Adv. in Fuzzy Sets & Systems **2**(1) (2007), 41-56

GINKGO, A PROGRAM FOR NON-STANDARD MULTIVARIATE FUZZY ANALYSIS

M. DE CÁCERES[*, †], X. FONT[†], F. OLIVA[*] and S. VIVES[*]

[*]Department of Statistics
Barcelona University
Barcelona, Spain
e-mail: mcaceres@ub.edu

[†]Department of Plant Biology
University of Barcelona
Barcelona, Spain

Abstract

Many multivariate programs are expensive commercial packages or require expensive third party software. Other applications are freely available to academic researchers but are limited to one operating system. This paper presents GINKGO, a free easy-to-use multi-platform Java application mainly oriented towards to non-standard classification analysis. This orientation is pursued (1) by providing users with several similarity and dissimilarity measures, (2) by allowing the execution of several crisp or fuzzy prototype-based clustering methods on arbitrary distance matrices, and (3) by including standard and non-standard facilities for evaluating the quality of clusters. Along with the software presentation, two methodological improvements are also given, which are available in the program. First, the implementation of a new

2000 Mathematics Subject Classification: 62H30, 62P10.

Keywords and phrases: classification, multivariate data representation, fuzzy clustering analysis, statistical software.

This work was supported in part by University and Research Awards Agency CGL 2006-03987(S.V).

Communicated by K. K. Azad

Received January 29, 2007

parameter initialization strategy for possibilistic C-means. As an example to illustrate both the methodological advancement and the capabilities of the program, a clustering analysis of human fibroblast DNA microarray expression data is presented. Second, a fuzzy generalization of the Rand and corrected Rand indices to allow the comparison of fuzzy partition matrices. GINKGO is available at the website http://biodiver.bio.ub.es/ginkgo, and software updates are automatically done via Java Web Start technology.

1. Introduction and Purpose

There are different commercial and non-commercial software packages and web applications available with implementations of different multivariate techniques, specially clustering methods. However, most implementations of the common prototype-based clustering methods (specifically: K-means and Fuzzy C-means) do not allow to use dissimilarity measures other than the Euclidean Distance. On the other hand, they sometimes lack facilities for evaluating the quality of the clusters obtained. Also most programs are expensive commercial packages or require expensive third party software. Finally, some are freely available to academic researchers but are limited to one operating system.

This paper is mainly concerned in the presentation of GINKGO, a new free easy-to-use multi-platform Java application mainly oriented to multivariate non-standard classification. This orientation is pursued in three ways: Firstly, by providing users with several similarity and dissimilarity measures. Secondly, by implementing algorithmic equivalents of well-known prototype-based clustering methods, such as K-means, Fuzzy C-means or Possibilistic C-means. Algorithmic equivalents of the usual methods allow the user to start them from arbitrary distance matrices, a rarely encountered option in other software packages. An analogous distance-based discriminant analysis method is also offered. Thirdly, by including standard and non-standard facilities for evaluating the quality of classification structures. All these capabilities are embedded into an integrated user interface framework that allows an easy and intuitive usage of the program.

Apart from presenting this software, our contribution is also devoted to the description of two improvements on fuzzy clustering methodology which have been implemented in the program:

- An improvement of a parameter initialization in Possibilistic C-means.

- A fuzzy generalization of the corrected Rand index for validating fuzzy partitions.

The rest of the paper is organized as follows. Section 2 describes the program's interface and capabilities. To start this section, we briefly describe the program's interface and data edition functions. We then mention most of the multivariate data analysis methods available, focusing on those clustering methods based on distance matrices. Afterwards, the implemented methods of comparison and evaluation of classification structures as well as plotting capabilities are listed. In Section 3 an effort is made to explain the two improvements of fuzzy-type clustering implemented in the program. In the final section we explain some technical issues.

2. Program Description

GINKGO is a program for the representation and classification of multivariate data. It is particularly well suited for academic labs and educational purposes, having several advantages over other analysis packages:

- The program is freely distributed and runs on multiple platforms (Windows, Mac OSX, Linux and Solaris), without need for compilation or specialized configuration.

- Installation is easy and updates are automatically done.

- The user interface is an intuitive easy-to-use integrated framework including three main windows: (1) a **Data Editor** to create, exchange and modify data matrices, either rectangular or symmetric; (2) an **Analysis Manager**, which stores all performed analysis, including; and finally (3) a **Graphic Editor** that keeps all plots, allowing printing or exporting.

- The user interface is designed to illuminate the analysis procedures and algorithms, encouraging the user to understand each step rather than running data through a “black box”.
- Flexible entry and exit points allow users to run GINKGO exclusively, or in conjunction with other programs. However, using a single program for all facets of data analysis greatly simplifies software management and user training.
- It allows data-mining complex data by hierarchically combining the execution of multivariate exploration tools (data representation and clustering methods) with the extraction of data subsets for subsequent analysis.

In the following subsections we describe the program’s main capabilities.

2.1. The Data Editor

The **Data Editor** window allows the creation, exchange, and modification of multivariate data matrices. Two matrix types are accepted: rectangular and symmetric. Symmetric matrices are usually created from a rectangular matrix and selecting among 9 different similarity or 13 different dissimilarity indices. Several file import options are available, including ASCII plain text using different value and decimal delimiter characters. Data matrices are exported by writing ASCII files or through the system clipboard. As with other statistical programs, univariate descriptive statistics and plots, as well as variable correlation and covariance matrices, can be obtained. The resemblance between two symmetric matrices can also be measured by calculating their matrix correlation value and significance level or through stress measures.

2.2. Multivariate analysis methods

Two kinds of multivariate analysis procedures are available in GINKGO. First, several multivariate data representation methods are offered (such as principal components analysis, metric and non-metric multidimensional scaling, correspondence analysis or canonical methods like redundancy analysis or canonical correspondence analysis). Second, user is provided with multiple classification options, including several clustering methods which can be run on both rectangular or arbitrary dissimilarity matrices. The results of all analysis are stored in the **Analysis**' **Manager** window and can be saved, along with input data matrices, for subsequent executions of the program.

Focusing on clustering, GINKGO allows the application of three different cluster models: hierarchical, partitioning, and cluster independent. The hierarchical clustering methods available include all the common agglomerative methods (Sneath and Sokal [17], Legendre and Legendre [13]): single linkage, complete linkage, UPGMA, WPGMA, UPGMC, WPGMC, Ward's minimum variance, and the β-flexible clustering, all of which can be executed using similarity or dissimilarity matrices as inputs. These hierarchical clustering methods yield ultrametric matrices, which are graphically depicted in the form of dendrograms. Typically, crisp partitions are obtained by "cutting" the dendrograms at the desired level of resemblance or into the desired number of groups.

The second clustering model aims at representing classifications as partitions of data. A data set of n objects is thus partitioned into a pre-specified number of clusters, c. Two common partitioning prototype-based clustering methods are available in GINKGO: classical K-means (KM, MacQueen [14]) and Fuzzy C-means (FCM, Bezdek [1]), which is a well-known generalization of K-means algorithm in the context of fuzzy logic. It is important to note that GINKGOs algorithmic implementation of both methods permits the use of arbitrary dissimilarity matrices as input, in addition to the standard rectangular matrices. This is made possible by computing the distance between an object and a cluster centroid as a function of inter-object distances, which avoids the need of the centroid (prototype) coordinates (Hathaway et al. [7]). Users are rarely allowed to use this distance-based equivalence in other multivariate software tools. To utilize these same methods for similarity matrices, users must first transform them into dissimilarity matrices inside the **Data Editor**. Additionally, corresponding versions of KM and FCM using medians instead of centroids as prototypes are also available (i.e., K-medians and Fuzzy C-medians).

Finally, the third classification model is cluster independent, signifying that each cluster is determined independently from the others. One clustering method that provides this classification model is Possibilistic C-means (PCM, Krishnapuram and Keller [11, 12]). This prototype-based clustering algorithm arose from a relaxation of the fuzzy partition concept. That is, in a common FCM partition each clustered object is limited to an overall membership of one. At the same time, the membership degree for a cluster is calculated by comparing the distance from the object to the cluster prototype with the distance to the other prototypes. Thus, membership values (sometimes called *probabilistic memberships*) are considered relative. In contrast, PCM clustered objects are not limited to a total membership of one and their membership degree can be considered absolute, sometimes referred to as "typicality degree". The membership degree an object exhibits for a PCM cluster i is obtained by comparing the distance from the object to the cluster prototype with a cluster reference distance (η_i). The cluster reference distance is a clustering parameter particular of each PCM cluster, related to cluster size. In short, PCM is a robust clustering method which can be used to identify *dense regions* in data (i.e., point-dense regions of the multivariate space). In Subsection 3.1 we describe an improvement of the initialization of η_i in PCM which has been implemented in the program.

When a previously reliable partition of objects is already available, users may be interested in the creation of discrimination functions to classify new data samples. Three discriminant analysis functions are available in GINKGO: canonical linear discriminant, quadratic discriminant, and distance-based discriminant (Cuadras et al. [3]). In distance-based discriminant analysis, a dissimilarity matrix is used as input, and discriminant functions are very much like the membership functions found in KM or FCM. In fact, distance-based discriminant analysis is the supervised learning counterpart of the distance-based versions of those clustering methods. It is worth noting that all discriminant analysis methods can work using either fuzzy or crisp training partitions.

2.3. Validation of clustering structures

The validity of a clustering structure can be expressed in terms of three criteria: internal, external or relative (Jain and Dubes [10]). Internal criteria not only assess the fit between the clusters and the original data, but may also be used to determine, for example, the number of groups to be sought. External criteria are used when matching a clustering structure to a priori external information (typically another classification structure). Finally, relative criteria determine which of two cluster structures is better in some qualitative or quantitative sense.

GINKGO incorporates two well-known indices that may be used as validating internal criteria: Calinsky-Harabasz [2] pseudo-F statistic, and non-parametric silhouette (Rousseuw [16]). The latter may also be used to detect individual object misclassifications. For the internal evaluation fuzzy partitions, users can also compute the partition coefficient (Bezdek [1]) or the fuzzy normalized entropy (Dunn [5]).

In order to evaluate crisp partitions using external criteria, the program provides the corrected Rand (Rand [15], Hubert and Arabie [8]) and Fowlkes-Mallows [6] indices. These indices can assess the level of agreement shown between two crisp partitions. To compare fuzzy partitions using these indices, they must first be defuzzified. Alternatively, GINKGO boasts a modification of the corrected Rand index which directly compares fuzzy matrices (see Subsection 3.2). Comparisons between hierarchical classifications (i.e., dendrograms) can be made by generating partitions at different cutting levels and then assessing their agreement with the aforementioned indices. In order to facilitate the validation by means of external criteria, external classification matrices can be imported into the **Analysis Manager** from ASCII text files. Note that those classification matrices, after import, could be also internally validated against the data set using the indices enumerated above.

2.4. Graphic Editor

Among other graphical outputs, GINKGO allows building 2D or 3D scatter diagrams from arbitrary data variables. Additionally, some multivariate data representation methods allow the creation of biplots. In order to depict the relationship between symmetric matrices, Shepard diagrams are offered. It is important to say that any available partition in the **Analysis' Manager**, whether fuzzy or crisp, may be used to label objects in a scatter diagram. It is therefore possible to "explore" the structure of multivariate data by combining the patterns revealed from both classification and representation methods. All plots are displayed in the **Graphics Editor** window, which permits editing some of their properties, exporting image files, and sending print jobs.

3. Two Improvements for Fuzzy Clustering

3.1. An improvement for PCM clustering

One of the main disadvantages of PCM is that it needs a good initialisation of the reference distance parameter (η_i) in order to provide accurate clustering results (Krishnapuram and Keller [12]). The usual way to initialise this parameter for a given cluster i is to make it proportional to the cluster variance V_i:

$$V_i = \left(\left(\sum_{j}^{n} u_{ij}^{m} \cdot e_{ij}^{2} \right) \Big/ \left(\sum_{j}^{n} u_{ij}^{m} \right) \right),$$

$$\eta_i = K \cdot V_i,$$

where e_{ij}^2 is the squared distance between the object j and the centroid of cluster i, u_{ij} is the membership of object j to cluster i, and m is the fuzziness exponent. Unfortunately, with the normal initialization of η_i sometimes the cluster size is overestimated, and thus a small cluster located beside a larger could be easily missed. A refined reference distance initialization strategy for PCM was suggested in (Cáceres et al. [4]). The new approach can be explained using the following rationale: For very small values, the size of the cluster is obviously underestimated in the dense region. Then, each increment of η_i provokes an increase in the possibilistic memberships, followed by an increase in cluster variability (i.e., V_i). However, when cluster growing reaches a less dense (or even empty) region, new increments in η_i do not include so many new objects and cluster variability progressively stops increasing. As soon as this low density region is stepped over and new external objects are about to be included, cluster variability increases again. Therefore, a heuristic criterion to provide suitable reference distance values is to search for those values which correspond to local minima of the partial derivative of cluster standard deviation with respect to η_i, that is

$$\delta Std_i / \eta_i = (\delta V_i / \delta\eta_i)/(2 \cdot Std_i),$$

where

$$\delta V_i / \delta\eta_i = \left(\left(\sum_j^n u_{ij}^m \cdot e_{ij}^2 \cdot \alpha_{ij} \right) \Bigg/ \left(\sum_j^n u_{ij}^m \right) \right)$$

$$- V_i \cdot \left(\left(\sum_j^n u_{ij}^m \cdot \alpha_{ij} \right) \Bigg/ \left(\sum_j^n u_{ij}^m \right) \right)$$

and $\alpha_{ij} = (m/(m-1)) \cdot \eta_i^{-1} \cdot u_{ij} \cdot (e_{ij}^2/\eta_i)^{1/(m-1)}$.

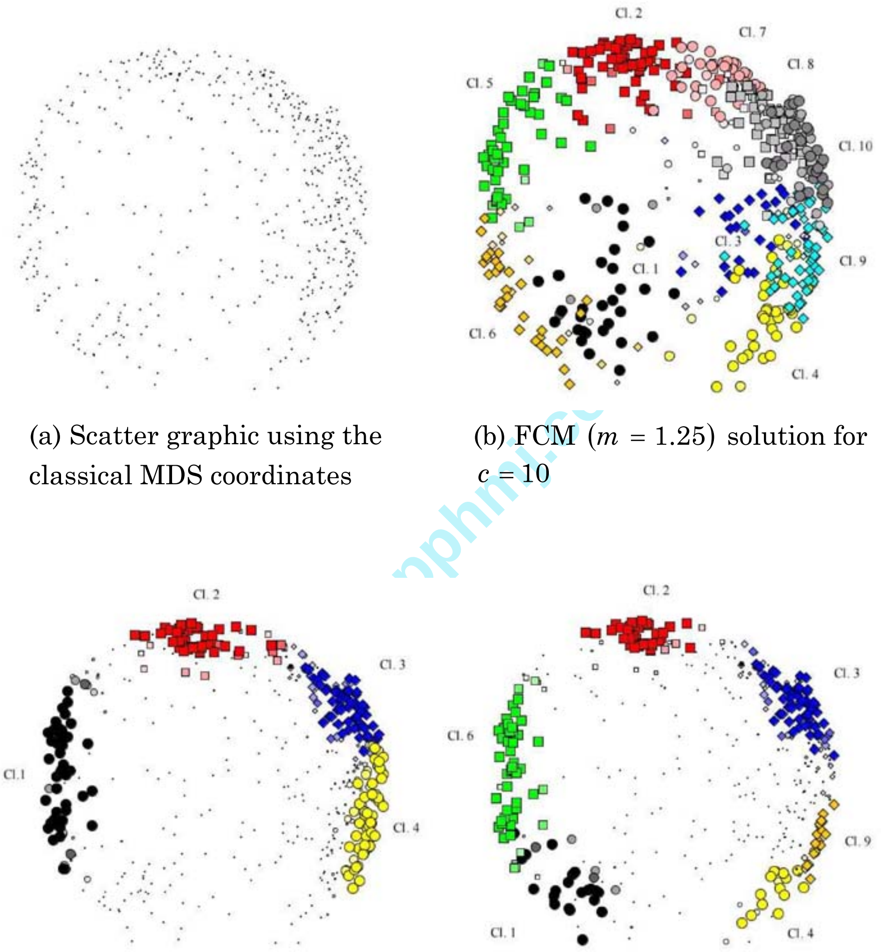
GINKGOs implementation of PCM incorporates both the common and new reference distance initialization strategy, which is as follows:

(1) Starting from a suitable initial membership matrix, calculate for each cluster the usual estimate of the reference distance.

(2) For each cluster, find the closest reference distance that yields a minimum in the cluster standard deviation, avoiding the trivial solutions (e.g., zero reference distance).

For illustrative purposes, we include here the clustering analysis of microarray data from a study of human fibroblast differential expression after serum addition (Iyer et al. [9]). This data set can be downloaded at http://www.sciencemag.org/feature/data/984559.shl. We chose for our analysis a subset of 517 genes which was studied in (Iyer et al. [9]). Our aim is to compare the performance of PCM, when using the usual or the reference distance initialization strategy suggested in (Cáceres et al. [4]). We will also show FCM results because this clustering method is normally used to provide starting cluster memberships for PCM.

To begin with, we computed the complementary dissimilarity of Pearson correlation coefficient between genes. In order to depict graphically the scatter of the dissimilarity matrix obtained, we performed a metric MDS, whose first two principal coordinates are shown in Figure 1(a). To start exploring the cluster structure of this data set, we first ran FCM on the dissimilarity matrix using $c = 10$ and $m = 1.25$. The relative membership matrix obtained is displayed in Figure 1(b). Note that the ten FCM clusters appear as segmentations of the global circular structure but not all of them correspond to dense regions of genes with correlated expression patterns. In our opinion PCM may be a useful tool to avoid the effect of loosely related genes (i.e., outliers) on the clustering solution of the others. We therefore ran PCM, using the ten FCM clusters as the starting configuration and with the fuzziness parameter set to $m = 1.2$. PCM cluster names keep the number of the FCM cluster from which the algorithm was initialized. PCM was run twice, as with the synthetic data examples, first the usual initialization of the cluster reference distances, and secondly initializing them with the strategy proposed in (Cáceres et al. [4]). Cluster standard deviation derivative profiles can be seen in Figure 2. The difference between cluster derivative values at the local minima can be interpreted as differences in cluster compactness and isolation. Some clusters (e.g., 8, 9 and 10) show only shallow minima. The reference distances used in each PCM run are signalled in Figure 2 with arrows for the first strategy and dots for the second. In seven clusters the reference distances resulting from the first method are higher than those of the proposed initialization strategy. While in both cases there is a certain amount of fuzzy overlap and inclusion, the classical reference distance initialization gives more cases of partial overlap, because cluster size is usually inadequately assessed. One can conclude that four main structures, i.e., clusters 1-4, are identified in this run. In contrast, using the proposed reference distance estimation substantially reduces the amount of cluster overlap, though there are still some cases of inclusion. Moreover, this time six distinct clusters (1-4, 6 and 9) can be recognized.



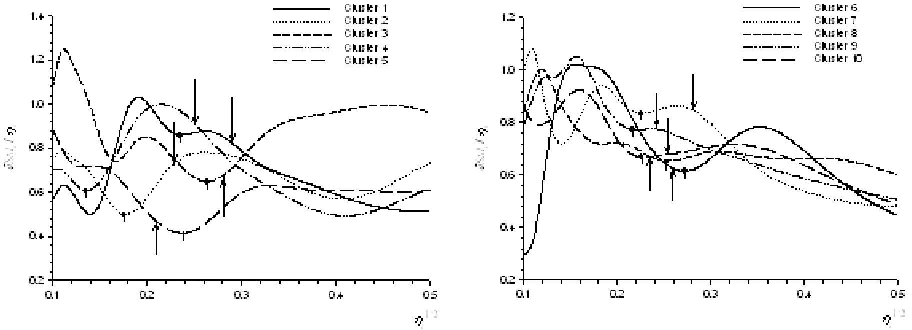
(a) Scatter graphic using the classical MDS coordinates

(b) FCM $(m = 1.25)$ solution for $c = 10$

(c) Four top-level clusters for the PCM $(m = 1.2)$ solution with initialization of using equation

(d) Six top-level clusters for the PCM $(m = 1.2)$ solution with initialization of finding the closest minimum

Figure 1. FCM and PCM solutions.



(a) (b)

Figure 2. Cluster standard deviation derivatives computed using different reference distances for FCM clusters 1-5 (a) and 6-10 (b). Reference distances used in the first PCM run are indicated by arrows and those used in the second run are indicated by dots.

3.2. A fuzzy generalization of the Rand index

In order to evaluate crisp partitions using external criteria, it is usual to make comparisons using the Rand index (Rand [15]) as well as its version corrected by chance effects (Hubert and Arabie [8]). Let **U** and **V** be two crisp partitions of the same set of n objects into c and c' clusters. The crisp memberships of objects to groups are indicated using values '1' and '0'. Now let $\mathbf{T}_{cxc'}$ be the confusion table where these two partitions are crossed (i.e., a cross-classification table). Each element $t_{ii'}$ contains the number of objects classified in group i of **U** and i' of **V**. Both the Rand and the corrected Rand indices can be expressed from matrix **T**:

$$\text{Rand}(\mathbf{T}(\mathbf{U},\ \mathbf{V})) = \frac{\binom{n}{2} + \sum_{i=1}^{c}\sum_{i'}^{c'} t_{ii'}^{2} - \frac{1}{2}\left(\sum_{i=1}^{c} t_{i.}^{2} + \sum_{i'=1}^{c'} t_{.i'}^{2}\right)}{\binom{n}{2}},$$

$$\text{Corrected Rand}(\mathbf{T}(\mathbf{U},\ \mathbf{V}))$$

$$= \frac{\text{Rand} - \text{Expected(Rand)}}{\text{Maximum(Rand)} - \text{Expected(Rand)}}$$

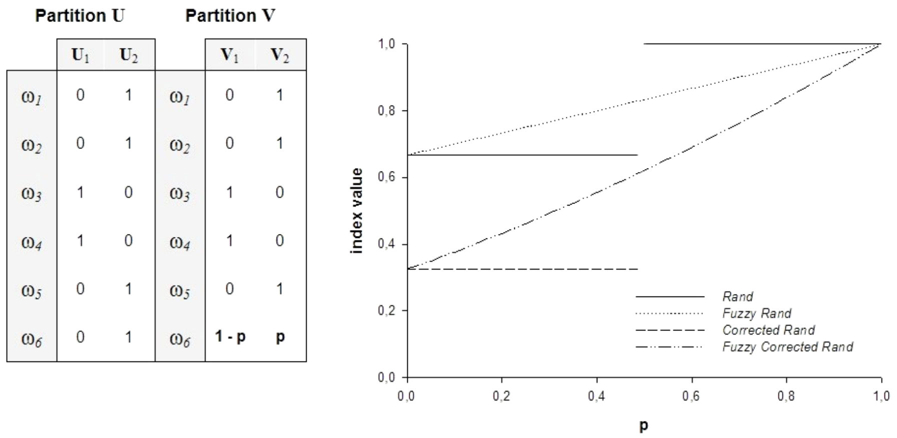
$$= \frac{\sum_{i=1}^{c}\sum_{i'}^{c'}\binom{t_{ii'}}{2} - \sum_{i=1}^{c}\binom{t_{i.}}{2}\sum_{i'=1}^{c'}\binom{t_{.i'}}{2}\Big/\binom{n}{2}}{\frac{1}{2}\left(\sum_{i=1}^{c}\binom{t_{i.}}{2} + \sum_{j=1}^{c'}\binom{t_{.i'}}{2}\right) - \sum_{i=1}^{c}\binom{t_{i.}}{2}\sum_{i'=1}^{c'}\binom{t_{.i'}}{2}\Big/\binom{n}{2}},$$

where $t_{i.}$ and $t_{.i'}$ are the sum of **T** row values and column values, respectively, and $\binom{x}{2} = x\,(x-1)/2.$

On the other hand and when dealing with fuzzy partitions, it is a common practice to defuzzify them (normally by choosing for each object the highest membership group) before applying crisp comparison indices. This additional defuzzification step implies a loss of information contained in the fuzzy memberships. Alternatively, GINKGO boasts a generalization of the Rand and corrected Rand indices which enables users to directly compare fuzzy partition matrices, thus avoiding the defuzzification step. This generalization is simply obtained by defining the confusion matrix **T** as the matrix product of **U** and **V**:

$$\mathbf{T}(\mathbf{U},\ \mathbf{V}) = \mathbf{U}'\mathbf{V} = \mathbf{V}'\mathbf{U}.$$

It can be easily seen that this product can be computed using both crisp and fuzzy partition matrices. When using **T** obtained with this matrix product in either the Rand or corrected Rand formulae does not change in the crisp partition case, but one obtains the corresponding fuzzy generalization of those indices in the fuzzy partition case. The following two figures exemplify the behavior of the fuzzy Rand and corrected fuzzy Rand indices. In Figure 3 two simple 6×2 partitions are compared. They only differ in the classification of the last object, whose memberships in partition **V** are made dependent on parameter p. The right graph of the figure shows the Rand and corrected Rand index values in both crisp and fuzzy versions, computed for different values of p. The extreme points correspond to the crisp case of matrix **V** and therefore crisp and fuzzy indices are equal. On the other hand, fuzzy comparison indices allow a continuous gradation of the comparison index values in the fuzzy case.



Partition U

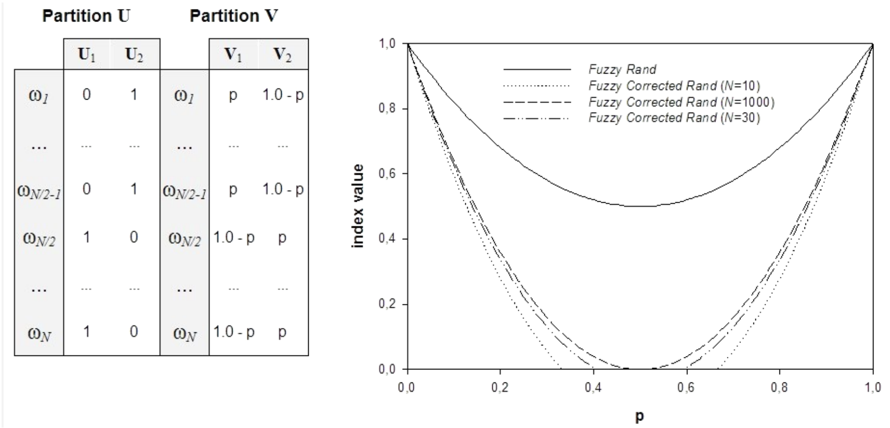
	U_1	U_2
ω_1	0	1
ω_2	0	1
ω_3	1	0
ω_4	1	0
ω_5	0	1
ω_6	0	1

Partition V

	V_1	V_2
ω_1	0	1
ω_2	0	1
ω_3	1	0
ω_4	1	0
ω_5	0	1
ω_6	**1 - p**	**p**

Figure 3. "Behavior" of the fuzzy Rand and corrected fuzzy Rand indices.

In Figure 4 matrices **U** and **V** have dimensions $n \times 2$. Again, the second matrix depends on the value of parameter *p*. Only the fuzzy versions of Rand and corrected Rand indices are shown. As expected, the lowest value for the (uncorrected) fuzzy Rand index is obtained for $p = 0.5$, i.e., when **V** is a completely fuzzy partition. In the case of the corrected fuzzy Rand index, when one partition (or both) are completely fuzzy the value of the index is zero, which indicates the lack of information in **V**. However, and like the crisp corrected Rand index does, the amount of penalization depends on the number of objects of the data set, because the amount of agreement due to chance effects is different.



Partition U

	U_1	U_2
ω_1	0	1
…	…	…
$\omega_{N/2-1}$	0	1
$\omega_{N/2}$	1	0
…	…	…
ω_N	1	0

Partition V

	V_1	V_2
ω_1	p	1.0 - p
…	…	…
$\omega_{N/2-1}$	p	1.0 - p
$\omega_{N/2}$	1.0 - p	p
…	…	…
ω_N	1.0 - p	p

Figure 4. Number of objects and "behavior" of the fuzzy Rand and corrected fuzzy Rand indices.

4. Technical Information

The program GINKGO is the statistical module of a software package for vegetation edition and analysis called *VEGANA*. While it has been developed within the field of numerical ecology, the program enables investigators to explore and analyze any kind of multivariate data with no limitation.

GINKGO is written in Java programming language. Thus, compiled code can be run under different operating systems (Windows, Linux, Mac, etc.). Any platform supporting Java Runtime Environment version 1.5.0 or higher is capable of for running it (http://www.java.com/). It is freely distributed and continuously improved. Installation instructions and sample data sets are available at the program's website: http://biodiver.bio.ub.es/ginkgo/. Once the program is downloaded and installed, subsequent updates are automatically done via Java Web Start technology (http://java.sun.com/products/javawebstart/). The minimum estimated hardware requirements estimated are a Pentium III processor and 256 MB of memory. The user's manual (in English) can be obtained from the web page in pdf format. The languages currently supported in the program are English, Spanish and Catalan.

References

[1] J. C. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[2] T. Calinski and J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1974), 1-27.

[3] C. M. Cuadras, J. Fortiana and F. Oliva, The proximity of an individual to a population with applications in discriminant analysis, J. Classification 14 (1997), 117-136.

[4] M. De Cáceres, F. Oliva and X. Font, On relational possibilistic clustering, Pattern Recognition (2006) (accepted for publication).

[5] J. C. Dunn, Indices of partition fuzziness and detection of clusters in large data sets, Fuzzy Automata and Decision Processes, Elsevier, New York, 1976.

[6] E. B. Fowlkes and C. L. Mallows, A method for comparing two hierarchical algorithms, J. Amer. Statist. Assoc. 78 (1983), 553-569.

[7] R. J. Hathaway, J. W. Davenport and J. C. Bezdek, Relational duals of the c-means clustering algorithms, Pattern Recognition 22 (1989), 205-212.

[8] L. Hubert and P. Arabie, Comparing partitions, J. Classification 2 (1985), 193-218.

[9] V. R. Iyer, M. B. Eisen, D. T. Ross, G. Schuler, T. Moore, J. C. F. Lee, J. M. Trent, L. M. Staudt, J. Hudson, M. S. Boguski, D. Lashkari, D. Shalon, D. Botstein and P. O. Brown, The transcriptional program in the response of human fibroblasts to serum, Science 283(5398) (1999), 83-87.

[10] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[11] R. Krishnapuram and J. M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Systems 1 (1993), 98-110.

[12] R. Krishnapuram and J. M. Keller, The possibilistic c-means algorithm: Insights and recommendations, IEEE Trans. Fuzzy Systems 4 (1996), 385-393.

[13] P. Legendre and L. Legendre, Numerical Ecology. Developments in Environmental Modelling, 2nd ed., Elsevier, 1998.

[14] J. MacQueen, Some methods for classification and analysis of multivariate observation, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281-297.

[15] W. M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Statist. Assoc. 66 (1971), 846-850.

[16] P. J. Rousseuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987), 53-65.

[17] P. H. A. Sneath and R. R. Sokal, Numerical Taxonomy. The Principles and Practice of Numerical Classification, W. H. Freeman, San Francisco, 1973.

■