks. Academic Press Inc. https://doi.org/10.1016/c2013-0-06161-0

Bagirov, A. M., Karmitsa, N., & Taheri, S. (2020). Introduction to clustering. In A. M. Bagirov, N. Karmitsa, & S. Taheri, *Partitional clustering via non-smooth optimization* (pp 3-13). Unsupervised and Semi-Supervised Learning. Springer. https://doi.org/10.1007/978-3-030-37826-4_1

Bai, L., Liang, J., Dang, C., & Cao, F. (2012). A cluster centers initialization method for clustering categorical data. *Expert Systems with Applications*, *39*(9), 8022–8029. https://doi.org/10.1016/j.eswa.2012.01.131

Bailey, K. D. (1975). Cluster Analysis. Sociological Methodology, 6, 59-128. https://doi.org/10.2307/270894

Behzadi, S., Müller, N. S., Plant, C., & Böhm, C. (2020). Clustering of mixed-type data considering concept hierarchies: Problem specification and algorithm. *International Journal of Data Science and Analytics*, *10*(3), 233–248. https://doi.org/10.1007/s41060-020-00216-2

Bradley, P. S., & Fayyad, U. M. (1998). Refining initial points for k-means clustering. In J. Shavlik (Ed.), *Proceedings of the 15th International Conference on Machine Learning (ICML98), Volume 98* (pp. 91-99). San Francisco: Morgan Kaufmann. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.8528&rep=rep1&type=pdf

Cao, F., Liang, J., & Bai, L. (2009). A new initialization method for categorical data clustering. *Expert Systems with Applications*, *36*(7), 10223–10228. https://doi.org/10.1016/j.eswa.2009.01.060

Chiu, T., Fang, D. P., Chen, J., Wang, Y., & Jeris, C. (2001). A robust and scalable clustering algorithm for mixed type attributes in large database environment. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)* (pp. 263–268). San Francisco: ACM. https://doi.org/10.1145/502512.502549

Dinh, D. T., & Huynh, V. N. (2020). K–PbC: An improved cluster center initialization for categorical data clustering. *Applied Intelligence*, *50*(8), 2610–2632. https://doi.org/10.1007/s10489-020-01677-5

Dorman, K. S., & Maitra, R. (2020). An efficient k-modes algorithm for clustering categorical datasets. *arXiv preprint arXiv:2006.03936*. https://arxiv.org/pdf/2006.03936

Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository*. University of California, Irvine. https://archive.ics.uci.edu/ml/citation_policy.html

Duda, O., Hart, E., & Stork, D. G. (1973). *Pattern classification*. Wiley.

Fleiss, J. L., Cohen, J., & Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, *72*(5), 323–327. https://doi.org/10.1037/h0028106

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378–382. https://doi.org/10.1037/h0031619

Fleiss, J. L., Levin, B., & Paik, M. C. (2003). *Statistical methods for rates and proportions* (3rd ed.). Wiley Series in Probability and Statistics. John Wiley & Sons, Inc. https://doi.org/10.1002/0471445428

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency vs interpretability of classification. *Biometrics*, *21*(3), 768–780.

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, *41*(8), 578–588. https://doi.org/10.1093/comjnl/41.8.578

Gan, G., Yang, Z., & Wu, J. (2005). A genetic k-modes algorithm for clustering categorical data. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3584*, 195–202. https://doi.org/10.1007/11527503_23

Guha, S., Rastogi, R., & Shim, K. (2000a). ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, *25*(5), 345–366. https://doi.org/10.1016/S0306-4379(00)00022-3

Guha, S., Rastogi, R., & Shim, K. (2000b). CURE: An efficient clustering algorithm for large databases. *Information Systems*, *26*(1), 35–58. https://doi.org/10.1016/S0306-4379(01)00008-4

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques*. Morgan Kaufmann, Elsevier.

Huang, Z. (1997a). A fast clustering algorithm to cluster very large categorical data sets in data mining. *Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery* (pp. 1–8). Vancouver, BC: The University of British Columbia, Dept. of Computer Science. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.83&rep=rep1&type=pdf

Huang, Z. (1997b). Clustering large data sets with mixed numeric and categorical values. *Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference* (pp. 21–34). Singapore: World Scientific. https://grid.cs.gsu.edu/~wkim/index_files/papers/kprototype.pdf

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, *2*(3), 283–304. https://doi.org/10.1023/A:1009769707641

Huang, J. Z., Ng, M. K., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(5), 657–668. https://doi.org/10.1109/TPAMI.2005.95

Huang, Z. (2009). Clustering categorical data with k-modes. In *Encyclopedia of data warehousing and mining* (2nd ed.) (pp. 246–250). https://doi.org/10.4018/978-1-60566-010-3.ch040

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data*. Prentice Hall.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A survey. *ACM Computing Surveys*, *31*(3), 264–323. https://doi.org/10.1145/331499.331504

Jiang, F., Liu, G., Du, J., & Sui, Y. (2016). Initialization of k-modes clustering using outlier detection techniques. *Information Sciences*, *332*, 167–183. https://doi.org/10.1016/j.ins.2015.11.005

Ji, J., Pang, W., Li, Z., He, F., Feng, G., & Zhao, X. (2020). Clustering mixed numeric and categorical data with cuckoo search. *IEEE Access*, *8*, 30988–31003. https://doi.org/10.1109/ACCESS.2020.2973216

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. John Wiley and Sons. https://doi.org/10.1002/9780470316801

Khan, S. S., & Ahmad, A. (2012). Cluster center initialization for categorical data using multiple attribute clustering. In *3rd MultiClust Workshop @ 2012 SIAM International Conference on Data Mining* (pp. 3–10). https://www.academia.edu/download/30720297/WS06.pdf#page=9

Khan, S. S., & Ahmad, A. (2013). Cluster center initialization algorithm for k-modes clustering. *Expert Systems with Applications*, *40*(18), 7444–7456. https://doi.org/10.1016/j.eswa.2013.07.002

Khan, S. S., & Ahmad, A. (2015). Computing initial points using density based multiscale data condensation for clustering categorical data. In *2nd International Conference on Applied Artificial Intelligence (ICAAI), Volume 3* (pp. 1-7). https://pdfs.semanticscholar.org/cd38/45e913e369985dba2d3aac933834a520b0c4.pdf

Kim, B. (2017). A fast k-prototypes algorithm using partial distance computation. *Symmetry*, *9*(4), 58. https://doi.org/10.3390/sym9040058

King, R. S. (2015). *Cluster analysis and data mining: An introduction*. Mercury Learning & Information.

Knor, E. M., & Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases* (pp. 392-403). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.5746&rep=rep1&type=pdf

Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, *30*(2/3), 271–274. https://doi.org/10.1023/A:1017181826899

Makhabel, B. (2015). *Learning data mining with R*. Packt Publishing Ltd.

McQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *Volume 233* (pp. 281–297). https://www.cs.cmu.edu/~bhiksha/courses/mlsp.fall2010/class14/macqueen.pdf

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Density-based multiscale data condensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(6), 734–747. https://doi.org/10.1109/TPAMI.2002.1008381

Ng, M. K., & Wong, J. C. (2002). Clustering categorical data sets using tabu search techniques. *Pattern Recognition*, *35*(12), 2783–2790. https://doi.org/10.1016/S0031-3203(02)00021-3

Pietrzykowski, M. (2017). Local regression algorithms based on centroid clustering methods. *Procedia Computer Science*, *112*, 2363–2371. https://doi.org/10.1016/j.procs.2017.08.210

Powers, D. M. W. (2007). Evaluation: from precision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63. https://dspace2.flinders.edu.au/xmlui/bitstream/handle/2328/27165/Powers%20Evaluation.pdf

Prakash, K., Anuradha, K., & Vasumathi, D. (2016). A survey on clustering techniques for multi-valued data sets. *Global Journal of Computer Science and Technology: C Software & Data Engineering*, *16*(1), 43-50. https://www.computerresearch.org/index.php/computer/article/download/1463/1450

Sajidha, S. A., Chodnekar, S. P., & Desikan, K. (2018). Initial seed selection for k-modes clustering – A distance and density based approach. *Journal of King Saud University – Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2018.04.013

Selosse, M., Jacques, J., & Biernacki, C. (2020). Model-based co-clustering for mixed type data. *Computational Statistics and Data Analysis*, *144*, 106866. https://doi.org/10.1016/j.csda.2019.106866

Shung, K. P. (2018, March 15). Accuracy, precision, recall or F1? *Towards Data Science*. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Sibson, R. (1976). Reviewed work: Clustering algorithms. By J. A. Hartigan. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *25*(1), 70. Wiley. https://doi.org/10.2307/2346526

Sneath, P. H. A., & Sokal, R. R. (1973). *Numerical taxonomy: The principles and practice of numerical classification*. W. H. Freeman.

Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, *240*(4857), 1285–1293. American Association for the Advancement of Science (AAAS). https://doi.org/10.1126/science.3287615

Szepannek, G. (2018). clustMixType: User-friendly clustering of mixed-type data in R. *The R Journal*, *10*(2), 200-208. https://doi.org/10.32614/RJ-2018-048

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2018.08.003

Townsend, J. T. (1971). Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics*, *9*(1), 40–50. https://doi.org/10.3758/BF03213026

Trevethan, R. (2017). Sensitivity, specificity, and predictive values: Foundations, pliabilities, and pitfalls in research and practice. *Frontiers in Public Health*, *5*, 307. https://doi.org/10.3389/fpubh.2017.00307

Upton, G. J. G. (2017). *Categorical data analysis by example*. Wiley. https://doi.org/10.1002/9781119450382

Van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). Butterworths.

Visa, S., Ramsay, B., Ralescu, A., & Knaap, E. (2011). Confusion matrix-based feature selection. *Proceedings of the 22nd Midwest Artificial Intelligence and Cognitive Science, Volume 710* (pp. 120–127). http://ceur-ws.org/Vol-710/paper37.pdf

Wierzchoń, S. T., & Kłopotek, M. A. (2018). *Modern algorithms of cluster analysis*. Springer. https://doi.org/10.1007/978-3-319-69308-8

Wu, S., Jiang, Q., & Huang, J. Z. (2007). A new initialization method for clustering categorical data. In Z. H. Zhou, H. Li, & Q. Yang (Eds.), *Advances in knowledge discovery and data mining* (pp. 972–980). Springer. https://doi.org/10.1007/978-3-540-71701-0_109

Yevseyeva, I., Basto-Fernandes, V., Ruano-Ordás, D., & Méndez, J. R. (2013). Optimizing anti-spam filters with evolutionary algorithms. *Expert Systems with Applications*, *40*(10), 4010–4021. https://doi.org/10.1016/j.eswa.2013.01.008

Yuan, F., Yang, Y., & Yuan, T. (2020). A dissimilarity measure for mixed nominal and ordinal attribute data in k-modes algorithm. *Applied Intelligence*, *50*(5), 1498–1509. https://doi.org/10.1007/s10489-019-01583-5

Yu, S. S., Chu, S. W., Wang, C. M., Chan, Y. K., & Chang, T. C. (2018). Two improved k-means algorithms. *Applied Soft Computing Journal*, *68*, 747–755. https://doi.org/10.1016/j.asoc.2017.08.032

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *ACM SIGMOD Record (ACM Special Interest Group on Management of Data)*, *25*(2), 103–114. https://doi.org/10.1145/235968.233324

Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, *1*(2), 141–182. https://doi.org/10.1023/A:1009783824328

Zhu, L., Lei, J. S., Bi, Z. Q., & Yang, J. (2013). Soft subspace clustering algorithm for streaming data. *Ruan Jian Xue Bao/Journal of Software*, *24*(11), 2610–2627. https://doi.org/10.3724/SP.J.1001.2013.04469

Zhu, E., & Ma, R. (2018). An effective partitional clustering algorithm based on new clustering validity index. *Applied Soft Computing Journal*, *71*, 608–621. https://doi.org/10.1016/j.asoc.2018.07.026

BIOGRAPHIES

Carlos Santos-Mangudo has been associate professor at the Complutense University of Madrid, Carlos III University of Madrid and European University of Madrid, teaching Statistics and Mathematics for more than 15 years. He also has more than 30 years of experience in senior management positions in companies of the technology sector. He has published the book Santos- Mangudo, C. (2017) Técnicas Cluster en Entornos Big Data, ed EAE (in Spanish).

Antonio J. Heras is professor and head of the Department of Financial and Actuarial Economics & Statistics at the Complutense University of Madrid, Spain, where he teaches actuarial and financial mathematics. His publications can be found in https://bibliometria.ucm.es/fichaInvestigadorParams/fe/2456?hidPubType=SCO&anyoInicio=1980&anyoFin=2020