International Journal of Statistics and Applications 2012, 2(4): 24-32
DOI: 10.5923/j.statistics.20120204.01

Clustering Algorithms for Categorical Data: A Monte Carlo Study

Sueli A. Mingoti*, Renata A. Matos

Department of Statistics, Federal University of Minas Gerais, Belo Horizonte, 31270-901, Brazil

Abstract In this paper the clustering algorithms: average linkage, ROCK, *k*-modes, fuzzy *k*-modes and *k*-populations were compared by means of Monte Carlo simulation. Data were simulated from Beta and Uniform distributions considering factors such as clusters overlapping, number of groups, variables and categories. A total of 64 population structures of clusters were simulated considering smaller and higher degree of overlapping, number of clusters, variables and categories. The results showed that overlapping was the factor with major impact in the algorithm's accuracy which decreases as the number of clusters increases. In general, ROCK presented the best performance considering overlapping and non-overlapping cases followed by *k*-modes and fuzzy *k*-Modes. The *k*-populations algorithm showed better accuracy only in cases where there was a small degree of overlapping with performance similar to the average linkage. The superiority of *k*-populations algorithm over *k*-modes and fuzzy *k*-modes presented in previous studies, which were based only in benchmark data, was not confirmed in this simulation study.

Keywords Clustering, Categorical Data, Monte Carlo Simulation

1. Introduction

Clustering algorithms have been used in a variety of fields such as chemistry, engineering, medicine, etc.[2]. Given a data set the main goal is to produce a partition with high internal intra-class similarity and high inter-cluster dissimilarity. The majority of papers available in the literature deals with clustering algorithms for numerical data even though categorical data are very common in practical applications. Two well-known non-hierarchical algorithms to cluster categorical data are *k*-modes[20] and fuzzy *k*-modes[19], which are direct extensions of the *k*-means[26] and fuzzy *c*-means[23], respectively. Some alternative algorithms have been developed to increase the accuracy of *k*-modes and its efficiency to handle larger data sets. Some of these algorithms are direct variations of the *k*-modes or fuzzy *k*-modes (see[4] and[5], for examples). The variations include changing the dissimilarity measure used to compare the objects to the cluster centroids or the cluster centroids themselves by adding more information from the data set in their definition besides of using the hard mode. The *k*-representatives[12] and the *k*-populations ([9],[1]) are examples of fuzzy *k*-modes variations. Among the algorithms based on different concepts to cluster data we find the hierarchical method ROCK[17], which uses the number of links between objects to identify which are neighbors and belong to the same cluster; the entropy-based algorithms LIMBO[13] and COOLCAT[14]; STIRR[16] based on nonlinear dynamical systems from multiple instances of weighted hypergraphs; GoM a parametric procedure based on the assumption that the data follow a multivariate multinomial distribution ([3],[25])); MADE[3] which uses concepts of the rough set theory to handle uncertainty of the partition; CACTUS[18] based on the idea of co-occurrence between attributes and pairs defining a cluster in terms of a cluster´s 2D projections; the subspace algorithms SUBCAD[11] and CLICK[7] whose main goal is to locate clusters in different subspaces of the data set with the purpose of overcome the difficulties found in clustering high-dimensional data; many other algorithms can be found. Hierarchical agglomerative clustering algorithms such as Single, Complete or Average linkage[15] have also been used to cluster categorical data although non-hierarchical methods are more efficient to handle larger data sets.

Many studies are found in the literature comparing the efficiency of clustering algorithms for categorical data. However, most of them uses benchmark data as references to evaluate the performance of the algorithms. The accuracy of the algorithm is then achieved by comparing the reference data with the partition produced by the application of the algorithm to the data. By doing so, the performance of the algorithm is dependent upon the relationship between the reference (previous) partition with the categorical variables used to cluster the data. By using this approach, a reasonable clustering algorithm may be

* Corresponding author:
suelimngt@gmail.com (Sueli A. Mingoti)
Published online at http://journal.sapub.org/statistics

considered of poor performance if the previous partition was created by using criteria with no relationship with the observed categorical variables. On the other hand, any algorithm which works well for these particular benchmark data may be considered very accurate. Therefore, simulation studies are also necessary.

In Kim et al.[9] the *k*-populations algorithm was proposed as an extension of fuzzy *k*-modes. In this method a degree of importance of each category of the variables used to group the data was incorporated in the definition of the clusters centroids. The effectiveness of the algorithm was tested using four benchmark data sets[22] having different composition with respect to the number of clusters (*k*), number of variables (*m*) and number of data points (*n*): the soybean, the zoo database, the credit approval and the hepatitis data. The results showed that *k*-populations algorithm was more accurated than *k*-modes, fuzzy *k*-modes and the hierarchical algorithm using Gower's similarity coefficient[21]. Although the superiority of *k*-populations was outstanding, further research is necessary to confirm its accuracy since the algorithm was tested only in four particular data sets. Taking that into consideration, the purpose of this paper is to evaluate the *k*-populations performance by using Monte Carlo simulation as well as to compare it to four clustering algorithms for categorical data: average linkage, ROCK, *k*-modes and fuzzy *k*-modes. The average linkage is a hierarchical procedure available in the majority of the statistical software and it has been proved to be a reasonable method to cluster numerical data (see[8] and[24]). ROCK is very appealing and its good performance has also been demonstrated in studies by using benchmark data sets. However, as far as we know, no paper has been published in the literature comparing these 5 algorithms using reference data sets or Monte Carlo simulation.

2. Clustering Algorithms for Categorical Data

2.1. *k*-Modes

The *k*-modes algorithm proposed by Huang[20] to cluster data on categorical variables is an extension of *k*-means[23]. It uses a dissimilarity coefficient to measure the proximity of the clusters, modes instead of means, and a frequency-based method to update the modes in each step. Let X_i be a sample observation described by $A_1, A_2, ..., A_m$ categorical variables being the domain of each A_j denoted by $DOM(A_j), j \in \{1,2,...,m\}$, and let $x_{i,j}$ be the observed value of A_j for X_i. The objective function of *k*-modes which has to be minimized, is given by:

$$P(W,Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l} \; d(X_i, Q_l) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,l} \; \delta(x_{i,j}, q_{l,j}) \quad \text{subject to} \quad \sum_{l=1}^{k} w_{i,l} = 1, \; w_{i,l} \in \{0,1\}, \; 1 \le i \le n, \; 1 \le l \le k \tag{1}$$

where $Q_l = [q_{l,1}, q_{l,2}, ..., q_{l,m}]$ is the vector mode of the cluster l, $l = 1,2,...,k$, $Q = \{Q_1, Q_2, ..., Q_k\}$; m is the number of categorical variables and $\delta(.)$ is the dissimilarity measure defined as

$$\delta(x_{i,j}, q_{l,j}) = \begin{cases} 0 & \text{if } x_{i,j} = q_{l,j} \\ 1 & \text{if } x_{i,j} \neq q_{l,j} \end{cases} \tag{2}$$

The basic steps of the algorithm are as follows: (i) k initial modes are selected and each object is compared to them by means of (2) being allocated in the nearest group; (ii) after the allocation of all elements, the k modes are updated and the allocation step is repeated again. The algorithm stops when there is no reallocation among the objects into clusters. As pointed out by Huang[17], *k*-modes may terminate at a local optimum rather than a global optimum solution and the final partition depends on the initial modes and the ordering of the objects of the data set.

2.2. Fuzzy *k*-Modes

The fuzzy *k*-modes algorithm was proposed by Huang and Ng[19] as an extension of the *k*-modes and it is based on the fuzzy *c*-means algorithm used to cluster numerical data[23]. Two new parameters are added in this method: the degree of membership (w) of the observations to each cluster, $w \in [0,1]$, and the fuzzy parameter denoted by $\alpha, \alpha > 1$. The membership degrees are estimated in the execution of the algorithm; the fuzzy parameter is fixed in advance and controls the degree of overlapping expected among clusters. In the literature is very common to set $\alpha = 2$. The degree of chaos in the final partition increases as α goes to infinity.

The objective function of the fuzzy *k*-modes, which has to be minimized to produce the best partition, is given by

$$P(W,Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l}^{\alpha} \; d(X_i, Q_l) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,l}^{\alpha} \; \delta(x_{i,j}, q_{l,j}) \quad \text{subject to} \quad \sum_{l=1}^{k} w_{i,l} = 1, \; w_{i,l} \in [0,1], \; 1 \le i \le n, \; 1 \le l \le n \tag{3}$$

where Q and $\delta(.)$ are defined as in section 2.1.

According to Huang and Ng[16] the function (3) is minimized if and only if $q_{l,j} = c_{t,j}, t \in DOM(A_j)$ and such that

$$\sum_{i: x_{i,j} = c_{t,j}} w_{i,l}^{\alpha} \ge \sum_{i: x_{i,j} \neq c_{t,j}} w_{i,l}^{\alpha} \tag{4}$$

where $w_{i,l}$ is given as (5). As in *k*-modes the vector modes may not be unique.

$$w_{i,l} = \frac{[d(X_i, Q_l)]^{-1/(\alpha-1)}}{\sum_{l=1}^{k} [d(X_i, Q_l)]^{-1/(\alpha-1)}} \tag{5}$$

The basic steps of the fuzzy *k*-modes, for k and α fixed, are as follows: (i) for each object *i* and each *cluster l*, the degree of membership w_{il} is selected and normalized. It can be randomly chosen from the uniform distribution defined in the[0,1] interval for example; (ii) k initial modes are selected by using equation (4); (iii) all objects are compared to the modes using a dissimilarity measure and allocated to the nearest group; (iv) the w_{il} weights are recalculated according to equation (5) and the modes are

updated; (v) the steps (iii) and (iv) are repeated till there is no change in the clusters modes.

Fuzzy *k*-modes has been proved to be an efficient cluster algorithm with the advantage of pointing out the data points which share some similarity with different clusters of the partition and therefore could be misclassified by *k*-modes-type of algorithms.

2.3. *k*-Populations

A modification of fuzzy clustering algorithm, called *k*-populations, was proposed by Kim et. al.[9]. In this new procedure the mode of cluster *l*, called the population of the *lth* cluster centroid, is denoted by V_l and it is defined as $V_l = \{v_{l,1}, v_{l,2}, ..., v_{l,m}\}$, where

$$v_{l,j} = \{(c_{t,j}, \vartheta_{t,j}) \mid t \in DOM(A_j)\}, \forall \; j \in \{1,2,...,m\}, 0 \le \vartheta_{t,j} \le 1;$$

$$0 < \sum_{t=1}^{n_{A_j}} \vartheta_{t,j} < n .$$

Therefore, the composition of the fuzzy centroids is determined by the categories and by the contribution degree of these categories to the specific cluster, $\vartheta_{t,j}$, which is defined as

$$\vartheta_{t,j} = \frac{1}{\lambda_l} \sum_{i=1}^{n} \gamma(x_{i,j}) \qquad (6)$$

where

$$\lambda_l = \sqrt{\sum_{i=1}^{n} w_{i,l}^{2\alpha}} \quad ; \quad \gamma(x_{i,j}) = \begin{cases} w_{i,l}^{\alpha} & , \; if \quad x_{i,j} = c_{t,j} \\ 0 & , \; if \quad x_{i,j} \neq c_{t,j} \end{cases}$$

and $c_{t,j}$ denotes the category *t* of variable A_j. Briefly speaking, for each category *t* of the variable A_j, $\vartheta_{t,j}$ describes the category distribution of the attribute for data belonging to the *lth* cluster. The normalizing factor λ_l, is the length of the vector $w_l^{\alpha} = (w_{1,l}^{\alpha}, w_{2,1}^{\alpha}, ..., w_{n,l}^{\alpha})'$, where $w_{i,l}$ is the degree of membership as defined in section 2.2. According to Kim et. al.[9] the introduction of the $\vartheta_{t,j}$ minimizes the imprecision in the representation of cluster centroids and improve the efficiency of the fuzzy clustering algorithm in finding the optimal partition rather than a local optimal.

As an illustration suppose there is only one categorical variable with two possible categories $\{a,b\}$ and that for a given cluster *l*, there are 3 observations $x_{1,1} = \{a\}, x_{2,1} = \{b\}, x_{3,1} = \{a\}$. Therefore, $x_{1,1} = c_{1,1}$, $x_{2,1} = c_{2,1} \neq c_{1,1}$, $x_{3,1} = c_{1,1}$. Let $\alpha = 2$, $w_{1,l} = 0.70$, $w_{2,l} = 0.80$ and $w_{3,l} = 0.15$. Then, $\lambda_l = \sqrt{(0.7)^4 + (0.8)^4 + (0.15)^4} \approx 0.81$; $\vartheta_{1,1} = \frac{1}{0.81}[(0.7)^2 + (0.15)^2] \approx 0.63$ since for the category $t=1$ of A_1, $\gamma(x_{1,1}) = (0.7)^2$, $\gamma(x_{2,1}) = 0$ and $\gamma(x_{3,1}) = (0.15)^2$. Similarly, it is found $\vartheta_{2,1} \approx 0.79$ and the fuzzy centroid is given by $V_l = \{(a;0.63);(b;0.79)\}$.

In *k*-populations the dissimilarity measure used to compare any observation X_i to any fuzzy centroid V_l is given by

$$d(X_i, V_l) = \sum_{j=1}^{m} \delta(x_{i,j}, v_{l,j}) \qquad (7)$$

where

$$\delta(x_{i,j}, v_{l,j}) = \frac{1}{\eta_l} \sum_{t=1}^{n_{A_j}} \tau(x_{i,j}, c_{t,j}); \tau(x_{i,j}, c_{t,j}) = \begin{cases} \vartheta_{t,j} & if \quad x_{i,j} \neq c_{t,j} \\ 0 & if \quad x_{i,j} \neq c_{t,j} \end{cases}$$

$$\eta_l = \sqrt{\sum_{i=1}^{n_{A_j}} \vartheta_{t,j}^2}$$

being η_l a the normalization factor which corresponds to the length of the vector $\vartheta_j = (\vartheta_{1,j}, \vartheta_{2,j}, ..., \vartheta_{n_{A_j},j})'$.

For *k* and α fixed, the basic steps of the *k*-populations algorithm are given as follows: (i) choose *k* initial seeds (fuzzy centroids) and the degree of contribution $\vartheta_{t,j}$ at random; (ii) obtain the proximity matrix between the observations and the fuzzy centroids according to equation (7); (iii) estimate the degree of membership $w_{i,l}$ according to the equation (5) and update the fuzzy centroids according to equation (6); (iv) repeat steps (ii)-(iii) until there is no reallocation of objects into clusters.

2.4. The Average Linkage

The Average linkage is a well-known algorithm used to cluster categorical and numerical data[15]. It starts with each object as its own cluster (step 1); the similarity between clusters are calculated and the two most similar are merged (step 2); this last step is repeated over and over again till the desirable number of clusters *k* is achieved.

Let C_i and C_l be two clusters with sizes n_i and n_l, respectively. In the Average linkage the dissimilarity measure between these two clusters is defined as:

$$d(C_i, C_l) = \frac{\sum_{X_q \in C_i} \sum_{X_r \in C_l} d(X_q, X_r)}{n_i n_l} \qquad (8)$$

where $d(X_q, X_r)$ is any dissimilarity measure used to compare the X_q and X_r elements. Average linkage does not depend on the notion of initial seeds and therefore, its final partition is not affected by the ordering of the objects in the data set. However, computationally speaking it is less efficient than non-hierarchical algorithms for large data sets.

2.5. ROCK (RObust Clustering using linKs)

Different than other cluster algorithms ROCK, proposed by Guha et. al.[17] is based on the notion of *links* instead of distance or dissimilarity to cluster objects. The number of *links* between two objects represents the number of neighbors they have in common in the dataset. Let *s(.)* be a similarity measure. Two observations X_q and X_r are considered neighbors if $s(X_q, X_r) \ge \theta$, where $\theta \in [0,1]$ is a pre-specified threshold parameter which controls the amount

of similarity required for two observations to be considered neighbors. The $link(X_q,X_r)$ is the number of neighbors the two observations have in common; the higher is its value the more probable is that X_q and X_r belong to the same group. The main goal is to maximize the sum of *links* between the observations which belong to the same group and to minimize the sum of *links* between observations which belong to distinct groups.

In practice the implementation of ROCK is as follows: after an initial computation of the number of *links* between the data objects, the algorithm starts with each cluster being a single object and keeps merging clusters till the specified number of clusters is achieved or no *links* remain between the clusters. In each step of the algorithm the two clusters which maximizes the *goodness measure* given in (9) are merged, where C_i and C_l are the clusters being compared and $link(C_i,C_l)$ is given by (10). The quantity in the denominator of (9) is approximately the expected number of cross *links* between the pair of clusters and $f(\theta)$ is a pre-specified function. According to Guha et. al.[17] empirical work had been shown that the function $f(\theta)=(1-\theta)/(1+\theta)$ works well in practical situations. Under this function when $\theta=1$ the expected number of *links* in cluster C_l is approximately equal to n_{Cl}, i.e., each sample point is a neighbor of itself; when $\theta=0$ the expected number of *links* in cluster C_l is approximately equal to n_{Cl}^3, which corresponds to the situation where all the observations in C_l are neighbors of each other.

$$g(C_i,C_l)=\frac{link(C_i,C_l)}{(n_{C_i}+n_{C_l})^{1+2f(\theta)}-n_{C_i}^{1+2f(\theta)}-n_{C_l}^{1+2f(\theta)}} \quad (9)$$

$$link(C_i,C_l)=\sum_{X_q\in C_i,\ X_r\in C_l} link(X_q,X_r) \quad (10)$$

In Guha et. al.[17] ROCK was implemented using the Jaccard similarity measure. However, an improvement of the computational efficiency of the algorithm is achieved when the weighted similarity measure given in (11) is applied (see[10]). According to (11), for each attribute it is associated a weight proportional to its number of categories. By doing so a pair of observations differing in only one variable whose domain has two categories will have a smaller similarity value than a pair also differing in only one variable, but whose domain has a larger number of categories.

$$s(X_q,X_r)=\frac{|X_q\cap X_r|}{|X_q\cap X_r|+\left[2\sum_{j\notin X_q\cap X_r}\left(|DOM(A_j)|\right)^{-1}\right]} \quad (11)$$

3. Monte Carlo Simulation

A total of 64 population structures of clusters were simulated considering different degrees of overlapping among clusters, different number of clusters and categorical variables (k=2,3,5; m=2,4,15) and different number of categories (t=2,3,5,10). Each generated cluster had 50 observations and the population structures of clusters were simulated to possess features of internal cohesion and external isolation. Additionally, one population structure was simulated without pre-setting the values of k,m,t in advance, but by selecting them randomly from the sets: {2,…,8}, {2,…,15} and {2,…,10}, respectively.

The main objective of the simulation study was to evaluate the changes in the performance of the clustering algorithms from more stable situations (smaller degree of overlapping, number of clusters, variables and categories), to more complex situations (higher degree of overlapping, number of clusters, variables and categories).

3.1. Data Generation

Table 1 presents the simulated population structures grouped by cases. Each case represents a different situation in terms of overlapping. Clusters without overlapping were generated in cases 1 and 2 (degree 1: non-overlapping in the first variable) and case 3 (degree 2: non-overlapping in the first and second variables). Overlapping clusters were generated in cases 4 and 5 (degree 3: overlapping in the first variable), case 6 (degree 4: overlapping in the first and second variables) and case 7 (degree 5: overlapping in the first three variables). The non-overlapping clusters were built as follows: all observations of the same group had the same category on the first variable (cases 1 and 2) or for the first and the second variables (case 3). The categories of the remaining variables (for m>2) were generated randomly according to the Beta and the Uniform distributions. As an illustration consider case 1, k=2, m=2 and t=2, and suppose that {a,b} are the categories of the first variable. The two clusters were built as follows: for the first variable the category {a} was allocated to all the observations of the first cluster and the category {b} for all observations of the second cluster. For each object the category of the second variable was generated randomly for both clusters. Now consider the case 3, $k=5,m=4$, the first variable with categories {a,b,c,d,e} and the second variable with categories {f,g,h,i,j}. Then for the first cluster, the categories {a} and {f} were allocated to all objects for the first and second variables, respectively; for the second cluster the categories {b} and {g} were allocated to all objects; and following this procedure for the cluster five the categories {e} and {j}, respectively, were allocated to all objects. For the other two remaining categorical variables the observations were generated randomly for each cluster.

The overlapping, (cases 4-7), was performed in such way that all the categories of the variables used to build the overlapping had proportionally the same frequency for each cluster. For example, suppose $k=2$, $m=2$, each variable with 2 categories (case 4). Then, the simulation procedure assured that the first category of variable 1 would appear in half of the observations of cluster 1 and the second category in the other half. The observations of the second variable were

randomly generated. The same procedure was used for cluster 2. Now suppose *k=5*, *m=4*, each variable with 5 categories (case 7). Then for each cluster, for the first, second and third categorical variables, the simulation procedure assured that the frequency of each category would be equal to 20% from all observations from the respective cluster. The observations for the other two remaining categorical variables were generated at random. The proportionality was preserved even in situations where the variables used to build the overlapping had different number of categories.

In all situations the generation of the observations for the remaining variables not used in the overlapping procedure, was performed according to the Beta and the Uniform distributions as follows. For each cluster, and each category of the variable A_j, j=1,2,…,m, a random number was selected from the Uniform distribution defined on the[0,1] interval and the Beta distribution with parameters $\alpha = 1$ and $\beta = 0.1$. The vector of the generated numbers from each distribution was normalized to describe the probability of observing each category of A_j. Data for all categories of A_j were then generated randomly according to the correspondent normalized probability vector. The Uniform model represents the situation where all categories of A_j had approximately the same chance of being observed. The Beta distribution was chosen to describe situations where some categories of A_j would have higher chance to appear in the sample than others, which is common in practical applications. As an illustration, Figure 1 presents the results of 4 samples of size 50 of a variable A_j with 3 categories, generated by the described procedure. As it can be seen the Uniform distribution tends to generate classes with similar frequencies different than the Beta distribution which tends to produce classes with larger frequencies than others.

Due to the fact that the values for the variables not used to build the overlapping were generated randomly, there is a chance of possible overlapping among the clusters generated in cases 1-3.

In the case 8 (k=2,m=2,t=15) there was no control of the overlapping degree among clusters since for each cluster, the observations were randomly selected from the Uniform and Beta distributions. Finally, the case 9 represents the situation where there was no control in the simulation for the number of clusters, variables and categories as well as for the degree of overlapping. For this case, the number of clusters was randomly selected from the set {2,3,…,8}, the number of variables from the set {2,3,…,15} and the number of categories from the set {2,3,…,10}. The observations for each cluster were generated from the Beta and the Uniform distributions, as previously described.

From each cluster structure of Table 1 a total of 1000 runs were simulated. The elements of each run were clustered into k groups by using all five clustering algorithms presented in section 2. The resulted partitions were then compared with the true simulated population structure and the performance of the algorithm was evaluated by the average percentage of correct classification taken over 1000 runs (called recovery rate).

Table 1. Simulated structures: number of clusters, variables, categories and degree of overlapping

Cases	k	m	A_1	A_2	A_3	A_4
Degree 1- no-overlapping in the first variable						
1 (8 situations)	2	2	2	2,5,10		
	3	2	3	3,5,10		
	5	2	2	5,10		
2 (13 situations)	2	4	2	2,5	2,3,5	2,3,4,5
	3	4	3	2,3,5	3,5	3,5,10
	5	4	5	2,3,5,8	2,4,5,8	2,4,5,8,10
Degree 2- no-overlapping in the first and second variables						
3 (11 situations)	2	4	2	2	2,3,5	2,3,5,10
	3	4	3	3	2,3,5	2,3,10
	5	4	5	5	5	5
	5	4	5	5	2,3	2,3
	5	4	5	5	5	10
Degree 3 - overlapping in the first variables						
4 (8 situations)	2	2	2	2,5,10		
	3	2	3	3,5,10		
	5	2	5	5,10		
5 (9 situations)	2	4	2	2,5	3,5	3,5,10
	3	4	3	2,5	3,5	3,5,10
	5	4	5	2,5	5	5,10
Degree 4 - overlapping in the first and second variables						
6 (11 situations)	2	4	2	2	2,3,5	2,3,10
	3	4	3	3	2,3,5	2,3,10
	5	4	5	5	2,3,5	2,3,5,10
Degree 5 - overlapping in the first, second and third variables						
7 (3 situations)	5	4	5	2	3	3
	5	4	5	2	5	10
	5	4	5	5	5	5
no control on the overlapping degree						
8	2	15	two categories for each of all 15 variables			
9 *	[2,8]	[2,15]	number of categories randomly selected from[2,10]			

(k,m,t) randomdly selected from the respective sets.

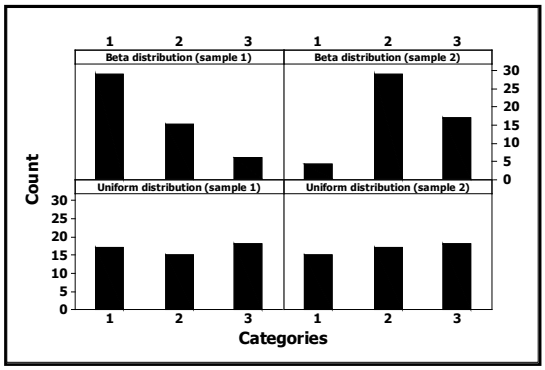

Figure 1. An illustration of samples from Beta and Uniform distributions. Simulated data – sample size 50

For each run of the Monte Carlo simulation the criterion used to interrupt the execution of the average linkage and and ROCK algorithms was the pre-specification of the number of clusters. For each simulated data the initial seeds needed for the execution of *k*-modes, fuzzy *k*-modes and

k-populations algorithms were randomly chosen from the simulated data by using sampling without replacement. The fuzzy parameter α was set as 2 and in the final partition each object was assigned to the cluster whose w_{il} estimate was the largest. The dissimilarity measure given in (11) was used in all algorithms.

4. Results and Discussion

To evaluate the performance of the algorithms the average of recovery rates (ARR) was calculated considering all factors involved in the simulation procedure: degree of cluster overlapping, number of clusters (k) and variables (m). The results are shown in Table 2 for each algorithm according to the degree of overlapping and the distribution used to generate the data. A weighted overall performance mean is also presented and it was calculated taking into account the fact that the number of simulated models were not the same for each combination of k, m and degree of overlapping. It can be seen that for each clustering algorithm and each k, the larger is the degree of overlapping the smaller is the ARR values, as expected. The same is true when the number of clusters increased. However, the performance loss due to the increase of overlapping degree from 1 to 4 does not necessarily increase with the number of clusters neither is similar for all algorithms. In all situations the average recovery rates were larger for data generated from the Beta distribution. The best results were achieved for k=2 and degrees 1 and 2 of overlapping (for Beta data; $84.46 \le ARR \le 100\%$; for Uniform data $71.43 \le ARR \le 100\%$). When k=2 and the degree of overlapping increased to 3 and 4, the ARR values dropped down ranging from 51.07 to 76.46% (Beta data) and 50 to 64.1% (Uniform data). When k increased to 3 the ARR values decreased ranging from 55.47 to 100% for degrees 1 and 2 of overlapping and from 34.65 to 75.2% for degrees 3 and 4, taking into account data from both distributions. The larger impact took place when k=5 since the majority of the ARR values belongs to theinterval[20,65%] except for average linkage and k-populations which presented ARR values between 70 to 80% for non-overlapping data. Considering all the situations, average linkage and k-populations were the most affected by overlapping as it can be seen by the efficiency loss values (Effl) which is defined as the difference between the ARR values from degrees 1 and 4 of overlapping (see Table 2). On the contrary, ROCK was the less affected and the only method resulting ARR values larger than 50% in all situations for Beta data except for the degree 5 of overlapping, although being also the best in this situation. For Uniform data k-modes was the less affected by overlapping.

Table 2. Average recovery rates for all algorithms according to the overlapping degree (m=4)

	Beta distribution						Uniform distribution					
	Degree of overlapping						Degree of overlapping					
	1	2	3	4	5	*Effl*	1	2	3	4	5	*Effl*
Algorithm	*k=2*							*k=2*				
Average	91.83	100.00	58.90	50.00	*	41.83	80.80	100.00	51.91	50.00	*	30.80
ROCK	84.99	96.64	76.46	52.61	*	32.61	79.08	100.00	64.10	50.00	*	29.08
k-Modes	84.46	98.21	66.92	52.71	*	31.75	71.43	93.55	62.27	53.94	*	17.49
Fuzzy *k*-Modes	86.98	98.73	59.73	51.07	*	35.91	75.69	95.86	54.51	51.57	*	24.12
k-Populations	90.17	98.08	60.19	53.40	*	36.77	79.76	95.40	52.44	50.75	*	20.00
	k=3							*k=3*				
Average	75.43	100.00	52.79	34.65	*	40.78	64.33	100.00	45.86	34.66	*	29.67
ROCK	78.22	79.31	75.20	68.56	*	9.66	66.80	69.78	47.23	47.17	*	19.63
k-Modes	68.82	88.62	54.46	44.00	*	24.82	55.78	77.40	49.29	44.62	*	11.16
Fuzzy *k*-Modes	71.84	82.39	49.09	44.65	*	27.19	55.47	69.51	42.10	43.01	*	12.46
k-Populations	83.71	94.48	50.35	39.66	*	44.05	68.00	91.63	41.63	36.61	*	31.39
	k=5							*k=5*				
Average	59.52	99.62	43.98	20.02	26.70	39.00	46.36	100.00	34.49	20.00	26.83	26.36
ROCK	64.79	56.75	67.06	61.45	43.63	3.34	51.12	39.44	38.09	35.23	34.61	15.89
k-Modes	54.56	72.89	42.29	34.69	34.22	12.27	44.21	61.21	36.41	33.04	33.04	11.17
Fuzzy *k*-Modes	54.27	64.86	36.64	32.72	34.63	21.55	41.20	56.47	29.41	30.26	30.26	10.94
k-Populations	67.46	87.29	44.00	25.71	29.96	41.75	51.99	86.98	38.29	23.86	23.86	28.13
	Average among all cases							*Average among all cases*				
Average	74.36	99.86	51.89	34.91	26.70	30.45	62.49	100.00	44.08	34.91	26.83	27.58
ROCK	75.14	77.41	72.91	60.18	43.63	14.96	64.55	69.74	49.81	43.86	34.61	20.69
k-Modes	68.15	86.39	54.56	43.78	34.22	24.37	56.14	77.38	49.32	43.80	33.04	12.34
Fuzzy *k*-Modes	69.74	81.96	48.49	42.65	34.63	27.09	56.20	74.35	42.00	41.49	30.26	14.71
k-Populations	79.45	93.17	51.51	39.58	29.96	39.87	62.84	91.31	44.12	37.12	23.86	25.72

Degrees: 1 - no overlapping in the first variable; 2- no overlapping in the first and second variables; 3 - overlapping in first variable; 4- overlapping in first and second variables; 5-overlapping in the first 3 variables. *Effl*: efficiency loss.

The 63 simulated population structures (cases 1 to 7) were grouped into the "no-overlapping cases" (i.e. degrees 1 and 2 of overlapping) and "overlapping cases" (degrees 3 and 4 of overlapping) and the average recovery rates were taken over all the simulated structures belonging to each respective group. The main results are shown in Tables 3 and 5 according to the number of clusters and variables for both types of generated data. The overall averages taken among all 63 simulated population structures are also shown. Table 4 presents the difference between the average recovery rates for all clustering algorithms calculated using the overall means shown in Table 3. From these results (Tables 3 and 4), it is seen that in average, for the "no-overlapping" group, the average linkage and *k*-populations were the best algorithms (overall means over 85%) compared to ROCK, *k*-modes and fuzzy *k*-modes (overall means between 75 and 78%). It is important to point out that the best average recovery rates were found for *k*=2 (overall means over 90%). For the "overlapping group" and data from a Beta distribution, ROCK was the best with *ARR* values larger than any other clustering method (overall mean 59.97%). The average linkage was the less accurate (overall means of 41.11% and 38.03% for Beta and Uniform data, respectively) followed by *k*-populations (overall means of 43.73% and 40.06% for Beta and Uniform data, respectively). Under overlapping the difference between the average recovery rates were smaller for data coming from the Uniform distribution being *k*-modes and ROCK the best algorithms although the overall means were very small (46.86% for k-modes and 44.57% for ROCK).

The results from Table 5 show that the *ARR* values were similar for *m*=2 and *m*=4 categorical variables. Comparing the *ARR* values from Tables 3 and 5 it is easily seen that the increase of the number of categorical variables had less impact in the accuracy of the algorithm than the increase of the number of clusters.

The efficiency loss measured by the difference between the "no-overlapping" and "overlapping" average recovery rates (see Tables 3 and 5) corroborate to the fact that average linkage and *k*-populations were the most affected by overlapping (average efficiency loss equal 45.82 and 41.94%, respectively) as long as ROCK (for Beta data) and *k*-modes (for Uniform data) were the less affected (average efficiency loss equal 18.25 and 20.20%, respectively).

Table 6 presents the results for cases 8 and 9, the two situations simulated with no control on the overlapping degree. For case 8 the average recovery rates were higher than 77% for data from the Beta distribution and between 55.19 to 68,14% for data from the Uniform distribution, results very reasonable for a situation whose number of clusters and categories were small (2) and the number of categorical variables was large (15). On the other hand, when the parameters (*k*,*m*,*t*) were chosen randomly (case 9), the average recovery rates were smaller (under 54.85%), specially for data from the Uniform distribution. The only exception was ROCK with an average recovery rate equal 72.83% (for Beta data).

5. Final Remarks

Table 3. Average recovery rate for all algorithms according to the overlapping degree

Beta distribution					Uniform distribution			
no-overlapping					no-overlapping			
Algorithm	*k*=2	*k*=3	*k*=5	overall mean	*k*=2	*k*=3	*k*=5	overall mean
Average	95.60	87.43	77.81	86.93	91.06	82,41	71.46	81.62
ROCK	91.19	80.23	63.43	78.22	89.75	73.28	50.76	71.20
k-Modes	90.12	75.67	62.03	75.95	82.68	65.63	52.75	67.06
Fuzzy	92.07	75.13	58.43	75.21	86.21	62.32	48.54	65.80
k-Populations	94.32	87.22	75.62	85.67	88.66	79.20	64.71	77.47
	Overlapping				Overlapping			
Algorithm	*k*=2	*k*=3	*k*=5	overall mean	*k*=2	*k*=3	*k*=5	overall mean
Average	54.12	42.60	29.14	41.11	50.86	39.40	26,32	38.03
ROCK	62.27	64.79	54.42	59.97	56.37	45.06	34.38	44.57
k-Modes	60.08	50.78	37.60	48.68	59.26	48.80	35.08	46.86
Fuzzy	54.80	46.47	34.18	44.40	52.97	42.74	30.92	41.46
k-Populations	56.33	44.76	32.47	43.73	52.15	39.97	30.06	40.06
	difference (*Effl*)				difference (*Effl*)			
Algorithm	*k*=2	*k*=3	*k*=5	overall mean	*k*=2	*k*=3	*k*=5	overall mean
Average	41.48	44.83	48.67	45.82	40.20	43.01	45.14	43.59
ROCK	28.92	15.44	9.01	18.25	33.38	28.22	16.38	26.63
k-Modes	30.04	24.89	24.43	27.27	23.42	16.83	17.67	20.20
Fuzzy	37.27	28.66	24.25	30.81	33.24	19.58	17.62	24.34
k-Populations	37.99	42.46	43.17	41.94	36.51	39.23	34.65	37.41

* difference = (*no-overlapping ARR – overlapping ARR*) is a measure of efficiency loss (*Effl*)

Table 4. Difference between the average recovery rates for all clustering algorithms

Beta distribution					Uniform distribution			
no-overlapping					no-overlapping			
Algorithm	Average	Rock	*k*-modes	Fuzzy	Average	Rock	k-modes	Fuzzy
Average	-				-			
ROCK	8.71	-			10.42	-		
k-Modes	10.98	2.27	-		14.56	4.14	-	
Fuzzy	11.72	3.01	0.74	-	15.82	5.40	1.26	-
k-Population	1.26	-7.75	-9.72	-10.47	4.15	-6.27	-10.41	-11.67
	overlapping				Overlapping			
Algorithm	Average	Rock	*k*-modes	Fuzzy	Average	Rock	*k*-modes	Fuzzy
Average	-				-			
ROCK	-18.86	-			-6.54	-		
k-Modes	-7.57	11.29	-		-8.83	-2.29	-	
Fuzzy	-3.29	15.57	4.28	-	-3.43	3.11	5.4	-
k-Population	-2.62	16..2	4.95	0.67	-2.03	4.51	6.8	1.4

* The differences were calculated using the overall means shown in Table 3. All algorithms were compared among themselves.

In this paper the performance of five clustering algorithms for categorical data was evaluated by means of Monte Carlo simulation being Beta and Uniform distributions used to

generate the artificial data. The results showed that overlapping is the factor with major impact on the accuracy of all the algorithms. Although the increase in the number of categorical variables affects the performance of the algorithms the impact is smaller than the respective impact due to the increase of the number of clusters. The average recovery rates were larger for data whose categories had difference frequency of occurrence (Beta data) compared to the data whose categories had approximately the same frequency (Uniform data).

Table 5. Average recovery rate according to overlapping degree and number of categorical variable

Beta distribution				Uniform distribution		
no-overlapping				no-overlapping		
Algorithm	m=2	m=4	overall mean	m=2	m=4	overall mean
Average	89.59	86.05	86.93	87.44	79.68	81.62
ROCK	84.35	76.18	78.22	84.02	66.92	71.20
k-Modes	74.27	76.51	75.95	70.62	65.87	67.06
Fuzzy *k*-Modes	74.83	75.34	75.21	69.66	64.52	65.80
k-Populations	85.46	85.74	85.67	82.22	75.88	77.47
overlapping				overlapping		
Algorithm	m=2	m=4	overall mean	m=2	m=4	overall mean
Average	42.89	40.48	41.11	39.73	37.44	38.03
ROCK	51.24	63.00	59.97	43.41	44.98	44.57
k-Modes	54.21	46.75	48.68	53.17	44.67	46.86
Fuzzy *k*-Modes	45.89	43.89	44.40	43.58	40.73	41.46
k-Populations	45.85	42.99	43.73	43.51	38.87	40.06
difference (*Effl*)				difference (*Effl*)		
Algorithm	m=2	m=4	overall mean	m=2	m=4	overall mean
Average	46.70	45.57	45.82	47.71	42.24	43.59
ROCK	33.11	13.18	18.25	40.61	21.94	26.63
k-Modes	20.06	29.76	27.27	17.45	21.20	20.20
Fuzzy *k*-Modes	28.94	31.45	30.81	26.08	23.79	24.34
k-Populations	39.61	42.75	41.94	38.71	37.01	37.41

* difference = (*no-overlapping ARR – overlapping ARR*) is a measure of efficiency loss (*Effl*)

Table 6. Average recovery rate – no control on the overlapping degree

	Beta distribution		Uniform	
Algorithm	Case 8	Case 9	Case 8	Case 9
Average	88.74	50.43	66.05	37.42
ROCK	89.34	72.83	68.14	41.87
k-Modes	81.18	46.83	58.37	36.65
Fuzzy *k*-Modes	77.75	42.01	55.19	31.30
k-Populations	90.15	54.85	55.22	48.08

* case 9: *(k,m,t)* were selected randomly

The superiority of *k*-populations over to *k*-modes and fuzzy *k*-modes presented by Kim et al.[9] was not confirmed by the results shown in this paper. In fact for Beta and Uniform data, the *k*-populations algorithm presented better accuracy only in cases where there was a small degree of overlapping (degree 1 and 2) having performance similar to the average linkage algorithm which was the best for these type of simulated situations. However, similar to the average linkage, the performance of *k*-populations decreased more than 40% in average when additional overlapping was introduced into the data (degrees 3 and 4 of overlapping) and its accuracy was smaller than ROCK, *k*-modes and fuzzy *k*-modes in these cases. Therefore, considering that the main goal is to use a clustering algorithm which in average has better accuracy for no-overlapping as well as for overlapping data, the simulation study has indicated that *k*-modes, fuzzy *k*-modes and ROCK should be preferred to *k*-populations, particularly ROCK which in general, was the algorithm less affected by overlapping and presenting average recovery rates larger or similar than any other algorithm discussed in this paper.

Finally, this paper shows that the evaluation of the accuracy of the clustering algorithms for categorical data should not be restricted to the studies which only use benchmark data to estimate their performance. It is also important to conduct studies using Monte Carlo simulation since by knowing in advance the characteristics of the simulated dataset it is possible to analyse better the impact of factors such as overlapping, number of clusters, categorical variables and categories in the final solution. The drawback of using only real data sets to compare clustering algorithms is the fact that not necessarily the pre-specified partition is consistent with the pattern of the data in terms of the variables used to perform the clustering.

ACKNOWLEDGEMENTS

The authors are very thankful to CNPq and CAPES Institutions.

REFERENCES

[1] Ali Seman, Zainab A. Bakar, Mohamed N. Isa, "Evaluation of k-Modes-Type Algorithms for Clustering Y-Short Tandem Repeats Data", Trends in Bioinformatics, vol. 5, n. 2, pp. 47-52, 2012.

[2] Caroline L. Wilson, Mathematical Modeling, Clustering Algorithms and Applications, Nova Science Publishers, USA, 2011.

[3] Tutut Herawan, Rozaida Ghazali, Iwan T. Yanto, Mustafa M. Deris, "Rough Set Approach for Categorical Data Clustering", International Journal of Database Theory and Application, vol. 3, n. 1, pp.33-51, 2010.

[4] W. Jiacai, Gu. Ruijunm, "An Extended Fuzzy k-Means Algorithm for Clustering Categorical Valued Data", in AICI´10 Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence, pp.504-507, 2010.

[5] Michael K.Ng, Liping Jing, "A New Fuzzy k-Modes Clustering Algorithm for Categorical Data", International Journal of Granular Computing, Rough Sets and Intelligent

Systems, vol. 1, n. 1, pp. 105 – 119, 2009.

[6] Miin S.Yang, Yu H. Chiang, Chiu C. Chen, Chien Y. Lai, "A Fuzzy k-Partitions Model for Categorical Data and its Comparison to the GoM Model", Fuzzy Sets and Systems, vol. 159, n. 4, pp. 390– 405, 2008.

[7] Mohammed Zaki, Markus Peters, Ira Assent, Thomas Seidl, "CLICK: An Effective Algorithm for Mining Subspace Clusters in Categorical Datasets", Data and Knowledge Engineering, vol. 60, n. 1, pp. 51-70, 2007.

[8] Sueli A. Mingoti and Joab O. Lima, "Comparing SOM Neural Network with Fuzzy c-Means, k-Means and Traditional Hierarchical Clustering Algorithms", European Journal of Operational Research, vol. 174, n. 3, pp. 1742–1759, 2006.

[9] Dae W. Kim, Ki Young Lee, Doheon Lee, Kwang H. Lee, "A k-populations Algorithm for Clustering Categorical Data", Pattern Recognition, vol. 38, n. 7, pp. 1131–1134, 2005.

[10] Mala Dutta, Anjana K. Mahanta, Arun K. Pujari, "QROCK: A Quick Version of the ROCK Algorithm for Clustering of Categorical Data", Pattern Recognition, vol. 26, n. 15 , pp. 2364–2373, 2005.

[11] Guojun Gan, Jianhong Wu, "Subspace Clustering for High Dimensional CategoricalData", ACM SIGKDD Explorations Newsletters, 6, pp. 87-94, 2004.

[12] Ohn M. San, Van H. Huynh, Yoshiteru Nakamori, "An Alternative Extension of the K-Means algorithm for Clustering Categorical Data", Journal of Applied Mathematics Computing Science, vol. 14, n. 2, pp. 241-245, 2004.

[13] P. Andritsos,"Scalable Clustering of Categorical Data and Applications". Doctor thesis. University of Toronto, Canada, 2004.

[14] Daniel Barbará, Yi Li and Julia Couto, "COOLCAT: an entropy-based algorithm for categorical clustering", in Proceedings of the 11th Symposium ACM Conference in Information and Knowledge Management CIKM (02), pp. 582-589, 2002.

[15] Brian S. Everitt, Cluster Analysis, John Wiley & Sons Inc., USA, 2001.

[16] David Gibson, Jon Kleinberg, Prabhakar Raghavan, "Clustering categorical data: an approach based on dynamical systems", The VLDB Journal, vol. 8, n. 3-4, pp. 311-322, 2000.

[17] Sudipto Guha, Rajeev Rastogi, R., Kyuseok Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes", Information Systems, vol. 25, n. 5, pp. 345–366, 2000.

[18] Venkatesh Ganti, Johannes Gehrke, Raghu Ramakrishnan, "CACTUS: Clustering Categorical Data using Summaries", in KDD´ 99 Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 73-83, 1999.

[19] Zhexue Huang, Michael K. Ng, "A Fuzzy k-Modes Algorithm forClustering Categorical Data", IEEE Transactions on Fuzzy Systems, vol. 7, n. 4, pp. 446–452, 1999.

[20] Zhexue Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery, vol. 2, n. 2, pp. 283–304, 1998.

[21] K. Chidananda Gowda , Edwin Diday, "Symbolic clustering using a new dissimilarity measure", Pattern Recognition, vol. 24, no. 6, pp. 567–578, 1991.

[22] Catherine L. Blake, Christopher J. Merz, "UCI Repository of machine learning databases", School of Information and Computer Science, Irvine, CA, 1989.

[23] James C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, USA, 1981.

[24] Glenn W. Milligan, "An examination of the effect of six types of error perturbation on fifteen clustering algorithms", Psychometrika, vol 45, n. 3, pp. 325–342, 1980.

[25] Max A. Woodbury, Jonathan Clive, "Clinical Pure Types as Fuzzy Partition", Journal of Cybernet, vol. 4, n. 3, pp. 111–121, 1974.

[26] James B. MacQueen, "Some methods for classification and analysis of multivariate observations", in Proceedings of 5-th Symposium of Mathematical Statistics and Probability, pp. 281-297, 1967.