Revista Eletrônica Sistemas & Gestão 4 (2) 89-107
Programa de Pós-graduação em Sistemas de Gestão, TEP/TCE/CTC/PROPP/UFF

Implementing Bootstrap in Ward´s Algorithm to estimate the number of clusters

Sueli A. Mingoti, sueliam@est.ufmg.br
Francisco N. Felix, fnfelix@unama.br

Universidade Federal de Minas Gerais (UFMG), Programa de Estatística
Belo Horizonte, MG, Brasil

*Received: February, 2009 / Accepted: August, 2009

ABSTRACT

In this paper we show how bootstrap can be implemented in hierarchical clustering algorithms as a strategy to estimate the number of clusters (k). Ward´s algorithm was chosen as an example. The estimation of k is based on a similarity coefficient and three statistical stopping rules, pseudo F, pseudo T^2 and CCC. The performance of the estimation procedure was evaluated through Monte Carlo simulation considering data consisting of correlated and uncorrelated variables, nonoverlapping and overlapping clusters. The estimation procedure discussed in this paper can be used with clustering algorithms other than Ward´s and also to provide initial solutions for non-hierarchical grouping methods.

Keywords: Ward´s Algorithm. Estimation of Number of Clusters. Bootstrap.

1. INTRODUCTION

Cluster analysis is used to classify objects into groups based on their similarities. Applications can be found in a variety of fields such as Data Mining, Marketing, Industry, Biology, Ecology, Medicine, Geology, among others. One of the most essential issues is the estimation of the number of clusters (k) since an improper choice might lead to bad clustering outcomes and mistaken decisions. Some procedures of estimation have been proposed in the literature based on statistical methods and neural networks (see for example, Fraley and Raftery, 2002; Guo, Chen and Lyu, 2002); Hruschkaa *et. al*.,2006; Katosa, 2007; Rosenberger and Cheddi, 2000; Steinley and Brusco, 2008; Teboulle *et. al.* 2006). However, most of the studies consider only a small number of real or simulated data sets, a few different structures of correlation between the classification variables and a small number of dimensions (p=2 or 3).

One common procedure in practical data analysis is to use an agglomerative hierarchical cluster algorithm such as single linkage, complete linkage, average linkage, centroid and Ward´s (EVERITT, 2001) as an exploratory method to estimate the value of *k.* However, these algorithms require some criteria to interrupt them in order to obtain an estimate. Many stopping rules based on clusters similarity or the internal variability measures

of the cluster partition have been explored in the literature (see MARRIOTT,1971; KRZANOWSKI and LAI 1988; GORDON, 1999; TIBSHIRANI *et. al.*, 2001, among others). Some of the simplest and very popular statistical stopping rules are *pseudo-F* (CALINSKI and HARABASZ, 1974), *pseudo-*T^2 (DUDA and HART,1973) and *CCC-Cubic Clustering Criterion* (SARLE,1983). In the study presented by MILLIGAN and COOPER (1985) these three stopping rules were the best among 30 different rules which were compared by using Monte Carlo simulation. In their study a total of 108 data sets (with 50 points each) were generated and Euclidian distance was used to compare clusters. The good performance of *pseudo F* was also indicated in ATLAS and OVERALL (1994) although their study was based on a small number of simulated data sets, and in BOWMAN *et al*. (2004) who analysed some neuroimaging data.

In practical data analysis after obtaining a point estimate of *k*, by any procedure, is common to evaluate the partitions in some neighborhood of the estimate in order to find the final solution considering the nature of the data in the specific field. Therefore, an interval estimate for *k* is needed.

In PECK, FISHER and VAN NESS (1989) an approximated confidence interval for the true number of clusters of the partition was built by using bootstrap procedure. For each bootstrap sample their method required a minimization of an objective function in order to obtain the optimal estimate of *k*. In their paper the objective function of the K-Means non-hierarchical cluster procedure (EVERITT, 2001) was used. For each bootstrap sample the number of clusters was estimated (say $\hat{k}$) and the empirical distribution of $\hat{k}$ was used to define the confidence interval for the true value of *k*. Data were simulated considering the univariate and bivariate normal distributions and the respective number of clusters were *k*=5 and 10 for the univariate case and *k*=4 and 9 for the bivariate. From each simulated cluster structure, $R = 30$ random samples of sizes $n = 75$ for $k = 5$, $n = 100$ for $k = 10$, $n = 60$ for $k = 4$ and $n = 90$ for $k = 9$ were selected. The number of bootstrap samples was set as *B*=75 for computational reasons according to the authors. The squared Euclidian distance was used to measure the dissimilarity among the sample clusters and three basic configurations were simulated defined as: "near"- groups with means one standard deviation apart; "far"- groups with means four standard deviations apart and "combined"- groups which was a mixture of these two types of configurations. The results presented in the paper showed that the bootstrap combined with the K-means clustering algorithm was a good strategy to estimate the true number of clusters since in the majority of the cases the proposed procedure was able to identify approximately, the true number of clusters of the simulated structure. The performance was better for groups very far apart as expected. In all cases the range of the confidence intervals increased as the value of the true number of clusters *k* increased. Although only the K-Means method was evaluated in PECK, FISHER and VAN NESS (1989) the idea of using the bootstrap procedure to generate a confidence interval for *k* can be implemented with any other clustering procedure. In this paper we will show how the bootstrap can be implemented in Ward´s hierarchical cluster algorithm (1963) when the goal is to estimate *k* by using some of the stopping rules: similarity coefficient, *pseudo F*, *pseudo* T^2 and *CCC*. To understand some features of these stopping rules a Monte Carlo simulation study was performed. Many different clusters structures were simulated considering spherical and nonspherical clusters, with and without overlapping, with a larger number of points and variables. The number of bootstrap samples was set as *B*=1000 for each simulated cluster structure. Some examples using real data sets are also presented.

There are other algorithms and stopping rules that can be used for grouping and to estimate the number of clusters but they will not be part of this presented paper. The readers can obtain more information in MILLIGAN and COOPER (1985) or XU and WUNSCH (2005), among others.

2. WARD´S CLUSTERING ALGORITHM

Agglomerative hierarchical cluster algorithms are largely used as an exploratory statistical technique to determine the number of clusters of data sets and to create the groups. Basically they work as follows: in the first stage each of the *n* objects to be clustered is considered as a single 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 (or larger similarity) are joined. This 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. WARD´s (1963), called also as minimum variance, is one of the most popular and important algorithms. Briefly speaking let C_l and C_m be two clusters with means vectors (centroids) $\overline{X}_l$ and $\overline{X}_m$ and sizes n_l, n_m, respectively. In Ward´s method the distance between clusters C_l and C_m is a function of the squared Euclidean distance between the cluster centroids and it is defined as

$$d_{l,m} = \frac{n_l n_m}{n_l + n_m} (\overline{X}_l - \overline{X}_m)' \ (\overline{X}_l - \overline{X}_m) \qquad (1)$$

It can be shown that the distance in (1) represents the additional within sum of squares of the partition resulted from the combination of clusters C_l and C_m in only one cluster. In each step of the algorithm the distance as (1) is calculated for every pair of clusters that could be joined in the particular step. The two clusters with the smallest distance are combined. This is equivalent to combine the two clusters that increased the most the sum of squares between clusters of the partition or that minimizes the within sum of squares of the partition. Ward´s algorithm has a good performance for recovering the original clusters (see MINGOTI and LIMA, 2006; MILLIGAN and COOPER, 1980) and provides a solution equivalent to the maximization of the multivariate normal distribution when the covariance matrix for all clusters are equal and proportional to the identity matrix. However, the use of Ward´s method variables does not require multivariate normality.

3. STOPPING RULES TO ESTIMATE THE NUMBER OF CLUSTERS

3.1. SIMILARITY OR DISTANCE LEVEL

A simple rule that can be used to decide which is the appropriate step to stop Ward´s or any other hierarchical cluster algorithm is the analysis of the similarity or distance values of each step. The main objective is to produce a partition such that the elements within a group are similar and elements in different groups are dissimilar. Let g be the respective number of cluster of the particular algorithm step. Then, if from step g to the step (g-1) the similarity (or distance level) decreased (or increased) significantly then the clusters that were joined in (g-1) step were not very similar and they should not be combined. Therefore, the algorithm should be interrupted in step g and the respective number of clusters used as a point estimate of *k*. There are many similarity coefficients in the literature (see JOHNSON and WICHERN, 2002) but in this paper the similarity measure between any two clusters C_l and C_m is defined as

$$S_{lm} = \left(1 - \frac{d_{lm}}{\max\left(d_{ij}, i, j = 1,2,...,n, i \neq j\right)} \right) \times 100 \qquad (2)$$

where $\max\left(d_{ij}, i, j = 1,2,...,n, i \neq j\right)$ is the largest distance between sampled elements e.g., the maximum value of the distance matrix used in the first step of the clustering algorithm and d_{lm} is the distance between clusters C_l and C_m. When the similarity level is below a certain pre-specified level, the cluster algorithm should be interrupted and the respective number of clusters adopted as an estimate of the number of clusters of the partition. Due to the fact that (2) has a maximum value (100) its use makes easier to evaluate the loss of the quality of the partition from one step to the next in the cluster algorithm. However, it is just a start since it reflects only the similarity of the clusters joined in the respective step and it does not take into consideration the internal variability of the data partition that was create in the step. In fact, the index (2), which is implemented in Minitab for Windows statistical software, is a pseudo-similarity because it may take negative values since the distance d_{lm} can be larger than $\max\left(d_{ij}, i, j = 1,2,...,n, i \neq j\right)$. Therefore, only the steps of the clustering algorithm where (2) is positive should be considered to estimate *k*. In Ward´s algorithm the distance d_{ij} between sampled elements is taken as the squared Euclidean distance and d_{lm} is define as in (1).

The main problem is how to choose the proper similarity level to stop the clustering algorithm. Usually the choice is subjective. Of course the best would be to estimate k with high similarity which means to stop the algorithm when similarity is around 90% or more. However, this procedure can produce an estimate of *k* too large, much larger than the necessary as the results in section 4 will show.

3.2. PSEUDO F

The statistics known as *pseudo F* was proposed by CALINSKI and HARABASZ (1974). It is a function of the number of clusters *g* produced in each step of the clustering algorithm and it is defined as

$$F = \frac{SSB/(g-1)}{SSW/(n-g)} = \left((n-g)/(g-1)\right)\left(R^2/(1-R^2)\right) \qquad (3)$$

where $R^2 = (SSB/SST)$ is the squared intraclass correlation coefficient, $SSB = \sum_{j=1}^{g} d_{jo}^2$ and $SST = \sum_{l=1}^{n} d_l^2$ are called the total sum of squares between clusters and the total sum of squares of the partition, respectively; d_{jo} is the Euclidean distance between the *j*th cluster mean vector and the overall sample mean vector; d_l is the Euclidean distance between the *l*th observation and the overall sample mean vector; *n* is the number of observed vectors (sample size) and $SSW = SST - SSB$ is the within sum of squares of the partition. As R^2 increases the intraclass dispersion of the partition decreases. The coefficient R^2 is an increasing function of the number of clusters *g* and decreases when *SSW* increases. In each step of the cluster algorithm the statistics *pseudo F* is calculated. If this function has a

maximum value then the number of clusters should be estimated as the respective value of *g* corresponding to the maximum value of *pseudo F*. However, if the function is directly proportional to *g* then no existence of a natural cluster partition is suggested by the data.

3.3. PSEUDO T^2

The statistics *pseudo* T^2 was proposed by DUDA and HART in 1973. Let $C_t = C_l \cup C_m$ be the union of the clusters C_l and C_m. Let $SS_j = \sum_{i=1}^{n_j} d_{ij}^2$, where d_{ij} is the Euclidean distance between the *i*th observation of cluster j and the sample mean of vector of cluster *j*, n_j is the number of elements in cluster *j*, *j=l,m*. The statistics *pseudo* T^2 is defined as

$$T^2 = \frac{d_{lm}}{\left[SS_l + SS_m\right]\left(n_l + n_m - 2\right)^{-1}} \tag{4}$$

where d_{lm} is the Ward´s distance defined in (1) and the denominator of (4) represents the total sample variance of observations in the new cluster C_t. The statistics in (4) reduces to the square of a t-Student when *p*=1. In each step of the cluster algorithm the statistics *pseudo* T^2 is calculated. When it reaches its maximum value the cluster algorithm should be interrupted and the number of clusters of the partition should be estimated as the respective value of *g* corresponding to the maximum value of *pseudo* T^2 or (*g+1*) which is the number of clusters related to the previous step.

3.4. CUBIC CLUSTERING CRITERION (CCC)

According to SARLE (1983) the *Cubic Clustering Criterion (CCC)* is based on the assumption that clusters obtained from a *p*-dimensional uniform distribution defined in a hyperbox are hypercubes of same size. The *CCC* value is obtained comparing the observed value of R^2 squared intraclass coefficient with an approximation of the expected value of R^2 calculated under the assumption that clusters are generated by a uniform p-dimensional distribution. Positive values of *CCC* indicate that the observed R^2 is larger than the expected under the uniform distribution and the cluster structure of the data is different from the uniform partition. The statistics *CCC* is defined as

$$CCC = ln\left[\frac{1 - E\left(R^2\right)}{1 - R^2}\right] \frac{\sqrt{\frac{np^*}{2}}}{\left(0.001 + E\left(R^2\right)\right)^{1.2}} \tag{5}$$

$$E(R^2) = 1 - \left[\frac{\sum_{j=1}^{p^*} \frac{1}{n+u_j} + \sum_{j=p^*+1}^{p} \frac{u_j^2}{n+u_j}}{\sum_{j=1}^{p} u_j^2}\right]\left[\frac{(n-k)^2}{n}\right]\left[1 + \frac{4}{n}\right] \tag{6}$$

where s_j is the square root of the j^{th} eigenvalue of the matrix *T= (X'X)/n-1*, *X* is the *nxp* data matrix, $v^* = \prod_{j=1}^{p^*} s_j$, $u_j = s_j / c$, $c = (v^* / k)^{1/p^*}$, *p** < *p* is chosen to be the largest integer less than *k* such that u_{p^*} is not less than one.

The *CCC* provides a crude test for the null hypothesis that the data have been sampled from a uniform distribution on a hyperbox, against the alternative that the data have been sampled from a mixture of spherical multivariate normal distributions with equal variances. SARLE (1983) presented some simulations which showed that the *CCC* criterion works well for clusters very far-apart but its performance decreases as the number of variables increases and the number of observations per cluster decreases.

3.5. A PRACTICAL STRATEGY TO ESTIMATE THE NUMBER OF CLUSTERS

In practical data analysis it is not recommended to use the *pseudo F*, *pseudo* T^2 or *CCC* directly to estimate *k,* e.g., by creating all steps of the hierarchical clustering algorithm (from *g=n* to *g=1*) and taking the value of *g* correspondent to the best value of any of these stopping rules as described in sections 3.2-3.4. This is due to the fact that these rules have a tendency to result in an estimate of *k* larger than necessary when all the steps of the algorithm are taken into consideration. As a practical strategy is better to first use some criterion to define a neighborhood for the true number of clusters *k* (initial solution) and then use one of the stopping rules *pseudo F*, *pseudo* T^2 or *CCC* to choose the best partition among those in the defined neighborhood. In this paper the similarity coefficient (2) combined with the bootstrap methodology is used to define the neighborhood of interest.

4. IMPLEMENTING THE BOOTSTRAP WITH WARD´S ALGORITHM

PECK, FISHER and VAN NESS (1989) proposed the bootstrap procedure to construct confidence interval for the true number of clusters *k* of the partition. The basic idea is as follows: first the researcher defines an objective function *L(.)* used to produce a point estimate of *k*. Given the observed sample of size *n* denoted by $\mathbf{x} = (x_1, x_2, \mathrm{K}, x_n)$, where each x_i is a vector of *px1*, a certain number *B* of bootstrap samples are generated. They are denoted by $\mathbf{x}^*$, $\mathbf{x}^* = (x_1^*, x_2^*, \mathrm{K}, x_n^*)$. For each bootstrap sample the parameter *k* is estimated according to the criteria *L(.)*. Therefore, each bootstrap sample produces a value $\hat{k}$. At the end, from the empirical distribution of $\hat{k}$, a confidence interval for the true *k* is built. In PECK, FISHER and VAN NESS this approach was used considering the objective criterion function $L_n(\hat{k}) = \alpha\hat{k} + (1/n)\,SSW$, where $\alpha > 0$ is a penalty function and *SSW* is the within sum of squares of the partition when the data set is divided into $\hat{k}$ clusters by K-Means clustering algorithm. For each bootstrap sample the value of $\hat{k}$ is the one that minimizes the function $L_n(\hat{k})$.

The bootstrap approach proposed in PECK, FISHER and VAN NESS (1989) can applied in a more general sense. The user just needs to define a criteria that will be used to estimate *k* for each bootstrap sample. In this paper a strategy to estimate *k* was implemented as follows: for each bootstrap sample the Ward´s algorithm was used to cluster the data; in each step of the algorithm, the similarity coefficient defined in (2) was calculated and the candidates to be an estimate of *k* were chosen in each of the three similarity intervals: I1:[60;80); I2:[80;90) and I3:[90;95] according to the observed values of pseudo *F*, *p*seudo T^2 and *CCC* stopping rules. If the similarity interval was not empty then it was

possible to obtain a point estimate of *k*. Therefore, for each bootstrap sample and for each similarity interval an estimate of *k* was produced by each stopping rule. The empirical distribution of $\hat{k}$ was then obtained by using all *B*=1000 bootstrap samples and a 80% confidence interval for the true value of *k* was constructed by using the percentile method.

A Monte Carlo simulation study was implemented. Several populations were generated with *k*=2,5,10,20 clusters containing 50 observations each. The number of random variables (dimensions) were *p*=2,5,10. Each cluster had its own mean vector μ_i and covariance matrix $\Sigma^i{}_{pxp}$, *i=1,2,…,k*. Different degrees of correlation among the *p* variables were investigated and 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.

The same algorithm described by MINGOTI and LIMA (2006), which is a modification of the MILLIGAN´S (1985), was used to generate clusters with and without overlapping. Basically in each structure the clusters were simulated to possess features of internal cohesion and external isolation. The basic steps are described next.

4.1. SIMULATING THE BOUNDARIES FOR NONOVERLAPPING CLUSTERS

For each cluster, boundaries were determined for each variable. To be part of a specific cluster, the sample observation 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 was then defined as 6 times the standard deviation and the cluster average was the midpoint. Therefore, the boundaries were 3 standard deviations away from the cluster mean. 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 a uniform distribution in the interval $(0.75,1)$ and $s_i, s_j, i \neq j,$ are the standard deviations of the clusters *i* and *j*. For the remaining variables the boundaries were determined by the same procedure with the maximum range being limited by 3 times the range of the first variable. The ordering of the clusters was chosen randomly. See Figure 1 for a general illustration.

4.2. SIMULATING THE BOUNDARIES FOR OVERLAPPING CLUSTERS

For a specific dimension let LI_i and LI_j be the lower limits of clusters *i* and *j*, respectively, $i \neq j$, where $LI_j = (1-m)range_i + LI_i$, *m* being the quantity specifying the intersection between clusters *i,j*, and $range_i$ the range of cluster *i*, *0<m<1*. Let the length of the interval of the intersection be defined as $R_i = m(range_i)$, *i=1,2,…,(k-1)*. 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 Figure 2 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).

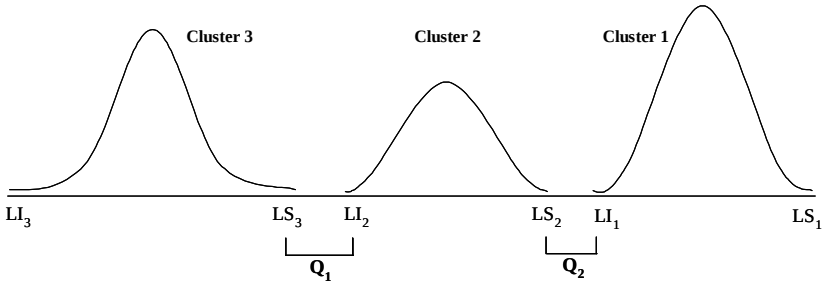

Figure 1: Example of nonoverlapping clusters. LI_i and LS_i are the boundaries of cluster i, i=1,2,3.

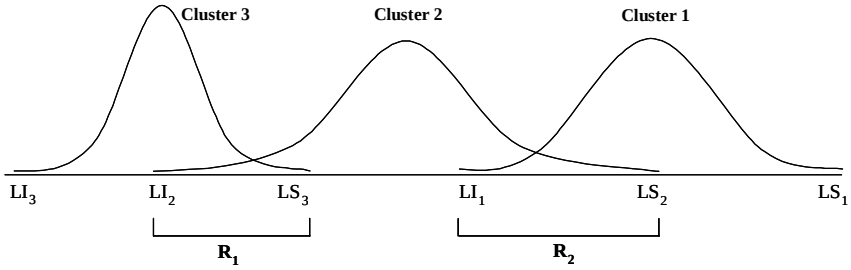

Figure 2: Example of overlapping clusters. . LI_i and LS_i are the boundaries of cluster i, i=1,2,3.

4.3. 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. For each cluster the diagonal elements of the covariance matrix were the square of the standard deviations obtained in the simulation algorithm described in sections 4.1 and 4.2. For *p*=2 and 5 the off diagonal elements were selected according to the following structures: **S1:** all clusters had a correlation matrix equals to the identity (uncorrelated case); **S2:** all clusters had different correlation matrices and for any cluster the correlation coefficients were generated from a uniform distribution in the interval (0,1). For *p=10* all clusters had different correlation matrices and for any cluster the correlation coefficients were generated from a uniform distribution in the interval (0.5,1). All generated correlation matrices used in the simulation procedure were positive definiteness. Due to the computational time, for *p*=2 and 5, 100 different simulations were obtained for each of the structures **S1** and **S2;** for *p*=10 only one structure was simulated and 100 random samples were taken from it. For each *p* and *k* the covariance matrix ${\Sigma^i}_{pxp}$, *i=1,2,…,k.*, was obtained by using the fact that: $cov(X_l, X_j) = \rho \sigma_l \sigma_j, l \neq j$, where *cov* denotes de covariance between the random variables X_l and X_j, ρ, σ_l, σ_j, are respectively the generated correlation and the standard deviations of these variables. The number of bootstrap samples was set as *B*=1000 for each simulated cluster structure.

5. RESULTS AND DISCUSSION

For the discussion presented in this section the criteria for comparison of the stopping rules and the similarity intervals were the average value of $\hat{k}$ distribution and the average confidence interval range. Table 1 presents the average results (considering all the

simulated structures) for nonoverlapping for each *k* and *p*. Since the results for 40 and 60% overlapping are similar only the results for 40% are shown in Table 2. The best results according to the proximity with *k* and the length of the average confidence interval range are marked as a "*" in Tables 1 and 2.

In general, the best similarity interval to estimate *k* was [60,80) for every stopping rule and *p*. For nonoverlapping the *pseudo F*, *pseudo* T^2 and *CCC* stopping rules had similar performance with some advantage for pseudo *F* when k=2. The quality of the stopping rules decreased for overlapping when k=2 but it was still possible to obtain good estimates for *k* in the other cases and again the stopping rules presented similar results. In general the average confidence interval range increased with overlapping compared to nonoverlapping. For a fixed *k* the average range increased as *p* increased. As expected the estimates of *k* are larger for [90,95] similarity interval. Similar to MINGOTI and LIMA (2006) the correlation structure of the variables did not affect much the performance of the stopping rules (results not shown).

Table 1. Results for nonoverlapping

Similarity Interval (%)	CCC	PF	P T^2	CCC	PF	P T^2	CCC	PF	P T^2
	k=2 ; p=2			k=2 ; p=5			k=2 ; p=10		
60|-- 80	2.8; 2.53	2.8; 2.37 *	2.9; 2.61	5.1; 7.54	3.0; 3 *	4.6; 6.08	3.7; 3.8	2.8; 0.86 *	4.9; 6.27
80|-- 90	10.6;13.2	12; 14.7	6.3; 9.8	7.0; 12.4	4.5; 6.09	7.0; 14.8	5.7; 7.5	5.5; 7.47	7.2; 12.3
90|--| 95	19.4; 1.5	32.4; 8.1	19.3; 16,2	18.4; 4.4	55.6; 8.2	28.1; 24.6	13.1; 12.6	60.4; 8	24.4; 33.5
	k=5 ; p=2			k=5 ; p=5			k=5 ; p=10		
60|-- 80	4.3; 0.54 *	4.3; 0.54 *	4.2; 0.65 *	5.0; 0.89 *	5.0; 0.89 *	5.0; 0.89 *	6.0; 1.8 *	6.0; 1.8 *	6.0; 1.8 *
80|-- 90	6.6; 4.9	8.7; 10.6	7.1; 6.1	9.4; 6.4	7.3; 7.8	11.0; 14.2	6.9; 2.9	9.0; 9.2	11.5; 14.5
90|--| 95	14.4; 16	36.8; 17	13.5; 17.1	11.4; 14.9	10.6; 13.5	18.1; 36.7	12.2; 15.9	12.0; 15.6	20.0; 30.1
	k=10 ; p=2			k=10 ; p=5			k=10 ; p=10		
60|-- 80	8.4; 0.34	8.4; 0.34	7.3; 1	9; 0.04 *	9; 0.04 *	8.9; 0.1	8.9; 0.25 *	8.9; 0.25 *	8.9; 0.25 *
80|-- 90	10.3; 2.7	10.4; 2.8	10.2; 2.6 *	13.4; 6.4	13.4; 6.4	13.2; 6.1	13.1; 6.2	13.1; 6.2	13.1; 6.2
90|--| 95	13.3; 5.8	27.4; 29.9	13.6; 10.5	18.2; 18.1	13.8; 9.6	19.3; 24.4	19.7; 21.5	18.5; 18.9	22.7; 27
	k=20 ; p=2			k=20 ; p=5			k=20 ; p=10		
60|-- 80	15.2; 0.72	15.2;0.72	12.3; 2.2	18.8; 0.24 *	18.8; 0.24 *	18.3; 0.9	19.4; 0.20 *	19.4; 0.20 *	19.4; 0.20*
80|-- 90	18.6; 0.23 *	18.6; 0.23 *	16.7; 1.2	19.0; 0 *	19.0; 0 *	18.9; 0.08 *	----	----	----
90|--| 95	24.7; 12.3	24.9; 12.6	24.3; 11.5	29.9; 20.1	29.9; 18.4	29.0; 18.4	27.9;15.3	27.8;15.3	27.8;15.3

Notes: In each cell the first number is the average of $\hat{k}$ and the second is the average confidence interval range for the true k.

----: indicates that no solution was found in the interval ; * indicates de best solution according to the proximity with k and the length of the average confidence interval range.

Table 2. Results for 40% overlapping

Similarity Interval (%)	CCC	PF	P T^2	CCC	PF	P T^2	CCC	PF	P T^2
	k=2 ; p=2			k=2 ; p=5			k=2 ; p=10		
60|-- 80	7.5; 7.6	8.1; 9.3	4.7; 3.5 *	11.3; 15.7	9.0; 10.1 *	10.2; 12.9	6.6; 2.5 *	11.8; 15.2	12.1; 15.6
80|-- 90	26.0; 19.4	30.1; 14.1	11.5; 17.5	16.3; 13.4	14.5; 26.8	21.7; 43.3	12.0; 15.8	22.7; 39.7	22.0; 35.5
90|--| 95	46.3; 5.1	67.8; 15.3	36.4; 16.6	43.0; 17.2	40.8; 13.2	73.1;16.2	21.7; 18.8	53.8; 12.7	68.6; 19.7
	k=5 ; p=2			k=5 ; p=5			k=5 ; p=10		
60|-- 80	14.5; 19.5	20.3; 21.4	6.4; 6.2 *	16.5; 16.9	5.1; 0.33 *	14.7; 30.2	11.3; 11.1	5.7; 12.2 *	15.3; 22.3
80|-- 90	46.5; 6.7	57.1; 15.8	27.1; 35.1	38.0; 19.6	129.7;17.7	54.5; 70.5	17.2; 11.9	144.4; 13.7	40.6; 71.1
90|--| 95	49.4;1.8	97.9; 14.7	62.5; 38.2	44.6; 11.5	154.8; 12.6	107.4; 49.2	43.2; 13.4	156.6; 12.9	115.2; 48.9
	k=10 ; p=2			k=10 ; p=5			k=10 ; p=10		
60|-- 80	8.8; 1.6 *	8.8; 1.6 *	7.2; 1.3	11.2; 4.5	11.2; 4.5	10.9; 4.1 *	13.8; 7.3 *	13.8; 7.2 *	13.8; 7.2 *
80|-- 90	17.1; 14.2	32.4; 31.9	11.8; 6.9	21.2; 24.4	12.5; 6.8	18.8; 24.8	18.2; 17.6	13.9; 8.6	19.6; 19.9
90|--| 95	87.4; 20.4	95.1; 24.6	40.6; 60.4	34.0; 16.8	98.8; 139	60.0; 103	20.0; 12.3	140.8; 127	44.7; 70.6
	k=20 ; p=2			k=20 ; p=5			k=20 ; p=10		
60|-- 80	15.2; 1	15.2; 1	12.1; 2.6	18.9; 0.19 *	18.9; 0.19 *	18.3; 0.78	18.9; 0.12 *	18.9; 0.12 *	18.8; 0.15 *
80|-- 90	21.2; 8.1 *	21.5; 8.8 *	16.9; 1.8	21.8; 2.3	20.5; 1.7 *	31.5; 26.6	27.8; 18.1	24.2; 10.5	28.9; 19.1
90|--| 95	62.5; 65.7	93.0; 41.8	28.6; 28.4	32.4; 28.4	32.4; 28.2	43.5; 68.4	30.9; 20.5	30.7; 20.3	51.3; 59.7

Notes: In each cell the first number is the average of $\hat{k}$ and the second is the average confidence interval range for the true k ;

* indicates de best solution according to the proximity with k and the length of the average confidence interval range.

6. EXAMPLES OF APPLICATION

In this section we will illustrate the application of bootstrap using two real examples involving a small and a moderate sample size. The classification variables are practically not correlated in the first example and correlated in the second.

Example 1. The data set is very well-known and presented in Hartigan (1975). It contains the nutrients in samples of 27 different types of meat, fish or fowl. The nutrients are: food energy (calories), protein (grams), fat (grams), calcium (milli grams) and iron (milli grams) and they are presented in percentage (the data is divided by the daily recommendable quantity and transformed in percentage). Some few outliers are observed (see Figure 3) in protein (clams canned), calcium (sardines, salmon and mackerel canned) and iron (clams raw and canned, beef heart). The bootstrap results are shown in Table 3. According to the discussion presented in section 4 we would expect to obtain better estimates for *k* in the [60,80) similarity interval. Now, the estimate of *k* will depend upon the choice of the stopping rule. *CCC* produces an estimate equals to 4 or 5. Pseudo *F* suggests a 80% confidence interval [7;12] and a average of 10 and pseudo T^2 resulted in an estimate in the 80% interval [5;9] and an average $\hat{k}$=6 or 7. The *CCC* had indicated the smallest partition ($\hat{k}$=5), pseudo *F* the largest ($\hat{k}$=9) and *pseudo* T^2 a value between these two rules ($\hat{k}$=7). *Pseudo F* resulted in the highest similarity average in the [60, 80) interval and *CCC* the lowest. The average confidence interval range were similar for *Pseudo F* and T^2 but *CCC* resulted in the smallest value. The partitions for $\hat{k}$=5,6 and 7 are shown in Table 4 and some descriptive statistics are shown in Table 5. The grouping produced by $\hat{k}$=6 and 7 makes sense in terms of nutrition aspects and also recognized very well the outliers of the data isolating them in clusters. It can be seen that the increase of the number of groups as suggested by *pseudo F* is not really necessary. As an example, the average nutrition aspects for $\hat{k}$=6 is described taking into account the maximum and minimum values observed for each nutrient as a reference for clusters comparison (see Table 5). All clusters have similar average content of protein. However, they differ in the other nutrients. Cluster 1 is richer in fat, energy and iron but has low value for calcium; Cluster 2 is rich in iron, has moderate values for fat and energy and low value for calcium; Cluster 3 has a moderate value for iron and low values for fat, energy and calcium; Cluster 4 is very rich in iron, has a moderate value for calcium and low values for fat and energy; Cluster 5 is rich in calcium, has a moderate value for iron and low values for fat and energy and finally Cluster 6 (sardine) is very rich in calcium, has low values for energy and fat, moderate value for iron and the highest value for protein.

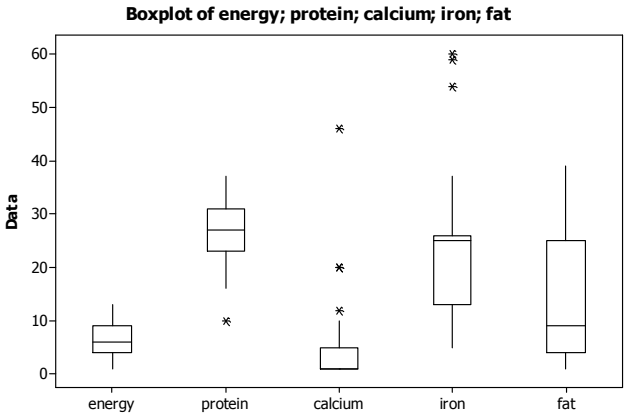

Figure 3. Box plot for each variable of the food data set.

Table 3. Bootstrap results for food example

Similarity Interval (%)	Lower 80% confidence limit	Upper 80% confidence limit	Similarity average	Average $\hat{k}$
CCC				
60 |--- 80	4	5	65.52	4.75
80 |--- 90	4	5	81.91	4.94
90 |---| 95	----	----	----	----
Pseudo F				
60 |--- 80	7	12	75.38	9.73
80 |--- 90	11	15	87.58	13.24
90 |---| 95	13	17	92.69	14.97
Pseudo T^2				
60 |--- 80	5	9	69.17	6.64
80 |--- 90	8	14	85.44	10.99
90 |---| 95	11	16	92.87	13.96

Table 4. Estimated partitions for k=5,6,7 - Food example

k=5
Cluster 1: Beef braised, beef roast, beef steak, lamb shoulder roast, smoked ham, pork roast, pork simmered. Cluster 2: Hamburguer, beef canned, bamb leg roast, beef tongue, veal cutlet. Cluster 3: Chicken broiled, chicken canned, bluefish baked, crabmeat canned, haddock fried,mackerel broiled, perch fried, tuna canned, salmon canned, mackerel canned, shrimp canned Cluster 4: Beef heart, clams raw, clams canned. Cluster 5: Sardines canned
k=6
Cluster 1: Beef braised, beef roast, beef steak, lamb shoulder roast, smoked ham, pork roast, pork simmered. Cluster 2: Hamburguer, beef canned, lamb leg roast, beef tongue, veal cutlet Cluster 3: Chicken broiled, Chicken canned, Bluefish baked, Crabmeat canned, haddock fried, mackerel broiled, perch fried, tuna canned. Cluster 4: Beef heart, clams raw, clams canned. Cluster 5: Mackerel canned, salmon canned, shrimp canned. Cluster 6: Sardines canned
k=7
Cluster 1: Beef braised, beef roast, beef steak, lamb shoulder roast, smoked ham, pork roast, pork simmered. Cluster 2: Hamburguer, beef canned, lamb leg roast, beef tongue, veal cutlet Cluster 3: Chicken broiled, chicken canned, bluefish baked, crabmeat canned, haddock fried, mackerel broiled, perch fried, tuna canned. Cluster 4: Beef heart. Cluster 5: Clams raw, clams canned. Cluster 6: Mackerel canned, salmon canned, shrimp canned. Cluster 7: Sardines canned

Table 5. Descriptive Statistics Food example – k=6 clusters

		Fat		Energy		Protein		Calcium		Iron	
Cluster	size	Mean	st.	Mean	st.	Mean	st.	Mean	st.	Mean	st.
1	7	30.14	4.45	11.14	1.21	26.57	2.70	1.00	0.00	24.14	2.11
2	5	14.00	4.64	6.80	1.09	29.80	2.59	1.20	0.44	28.40	4.88
3	8	5.25	3.81	4.62	1.06	28.13	6.01	2.12	1.35	10.38	3.74
4	3	2.33	2.31	2.67	2.08	21.00	14.18	7.00	4.36	57.67	3.21
5	3	5.00	4.00	4.00	1.00	26.67	5.51	17.33	4.62	17.00	9.54
6	1	9.00	0.00	6.00	0.00	31.00	0.00	46.00	0.00	25.00	0.00
min *	27	1.00		1.00		10.00		1.00		5.00	
max *	27	39.00		13.00		37.00		46.00		60.00	

* min and max are the minimum and maximum values observed for each nutrient in the whole data set.

Example 2. We explored the data set presented by Spiehler (1987) which contains 214 measurements of several types of glasses and it is available in the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html). The classification of types of glass is important for criminal investigation since at the scene of the crime, the glass left can be used as an evidence if it is correctly identified. Nine (9) continuous variables were considered: refractive index (RI), Sodium (Na), Magnesium (Mg), Aluminum (Al), Silicon (Si), Potassium (K), Calcium (Ca), Barium (Ba), Iron (Fe). Originally the data set had 6 types of glasses defined as: 1. building windows float processed; 2. building windows non float processed; 3. vehicle windows float processed; 4. containers; 5. tableware; 6. headlamps. Due to the high difference in variability the data was standardized. Figure 4 shows a plot of the scores of the two first principal components (JOHNSON and WICHERN, 2002) of the standardized data. The sample points are very spread out due to the presence of outliers. The bootstrap results are shown in Table 6. According to section 4 the best estimate for *k* is probably in the [60,80) interval. Due to large number of outliers observed in the data the estimates produced by *pseudo F* and *CCC* are too large as well the average

range of the confidence intervals produced for all rules. *Pseudo* T^2 was less affected by the outliers and resulted an estimate is the interval [1;27] with 80% of confidence and the average is $\hat{k}$=8 or 9. Is interesting to see that *pseudo* T^2 was the only stopping rule that produced a confidence interval that covered the number 6 (the number of glasses types). From Figure 4 it is easily seen that *pseudo F* and *CCC* had indicated non-appropriated values for *k* and that *pseudo* T^2 resulted in a more reasonable estimate. Except for *CCC* the estimate $\hat{k}$ increased highly for the similarity intervals other than [60,80).

Table 6. Bootstrap results for Glass example

Similarity Interval (%)	Lower 80% confidence limit	Upper 80% confidence limit	Similarity Average	Average $\hat{k}$
CCC				
60 |--- 80	33	42	76.40	38.13
80 |--- 90	40	42	85.46	38.42
90 |---| 95	32	42	91.54	41.34
Pseudo F				
60 |--- 80	34	53	77.47	42.42
80 |--- 90	67	88	89.14	77.05
90 |---| 95	104	120	94.70	112.20
Pseudo T^2				
60 |--- 80	1	27	71.96	8.44
80 |--- 90	5	55	84.66	28.78
90 |---| 95	19	87	91.77	49.92

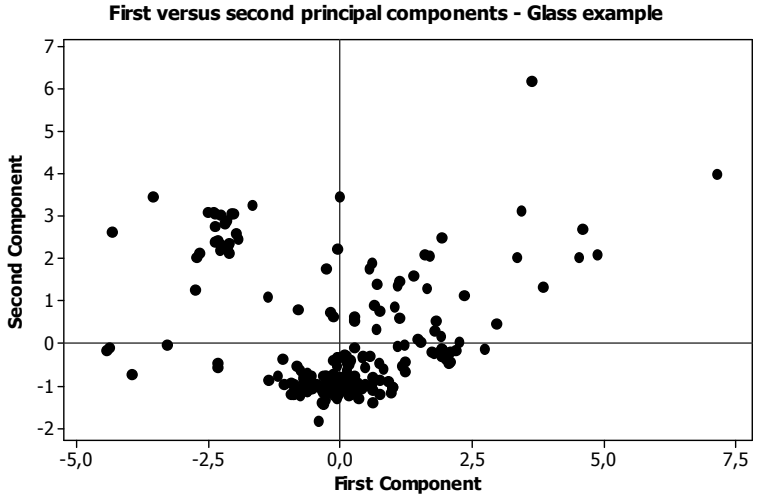

Figure 5. Scores of the two principal components of the glasses standardized data.

7. FINAL REMARKS

The results presented in this paper reinforce the difficulty in estimating accurately the number of clusters *k* of a data set. However, for hierarchical clustering algorithms, the right combination of the similarity interval and the statistical stopping rule makes it possible to obtain a reasonable estimate for *k* using bootstrap procedure. The results of the simulation study show that the stopping rules considered in this paper had similar performance. The quality of the estimates was affected by overlapping as expected but even in those situations it was still possible to obtain reasonable estimates for *k*, except for *k*=2. In general an increase in the number of groups *k* had more effect on the estimates than an increase in the

number of variables *p*. It is interesting to point out that the researcher should not look for a solution in a higher similarity interval ([90,95]), because the estimate of *k* will be probably larger than the necessary. The length of the confidence interval range was very affected by the presence of outliers in the data set.

Some of the results of this paper agreed with MILLIGAN and COOPER (1985) and ATLAS and OVERALL (1994) as far as the good performance of *pseudo F* is concerned, although the effects of the amount of overlapping in the stopping rules were not discussed in those two previous papers. The presented results also agreed with BOWMAN et al. (2004) who had also noticed the good performance of *pseudo* T^2 for noisy data.

This paper also showed that the bootstrap approach suggested in PECK, FISHER and VAN NESS (1989) is an interesting procedure and it can be used in a more general sense for any cluster algorithm as long as some criterion is well defined to pursue the optimal estimate of *k*. Other similarity measures, stopping rules and clustering algorithms can be considered. The bootstrap methodology is not just helpful to provide a point estimate for *k* but also to give an information about the stability of the solution by the observation of the confidence interval range.

ACKNOWLEDGMENTS

This research was partially financed by the Brazilian Institution CNPq and Universidade da Amazonia (Unama).

REFERENCES

ATLAS, R. S.; OVERALL, J. E Comparative evaluation of two superior stopping rules for hierarchical cluster analysis. **Psychometrika**, v. 59, n. 4, p. 581-591, 1994.

BOWMAN, D. F.; PATEL, R.; CHENGXING, L.. Methods for detecting functional classifications in neuroimaging data. **Humain brain mapping.**, v. 23. n.2, p.109-119, 2004.

CALINSKI, T.; HARABASZ, J. A. Dendrite method for cluster analysis. **Communication in** s**tatistics**, n. 3, p. 1-27, 1974.

DUDA, R. O.; HART, P. E., 1973. **Pattern recognition and scene analysis**. New York: John Wiley, 1973.

EVERITT, B. S. **Cluster analysis**. New York: John Wiley, 2001.

FRALEY, C.; RAFTERY, A. E. Model-based clustering, discriminant analysis,and density estimation. **Journal of american statistical association**, v. 97, n. 458, p. 611-631, 2002.

GORDON, A. D. **Classification**. London: Chapman and Hall, 1999.

GUO, P.; CHEN, C. L; LYU, M. L. Cluster number selection for a small set of samples using the bayesian Ying-Yang model, **IEEE transactions on neural networks**, v. 13, n. 3, p. 757-763, 2002.

HARTIGAN, J. A. **Clustering algorithms**. New York: John Wiley, 1975.

HRUSCHKAA, E. R.; CAMPELLOA, R. J. G. B.; CASTRO, L. N. Evolving clusters in gene-expression data. **Information sciencies**, v.176, n.13, p. 1898-1927, 2006.

JOHNSON, R. A.; WICHERN, D. W. **Applied multivariate statistical analysis**. New Jersey: Prentice Hall, 2002.

KATOSA, V. Network intrusion detection: Evaluating cluster, discriminant, and logit analysis. **Information sciences**, v.15, n. 1, p. 3060-3073, 2007.

KRZANOWSKI, W. J.; LAI, Y. T. A criterion for determining the number of groups in a data set using sum-of–squares clustering. **Biometrics**, v. 44, p. 23-34, 1988.

MARRIOTT, F. H. C. Practical problems in a method of cluster analysis, **Biometrics**, v.27, p. 501-514, 1971

MILLIGAN, G. W. An algorithm for generating artificial test clusters, **Psychometrika**., v. 50, n. 1, p. 123-127, 1985.

MILLIGAN, G. W.; COOPER, M. C. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. **Psychometrika**, v. 45, n. 3, p. 159-179, 1980.

_____. An examination of procedures for determining the number of cluster in a data set. **Psychometrika**, v. 50, n.2, p. 159-179, 1985.

MINGOTI, S. A.; LIMA, J. O. Comparing som neural network with fuzzy c-means, k-Means and traditional hierarchical clustering algorithms. **European journal of operational research**, n. 174, p. 1742-1759, 2006.

PECK, R.; FISHER, L.; VAN NESS, J. Approximate confidence interval for the number of clusters. **Journal of american statatistical association**, v. 84, n. 405, p. 184-191, 1989.

ROSENBERGER, C.; CHEDDI, K. Unsupervised clustering method with optimal estimation of the number of clusters: application to image segmentation, **Pattern recognition**, n.1. p. 656-659, 2000.

SARLE, W. S. Cubic clustering criterion. SAS Technical Report A-108, SAS Institute Inc., Cary, (full text is provided in www.sas.com/apps/pubscat/bookdetails.jsp?catid=1&pc=5903, 1983.

STEINLEY, D.; BRUSCO, M. J. Selection variables in cluster analysis: an empirical comparison of eight procedures. **Psychometrika**., v. 73, n.1, p. 125-144, 2008.

TEBOULLE, M.; BERKHIN, P.; DHILLON, I.; GUAN, Y.; KOGAN, J. Clustering with entropy-like k means algorithm. In Grouping Multidimensional Data. **In: Recent advances in clustering**, Ed. J. Kogan, C. Nicholas and M. Teboulle, 127-160, 2006.

TIBSHIRANI, R.; WALTHER, G.; HASTIE, G. T. Estimating the number of clusters in a data set via gap statistic. **Journal of royal statistics society B**, v. 63, part 2, p. 411-423, 2001.

XU, R.; WUNSCH, D. I. Survey of clustering algorithms. **IEEE Transactions on neural networks**, v.16, n. 3, p. 645-676, 2005.

WARD, J. H. Hierarchical grouping to optimize an objective function, **Journal of american statistical association**, 58, p. 236-244,1963.

Implementando Bootstrap no algoritmo de agrupamento de Ward para estimar o número de Clusters

Sueli A. Mingoti, sueliam@est.ufmg.br
Francisco N. Felix, fnfelix@unama.br

Universidade Federal de Minas Gerais (UFMG), Programa de Estatística
Belo Horizonte, MG, Brasil

*Recebido: Fevereiro, 2009 / Aceito: Agosto, 2009

RESUMO

Neste artigo apresentamos como a metodologia bootstrap pode ser implementada em métodos de agrupamentos hierárquicos como uma estratégia para estimar o número de grupos (k).O algoritmo de Ward foi escolhido como exemplo. A estimação de k é baseada num coeficiente de similaridade e em três regras de parada: pseudo F, pseudo T^2 e CCC. O desempenho do procedimento de estimação foi avaliado através de simulações de Monte Carlo considerando dados provenientes de variáveis correlacionadas e não correlacionadas e de grupos com e sem sobreposição. O procedimento de estimação discutido neste artigo pode ser utilizado com outros algoritmos de agrupamento e também para a escolha de soluções iniciais para uso em métodos de agrupamentos não-hierárquicos.

Palavras-Chave: Algoritmo de Ward. Estimação do Número de Grupos. Bootstrap.