Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 174 (2006) 1742–1759
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Stochastics and Statistics

Comparing SOM neural network with Fuzzy *c*-means, *K*-means and traditional hierarchical clustering algorithms

Sueli A. Mingoti *, Joab O. Lima

Departamento de Estatística, Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Av. Antonio Carlos 6627, Belo Horizonte, 31270-901 Minas Gerais, Brazil

Received 5 January 2004; accepted 15 March 2005
Available online 27 June 2005

Abstract

In this paper we present a comparison among some nonhierarchical and hierarchical clustering algorithms including SOM (Self-Organization Map) neural network and Fuzzy *c*-means methods. Data were simulated considering correlated and uncorrelated variables, nonoverlapping and overlapping clusters with and without outliers. A total of 2530 data sets were simulated. The results showed that Fuzzy *c*-means had a very good performance in all cases being very stable even in the presence of outliers and overlapping. All other clustering algorithms were very affected by the amount of overlapping and outliers. SOM neural network did not perform well in almost all cases being very affected by the number of variables and clusters. The traditional hierarchical clustering and *K*-means methods presented similar performance.

Keywords: Multivariate statistics; Hierarchical clustering; SOM neural network; Fuzzy *c*-means; *K*-means

1. Introduction

Cluster analysis have been used in a variety of fields. Some examples appear in data mining where the organization of larger data sets makes the statistical analysis easier and more efficient; in the identification of different consumer's profiles in marketing surveys, in helping the researchers to build up the strata in stratified sampling or even in the identification of the variables that are more important to describe a phenomenon. However, it is well known that the accuracy of the final partition depends upon the method used to cluster the objects. Because of that, studies have been conducted to evaluate the performance of the clustering algorithms (Milligan and Cooper, 1980; Gower, 1967). Most of them are related to the

* Corresponding author. Tel.: +55 31 3499 5948; fax: +55 31 3499 5924.
E-mail address: sueli@est.ufmg.br (S.A. Mingoti).

0377-2217/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2005.03.039

classical hierarchical techniques (Gordon, 1987) and the nonhierarchical *K*-means method (Everitt, 2001). Very few papers examine the performance of the Fuzzy *c*-means (Bezdek et al., 1999) and the artificial neural networks methods for clustering (Kohonen, 1995; Kiang, 2001). Usually, the comparison of the algorithms involves a simulation of several multidimensional structures, with nonoverlapping and overlapping clusters. The clustering algorithms are then used to cluster the data and the final partition is compared with the true simulated structure. Criteria as the percentage of observations that are correctly classified and internal dispersion of the groups in the partition are in general used to access the accuracy of the clustering algorithm. In general the population structure is simulated from a multivariate normal distribution although the application of clustering methodology does not require the assumption of normality (Johnson and Wichern, 2002).

Milligan and Cooper (1980) presented an algorithm to simulate multidimensional clusters partitions and a comparison among some hierarchical clustering procedures. The data were simulated according to a three-factor design: the first factor controls the number of clusters $k = 2, 3, 4, 5$; the second the number of variables $p = 4, 6, 8$ and the third the pattern for the distribution of points to the clusters. Three patterns were considered: uniform distribution of points among all clusters, 10% of the observations concentrated in only one cluster of the partition and 60% of the observations in only one cluster of the partition. The algorithm used to generate the data was also discussed in Milligan (1985). Clusters were simulated in such way that overlap of cluster boundaries was not permitted in the first dimension of the variable space but permitted in the other $(p - 1)$ dimensions. The degree of overlapping was related to the clusters variances. All p variables were considered independent (spherical clusters) and simulated according to a normal distribution. A total of 108 error free data sets were generated, 3 for each of the 36 cells of the three-factor design. Each data set contained a total of 50 points. Clusters were also simulated with the following error perturbation: (i) inclusion of outliers, (ii) inclusion of random error in the distance matrix, (iii) addition of irrelevant variables, (iv) computation of distances with a noneuclidean index, (v) standardization of the variables. A total of 15 algorithms were evaluated, 14 hierarchical and the *K*-means method. In general the paper showed that the *K*-means method had a good performance especially when the initial seeds were generated from one of the hierarchical methods. In the situation of error free data all the clustering algorithms had good performance (average recovery rate over 90%). However, when the data were perturbed the algorithms were influenced differently according to the type of perturbation. The Ward and Complete linkage methods were very affected by the inclusion of outliers but the single and the average linkages, the centroid and *K*-means methods were very robust against this type of error. The single linkage was very affected by the inclusion of random error in the distance matrix. All methods were affected by the inclusion of irrelevant variables. Standardization and the use of a noneuclidean distance index had very few perturbation in all the methods (average recovery rate over 90%). In Balakrishnan et al. (1994) SOM neural network (Kohonen, 1989) was compared to the nonhierarchical *K*-means method by using a design and a simulation procedure similar to Milligan's (1980, 1985). The data were simulated according to a normal distribution with no correlation among the variables and considering 3 factors: numbers of clusters $k = 2, 3, 4, 5$, number of variables $p = 4, 6, 8$ and perturbance in the distance matrix (error structure) measured in 3 levels: free, low and high. A total of 108 data sets were generated in the simulation process. It was shown that in general SOM did not have a good performance. Considering the error factor the best and the worst performance were observed for the error free structure (89.34%) and for the high error structure (86.44%) respectively. For the number of clusters the best average recovery rate was observed for $k = 2$ (97.04%) and the worst for $k = 5$ (74.82%). For the number of variables the best result was for $p = 8$ (88.78%) and the worst for $p = 6$ (86.22%). The overall average recovery rate was 98.77% for *K*-means and 87.79% for SOM. Considering the 3 factors (error, number of clusters and number of variables) the average recovery rate

ranged from 100% to 96.22% for *K*-means and from 97.04% to 74.82% for SOM. Another similar study was conducted by Balakrishnan et al. (1996) comparing the *K*-means algorithm with the Frequency-Sensitive Competitive Learning (FSCL) neural net (Krishnamurthy et al., 1990). The *K*-means performed better in all simulated situations with overall recovery rate equals to 98.67% against 90.81% for FSCL. The FSCL was affected by the increased in the number of clusters (recovery rate drop from 95.04 for $k = 2$ to 84.74 to $k = 5$ clusters), by the number of variables (recovery rate of 87.17% for $p = 2$ variables and 93.72% for $p = 4$) and by the error structure (recovery rate of 92.72% for error free to 86.22% for high error structure). In Mangiameli et al. (1996) agglomerative hierarchical clustering procedures were also compared with SOM artificial network. Seven clustering algorithms were compared including the single, complete, average, centroid and Ward methods. Data were generated according to Millligan's algorithm (1980, 1985) considering $k = 2, 3, 4, 5$ clusters, $p = 4, 6, 8$ variables, and three different intracluster dispersion degrees called high, medium and low. The choice of the dispersion degree determines the rate of cluster overlap. The addition of irrelevant variables and outliers were also investigated. The normal distribution with zero correlation was used to generate the observations for each cluster in the population. A total of 252 data sets were generated, each cluster with 50 observations. For low intracluster degree of dispersion the analysis presented in Mangiameli et al. (1996) showed that all the algorithms had a good average recovery rate (over 90%) except for the single linkage (76.9%). For medium degree of dispersion SOM still had a good average recovery rate (98%) but all the others methods decreased in accuracy. The Ward was the best among the classical with a recovery average rate of 86.2%. The majority of the other algorithms had the average recovery rate dropped down to less than 45%. For high intracluster dispersion degree the overall percentage average of correct classification of SOM was 82.5% higher than the Ward's method (50.4%) which was the best among the hierarchical procedures. Single linkage as well the centroid and average linkages performed very bad in high and medium intraclass clusters dispersion. When outliers and irrelevant variables were added to the data, SOM average recovery rate decreased to about 80% and it was similar to Ward's method. The others hierarchical methods were very affected most of them, presenting average recovery rates under 40% when outliers were included in the data. In general the results showed that the average recovery rate decreases as the number of clusters and the degree of intracluster dispersion increase. No results were shown in the paper about the effect of the number of variables in the accuracy of clustering algorithm. In Schreer et al. (1998) a comparison of *K*-means with Fuzzy *c*-means, SOM and ART artificial neural networks was presented using artificial and real data. The study involved three types of situation. In the first, the data were generated according to a three-factor design: the number of clusters $k = 2, 3, 4, 5$, the number of variables $p = 4, 6, 8, 10$, and three degrees of overlapping called high, medium and low. For each cluster the variables were independent and simulated according to a normal distribution. Each data set had 100 observations and equal number of points per cluster. A total of 144 data sets were generated, 3 per level of the design. The second type of data consisted of $k = 5$ shapes, described by $p = 10$ depths, commonly observed as dive profiles for the species treated in Schreer et al. (1998). According to the authors the data were generated from a multivariate normal distribution with autocorrelated depths similar to those observed from real data. Three data sets with 1000 observations each, were generated. The pattern of the distribution of points per cluster was: 37%, 20%, 13%, 13% and 17%. The authors were not very specific about the algorithm used to generate the artificial data. The third type of data consisted of subsamples from a real diving data from Adélie penguins, southern elephant seals and Weddell seals. Three data sets, each containing a subsample of 3000 dives, were taken from the diving data recorded for each of the different species. For the artificial data of the first type the results indicated that SOM network had good performance equivalent to *K*-means and Fuzzy *c*-means methods (average recovery rate over 90%). The Fuzzy Art

(Carpenter et al., 1991) did not performed well (recovery rate between 80% and 90%). In general, for all methods, the average recovery rate decreased as the number of clusters and the degree of overlapping increased. However, the results were still good for high degree of intracluster dispersion (average recovery rate over 90%) except for Fuzzy Art. The average recovery rate increased as the number of variables increased. For the second type of artificial data the results were very similar to those obtained for data of first type. For the real data the methods had similar performance but with more dispersion than the artificial data. The *K*-means method created clusters more logical when compared to the actual dive profiles and it was considered by the authors as "the most suited for grouping multivariate diving data". The SOM and Fuzzy *c*-means performed similar as *K*-means but had poorer boundaries separating the clusters because the observations were classified in such way that some clusters were very close together.

All papers presented very interesting results. However, (i) none of them compared the hierarchical with the nonhierarchical algorithms simultaneously; (ii) the number of data sets for each cell in the three-factor design was small: only three replicates for each population structure (cell); (iii) the number of objects in each simulated data set was small: only 50 points in Milligan and Cooper (1980) and Balakrishnan et al. (1994), 100 points in Schreer et al. (1998) and from 100 to 250 in Mangiameli et al. (1996); (iv) the simulated variables were independent (spherical clusters) and the only paper that simulated correlated variables, did it for a very specific situation (Schreer et al., 1998).

In this article we will extend the results comparing the traditional hierarchical clustering procedures with the nonhierarchical *K*-means, Fuzzy *c*-means and SOM artificial neural networks. The simulation involved many different clusters structures (spherical and nonspherical clusters with and without overlapping and outliers), data sets with a larger number of points (500 each) and larger number of variables and clusters. It goes much beyond the studies previously published. It will be shown that in general Fuzzy *c*-means and *K*-means methods have a good performance and SOM did not performed very well. In some extent our study agrees with the results obtained by Milligan and Cooper's (1980) and Balakrishnan et al. (1994) as far as the neural network SOM is concerned.

2. Clustering methods: A brief explanation

2.1. The agglomerative hierarchical clustering

The agglomerative hierarchical algorithms are largely used as an explanatory statistical technique to determine the number of clusters of data sets (Anderberg, 1972). They basically work in the following way: in the first stage each of the n objects to be clustered is considered as a unique cluster. The objects are then, compared among themselves by using a measure of distance such as Euclidean, for example. The two clusters with smaller distance are joined. The same procedure is repeated over and over again until the desirable number of clusters is achieved. Only two clusters can be joined in each stage and they cannot be separated after they are joined. A linkage method is used to compare the clusters in each stage and to decide which of them should be combined. Some very common procedures are: Single, Complete and Average linkages, which can be used for quantitative or qualitative variables, Centroid and Ward's methods which are appropriate only for quantitative variables (Johnson and Wichern, 2002). A graphical called dendogram is available showing the clustering results of each stage.

2.2. The nonhierarchical clustering

Contrary to the hierarchical procedures, to perform the nonhierarchical clustering algorithm, the desired number of clusters k has to be pre-defined. The purpose then is to cluster the n objects into k clusters in such way that the members of the same cluster are similar in the p characteristics used to cluster the data and the members of different clusters are heterogeneous. Next we will present the three nonhierarchical procedures which will be discussed in this paper.

2.2.1. K-means

The *K*-means clustering (Johnson and Wichern, 2002) method is probably the most well known. The algorithm starts with k initial seeds of clustering, one for each cluster. All the n objects are then compared with each seed by means of the Euclidean distance and assigned to the closest cluster seed. The procedure is then repeated over and over again. In each stage the seed of each cluster is recalculated by using the average vector of the objects assigned to the cluster. The algorithm stops when the changes in the cluster seeds from one stage to the next are close to zero or smaller than a pre-specified value. Every object is assigned to only one cluster.

The accuracy of the *K*-means procedure is very dependent upon the choice of the initial seeds (Milligan and Cooper, 1980). To obtain better performance the initial seeds should be very different among themselves. One efficient strategy to improve the *K*-means performance is to use, for example, the Ward's procedure first to divide the n objects into k groups and then use the average vector of each of the k groups as the initial seeds to start the *K*-means. As all the agglomerative clustering procedures, this method is available in a majority of statistical software.

2.2.2. Fuzzy c-means

As the *K*-means algorithm the desired number of clusters c has to be pre-defined and c initial seeds of clustering are required to perform the Fuzzy *c*-means (Bezdek, 1981; Roubens, 1982). The seeds are modified in each stage of the algorithm and for each object a degree of membership to each of the c clusters is estimated. A metric is also used to compare every object to the cluster seed but the comparison is made using a weighted average that takes into account the degree of membership of the object to each cluster. In the end of the algorithm, a list of the estimated degree of membership of the object to each of the c clusters is printed. The object can be assigned to the cluster for which the degree of membership is higher. Contrary to the *K*-means method the Fuzzy *c*-means is more flexible because it shows those objects that have some interface with more than one cluster in the partition as can be seen in the illustration of Fig. 1. These objects usually deserve further investigation in order to find out the reasons that contributed for them to be in the interface. Mathematically speaking, Fuzzy *c*-means minimizes the objective function defined as

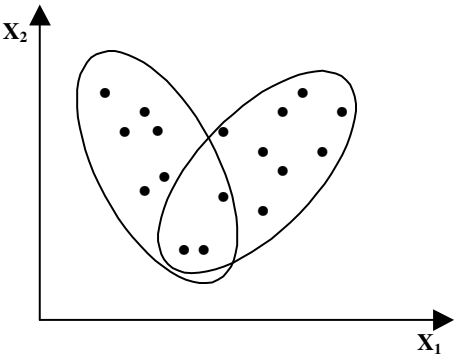

Fig. 1. Illustration of fuzzy clustering.

$$J = \sum_{i=1}^{n} \sum_{l=1}^{c} (w_{il})^{\lambda} d_{il}^2$$

restricted to the condition $\sum_{l=1}^{c} w_{il} = 1, i = 1, 2, \ldots, n$, where w_{il} is the degree of membership of object i to the cluster l, $\lambda > 1$ is the fuzzy exponent that determines the degree of fuzziness of the final partition, or in other words the degree of overlap between groups, d_{il}^2 is the squared distance between the vector of observations of object i to the vector representing the centroid (prototype) of cluster l and n is the number of sample observations. The solution with highest degree of fuzziness is related to λ approaching to infinity. Some additional references in Fuzzy *c*-means are Hathaway and Bezdek (2002), Bezdek et al. (1999), Susanto et al. (1999) and Zhang and Chen (2003) among others.

2.2.3. Artificial neural network SOM (Kohonen)

The first model in artificial neural netwroks (ANN) dated from the 1940s (McCulloch and Pitts, 1943) which was explored by Hebb (1949) who proposed a model based on the adjustment of weights in inputs neurons. Rosenblatt (1958) introduced the Perceptron model. But only in the 1980s the ANN started been more used. In clustering problems, the ANN clusters observations in two main stages. In the first the learning rule is

used to train the network for a specific data set. This is called a training or learning stage. In the second the observations are classified, which is called a recall stage. Briefly speaking the ANN work into layers. The input layer contains the nodes through which data are input. The output layer generated the output interpreted by the user. Between these two layers there can be more layers called hidden layers. The output of each layer is an input of the next layer until the signal reaches the output layers as shown in Fig. 2. One of the more important ANN is the Self-Organization Map (SOM) proposed by Kohonen. In this network there is an input layer and the Kohonen layer which is usually designed as two-dimensional arrangement of neurons that maps n-dimensional input to two dimensional. It is basically a competitive network with the characteristic of self-organization providing a topology-preserving mapping from the input space to the clusters (Kohonen, 1989, 1995; Gallant, 1993). Mathematically speaking, let $x = (x_1 x_2 \dots x_p)'$ be the input vector (training case), $w_l = (w_{l1} w_{l2} \dots w_{lp})'$ the weight vector associated with the node l where w_{lj} indicates the weight assigned to input x_j to the node l, where k is the number of nodes (cluster seeds) and p is the number of variables. Each object of the training data set is presented to the network in some random order. Kohonen's learning law is an online algorithm that finds the node closest to each training case and moves that "winning" node closer to the training case. The node is moved some proportion of the distance between it and the training case. The proportion is specified by the learning rate. For each object i in the training data set, the distance d_i between the weight vector and the input signal is computed. Then the competition starts and the node with the smallest d_i is the winner. The weights of the winner node are then updated using some learning rule. The weights of the nonwinner nodes are not changed. Usually, the Euclidean distance is used to compare each node with each object although any other metric could be chosen. The Euclidean distance between an object with observed vector $x = (x_1 x_2 \dots x_p)'$ and the weight vector $w_l = (w_{l1} w_{l2} \dots w_{lp})'$ is given by

$$d(x, w_l) = \left[\sum_{j=1}^{p} (x_j - w_{lj})^2 \right]^{\frac{1}{2}}.$$

Let w_l^s be the weight vector for the lth node on the sth step of the algorithm, X_i be the input vector for the ith training case, and α^s be the learning rate for the sth step. On each step, a training case X_i is selected, and the index q of the winning node (cluster) is determined by

$$q = \arg \min_l \|w_l^s - X_i\|.$$

The Kohonen update rule for the winner node is given by

$$w_q^{s+1} = w_q^s (1 - \alpha^s) + X_i \alpha^s = w_q^s + \alpha^s (X_i - w_q^s). \quad (1)$$

For all nonwinning nodes, $w_l^{s+1} = w_l^s$. Several others algorithms have been developed in the neural net and machine learning literature. Neural networks which update the weights of the winner node and the weights of nodes in a pre-specified neighborhood of the winner are also possible. See Hecht-Nielsen (1990) and Kosko (1992) for a historical and technical overview of competitive learning.

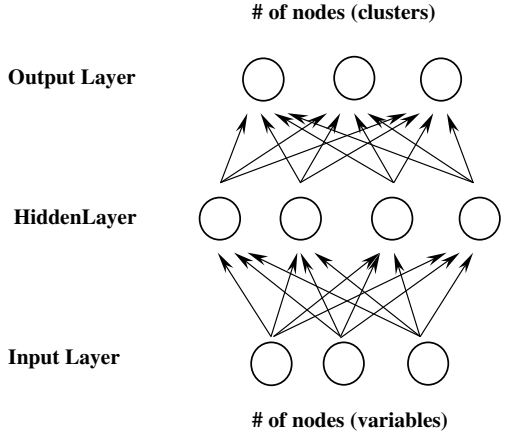

Fig. 2. Illustration of a neural network for clustering.

3. Monte Carlo simulation

In this study several populations were generated with number of clusters $k = 2, 3, 4, 5, 10$, with

equal sizes and number of random variables $p = 2, 4, 6, 8, 10, 20$. The total number of observations for each population was set as $n = 500$ and the number of observations generated for each cluster was equals to n/k. Each cluster had its own mean vector μ_i and covariance matrix Σ^i_{pxp}, $i = 1, 2, \ldots, k$. Different degrees of correlation among the p variables were investigated. The normal multivariate distribution was used to generate the observations for each cluster. First, the clusters were simulated very far apart. Next, many degrees of overlapping among clusters were introduced. Contamination of the original data by the inclusion of outliers was also conducted to analyse the robustness of the clustering algorithms. Clusters were generated according to the procedure proposed by Milligan and Cooper (1980). A total of 1000 samples were selected from each simulated population.

The elements of each sample were clustered into k groups by using all eight clustering procedures presented Section 2. The resulted partition was then compared with the true population. The performance of the algorithm was evaluated by the average percentage of correct classification (recovery rate) and the internal cluster dispersion rate of the final partition defined as

$$\text{icdrate} = 1 - \frac{\text{SSB}}{\text{SST}} = 1 - R^2, \tag{2}$$

where $R^2 = (\text{SSB}/\text{SST})$; $\text{SSB} = \sum_{j=1}^{k} d_{j0}^2$; $\text{SST} = \sum_{l=1}^{n} d_l^2$, d_{j0} is the Euclidean distance between the jth cluster center vector and the overall sample mean vector, d_l is the Euclidean distance between the lth observation vector and the overall sample mean vector, k is the number of clusters, n is the number of observed vectors. The SSB and SST are called respectively, the total sum of squares between clusters and the total sum of squares of the partition (Everitt, 2001). The smaller the value the icdrate the smaller is the intraclass clusters dispersion.

In all clustering algorithms discussed in this paper the Euclidean distance was used to measure similarity among clusters. In the next section the simulation procedure as well the generated populations will be described with details.

3.1. The algorithm to simulate clusters

The population structure of clusters were simulated to possess features of internal cohesion and external isolation. The algorithm proposed by Milligan and Cooper (1980) was used to generate clusters far apart and the same algorithm with modifications was used to generate clusters with overlapping. The basic steps involved in the simulation are described next.

3.1.1. Simulating the boundaries for nonoverlapping clusters

For each cluster, boundaries were determined for each variable. To be part of a specific cluster, the sampled observations had to fall into these boundaries. For the first cluster the standard deviation for the first variable was generated from a uniform distribution in the interval (10; 40). The range of the cluster in the specific variable is then defined as three times the standard deviation and the average is the midpoint of the range. Therefore, the boundaries were 1.5 standard deviation away from the cluster mean in each variable. The boundaries for the other clusters in the specific variable were chosen by a similar procedure with a random degree of separation $Q_i = f(s_i + s_j)$ among them where f is a value of an uniform distribution in the interval (0.25, 0.75) and s_i, s_j, $i \neq j$ are the standard deviations of the clusters i and j, $i, j = 1, 2 \ldots, k - 1$. For the remaining variables the boundaries were determined by the same procedure with the maximum range being limited by three times the range of the first variable. The ordering of the clusters was chosen randomly. See Fig. 3 for a general illustration.

3.1.2. Simulating the boundaries for overlapping clusters

To generate the boundaries for overlapping clusters, Milligan and Cooper's (1980) procedure was used with the following modification: for a specific dimension let LI_i and LI_j be the lower limits of clusters i and j, respectively, $i \neq j$, where

$$LI_j = (1 - m)\text{range}_i + LI_i, \tag{3}$$

m being the quantity specifying the intersection between clusters i, j and range_i the range of cluster i,

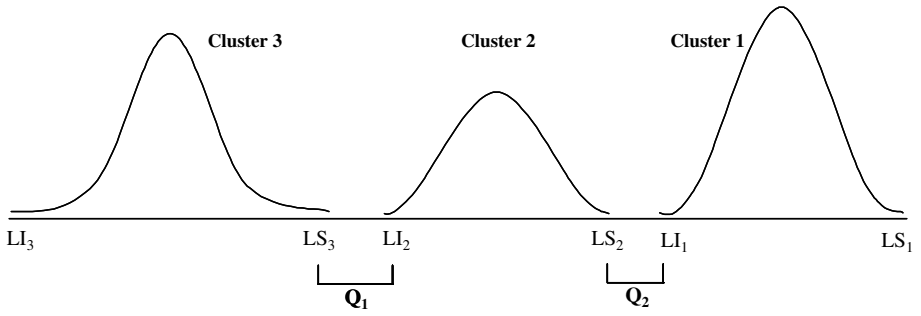

Fig. 3. Nonoverlapping clusters population.

$0 < m < 1$. Let the length of the interval of the intersection be defined as

$$R_i = m\,\text{range}_i, \quad i = 1, 2, \ldots, (k-1). \tag{4}$$

First 40% (i.e. $m = 0.40$) of the observations were generated in the intersection region between any two clusters. Next this amount was increased to 60% (i.e. $m = 0.60$). In Fig. 4 a general illustration is presented for the case where there are $k = 3$ clusters with overlapping between clusters 3 and 2 (area denoted by R_1) and clusters 2 and 1 (area denoted by R_2). To assure that all the clusters had m% observations in the respective region of overlapping the following procedure was used: first the clusters were generated with boundaries according to (3). Next random observations were generated from a Uniform distribution with support defined in the overlapping region as defined in (4) for the pre-specified value of m. Finally, the clusters overlapping regions were identified and the observations in the region were randomly substituted by those generated from the Uniform distribution, half of the observations for each cluster, in such way that in the end of the procedure there was m% observations in the intersection area between clusters.

3.1.3. Data generation

In both, nonoverlapping and overlapping cases, the observations for each cluster were generated from a multivariate normal distribution with the mean vector equals to the vector containing the midpoints of the boundaries length for each of the p variables. Population compose by clusters with the same and different shapes were simulated. For each cluster the diagonal elements of the covariance matrix are the square of the standard deviation obtained in the simulation algorithm described in Sections 3.1.1 and 3.1.2. The off diagonal elements are selected according to the following structures: **S0**: all clusters have a correlation matrix equals to the identity (uncorrelated case); **S1**: all clusters have the same correlation matrix and the correlation between any two variables are the same. The correlation coefficients $\rho = \text{Corr}(X_i, X_j)$, $i \neq j$, were generated from a

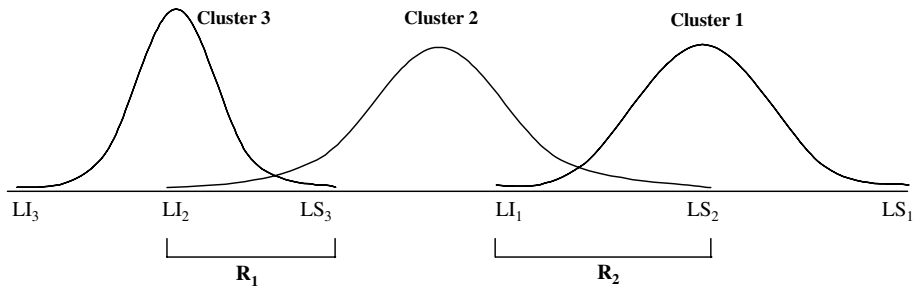

Fig. 4. Overlapping clusters population.

uniform distribution in the intervals (0.25, 0.5), (0.5, 0.75) and (0.75, 1) which characterize small, medium and high correlation structures; **S2**: all clusters have the same correlation matrix but the correlation between any two variables is not necessarily the same. The values of the correlation coefficients ρ_{ij} were generated according to the uniform distribution as described in case **S1**; **S3**: all clusters have different correlation matrices and for any cluster the correlation coefficients are generated from a uniform distribution as in case **S1**; **S4**: clusters have different correlation matrices in such way that half of the clusters in the population have correlation coefficients generated from an uniform distribution in the interval (0.25; 0.5) and the other half from an uniform in the interval (0.75, 1); **S5**: clusters have different correlation matrices in such way that one-third of the clusters in the population have correlation coefficients generated from an uniform distribution in the interval (0.25; 0.5), one-third from an uniform in the interval (0.5, 0.75) and one-third from an uniform distribution in the interval (0.75; 1); **S6**: all clusters have different correlation matrices and the correlation coefficients were generated from an uniform distribution in the (0, 1) interval.

Data were generated with and without outliers. Three percentage of contamination of the original data were considered: 10%, 20% and 40%. For the study of the effect of outliers only data sets with nonoverlapping clusters were generated. A total of 2530 data sets were simulated for the complete study presented in this paper.

3.1.4. Fuzzy c-means and SOM implementation

Fuzzy c-means was implemented using a degree of fuzziness $\lambda = 2$. SOM network was implemented by using SAS's statistical software (1999). Incremental training was used. The learning rate was initialized as 0.5 and was linearly reduced to 0.02 during the first 1000 training steps. The maximum number of steps was set to 500 times the number of clusters. A step is the processing that is performed on a single case. The maximum number of iterations was set to 100. An iteration is the processing that is performed on the entire data set. The convergence criterion was set to 0.0001. Training stops when any one of the termination criteria (maximum number of steps, maximum number of iterations, or convergence criterion) is satisfied. The updating Kohonen rule given in (1) was implemented using as a learning rate $\frac{1}{m^*}$, where m^* is the number of cases that have been assigned to the winning cluster. Let us suppose that when processing a given training case, N_n cases have been previously assigned to the winning seed. In this case the updating Kohonen rule is given by

$$w_q^{s+1} = w_q^s \frac{N_n}{N_n + 1} + X_i \frac{1}{N_n + 1}. \tag{5}$$

This reduction of the learning rate guarantees convergence of the algorithm to an optimum value of the error function, i.e., the sum of squared Euclidean distances between cases and seeds, as the number of training cases goes to infinity. For each generated population the network was trained by using 40% randomly selected observations from the original data set.

4. Results and discussion

To simplify the presentation of the results the structures **S0**–**S6** were grouped into four categories: data simulated with independent variables (**Case 0**), data simulated with medium (**Case 1**) and high (**Case 2**) correlation between variables, and finally data simulated with correlated variables with the correlation coefficient chosen randomly from the uniform in the (0, 1) interval (**Case 3**). Table 1 presents the average results of the correct classification rate considering all the cluster correlation structures evaluated for clusters with nonoverlapping. It can be seen that all the clustering procedures performed very well for all values of p and k, (the majority of average recovery rates were higher or equal to 99%), except for SOM network which had lower recovery rates (some are lower than 80%) being affected by the amount of variables and clusters. The best results were for $p = 4$ (94.99% recovery rate) and for $k = 2$ (99.9% recovery rate). The worst results were 74.98% for $p = 20$ and 76.43 for $k = 10$. Basically the addition of correlation structures did not affected the performance of the algorithms. Table 2 shows the overall average of recovery rate and

Table 1
Average rate of correct classification per number of variables and clusters (nonoverlapping clusters)

Clustering method	Number of variables p						Overall mean	Number of clusters k				
	2	4	6	8	10	20		2	3	4	5	10
Case 0												
Single	99.58	99.98	100.00	100.00	100.00	100.00	99.92	99.96	99.92	99.96	99.90	99.88
Complete	98.09	99.37	100.00	100.00	100.00	100.00	99.58	98.96	99.72	99.96	99.90	99.33
Centroid	99.29	99.98	100.00	100.00	100.00	100.00	99.88	99.88	99.86	99.97	99.83	99.85
Average	99.33	99.99	100.00	100.00	100.00	100.00	99.89	99.88	99.86	99.96	99.83	99.88
Ward	99.42	99.99	100.00	100.00	100.00	100.00	99.90	99.92	99.86	99.97	99.83	99.93
K-means	92.21	99.78	100.00	100.00	100.00	100.00	98.66	99.83	96.56	98.33	99.11	99.48
Fuzzy	99.47	99.98	100.00	100.00	100.00	100.00	99.91	99.87	99.86	99.93	99.93	99.95
SOM	88.55	94.99	86.76	77.12	79.03	74.98	83.57	99.90	86.03	78.78	76.71	76.43
Mean	96.99	99.26	98.34	97.14	97.38	96.87	97.66	99.78	97.71	97.11	96.88	96.84
Case 1												
Single	98.99	99.96	99.96	99.97	99.96	99.96	99.80	99.81	99.81	99.80	99.79	99.79
Complete	98.04	99.90	99.97	99.95	99.93	99.93	99.62	99.38	99.70	99.85	99.74	99.45
Centroid	98.90	99.97	99.97	99.95	99.94	99.95	99.78	99.78	99.76	99.83	99.76	99.77
Average	99.08	99.94	99.96	99.96	99.93	99.94	99.80	99.79	99.78	99.86	99.78	99.80
Ward	98.89	99.97	99.97	99.95	99.93	99.94	99.78	99.78	99.75	99.86	99.73	99.76
K-means	91.91	99.67	99.97	99.96	99.94	99.94	98.57	99.79	96.45	98.20	99.02	99.38
Fuzzy	99.30	99.96	99.97	99.97	99.97	99.96	99.86	99.83	99.82	99.89	99.87	99.89
SOM	88.28	88.64	86.83	76.98	78.81	74.48	82.34	99.82	84.82	77.47	74.97	74.60
Mean	96.67	98.50	98.33	97.09	97.30	96.76	97.44	99.75	97.48	96.84	96.58	96.55
Case 2												
Single	98.63	99.85	99.92	99.95	99.94	99.94	99.71	99.71	99.71	99.70	99.69	99.71
Complete	97.51	99.83	99.93	99.93	99.91	99.91	99.50	99.33	99.62	99.63	99.59	99.35
Centroid	98.63	99.90	99.94	99.93	99.92	99.93	99.71	99.71	99.70	99.75	99.69	99.68
Average	98.82	99.87	99.93	99.95	99.92	99.93	99.73	99.73	99.72	99.82	99.72	99.69
Ward	98.52	99.89	99.94	99.93	99.91	99.92	99.69	99.70	99.67	99.73	99.65	99.67
K-means	91.55	99.62	99.94	99.94	99.92	99.93	98.48	99.69	96.37	98.15	98.92	99.29
Fuzzy	98.75	99.91	99.95	99.96	99.96	99.95	99.75	99.73	99.74	99.77	99.72	99.78
SOM	87.64	85.45	86.26	76.87	78.64	74.10	81.49	99.64	83.77	76.60	74.15	73.32
Mean	96.26	98.04	98.23	97.06	97.27	96.70	97.26	99.65	97.29	96.64	96.39	96.31
Case 3												
Single	98.62	99.87	99.85	99.89	99.88	99.88	99.67	99.75	99.75	99.71	99.57	99.56
Complete	97.43	99.74	99.86	99.86	99.85	99.84	99.43	99.20	99.54	99.61	99.62	99.18
Centroid	98.17	99.88	99.86	99.88	99.85	99.88	99.59	99.61	99.59	99.62	99.57	99.55
Average	98.23	99.86	99.86	99.89	99.88	99.87	99.60	99.61	99.58	99.65	99.59	99.56
Ward	98.19	99.88	99.88	99.87	99.86	99.87	99.59	99.62	99.57	99.62	99.57	99.57
K-means	90.75	99.51	99.87	99.89	99.86	99.88	98.29	99.57	96.08	97.87	98.76	99.18
Fuzzy	98.33	99.93	99.91	99.93	99.91	99.91	99.65	99.65	99.65	99.63	99.63	99.71
SOM	85.57	81.42	86.14	76.25	78.28	73.51	80.19	99.36	82.21	74.77	72.52	72.11
Mean	95.66	97.51	98.15	96.93	97.17	96.58	97.00	99.54	97.00	96.31	96.10	96.05

the overall average of internal dispersion for all clustering algorithms. SOM is the method with the highest average dispersion rate (0.1334) and the lowest overall average recovery rate (81.39%). Fuzzy c-means presented the smallest average dispersion rate (0.0387) and the highest average

Table 2
Average results for correct classification and internal cluster dispersion rates (nonoverlapping clusters)

Clustering method	Number of variables p						Overall mean	Number of clusters k				
	2	4	6	8	10	20		2	3	4	5	10
Correct classification (%)												
Single	98.82	99.90	99.93	99.95	99.94	99.94	99.75	99.77	99.76	99.75	99.72	99.74
Complete	97.74	99.81	99.94	99.93	99.91	99.91	99.54	99.30	99.64	99.73	99.68	99.35
Centroid	98.73	99.93	99.94	99.93	99.92	99.93	99.73	99.74	99.72	99.79	99.71	99.70
Average	98.83	99.90	99.93	99.95	99.93	99.93	99.75	99.75	99.73	99.81	99.73	99.71
Ward	98.70	95.36	99.95	99.94	99.92	99.93	98.96	99.74	98.43	99.80	98.42	98.44
K-means	91.59	99.64	99.95	99.94	99.92	99.93	98.50	99.72	96.36	98.14	98.94	99.31
Fuzzy	98.95	99.94	99.96	99.96	99.96	99.95	99.79	99.77	99.77	99.80	99.77	99.83
SOM	87.66	84.50	86.45	76.83	78.67	74.21	81.39	98.32	83.97	76.64	74.26	73.75
Mean	96.38	97.37	98.26	97.05	97.27	96.72	97.17	99.52	97.17	96.68	96.28	96.23
Internal dispersion rate												
Single	0.0310	0.0560	0.0544	0.0584	0.0483	0.0468	0.0492	0.0821	0.0650	0.0481	0.0316	0.0189
Complete	0.0281	0.0572	0.0593	0.0621	0.0594	0.0509	0.0529	0.0871	0.0729	0.0529	0.0340	0.0174
Centroid	0.0291	0.0573	0.0546	0.0591	0.0512	0.0468	0.0497	0.0830	0.0688	0.0475	0.0313	0.0179
Average	0.0281	0.0513	0.0558	0.0570	0.0493	0.0455	0.0478	0.0802	0.0632	0.0463	0.0323	0.0172
Ward	0.0271	0.0535	0.0545	0.0579	0.0478	0.0478	0.0481	0.0818	0.0630	0.0484	0.0313	0.0160
K-means	0.0362	0.0545	0.0577	0.0608	0.0485	0.0476	0.0509	0.0808	0.0661	0.0495	0.0382	0.0198
Fuzzy	0.0046	0.0458	0.0502	0.0499	0.0399	0.0387	0.0382	0.0677	0.0529	0.0367	0.0260	0.0077
SOM	0.0621	0.1363	0.1855	0.1261	0.1893	0.1014	0.1334	0.1238	0.1218	0.1270	0.1472	0.1475
Mean	0.0308	0.0640	0.0715	0.0664	0.0667	0.0532	0.0588	0.0858	0.0717	0.0570	0.0465	0.0328

recovery rate (99.79%). The other methods had similar results with average recovery rates over 99% and average dispersion rate around 0.05. Tables 3 and 4 present the results for overlapping clusters. The performance decreased substantially for all the algorithms except for Fuzzy *c*-means which still presented an average recovery rate over or close to 90% for 40% degree of overlapping, and

Table 3
Average correct classification rate by number of variables and clusters (clusters with 40% overlapping)

Clustering method	Number of variables p						Overall mean	Number of clusters k				
	2	4	6	8	10	20		2	3	4	5	10
Case 0												
Single	85.43	82.83	81.70	81.23	79.23	78.90	81.55	82.96	82.46	81.59	81.19	79.58
Complete	83.63	82.47	81.01	80.74	79.24	78.64	80.96	82.72	81.96	81.16	80.17	78.78
Centroid	84.49	83.47	81.36	80.79	79.27	78.91	81.38	83.24	82.46	81.58	80.42	79.22
Average	84.53	83.54	82.17	81.78	80.07	79.18	81.88	83.42	82.93	82.38	81.09	79.57
Ward	83.87	82.03	80.48	80.00	78.61	78.42	80.57	81.90	81.05	80.77	80.19	78.93
K-means	84.70	83.87	82.20	81.94	80.03	79.67	82.07	83.77	83.09	82.20	81.77	79.51
Fuzzy	91.38	91.03	90.92	90.78	90.67	90.56	90.89	92.40	92.16	90.97	89.84	89.09
SOM	78.80	76.93	74.40	74.03	72.79	71.27	74.70	78.40	76.80	75.94	73.53	68.85
Mean	84.60	83.27	81.78	81.41	79.99	79.44	81.75	83.60	82.86	82.07	81.02	79.19
Case 1												
Single	81.99	82.52	81.52	81.11	79.02	78.75	80.82	82.12	81.79	80.83	80.52	78.84
Complete	81.05	82.05	80.85	80.59	79.07	78.51	80.35	82.25	81.35	80.43	79.48	78.25
Centroid	82.02	82.99	81.24	80.59	79.10	78.82	80.79	82.73	81.99	80.85	79.86	78.54
Average	81.66	83.17	82.02	81.54	79.87	78.99	81.21	82.71	82.33	81.73	80.37	78.90
Ward	80.96	81.48	80.33	79.76	78.44	78.28	79.88	81.47	80.61	79.94	79.30	78.06
K-means	80.45	83.47	82.03	81.79	79.90	79.53	81.19	82.71	82.42	81.28	80.93	78.64
Fuzzy	90.71	90.84	90.92	90.78	90.66	90.54	90.74	92.28	91.98	90.80	89.70	88.97
SOM	76.37	76.08	74.26	73.82	72.61	71.05	74.03	77.50	76.41	75.41	72.55	68.30
Mean	81.90	82.83	81.64	81.25	79.84	79.31	81.13	82.97	82.36	81.41	80.34	78.56
Case 2												
Single	77.83	82.31	81.40	81.00	78.92	78.66	80.02	81.09	80.84	80.12	79.81	78.24
Complete	79.31	81.79	80.75	80.47	78.99	78.41	79.95	81.92	80.98	79.96	79.10	77.80
Centroid	78.95	82.78	81.15	80.48	79.01	78.73	80.18	81.97	81.28	80.23	79.46	77.99
Average	79.70	82.95	81.91	81.50	79.75	78.88	80.78	82.24	81.85	81.28	80.00	78.54
Ward	79.35	81.26	80.16	79.61	78.37	78.18	79.49	81.15	80.21	79.50	78.97	77.61
K-means	77.50	83.25	81.93	81.68	79.79	79.42	80.59	82.24	81.61	80.79	80.23	78.12
Fuzzy	89.65	90.71	90.91	90.77	90.66	90.56	90.54	92.00	91.68	90.63	89.58	88.83
SOM	74.08	75.38	74.16	73.70	72.52	70.92	73.46	77.01	75.79	74.71	71.81	67.97
Mean	79.55	82.55	81.55	81.15	79.75	79.22	80.63	82.45	81.78	80.90	79.87	78.14
Case 3												
Single	75.51	81.95	81.41	80.82	78.75	78.53	79.50	80.49	80.24	79.70	79.28	77.78
Complete	75.96	81.52	80.57	80.28	78.83	78.24	79.23	80.84	80.27	79.50	78.35	77.21
Centroid	75.17	82.50	81.03	80.30	78.85	78.57	79.40	81.01	80.42	79.52	78.66	77.41
Average	75.60	82.61	81.74	81.27	79.61	78.74	79.93	81.32	80.94	80.23	79.20	77.95
Ward	75.98	81.12	79.88	79.42	78.18	78.01	78.77	80.35	79.47	78.83	78.24	76.94
K-means	74.48	82.93	81.78	81.54	79.65	79.26	79.94	81.34	81.03	80.27	79.54	77.54
Fuzzy	88.47	90.44	90.85	90.75	90.64	90.52	90.28	91.74	91.32	90.35	89.40	88.57
SOM	72.45	74.64	74.02	73.50	72.35	70.69	72.94	76.49	75.32	74.14	71.63	67.13
Mean	76.70	82.21	81.41	80.99	79.61	79.07	80.00	81.70	81.13	80.32	79.29	77.57

Table 4
Average correct classification rate by number of variables and clusters (clusters with 60% overlapping)

Clustering method	Number of variables p						Overall mean	Number of clusters k				
	2	4	6	8	10	20		2	3	4	5	10
Case 0												
Single	66.78	66.47	65.91	65.64	65.33	64.91	65.84	68.91	67.67	65.27	64.49	62.86
Complete	66.37	65.80	65.46	65.08	64.86	64.43	65.33	68.57	67.41	64.71	63.07	62.90
Centroid	67.55	67.04	66.56	65.99	65.60	65.26	66.33	69.73	68.25	65.44	65.21	63.05
Average	67.00	66.34	66.12	65.63	65.27	64.88	65.87	69.53	68.47	64.91	64.49	61.98
Ward	67.06	66.05	65.78	65.43	65.12	64.76	65.70	69.15	67.80	64.87	63.69	62.99
K-means	66.87	66.41	66.22	65.60	65.23	64.84	65.86	70.79	66.55	64.92	63.88	63.17
Fuzzy	88.97	88.88	88.84	88.70	88.56	88.32	88.71	89.62	89.29	88.85	88.32	87.49
SOM	52.23	50.55	50.12	49.20	48.76	47.86	49.78	55.30	52.11	49.42	47.27	44.83
Mean	67.86	67.19	66.88	66.41	66.09	65.66	66.68	70.20	68.44	66.05	65.05	63.66
Case 1												
Single	66.64	66.32	65.74	65.49	65.18	64.80	65.69	68.78	67.50	65.14	64.33	62.74
Complete	66.21	65.67	65.31	64.97	64.75	64.31	65.20	68.48	67.27	64.59	62.95	62.73
Centroid	67.45	66.92	66.44	65.82	65.50	65.19	66.22	69.59	68.14	65.31	65.11	62.95
Average	66.84	66.03	65.81	65.53	65.16	64.81	65.70	69.15	68.33	64.79	64.37	61.85
Ward	66.92	65.85	65.58	65.30	65.04	64.63	65.55	69.00	67.64	64.71	63.55	62.88
K-means	66.73	66.27	66.02	65.46	65.12	64.71	65.72	70.64	66.38	64.77	63.76	63.04
Fuzzy	89.03	88.88	88.84	88.70	88.56	88.52	88.75	89.63	89.45	88.84	88.37	87.49
SOM	52.10	50.43	50.01	49.07	48.66	47.75	49.67	55.20	51.98	49.31	47.14	44.71
Mean	67.74	67.04	66.72	66.29	66.00	65.59	66.56	70.06	68.34	65.93	64.95	63.55
Case 2												
Single	66.55	66.20	65.62	65.41	65.09	64.73	65.60	68.69	67.40	65.04	64.23	62.64
Complete	66.12	65.59	65.21	64.90	64.67	64.24	65.12	68.42	67.18	64.50	62.87	62.63
Centroid	67.35	66.84	66.35	65.71	65.45	65.14	66.14	69.49	68.07	65.23	65.04	62.88
Average	66.75	65.97	65.72	65.45	65.07	64.73	65.61	69.05	68.25	64.68	64.33	61.76
Ward	66.81	65.65	65.47	65.22	65.00	64.55	65.45	68.90	67.53	64.60	63.41	62.81
K-means	66.63	66.17	65.93	65.36	65.05	64.62	65.62	70.54	66.27	64.67	63.68	62.96
Fuzzy	88.96	88.88	88.83	88.69	88.56	88.52	88.74	89.63	89.45	88.83	88.31	87.49
SOM	52.00	50.33	49.91	48.99	48.60	47.68	49.59	55.12	51.91	49.23	47.06	44.63
Mean	67.65	66.95	66.63	66.22	65.94	65.52	66.48	69.98	68.26	65.85	64.87	63.47
Case 3												
Single	66.39	66.01	65.48	65.24	64.96	64.60	65.45	68.54	67.24	64.90	64.09	62.47
Complete	65.95	65.47	65.02	64.74	64.50	64.08	64.96	68.33	67.02	64.35	62.65	62.46
Centroid	67.25	66.69	66.21	65.48	65.37	64.99	66.00	69.38	67.91	65.09	64.87	62.74
Average	66.63	65.81	65.47	65.34	64.92	64.56	65.46	68.82	68.09	64.49	64.30	61.60
Ward	66.65	65.56	65.24	65.04	64.88	64.39	65.29	68.73	67.38	64.43	63.30	62.65
K-means	66.47	65.95	65.73	65.21	64.93	64.44	65.45	70.41	66.09	64.48	63.48	62.83
Fuzzy	88.95	88.87	88.83	88.68	88.54	88.52	88.73	89.61	89.45	88.82	88.31	87.48
SOM	51.80	50.17	49.79	48.86	48.49	47.71	49.47	55.00	51.72	49.08	46.91	44.65
Mean	67.51	66.82	66.47	66.07	65.83	65.41	66.35	69.85	68.11	65.70	64.74	63.36

around 88% for 60% of overlapping. As expected the decreased in performance was higher for the 60% overlapping degree than for 40% for all methods. For the traditional hierarchical and the *K*-means methods the overall average of recovery rate dropped to about 80% for 40% degree of

Table 5
Average results of clusters internal dispersion rate (clusters with overlapping)

Clustering method	Number of variables						Overall mean	Number of clusters				
	2	4	6	8	10	20		2	3	4	5	10
Internal dispersion rate (40% overlapping)												
Single	0.1147	0.1030	0.1091	0.1103	0.1014	0.0967	0.1059	0.1937	0.1249	0.0880	0.0827	0.0402
Complete	0.0884	0.0889	0.0891	0.1014	0.0961	0.0941	0.0930	0.1960	0.0999	0.0723	0.0495	0.0473
Centroid	0.0927	0.0875	0.0910	0.0921	0.0938	0.0883	0.0909	0.1916	0.0950	0.0849	0.0481	0.0350
Average	0.0903	0.1023	0.0961	0.0981	0.0925	0.0865	0.0943	0.1918	0.0958	0.0784	0.0619	0.0437
Ward	0.0870	0.0857	0.0967	0.0984	0.0928	0.0898	0.0917	0.1971	0.0982	0.0804	0.0506	0.0324
K-means	0.1024	0.0864	0.0818	0.0977	0.0905	0.0866	0.0909	0.1649	0.0968	0.0935	0.0646	0.0347
Fuzzy	0.0776	0.0570	0.0454	0.0434	0.0347	0.0269	0.0475	0.1023	0.0704	0.0363	0.0221	0.0065
SOM	0.1990	0.2073	0.2119	0.2219	0.2410	0.2565	0.2229	0.3784	0.2540	0.1831	0.1589	0.1403
Mean	0.1065	0.1023	0.1026	0.1079	0.1054	0.1032	0.1046	0.2020	0.1169	0.0896	0.0673	0.0475
Internal dispersion rate (80% overlapping)												
Single	0.1312	0.1334	0.1300	0.1272	0.1225	0.1120	0.1260	0.2253	0.1302	0.1005	0.1001	0.0741
Complete	0.1181	0.1158	0.1137	0.1149	0.1153	0.1121	0.1150	0.2179	0.1016	0.1089	0.0771	0.0694
Centroid	0.1149	0.1169	0.1150	0.1130	0.1107	0.1034	0.1123	0.2217	0.1079	0.0908	0.0746	0.0667
Average	0.1096	0.1079	0.1048	0.1041	0.1049	0.0999	0.1052	0.2062	0.1012	0.0986	0.0658	0.0542
Ward	0.1041	0.1056	0.1041	0.1020	0.1016	0.0984	0.1026	0.2120	0.1103	0.0792	0.0661	0.0454
K-means	0.1140	0.1124	0.1103	0.1093	0.1072	0.1028	0.1093	0.2120	0.1042	0.1031	0.0687	0.0588
Fuzzy	0.0786	0.0766	0.0601	0.0546	0.0529	0.0558	0.0631	0.1186	0.0837	0.0514	0.0385	0.0232
SOM	0.2135	0.2230	0.2268	0.2275	0.2339	0.2269	0.2253	0.3956	0.2488	0.1840	0.1636	0.1343
Mean	0.1230	0.1240	0.1206	0.1191	0.1186	0.1139	0.1199	0.2262	0.1235	0.1021	0.0818	0.0658

overlapping and to 66% for 60% of overlapping. SOM network performed regularly for 40% of overlapping with average of recovery rate around 75% and very bad for 60% of overlapping reaching an average recovery rate around 50%. Table 5 shows the average dispersion rate for the overlapping cases. SOM had the highest overall averages (0.2229 and 0.2253) and Fuzzy *c*-means the smallest (0.0475; 0.0631). For the other methods the overall average are around 0.10. Fuzzy *c*-means had similar values of average internal dispersion rates for the overlapping data, contrary to the other methods which were very affected. The results for contaminated data with outliers are presented in Tables 6 and 7. When outliers were introduced the performance of all the algorithms decreased and SOM was more affected. For 10% of outliers the average recovery rates were over or similar to 95% for all methods except *K*-means (89.82%) and SOM (50.51%). Similar results were found for 20% of outliers. For 40% of outliers the average recovery rate of Fuzzy *c*-means was lower than single linkage (88.91% and 98.10% respectively) and SOM had the average recovery rate below 50%. All the other methods presented average recovery rate over 80%. The average dispersion rate increased substantially except for Fuzzy *c*-means which averaged about 0.10. The *K*-means and the hierarchical algorithms averaged about 0.20 except for the single linkage which had the highest averages ranging from 0.4303 for 10% to 0.6096 for 40% of outliers and the Ward's method which had the smallest averages among the hierarchical procedures (0.1213, 0.1410 and 0.1687 for 10%, 20% and 40% of outliers respectively). SOM averaged about 0.24 and it was higher than the majority of the other methods except to the centroid method for 20% and 40% of contamination.

5. Final remarks

The results presented in this paper show that in general the performance of the clustering algorithm is more affected by overlapping than by

Table 6
Average correct classification rate—clusters with outliers (nonoverlapping)

Clustering method	Number of variables						Overall mean	Number of clusters				
	2	4	6	8	10	20		2	3	4	5	10
Outliers: 10%												
Single	97.99	97.45	97.54	97.58	97.65	97.70	97.65	98.44	97.91	97.96	97.35	96.60
Complete	94.02	93.85	93.88	93.78	93.67	93.68	93.81	96.45	93.45	93.27	93.13	92.76
Centroid	96.72	96.31	96.31	96.25	95.85	95.71	96.19	98.56	97.51	96.68	94.75	93.47
Average	96.71	96.59	96.54	96.36	95.98	95.86	96.34	98.52	97.54	96.35	95.06	94.23
Ward	96.53	96.15	96.22	96.19	96.16	96.12	96.23	97.36	96.38	96.36	95.58	95.47
K-means	90.51	90.06	89.88	89.61	89.48	89.40	89.82	92.52	89.91	89.24	88.76	88.69
Fuzzy	97.11	97.11	96.87	96.89	96.85	96.79	96.94	98.36	97.21	96.78	96.43	95.90
SOM	50.78	50.72	50.58	50.43	50.32	50.24	50.51	61.25	56.37	49.59	45.57	39.80
Mean	90.05	89.78	89.73	89.64	89.49	89.44	89.69	92.68	90.78	89.53	88.33	87.12
Outliers: 20%												
Single	97.78	93.03	92.04	90.61	90.25	89.85	92.26	98.67	91.46	90.97	90.82	89.38
Complete	89.42	89.51	89.47	89.32	89.17	89.07	89.33	93.41	88.69	88.52	88.18	87.84
Centroid	95.21	95.43	95.33	95.23	95.10	94.96	95.21	99.05	96.50	95.33	93.60	91.58
Average	94.93	94.66	94.46	94.38	94.32	94.23	94.50	98.68	95.77	94.51	92.71	90.82
Ward	95.37	95.39	95.27	95.16	95.09	94.92	95.20	96.51	95.73	95.37	94.75	93.64
K-means	84.77	84.10	83.99	83.85	83.67	83.17	83.92	89.31	84.48	84.41	79.42	82.00
Fuzzy	96.00	96.00	95.94	95.91	95.89	95.85	95.93	97.83	96.98	96.31	95.70	92.86
SOM	48.70	48.35	48.02	47.84	47.57	47.49	47.99	61.67	55.12	45.49	39.09	38.60
Mean	87.77	87.06	86.82	86.54	86.38	86.19	86.79	91.89	88.09	86.36	84.28	83.34
Outliers: 40%												
Single	98.46	98.41	98.21	98.14	97.88	97.49	98.10	98.79	98.95	98.24	97.56	96.95
Complete	81.34	90.13	86.66	84.00	83.16	80.05	84.22	90.35	83.79	82.51	82.40	82.07
Centroid	92.40	95.68	94.05	93.91	93.03	91.97	93.51	98.72	96.27	92.88	90.82	88.83
Average	91.66	95.16	94.33	93.44	92.80	90.84	93.04	98.82	95.04	92.42	90.24	88.67
Ward	83.82	95.99	91.17	89.53	86.44	82.56	88.25	93.64	87.66	87.16	86.85	85.94
K-means	77.91	85.01	83.60	81.28	79.71	77.18	80.78	86.64	80.35	79.97	77.73	79.21
Fuzzy	84.16	95.88	91.61	90.23	87.60	83.98	88.91	93.74	89.10	88.49	87.43	85.77
SOM	48.52	48.97	48.68	48.48	48.22	46.81	48.28	61.87	54.90	45.80	39.61	39.22
Mean	82.28	88.15	86.04	84.88	83.60	81.36	84.39	90.32	85.76	83.43	81.58	80.83

the amount of outliers. For nonoverlapping situations all the methods had good performance except SOM network. The best results for average recovery and internal dispersion rates were found for Fuzzy *c*-means which was very stable in all situations achieving recovery averages over 90%. The traditional hierarchical algorithms presented similar performance among themselves and Ward's method was the more stable. The *K*-means method was very affected by the presence of a large amount of outliers (data with 40% of contamination). The overlapping increased substantially the average internal dispersion rate of the partition and decreased the average recovery rate to about 60% except for Fuzzy *c*-means. The correlation structures did not affect very much the performance of the algorithms. This is an interest result because only the Euclidean distance was used in the clustering algorithms. Therefore, although the Euclidean distance is suitable for uncorrelated variables with the same variances (i.e. spherical clusters) this study indicates that it was able to describe very well populations generated with nonspherical clusters with same and different shapes (cases S1–S6). The choice of the clustering algorithm is more crucial. In general for overlapping

Table 7
Average results of clusters internal dispersion rate—clusters with outliers (nonoverlapping)

Clustering method	Number of variables						Overall mean	Number of clusters				
	2	4	6	8	10	20		2	3	4	5	10
Outliers: 10%												
Single	0.4012	0.4105	0.4223	0.4379	0.4496	0.4601	0.4303	0.4948	0.4568	0.4269	0.4008	0.3721
Complete	0.1379	0.1497	0.1680	0.1840	0.1941	0.1910	0.1708	0.2628	0.2048	0.1500	0.1283	0.1081
Centroid	0.1751	0.1878	0.1950	0.2062	0.2152	0.2256	0.2008	0.2824	0.2474	0.2039	0.1471	0.1234
Average	0.1550	0.1636	0.1750	0.1834	0.1930	0.2009	0.1785	0.2538	0.2112	0.1814	0.1316	0.1145
Ward	0.0966	0.1046	0.1173	0.1315	0.1385	0.1392	0.1213	0.1712	0.1530	0.1168	0.0948	0.0707
K-means	0.1464	0.1600	0.1679	0.1816	0.1860	0.1957	0.1730	0.2527	0.2081	0.1710	0.1239	0.1090
Fuzzy	0.0542	0.0663	0.0749	0.0853	0.0912	0.0984	0.0784	0.1184	0.0899	0.0769	0.0640	0.0427
SOM	0.1991	0.2278	0.2424	0.2513	0.2654	0.2702	0.2427	0.3233	0.2760	0.2324	0.2025	0.1792
Mean	0.1707	0.1838	0.1953	0.2076	0.2166	0.2226	0.1995	0.2699	0.2309	0.1949	0.1616	0.1400
Outliers: 20%												
Single	0.5490	0.5633	0.5752	0.5895	0.5996	0.6117	0.5814	0.6432	0.6165	0.5726	0.5584	0.5162
Complete	0.1625	0.1729	0.1872	0.1964	0.2010	0.2066	0.1877	0.2669	0.2179	0.1760	0.1578	0.1201
Centroid	0.2237	0.2395	0.2505	0.2620	0.2692	0.2768	0.2536	0.3219	0.3061	0.2524	0.2103	0.1774
Average	0.1779	0.1958	0.2153	0.2262	0.2337	0.2388	0.2146	0.2665	0.2467	0.2094	0.1839	0.1667
Ward	0.1126	0.1258	0.1400	0.1501	0.1554	0.1622	0.1410	0.1875	0.1701	0.1429	0.1169	0.0876
K-means	0.1621	0.1801	0.1992	0.2094	0.2144	0.2194	0.1974	0.2612	0.2263	0.1952	0.1614	0.1431
Fuzzy	0.0877	0.0965	0.1008	0.1073	0.1121	0.1163	0.1034	0.1416	0.1204	0.1028	0.0849	0.0676
SOM	0.2134	0.2300	0.2505	0.2657	0.2697	0.2761	0.2509	0.3317	0.2848	0.2418	0.2169	0.1793
Mean	0.2111	0.2255	0.2398	0.2508	0.2569	0.2635	0.2413	0.3026	0.2736	0.2367	0.2113	0.1822
Outliers: 40%												
Single	0.5803	0.5988	0.6077	0.6141	0.6209	0.6356	0.6096	0.6737	0.6446	0.6158	0.5750	0.5387
Complete	0.1904	0.2023	0.2168	0.2232	0.2289	0.2383	0.2166	0.2870	0.2384	0.2131	0.1847	0.1600
Centroid	0.2765	0.2850	0.2934	0.2988	0.3088	0.3214	0.2973	0.3570	0.3312	0.3024	0.2662	0.2298
Average	0.2307	0.2472	0.2591	0.2725	0.2780	0.2885	0.2627	0.3123	0.2909	0.2534	0.2369	0.2199
Ward	0.1406	0.1610	0.1674	0.1729	0.1811	0.1891	0.1687	0.2230	0.1960	0.1674	0.1415	0.1155
K-means	0.1944	0.2057	0.2232	0.2286	0.2327	0.2411	0.2209	0.2911	0.2511	0.2112	0.1863	0.1650
Fuzzy	0.0948	0.0972	0.0999	0.1023	0.1049	0.1080	0.1012	0.1046	0.1087	0.1058	0.0958	0.0908
SOM	0.2081	0.2250	0.2446	0.2583	0.2641	0.2741	0.2457	0.3317	0.2766	0.2415	0.2039	0.1749
Mean	0.2395	0.2528	0.2640	0.2713	0.2774	0.2870	0.2653	0.3226	0.2922	0.2638	0.2363	0.2118

clusters the increase of the number of clusters and variables (dimensions) decreased the performance of the clustering algorithms. The same is true for data with outliers. SOM did not performed well in many cases being very affected by the amount of variables and clusters even for the nonoverlapping cases.

The results obtained in this paper agreed partially with Milligan and Cooper's (1980) for *K*-means and the hierarchical algorithms and partially with Schreer et al. (1998) for Fuzzy *c*-means and *K*-means. As far as SOM neural network is concerned the results are more concordant with those presented by Balakrishnan et al. (1994) and less with those shown in Mangiameli et al. (1996). One reason could be that we explore many different data structures and a number of data sets much higher than any other study published so far. Our study differs from others with respect to the clusters sizes. Contrary to the other published articles mentioned in the introduction of this paper, all the populations simulated in this study had the same size (500). As the number of clusters decreased the number of observations in each cluster increased. Therefore, we were able to test the clustering algorithms for situations where each cluster

had 250 observations (case where $k = 2$) up to situations where each cluster had 50 observations (case where $k = 10$). Only 50 observations in each data set were considered by Milligan (1985) and Balakrishnan et al. (1994), 100 in Schreer et al. (1998) and from 100 to 250 in Mangiameli et al. (1996). The number of replicates for each population structure was much higher in our study. We generate 1000 replicates for each structure and the other authors generate only three replicates. Another difference with the above mentioned papers is that in the nonoverlapping case, population were simulated with clusters far apart in all p dimensions and not only in the first dimension as Milligan's proposition (1980, 1985). In the simulation of the overlapping structures we had a good control of the amount of clusters overlapping in each variable. This was not done in the other papers. The simulation of the amount of outliers was also very well controlled. Finally, another possible reason for different results is the method used to implement SOM network. As described by many authors the performance of a neural network depends strongly upon the parameters set for the training stage. For this presented work the optimized routine of SOM implemented in the SAS statistical software was used to generate the clusters. Therefore, the authors believe that the bad performance of SOM was not a result of any inadequate learning process of the network but due to its own structure. Because of the extension of our study we had a better chance to test the performance of SOM in many different scenarios and the presented results indicate that some care should be taken when using SOM neural network to cluster data because its performance could be very poor in some cases. Methods such as Fuzzy c-means, K-means and Ward's for example presented good performance and are simpler to implement.

Many other studies still can be performed. Comparison of the clustering algorithms by using other metrics than the Euclidean distance, populations with clusters of different sizes and generated by a distribution different than the multivariate normal are some examples. The performance of SOM neural network in general situations has also to be better evaluated.

Acknowledgement

The authors were partially financed by the Brazilian Institutions CNPq and CAPES.

References

Anderberg, M.R., 1972. Cluster Analysis for Applications. Academic Press, New York.

Balakrishnan, P.V., Cooper, M.C., Jacob, V.S., Lewis, P.A., 1994. A study of the classification of neural networks using unsupervised learning: A comparison with K-means clustering. Psychometrika 59 (4), 509–525.

Balakrishnan, P.V., Cooper, M.C., Jacob, V.S., Lewis, P.A., 1996. Comparative performance of the FSCL neural net and K-means algorithm for market segmentation. European Journal of Operational Research 93 (1), 346–357.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

Bezdek, J.C., Keller, J., Krishnapuram, R., Pal, N., 1999. Algorithms for Pattern Recognition and Image Processing. Kluwer, Boston.

Carpenter, G.A., Grossberg, S., Rosen, D.B., 1991. Fuzzy art: Stable learning and categorization of analog patterns by adaptive resonance system. Neural Networks 4 (1), 759–771.

Everitt, B.S., 2001. Cluster Analysis. John Wiley & Sons, New York.

Gallant, S.I., 1993. Neural Network Learning and Expert Systems. MIT Press, Cambridge.

Gordon, A.D., 1987. A review of hierarchical classification. Journal of Royal Statistical Society 150 (2), 119–137.

Gower, J.C., 1967. A comparison of some methods of cluster analysis. Biometrics 23 (4), 623–638.

Hathaway, R.J., Bezdek, J.C., 2002. Clustering incomplete relational data using the non-Euclidean relational fuzzy c-means algorithm. Pattern Recognition Letters 23 (1–3), 151–160.

Hebb, D.O., 1949. The Organization of Behavior. John Wiley, New York.

Hecht-Nielsen, R., 1990. Neurocomputing. Addison-Wesley, Reading, MA.

Johnson, R.A., Wichern, D.W., 2002. Applied Multivariate Statistical Analysis. Prentice-Hall, New Jersey.

Kiang, M.Y., 2001. Extending the Kohonen self-organizing map networks for clustering analysis. Computational Statistics & Data Analysis 38 (2), 161–180.

Kohonen, T., 1989. Self-Organization and Associative Memory. Springer-Verlag, New York.

Kohonen, T., 1995. Self-Organizing Maps. Springer-Verlag, Berlin.

Kosko, B., 1992. Neural Networks and Fuzzy Systems. Prentice-Hall, Englewood Cliffs, NJ.

Krishnamurthy, A.K., Ahalt, S.C., Melton, D.E., Chen, P., 1990. Neural networks for vector quantization of speech

and images. IEEE Journal on Selected Areas in Communications 8, 1449–1457.

Mangiameli, P., Chen, S.K., West, D., 1996. A comparison of SOM neural network and hierarchical clustering methods. European Journal of Operational Research 93 (2), 402–417.

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5 (1), 115–133.

Milligan, G.W., Cooper, M.C., 1980. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. Psychometrika 45 (3), 159–179.

Milligan, G.W., 1985. An algorithm for generating artificial test clusters. Psychometrika 50 (1), 123–127.

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychology Review 65 (1), 386–408.

Roubens, M., 1982. Fuzzy clustering algorithms and their cluster validity. European Journal of Operational Research 10, 294–301.

SAS, 1999. SAS/STAT User's Guide (version 8.01). SAS Institute, Cary, NC.

Schreer, J.F., O'Hara, R.J.H., Kovacs, K.M., 1998. Classification of dive profiles: A comparison of statistical clustering techniques and unsupervised artificial neural networks. Journal of Agriculture Biological and Environmental Statistics 3 (4), 383–404.

Susanto, S., Kennedy, R.D., Price, J.H., 1999. A new fuzzy *c*-means and assignment technique based cell formation algorithm to perform part-type clusters and machine-type clusters separately. Production Planning and Control 10 (4), 375–388.

Zhang, D.-Q., Chen, S.-C., 2003. Clustering incomplete data using kernel-based fuzzy *c*-means algorithm. Neural Processing Letters 18 (3), 155–162.