Accepted for publication in the *Journal of Classification*

Optimal Variable Weighting for Ultrametric and Additive Trees and *K*-means Partitioning: Methods and Software

Vladimir Makarenkov
Université de Montréal
<makarenv@ere.umontreal.ca>

Pierre Legendre
Université de Montréal
<Pierre.Legendre@umontreal.ca>

Abstract: De Soete (1986, 1988) proposed some years ago a method for optimal variable weighting for ultrametric and additive tree fitting. This paper extends De Soete's method to optimal variable weighting for *K*-means partitioning. We also describe some new features and improvements to the algorithm proposed by De Soete. Monte Carlo simulations have been conducted using different error conditions. In all cases (i.e., ultrametric or additive trees, or *K*-means partitioning), the simulation results indicate that the optimal weighting procedure should be used for analyzing data containing noisy variables that do not contribute relevant information to the classification structure. However, if the data involve error-perturbed variables that are relevant to the classification or outliers, it seems better to cluster or partition the entities by using variables with equal weights. A new computer program, OVW, which is available to researchers as freeware, implements improved algorithms for optimal variable weighting for ultrametric and additive tree clustering, and includes a new algorithm for optimal variable weighting for *K*-means partitioning.

Undertaken at the suggestion of Professor Glenn W. Milligan, this research was supported by NSERC grant number OGP7738 to P. Legendre.

Authors' Addresses: Département de sciences biologiques, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada. V. Makarenkov is also associated to the Institute of Control Sciences, 65 Profsoyuznaya, Moscow 117806, Russia.

Résumé: De Soete a proposé il y a quelques années une méthode de pondération optimale des variables devant servir à la reconstruction d'arbres ultramétriques ou additifs. Notre article propose d'étendre cette méthode au partitionnement par la méthode des *K* centroïdes. Nous décrivons également des améliorations à l'algorithme de De Soete, ainsi que de nouvelles options de calcul. Des simulations de Monte Carlo ont été réalisées en utilisant différents types d'erreur. Dans tous les cas (i.e., arbres ultramétriques ou additifs, ou partition par la méthode des *K* centroïdes), les résultats des simulations indiquent qu'il est bon d'utiliser la méthode de pondération optimale des variables lors de l'analyse de tableaux de données susceptibles de contenir des variables-bruit qui ne contribuent que peu ou pas du tout à la classification. Cependant, si les données contiennent des variables pertinentes à la classification qui contiennent de l'erreur, ou encore des observations aberrantes, il est préférable de procéder à la classification en donnant des poids égaux à toutes les variables. Un nouveau logiciel, OVW, est mis gratuitement à la disposition des chercheurs désireux d'explorer la méthode. Ce programme met en oeuvre nos algorithmes améliorés pour la pondération optimale des variables pour la reconstruction d'arbres ultramétriques ou additifs; il permet également la pondération optimale des variables en vue du partitionnement par la méthode des *K* centroïdes.

Keywords: Additive tree; *K*-means partitioning; Optimal variable weighting; Ultrametric tree.

1. Introduction

In two pioneering papers, De Soete (1986, 1988) proposed a numerical method for estimating optimal weights for variables intended for ultrametric or additive tree reconstruction. The present paper extends De Soete's method to least-squares (*K*-means; MacQueen 1967) partitioning. We will also point out some properties of the algorithm proposed by De Soete that seem to have gone unnoticed; an understanding of these properties leads to improvements in the methods.

We carried out Monte Carlo studies for optimal variable weighting applied to additive tree reconstruction and *K*-means partitioning. These studies, conducted using different error conditions, confirmed the ability of the method to identify and reduce the effect of 'noisy' variables. We did not test the ability of the method for recovering clusters in the framework of ultrametric clustering procedures because a Monte Carlo study had already been carried out and discussed in detail by Milligan (1989). Considering the complexity of the algorithms that we discuss, a computer program is made available to the scientific community to encourage researchers to use optimal variable weighting.

There is an appreciable literature about variable weighting. DeSarbo, Carroll, Clark, and Green (1984) described SYNCLUS, a program that solves for both variable weights and produces *K*-means clustering. Fowlkes, Gnanadesikan, and Kettenring (1988) also proposed a method, here called FGK, for selecting weights — in that case, binary (0 and 1) weights. These authors proposed a model that selects subsets of variables from the original data and produces binary weights for the variables; their procedure was applied to complete linkage hierarchical clustering.

In a later paper, Gnanadesikan, Kettenring, and Tsao (1995) compared Fowlkes et al.'s (1988) FGK procedure to De Soete's OVWTRE and to DeSarbo et al.'s (1984) SYNCLUS models. Gnanadesikan et al. (1995) determined that the FGK forward selection procedure performed reasonably well compared to its competitors. Subsequent to the FGK algorithm, Carmone, Kara, and Maxwell (1999) proposed a variable subset selection method based on Hubert and Arabie's (1985) adjusted Rand index. Their method was designed for partitioning using continuous variables. The procedure proposed by Carmone et al. (1999) in the context of partitioning clustering, called HINoV, was described as a heuristic method based upon the adjusted Rand statistics. These authors conducted a series of Monte Carlo simulations, using synthetic data with noise of various kinds added, including masking variables. The results indicated that variables selected using the HINoV procedure outperformed the all-variable cases in 70 out of 72 different computer runs. In contrast to the good results found by Carmone et al. (1999), in real data set

analyses using HINoV, Green, Carmone, and Kim (1990) had earlier found mixed results in the ability of SYNCLUS to recover the correct variable weights.

Hubert and Arabie (1995) applied a least-squares optimization strategy to fit tree structures to symmetric proximity matrices among objects, using a heuristic optimization technique based on iterative projection. They considered extensions of this method beyond the analysis of a single symmetric proximity matrix. In this paper, we will explore how least-squares optimization can be applied to the analysis of two-way data matrices in the context of ultrametric and additive tree reconstruction as well as *K*-means partitioning.

2. Description of the Method

Given a rectangular (i.e., object-by-variable, or two-way, two-mode) data matrix **Y**, containing measurements of n objects on m variables, our algorithm computes weights $\mathbf{w} = \{w_1, w_2, \ldots w_m\}$ for the m variables such that the resulting matrix of predicted dissimilarities $\mathbf{D} = [d_{ij}]$ among objects, where

$$d_{ij} = \left[\sum_{p=1}^{m} w_p (y_{ip} - y_{jp})^2 \right]^{1/2}, \qquad (1)$$

optimally satisfies *either* (a) the ultrametric or (b) the additive inequality, or (c) optimally corresponds to a *K*-means partition with a fixed number of groups K. Equation (1) is the weighted form of the familiar Euclidean distance formula. The weights are constrained to be nonnegative with their sum equal to one.

The ultrametric inequality which defines dendrograms (Hartigan 1967) is satisfied when:

$$d_{ij} \leq \max(d_{ik}, d_{jk}) \qquad (2)$$

for all triplets i, j, and k, whereas the additive-tree inequality (four-point condition: Buneman 1974) is satisfied when:

$$d_{ij} + d_{kl} \leq \max\,(d_{ik} + d_{jl},\, d_{il} + d_{jk}) \tag{3}$$

for all quadruplets *i*, *j*, *k*, and *l*. The *K*-means partitioning problem can be defined as follows: Find a partition of *n* objects into *K* groups, or clusters, such that the sum, over all groups, of the sums of within-group squared distances to the centroids is minimum.

For each of the three clustering problems, a particular loss function (L) is defined to compute optimal weights. In the ultrametric case (dendrograms), optimal weights are found by solving the optimization problem as described by De Soete (1986):

$$L_U\,(w_1,\, w_2,\, \ldots\, w_m) = \frac{\sum_{\Omega U} (d_{ik} - d_{jk})^2}{\sum_{i<j} d_{ij}^2} \quad \rightarrow \quad \min, \tag{4}$$

where $\mathbf{\Omega}_U = \{(i, j, k) \mid d_{ij} \leq \min(d_{ik}, d_{jk})$, and $d_{ik} \neq d_{jk}\}$ denotes the set of ordered triplets for which the distances violate the ultrametric inequality (De Soete 1986). The minimization is done subject to the following constraints:

$$w_1,\, w_2,\, \ldots\, ,\, w_m \geq 0, \tag{5}$$

$$w_1 + w_2 + \ldots + w_m = 1. \tag{6}$$

In the case of additive trees, the optimization problem is also formulated as in De Soete (1986):

$$L_A\,(w_1,\, w_2,\, \ldots\, w_m) = \frac{\sum_{\Omega A} (d_{ik} + d_{jl} - d_{il} - d_{jk})^2}{\sum_{i<j} d_{ij}^2} \quad \rightarrow \quad \min, \tag{7}$$

subject again to constraints (5) and (6); $\mathbf{\Omega}_A = \{(i, j, k, l) \mid (d_{ij} + d_{kl}) \leq \min(d_{ik} + d_{jl}, d_{il} + d_{jk})$, and $d_{ik} + d_{jl} \neq d_{il} + d_{jk}\}$ denotes the set of ordered quadruplets for which the distances violate the additive inequality (De Soete 1986).

In the case of *K*-means partitioning, the minimization problem can be formulated as follows for a partition of *n* objects into a fixed number of clusters *K*:

$$L_P\,(w_1, w_2, \ldots w_m) = \sum_{k=1}^{K} \left[\sum_{i,j=1}^{n_k} d_{ij}^2 \right] / n_k \quad \rightarrow \quad \min, \tag{8}$$

subject to constraints (5) and (6); values d_{ij}^2 are the squared distances among objects in cluster k, and n_k is the number of objects in cluster k. The function L_P consists in the sum of the within-cluster sums of squared errors (the external sum in Equation 8), each one being computed as the mean of the squared distances among cluster's members (the internal sum in Equation 8).

We used the Polak-Ribière optimization procedure (see Press, Flannery, Teukolsky and Vetterling 1986, p. 303, and later editions, or Polak 1971, p. 53) to carry out the minimization of L_U, L_A and L_P. First, following De Soete (1986), we reduced the problem, which was originally formulated with constraints (5) and (6), to an unconstrained form, using the type of transformation of variables suggested by Gill, Murray, and Wright (1981, p. 270). The Polak-Ribière optimization method uses first partial derivatives of the functions L_U, L_A and L_P with respect to the introduced weights. It has proved successful in applications to unconstrained minimization problems; see Press et al. (1986, p. 277, and later editions).

When optimal variable weights have been obtained using L_U or L_A, the dissimilarity matrix **D** among objects can be computed using Equation 1 and subjected to any of the existing ultrametric or additive-tree fitting procedures; see, for example, Arabie, Hubert, and De Soete (1996, pp. 65-199) for an overview of existing fitting algorithms. Alternatively, matrix **D** can be subjected to *K*-means partitioning if optimization has been carried out using loss function L_P. *K*-means partitioning can

be computed from either a dissimilarity matrix or a rectangular data matrix; see for instance P. Legendre and L. Legendre (1998, p. 351). The latter option is the most commonly available in computer programs. There are two ways of passing the weights on to a *K*-means algorithm: (a) one can incorporate the weights into the calculation of distances and sums of squares in the *K*-means algorithm itself, as was done in Step 2.3 of the simulation procedure for *K*-means described in Section 5.3. Or (b), one can transform **D** into a rectangular object-by-variable matrix, preferably by metric scaling (also called principal coordinate analysis, Gower 1966), prior to *K*-means partitioning. Metric scaling is the only way of totally preserving the distance relationships among objects in the subsequent *K*-means procedure; nonmetric scaling would modify the distance relationships among objects.

The optimization methods described above may sometimes produce a local instead of a global minimum of L_U, L_A, or L_P. Hence, a good choice of initial weights is essential. While experimenting with our new program, we realized that making all weights equal to $1/m$ as an initial guess (where m is the number of variables), as implemented in the program OVWTRE, does not guarantee that the global minimum is always going to be reached. An interesting feature of our optimal variable weighting (OVW) program, compared to OVWTRE, is that it allows users to restart the optimization procedure any number of times, using different random initial configurations for the weights. As a consequence, OVW usually obtains better results than OVWTRE in the case of ultrametric clustering and additive tree reconstruction. Optimization for *K*-means partitioning, which is offered in program OVW, is not available in OVWTRE.

An important detail not reported in De Soete (1986, 1988) is that the global minimum of L_A or L_U can sometimes be reached with *several different sets* of optimal weights $\mathbf{w}$. This nonuniqueness may lead to different dissimilarity matrices $\mathbf{D}$, from which different clustering hierarchies or additive trees can be inferred.

Moreover, in the optimization for additive tree reconstruction, degenerate solutions, which are trivial, represent a pervasive problem. Such solutions, which consist in giving a weight of 1 to

any one of the variables and weights of 0 to all others, are frequently produced by De Soete's OVWTRE program. The theorem in Appendix 1 shows that any trivial solution of the type (1, 0, ..., 0), (0, 1, ..., 0), ..., or (0, 0, ..., 1) provides a perfect fit for the additive loss function L_A. In program OVW, we found a way of avoiding, where possible, this trivial solution which leads in most cases to a sub-optimal additive tree: users of the method can set a maximum value for the weight permitted for any single variable. This option effectively prevents obtaining a weight of 1 for a variable, which corresponds to a trivial solution. A numerical example in Section 4 illustrates how the program OVW works in practice.

An extensive Monte Carlo investigation of De Soete's variable weighting algorithm for hierarchical cluster analysis, based on results provided by De Soete's program OVWTRE, can be found in Milligan (1989). The simulations reported in the present study will focus on additive tree reconstruction and *K*-means partitioning.

3. Variants of the Optimization Problems Using Optimal Weights

Weights could be incorporated into distance coefficients other than the Euclidean distance. For instance, the Minkowski metric, which is a generalization of the Euclidean distance in which power 2 is replaced by an arbitrary positive power $r \geq 1$, and power 1/2 is replaced by $1/r$, may be weighted as for the Euclidean distance (Equation 1) to refine the computation of the dissimilarity matrix. Another case is Gower's (1971) general dissimilarity coefficient; weights w_p can be included in the coefficient either to handle the presence or absence of information ($w_p = 0$ when information about variable p is missing for one or the other object, or both; $w_p = 1$ when information is present for both objects) or to represent the importance to be given to the variables when estimating the dissimilarity (Gower 1971, Equation 5, P. Legendre and L. Legendre 1998, Equation 7.20). The development of an optimal variable weighting algorithm for the Minkowski metric or Gower's coefficient is an interesting topic for further investigation. As described above, partial derivatives of the of L_U, L_A, and L_P with respect to the weights have to be calculated for these dissimilarity coefficients. In the present paper, only the Euclidean distance is considered.

Variable weighting is not desirable in all cases. The theoretical foundations of a study may indicate whether differential weighting is warranted. Least-squares partitioning methods, such as *K*-means, consider the positions of the objects in Euclidean space, where they are divided into groups. If the Euclidean distances between pairs of objects are appropriate measures of the relative positions of the objects in variable space for the problem at hand, variables should not be differentially weighted prior to partitioning. For example, in ecological studies, when species abundance data have been transformed prior to clustering or partitioning using some appropriate transformation (e.g., those proposed by P. Legendre and Gallagher, in press), the transformed variables should not be differentially weighted using the present optimal variable weighting method. But in most other cases, when weighting is not specifically addressed by substantive theory, one may assume that some of the variables are noisy and should be eliminated or downweighted.

4. Numerical Example

To demonstrate the effectiveness of the OVW program, we carried out computations on the synthetic data considered by De Soete (1986) to illustrate the usefulness of his weighting procedure for ultrametric trees. De Soete's data, reported in Table 1, possess a clear predefined structure; the first two variables perfectly determine the separation of the objects into clusters. The three clusters {1, 2, 3, 4}, {5, 6, 7, 8} and {9, 10, 11, 12} can easily be deduced from the first two variables which have a clear partitioning structure. The values in variables 3 and 4 are uniform random deviates, unrelated to the other variables and, thus, should not be taken into account when creating the cluster structure, which should be based solely on variables 1 and 2. We will apply to this data set the variable weighting algorithm designed for *additive* and for *ultrametric* clustering as implemented in OVW; note that in his paper, De Soete (1986) only applied the optimal variable weighting procedure for *ultrametric trees* to this data set.

*** Table 1 here ***

First consider the case of the additive tree clustering. Results were produced by OVW using the following options: (a) the optimization procedure was restarted 10 times with different initial estimates; (b) to avoid a trivial solution when a weight of 1 was assigned to a single variable, the maximum allowed weight of a single variable was set to 0.9 (in fact, to force the program to skip a trivial solution, we could choose any other value smaller than 1). The following vector of optimal weights **w** was obtained: w_1=0.395, w_2=0.605, w_3=0.0, w_4=0.0; the value of the objective function L_A dropped from 0.329523 (when all weights were equal to 0.25) to 0.000007 (for the optimum weights). The correct additive tree structure effectively separating the three clusters could be found from the matrix of weighted distances provided by the program. For the same data set, De Soete's OVWTRE program failed to provide relevant results with the additive tree clustering option and produced only a trivial solution with w_1=0.0, w_2=0.0, w_3=1.0, w_4=0.0; the corresponding value of L_A was 0.

However, when OVWTRE was launched with the ultrametric clustering option, it was able to discover a good classification, finding the following set of optimal weights: w_1=0.558, w_2=0.439, w_3=0.000, w_4=0.003. Running the OVW program with the ultrametric clustering option provided a different set of optimal weights: w_1=0.708, w_2=0.292, w_3=0.000, w_4=0.000, which also led to the correct classification.

Finally, when OVW was run on the data from Table 1 using the *K*-means partitioning option, with a correct partition vector supplied to the program separating the 12 objects into 3 groups as (1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3), our *K*-means variable weighting procedure detected the 'noisy' variables in the data and assigned weights of zero to variables 3 and 4. The optimal weights assigned to variables 1 and 2 were respectively 0.906 and 0.094, after 10 starts of the optimization procedure using different initial random configurations for the weights, whereas the minimum value of the objective function L_P dropped from 1.815205 for all weights equal to 0.000000 for the optimal weights. When an incorrect classification vector (1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3) was supplied to OVW, the following weights were obtained for the four variables: w_1=0.909,

w_2=0.091, w_3=0.000, w_4=0.000; the minimum value of the objective function L_P corresponding to the solution was 0.937442. This value, which is remote from 0, indicated that the classification vector supplied to the program was not optimal.

The classification structures obtained for the data of Table 1 using optimal weights computed by OVW are depicted in Figure 1. The dendrogram is represented in Part A, the additive tree in Part B, and the *K*-means clusters in Part C of the Figure. In the dendrogram and the additive tree, the interior nodes are numbered 13 to 22.

*** Figure 1 here ***

5. Monte Carlo Studies

We carried out Monte Carlo simulations to identify the situations where the variable weighting algorithms would represent an advantage. We conducted extensive studies for the variable weighting algorithms in the context of additive tree reconstruction and *K*-means partitioning. We did not repeat the simulations published by Milligan (1989) for ultrametric clustering because the algorithm implemented in our OVW program is merely an improvement over that proposed by De Soete (1986, 1988) and used by Milligan (1989) for his simulations.

In contrast to additive tree reconstruction, the ultrametric clustering loss function L_U is not impaired by degenerate solutions consisting of assigning a weight of 1 to a single variable. However, L_U may possess several local minima. Milligan (1989) showed that the solutions obtained when using all equal weights as the starting set of weights in OVWTRE enabled good detection of 'noisy' variables. Unfortunately, the OVWTRE program proposes only one solution to the user, who receives no information from the program about the number of local minima and the combinations of weights that correspond to them. Using our OVW program, we conducted a brief exploration (not presented in detail) of the number and characteristics of the local minima that could be reached by L_U. When analyzing data sets that did not possess a clear cluster separation structure,

several local minima were usually found by OVW. Interestingly, each of the local minima usually identified the 'noisy' variables by assigning them weights of 0.

5.1 Additive Tree Simulations: Data Generation and Error Conditions

Our simulation study was designed as follows. The objective was to examine the ability of our additive tree weighting procedure to recover a variety of known underlying structures and to eliminate noise variables.

1. First, we generated random matrices **Y**(*n*x*m*) containing measurements of *n* objects on *m* variables. These matrices corresponded to an evolutionary process modeled by an additive tree. Matrices **Y** were of sizes: 8x4, 8x6, 8x8; 16x4, 16x10, 16x16; and 24x4, 24x14, 24x24. To obtain these random object-by-variable matrices, we first generated corresponding random additive binary trees using the algorithm of Pruzansky, Tversky, and Carroll (1982). In these trees, each leaf, or vertex of degree one, was associated with an object. Each tree was rooted using an internal vertex located in the center of the longest path of the tree.

Quantitative vectors of length *m*, where *m* is the number of variables, were then "evolved" along the trees, from the root and up, providing random realizations at each level of the tree up to the level of the leaves (objects). In each tree, we started with a sequence $\mathbf{v}_r$ of *m* 0's associated with the root. The values in vectors $\mathbf{v}_{11}$ and $\mathbf{v}_{12}$ associated with the root's two successors were obtained using the following formula: $v_{11,i}$ and $v_{12,i} = v_{r,i} + ran_i$, where ran was a vector of variables drawn from a random normal generator with mean zero. The standard deviation of each entry "ran_i" was equal to $10^{(1 - Level)}$, where the *Level* variable showed the level of the vertex under consideration, relative to the root. *Level* was equal to 1 in the case of $\mathbf{v}_{11}$ and $\mathbf{v}_{12}$, 2 for their successors, and so on. Thus proceeding, we attributed a particular sequence $\mathbf{v}_v$ to each vertex *v* of the tree. The set of sequences $\mathbf{v}_L$, associated with the set of *n* leaves *L* of the tree, formed the object-by-variable matrix **Y** of size *n* by *m*. This matrix was used in Step 2 (below). When required, different types of noise were added to **Y**, as described below.

Note that the farther we were from the root, the smaller was the difference in the pairs of sequences located side by side. The greatest amount of variability was generated among the sequences closest to the root. There were two reasons for using this type of additive trees in our simulations: first, this structure corresponds to the widely recognized biological fact that there is, in nature, more variability among higher taxa (e.g., orders and phyla) than among species or genera. Likewise, in a Euclidean ordination space, there is more variation among the centroids of the major groups than among the centroids of the smaller groups nested into them. Secondly, after a number of trials conducted with different types of trees, only the trees possessing the structure described above were properly reconstructed by the additive tree fitting methods. These methods took as input distance matrices **D** obtained from the object-by-variable matrices **Y** using all equal weights in Equation 1.

2. Given a rectangular data matrix **Y** containing a set of vectors corresponding to the leaves of a true tree *TT*, we computed optimal variable weights $\mathbf{w} = \{w_1, w_2, \ldots w_m\}$, using our algorithm OVW, such that the weighted matrix of dissimilarities among objects, $\mathbf{D}_w = [d_{ij}]$, defined by Equation 1 optimally satisfied the four-point condition. In parallel, we computed the dissimilarity matrix $\mathbf{D}_1$ using all weights equal in Equation 1. Then, we inferred additive trees T_w and T_1 from dissimilarities $\mathbf{D}_w$ and $\mathbf{D}_1$, respectively, by using each of the following tree fitting methods, as implemented in the T-REX software (Makarenkov and Casgrain 2000; Makarenkov 2001): ADDTREE by Sattath and Tversky (1977), Neighbor Joining by Saitou and Nei (1987), and the Method of Weights by Makarenkov and Leclerc (1999).

To assess the fit provided by each method, we used one topological and one metric criterion. We computed the value of the Robinson and Foulds (1981) topological distance between the true tree *TT* and the inferred trees T_w and T_1, as well as the value of the cophenetic correlation coefficient (Sokal and Rohlf 1962) between the true tree metric matrix **TT** associated with the true tree *TT* and the values of the tree metric matrices $\mathbf{T}_w$ and $\mathbf{T}_1$, associated with the inferred trees T_w and T_1, respectively. **TT** was obtained by computing by least squares the edge lengths along the

true tree *TT* corresponding to the dissimilarity values in $\mathbf{D}_1$; see Makarenkov and Leclerc (1999) for an overview of this technique.

3. As in Milligan's (1989) study on ultrametric clustering, we carried out simulations for 100 data sets with 6 different types of errors added to the initial object-by-variable matrix **Y**. The following error conditions (EC) were considered:

• *EC1: Error-free data.*

The first condition corresponded to the error-free data contained in matrix **Y**.

• *EC2: Inclusion in the data sets of 25% outliers.*

The second error condition involved replacing 25% of the real objects in the data set by outliers. A randomly selected object **y** from **Y** was replaced by an outlier whose values, denoted out_j (j = 1, ..., m), were obtained using the following formula: $out_j = y_j + ran_j$, where "**ran**" is a vector of variables drawn at random from a normal distribution with mean zero. The standard deviation of each entry in "**ran**" was equal to $10^{(2 - Level)}$ where, as above, *Level* showed the level of the object under consideration, relative to the root of the true tree *TT*. Hence, the standard deviation of a random variable added to an outlier was 10 times larger than the standard deviation of the replaced object. The outlier condition used in our study was different from that used by Milligan (1989) who *added* additional objects to the observed data matrices. One would expect the outliers to cause greater perturbation in our strategy than in Milligan's (1989) work. This strategy also provides greater comparability of the simulation results because the number of objects remains the same in all simulations of a series reported in Table 2.

• *EC3: Perturbation of the error-free coordinate values.*

The third error condition involved perturbing the error-free coordinate values. For y_{ij} representing the error-free coordinate value for object i on variable j, the error-perturbed value e_{ij} was computed as $e_{ij} = y_{ij} + 2ran_{ij}$ where "ran_{ij}" was a noise value drawn at random from a normal distribution with mean zero and standard deviation $10^{(1 - Level)}$; *Level* is the level of point i relative to the root of the true tree *TT*.

• *EC4-EC6: Addition of 1, 2 or 3 random noise dimensions (variables).*

Error conditions 4, 5, and 6 involved the addition of 1, 2, or 3 random noise dimensions to the basic variables which defined an additive tree structure in 4 to 24-dimensional space. The coordinates of the noise variables were drawn at random from a normal distribution with mean zero and the same standard deviation as in the error-free object-by-variable matrix **Y**.

5.2 Additive Tree Simulations: Results

Tables 2 and 3 report the mean values of the cophenetic correlation coefficients and the Robinson and Foulds (1981) topological distances obtained after 100 simulations, using the different types of error conditions described above. The maximum value of the cophenetic correlation coefficient, indicating maximum fit, is 1. The Robinson and Foulds distance corresponds to the number of bipartitions of the true tree which are not found in the inferred tree, plus the number of bipartitions of the inferred tree not found in the true tree. The maximum value of this distance between two binary trees with n leaves (representing n objects) is $2n–6$ in the case of different topologies, whereas the minimum value corresponding to topologically equivalent trees is 0. Actually, the Robinson and Foulds distances reported in Tables 2 and 3 were divided by the maximum value, $2n–6$, in order to provide a measure bounded in the interval [0, 1]. The cophenetic correlation (Cor) was computed between matrix **TT** and either $\mathbf{T}_1$ or $\mathbf{T}_w$, whereas the Robinson and Foulds distance (RF) was computed between the true tree TT and the inferred trees T_1 or T_w. The recovery values obtained by the tree fitting methods ADDTREE, NJ, and MW were very similar, according to the cophenetic correlation coefficient and the topological distance. The greatest difference among the three methods for the average cophenetic correlation, after 100 simulations, was 0.012 in the unweighted and 0.009 in the weighted case. Turning to the average Robinson and Foulds topological distance, the greatest difference was 0.071 in the unweighted and 0.066 in the weighted case. Thus, in Tables 2 and 3 we only report the results provided by NJ which is at the moment the most popular additive tree fitting algorithm.

*** Table 2 here ***

The first series of simulations were undertaken using error-free data, data with outliers, and error-perturbed data (see Table 2). The OVW program was run with the following parameters: the number of restarts (the number of different input configurations for the weights) of the variable weighting algorithm for each data set was set to 50, whereas the maximum weight value for any variable was set to 0.5. In the rare cases when OVW failed to provide a solution fulfilling the latter condition, we imposed a solution consisting of all weights equal. Examination of columns 3 and 4 of Table 2 shows that recovery of the additive tree structure was almost perfect when equal weights were assigned to all variables with error-free data. Recovery results in columns 5 and 6 correspond to the case where the optimal weighting algorithm was used on error-free data. There is a small decrease in recovery when the number of variables is high and the number of objects is small.

The following four columns of Table 2 provide information about the impact of error on the reconstruction of additive trees. In the columns corresponding to the presence of outliers, recovery for optimal weights dropped, compared to the equal-weight case. This finding was especially important when a large number of variables were considered. The results with the error-perturbed variables are reported in the last four columns of Table 2 and are similar to the results with outliers: the larger the number of variables, the worse the OVW results.

*** Table 3 here ***

The next types of data used in the study involved 1, 2, or 3 random noise variables which contributed no information to the additive tree structure. The mean recovery values for these conditions are presented in Table 3. There is a dramatic deterioration of the results for both criteria as the number of noise dimensions increases, compared to the error-free condition (Table 2). However, the optimal weights found by OVW allowed in most cases a significant improvement over the results obtained using equal weights. The topological improvement, measured by the Robinson and Foulds (RF) distance, is the most striking. The average gain in recovery for the topological distance across numbers of objects and variables in Table 3 obtained using optimal weights, compared to equal weights, was 0.263 for one noise dimension, 0.233 for two noise

dimensions, and 0.231 for three random noise dimensions. In most cases, OVW allowed recognition of noise variables by assigning them weights very close to 0. Of course, the trees constructed using optimal weights were not perfect, with sometimes as much as half their topology wrongly reconstructed (RF coefficients near 0.5, as in the results for 16 and 24 objects). However, the correctly reconstructed parts of the trees mainly comprised the edges located near the tree's roots, which is indeed the most informative part of a tree. Improvement from OVW, as measured by the cophenetic correlation (Cor), was large when the number of noise dimensions was large compared to the total number of variables; Cor was smaller when the noise dimensions represented but a small fraction of the number of variables.

The conclusions to be drawn from the results presented in Tables 2 and 3 are similar to those stated by Milligan (1989) for the ultrametric weighting procedure. First, the additive variable weighting algorithm should be used for analyzing data susceptible of comprising some noisy or masking variables. Second, if the data are perfectly error-free or involve lightly error-perturbed factors or outliers, no weighting should be used in the distance measure (Equation 1) prior to fitting an additive tree. However, if the data comprise compounded errors, for example outliers with some noise variables, the variable weighting technique is preferable.

5.3 *K*-means Simulations: Data Generation and Error Conditions

Another Monte Carlo study was conducted for the optimal weighting algorithm for *K*-means partitioning presented in Section 3. As in the case of the additive tree simulations, the presentation starts with an overview of the data generation strategy.

1. We generated random matrices $\mathbf{Y}(n \times m)$ containing measurements of n objects on m variables. Each data matrix defined a number of clusters in m-dimensional space. In the present study, matrices **Y** were of sizes: 10x2, 10x4, 10x6; 25x2, 25x4, 25x6; 50x2, 50x4, 50x6; 100x10; and 200x10. To generate the data, we used a modified version of the program developed by Milligan (1985) and later used by Milligan (1989) for generation of clustered data. We modified the source

code of Milligan's program, which is available on the Classification Society of North America's WWWeb site at URL <http://www.pitt.edu/~csna/Milligan/readme.html>, to make it possible to generate data containing the number of objects, variables, and clusters necessary for our simulations; the modifications did not imply any important change to Milligan's data generation procedure. Milligan's method allows the creation of 2 to 5 clusters in an *m*-dimensional Euclidean space. To ensure a minimum of separation, the clusters are designed to be nonoverlapping on the first dimension. Cluster boundaries can overlap along any or all the other variables of the space. As such, the generated error-free clusters possessed the properties of internal cohesion and external isolation and hence satisfied the definition of natural clusters as given by Cormack (1971), Everitt (1993, Chapter 1) and others.

2. Our optimal variable weighting algorithm applied to *K*-means partitioning requires as input an object-by-variable matrix **Y** as well as a vector of initial assignment of objects to clusters. Because in real-life situations we only possess the object-by-variable matrix and (usually) not the vector of object assignments, we imposed the same restriction on the input data in our simulations: we assumed that the vector of object assignments was unknown. To approach this issue we adopted the algorithmic strategy described below. Although this strategy does not guarantee optimal results, it implements the concept comparably to the analysis of real data sets. Partitioning was done using the program K-MEANS by P. Legendre (2000). This program implements a standard two-step alternating least-squares *K*-means algorithm which iterates between calculation of cluster centroids and assignment of objects to the centroids. At the beginning of an analysis, the objects are assigned at random to the clusters; the number of random assignments of the objects to clusters was fixed to 5 in the simulations. The programs allows users to search through different values of K in a cascade, starting with K_1 groups and ending with K_2 groups, with $K_1 \geq K_2$; $K_1 = 10$ and $K_2 = 2$ were used in the simulations. In the cascade from a larger to the next smaller number of groups, the two closest groups are identified and fused. Then the alternating least-squares algorithm is run until convergence, reallocating objects to the groups. For each number of groups (K), the Calinski-Harabasz (1974) pseudo-*F*-statistic was computed. We were interested in finding the number of

groups, *K*, for which the Calinski-Harabasz criterion was maximum; this value of *K* corresponded to the most compact set of groups. In a simulation study involving 30 stopping rules for cluster analysis, Milligan and Cooper (1985) found that the Calinski-Harabasz criterion was the one most often recovering the correct number of groups. The K-MEANS program can perform either unweighted or weighted optimization. In the latter case, the vector of weights associated with the variables can be supplied by the user. The simulation strategy was the following:

2.1. Run *K*-means partitioning on **Y**, as described above, with equal weights for all variables. Cluster membership for the number of clusters corresponding to the maximum value of the Calinski-Harabasz criterion is written out to vector $\mathbf{P}_1$.

2.2 Using **Y** and $\mathbf{P}_1$ as input parameters to the OVW program, compute the optimal variable weights $\mathbf{w} = \{w_1, w_2, \ldots w_m\}$ that minimize the *K*-means objective function L_p.

2.3. Compute a new *K*-means partition for **Y** using the vector of optimum weights **w** found in the previous step. Vector $\mathbf{P}_w$ describes the new group membership.

2.4. Vector **P*** describes the true cluster partition of the *n* objects among the *K* groups. This partition is specified by the data generation program. To assess the quality of the cluster recovery, we compare **P*** to $\mathbf{P}_1$, and **P*** to $\mathbf{P}_w$, using the corrected Rand index (Hubert and Arabie 1985).

The corrected Rand index measuring the agreement between two partitions was used as a primary numerical evaluation measure in a number of recent studies; see for example Milligan (1989) or Carmone et al. (1999). This index returns the value 1.0 if the two partitions are identical. Values near 0.0 correspond to the case where the match between partitions has fallen to chance level. Hence, larger values of the corrected Rand index point out a better recovery achieved by a clustering method.

3. We report mean cluster recoveries after 100 simulations, for data generated using six types of errors similar to those used in the additive tree simulations reported above. Because the error-free

data as well as the data affected by different types of error were provided by Milligan's data generation program, the reader is referred to Milligan (1989) for a detailed description of all error conditions. The conditions, compared to the additive tree simulations (above), were the following:

3.1. *Error-free data.* The first condition corresponded to the error-free data contained in matrix **Y**.

3.2. *Inclusion in the data sets of 40% of outliers.* The second error condition involved the inclusion in the data sets of 40% of additional points that were the outliers. An outlier was drawn from a normal distribution with a standard deviation three times larger than that of the given cluster.

3.3. *Perturbation of the error free coordinate value.* The third error condition consisted in adding a random standard normal deviate, multiplied by 2, to the error-free coordinate values.

3.4. *Addition of 1, 2, or 3 random noise dimensions.* Error conditions 4, 5, and 6 consisted of the addition of 1, 2, or 3 random noise dimensions to the basic variables which defined an additive tree structure in 2- to 10-dimensional space. For these dimensions, values were drawn at random from a standard normal distribution ($\mu = 0$, $\sigma^2 = 1$). The range of a random noise variable was then made equal to the range of the first dimension of the space for which cluster overlap was not allowed.

5.4 *K*-means Simulations: Results

*** Table 4 here ***

Table 4 reports the mean values of the corrected Rand index obtained after 100 simulations, using the different types of error conditions described above. The maximum value of the corrected Rand index is 1, indicating maximum fit. Strategies using all equal weights and the optimal weights found by OVW are compared. For the *K*-means simulations, OVW was run with the following parameters: the number of restarts (i.e., the number of different input configurations for weights) of the variable weighting algorithm for each data set was 10, whereas the maximum weight of any

single variable was set to 0.75. As in the additive tree simulations, we imposed a solution with all weights equal in the rare cases where OVW failed to provide a solution meeting the latter condition.

In the simulations carried out with error-free data, recovery was similar to using equal weights or the optimal weighting strategy (Table 4 and in Figure 2a). In contrast, for outliers and error-perturbed data, recovery for optimal weights dropped, compared to that for equal weighting.

*** Figure 2 here ***

With 1, 2, or 3 noise dimensions, recovery was much higher using optimal weights, compared to equal weights (Table 4 and in Figure 2b). The average gain in recovery across all simulation results presented in Table 4, using OVW optimal weights relative to equal weights, was 0.155 for one noise dimension, 0.152 for two noise dimensions, and 0.163 for three noise dimensions. In most instances, the optimal variable weighting procedure assigned weights very close to 0 to the noise variables.

The following important trend in recovery is observed across all six error conditions: the larger the number of objects or dimensions is in the object-by-variable matrix, the better is the cluster recovery. This result follows our expectations: each extra object which is not an outlier, and each extra variable, provide additional information to the clustering method and thus reduce the possibility of obtaining a wrong solution. This trend is particularly visible in the case of error-free data.

An interesting trend is found in Table 4: the larger the number of objects or variables in the data matrix, the better the cluster recovery, using either equal or OVW weights. On the other hand, putting the simulation results into graphs shows that recovery of partition structure stabilizes for $n \geq 50$ (Figure 2a) or $n \geq 100$ (Figure 2b), indicating that these simulation results are likely to be applicable to larger data sets. This result is important because, in most cases, real data sets comprise more than 100 objects.

From the results in Tables 4, we recommend using the optimal weighting algorithm for *K*-means partitioning for analyzing data that are likely to contain noisy variables not contributing relevant information about the real partition structure. However, if the data involve error-perturbed variables or outliers, it seems better to partition them by *K*-means using equal weights. For error-free data, equal or OVW weights can be used.

The performance of the variable weighting algorithm for *K*-means is likely to improve if, instead of a single classification (one vector of object assignments to the clusters, provided by the K-MEANS program), several classifications are used as input to OVW. An optimal strategy would consist in using as many classifications as possible as input to the OVW algorithm and selecting the solution that minimizes the *K*-means objective function L_p.

6. Discussion

In general, the optimal weighting algorithm should be used prior to ultrametric or additive tree clustering, or *K*-means partitioning, if one assumes that the data may contain irrelevant or noisy variables. When the data mostly include error-perturbed variables or outliers, we suggest processing such data using equal weights. Equal or optimal OVW weights can be employed when the data are supposed to be free of errors. The present paper extends to the case of additive trees and *K*-means partitioning the trends found by Milligan (1989) for ultrametric clustering.

It is very difficult to handle error-perturbed data, which is the most complicated case of error condition. As for the outlier condition, we would like to suggest a new strategy which could be tested through simulations. If the data being analyzed are likely to contain more noisy objects than noisy variables, the following strategy could be employed: instead of assigning weights to the variables, weights can be associated with the objects. Using a weighting function that assigns weights of 0 or 1 to the objects would lead to a new objective function to be minimized for ultrametric and additive trees as well as for *K*-means partitioning. Such a strategy may allow one to detect noisy objects rather than noisy variables; weights of 0 would be assigned to the noisy

objects. The resulting matrix of predicted dissimilarities **D** = [d_{ij}] among objects would be computed as follows:

$$d_{ij} = \left[\sum_{p=1}^{m} (v_i y_{ip} - v_j y_{jp})^2 \right]^{1/2}, \qquad (1')$$

where v_i and v_j are weights associated with the object *i* and *j*, respectively. Variants of the objective functions L_U, L_A and L_P should be considered: d_{ij} should be excluded from the objective function if v_i or v_j equal 0. A much more complicated model involving weights for both variables and objects may also be explored. Although the latter model would contain two sets of weights, it may allow one to reduce, at the same time, the effect of noisy variables and noisy objects or outliers. Investigation of weighting strategies implying weights for objects, or for both objects and variables, would constitute an interesting and relevant topic for future research.

Appendix 1: A Property of the Additive Loss Function

The following theorem states a property of the additive loss function L_A.

Theorem

Any single variable **y** *from an object-by-variable matrix* **Y** *having a weight of* 1 *defines an additive tree using the transformation described in Equation 1.*

Proof

We have to prove that assigning a weight of 1 to any one of the m variables and weights of 0 to the others always guarantees a perfect fit of the distance matrix to an additive tree, which means a value of 0 for the additive-tree loss function L_A of Equation (7). For convenience, assume that the weight corresponding to the first variable of an object-by-variable matrix $\mathbf{Y}(n \times m)$ is set to 1 and all others to 0. Let us consider any four entries of **Y** corresponding to this first variable. They will be denoted y_i, y_j, y_k, and y_l. Without loss of generality, we can suppose that $y_i \geq y_j \geq y_k \geq y_l$. As the weights of all variables except the first one are 0, then, from Equation 1, the following equations can be written for the corresponding distances: $d_{ij} = y_i - y_j$; $d_{il} = y_i - y_l$; $d_{ik} = y_i - y_k$; $d_{jl} = y_j - y_l$; $d_{jk} = y_j - y_k$; $d_{kl} = y_k - y_l$. Therefore, the term appearing in the numerator of L_A and associated with the quadruple of objects i, j, k, and l will consist of the difference between the two largest sums of two distances from among d_{ij}, d_{ik}, d_{il}, d_{jk}, d_{jl}, and d_{lk}. This term is the following: $((d_{il} + d_{jk}) - (d_{ik} + d_{jl}))^2 = ((y_i - y_k) + (y_j - y_l) - (y_i - y_l) - (y_j - y_k))^2 = 0$. Thus, any quadruple of objects i, j, k, and l of a single variable of **Y** will contribute a zero value to the sum appearing in the numerator of L_A. Consequently, any single variable of **Y** with a weight of 1 defines an additive tree distance using the loss function described in Equation 1. This tree can be represented graphically by a chain tree, i.e., a tree with all objects lying on the same axis. ♦

This theorem proves that any trivial solution of the type (1, 0, ..., 0), (0, 1, ..., 0), ..., or (0, 0, ..., 1) provides a perfect fit for the additive loss function L_A.

Appendix 2: A Program for Optimal Variable Weighting (OVW)

Program OVW performs optimal variable weighting for ultrametric and additive tree clustering, following the method proposed by De Soete (1986, 1988), as well as for least-squares (*K*-means) partitioning. The new program, which is available free of charge to academic researchers, provides improvements and extra options, compared to De Soete's (1988) program OVWTRE; the latter program only fits ultrametric and additive trees.

Input. The input data file is an ASCII text file which contains a data matrix **Y**(*n* x *m*) as well as the parameters *n* (number of objects) and *m* (number of variables). If the *K*-means partitioning option is selected, a vector of group assignments for the objects has to be provided in the same input file.

Output. The output consists of the weighted Euclidean dissimilarity matrix **D** of size (*n* x *n*) computed from **Y** using the optimum weights in Equation 1, the vector of optimal weights **w**(*m*) obtained using the Polak-Ribière minimization procedure, the minimum value of the objective loss function, and the number of iterations of the Polak-Ribière minimization that were needed to reach the optimal solution.

Language and computer. The current version of OVW is written in the C programming language. The program is distributed as freeware in a variety of formats: C source code for PC and Macintosh (the files are found in the folder “Source”) which can be compiled using a C/C++ compiler; compiled versions of the program for Win32-bit-compatible computers (OVW.exe); compiled version for PowerPC processors for Macintosh (file OVW_PPC); C source code for various versions of UNIX as well as the corresponding Make file.

Dimensionality and running time. There are no limitations to the size of matrix **Y**(*n*x*m*) in the program. The only existing limitation is the size of the random access memory (RAM) of the user’s computer. However, the Polak and Ribière optimization procedure uses the matrix of partial

derivatives ($\mathbf{D}_{ij}$ par $\mathbf{w}_k$). This intermediate matrix, which is repeatedly computed by the program, requires $O(m \times n^2)$ bytes for storage. For example, for an input matrix **Y** of size (100 x 100), the program requires about 4 MB of memory only to store the auxiliary matrix of partial derivatives. There are also some other auxiliary matrices and vectors occupying a substantial, but not so huge, amount of RAM. As to the running time, during the simulations involving a matrix **Y** with 300 objects and 166 variables, the program ran during approximately 4.5 hours on a Power Macintosh 604 at 350 MHz with 80 MB of RAM before providing a solution for the *K*-means partitioning problem; the optimization procedure was run only once for this problem.

Availability. Program OVW is freeware for researchers[1]. It is available via Internet on the WWWeb page of the Laboratory of Numerical Ecology at Université de Montréal:
<http://www.fas.umontreal.ca/biol/legendre/> or
<http://www.fas.umontreal.ca/biol/casgrain/en/labo/ovw.html>.

[1] This program has been developed as part of a university-based research program. Users who encounter problems with this program may report them to the authors who will be happy to help solve them. Researchers may use this program for scientific purposes, but the source code remains the property of Vladimir Makarenkov and Pierre Legendre (© 1999). Commercial users who want to use the program for profit should get in touch with the authors and pay royalties, or develop their own computer program based on the description of the method provided in this paper. Publications should give proper credit to the method by referring to this paper as well as De Soete's two papers. Users of program OVW may refer to the user's manual of Makarenkov and P. Legendre (1999).

References

ARABIE, P., HUBERT, L. J., and DE SOETE, G. (Eds.), (1996), *Clustering and Classification*, River Edge, New Jersey: World Scientific Publ.

BUNEMAN, P. (1974), "A Note on the Metric Properties of Trees," *Journal of Combinatorial Theory (B), 17,* 48-50.

CALINSKI, T., and HARABASZ, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics: Theory and Methods, A3,* 1-27.

*** Eva Whitmore: an acute accent has to be added on the "N" of "Calinski", here and everywhere it appears in the text. ***

CARMONE, F. J., KARA, A., and MAXWELL, S. (1999), "HINoV: A New Model to Improve Market Segment Definition by Identifying Noisy Variables," *Journal of Marketing Research, 36,* 501-509.

CORMACK, R. M. (1971), "A Review of Classification," *Journal of the Royal Statistical Society, Series A, 134,* 321-367.

DESARBO, W. S., CARROLL, J. D., CLARK, L. A., and GREEN, P. E. (1984), "Synthesized Clustering: A Method for Amalgamating Clustering Bases with Differential Weighting of Variables", *Psychometrika, 49,* 57-78.

DE SOETE, G. (1986), "Optimal Variable Weighting for Ultrametric and Additive Tree Clustering," *Quality & Quantity, 20,* 169-180.

DE SOETE, G. (1988), "OVWTRE: A Program for Optimal Variable Weighting for Ultrametric and Additive Tree Fitting," *Journal of Classification, 5,* 101-104.

EVERITT, B. S. (1993), *Cluster Analysis, 3rd edition*, London: Edward Arnold.

FOWLKES, E. B., GNANADESIKAN, R., and KETTENRING, J. R. (1988), "Variable Selection in Clustering," *Journal of Classification, 5,* 205-228.

GILL, P. E., MURRAY, W., and WRIGHT, M. H. (1981), *Practical Optimization*, London: Academic Press.

GNANADESIKAN, R., KETTENRING, J. R., and TSAO, S. L. (1995), "Weighting and Selection of Variables for Cluster Analysis," *Journal of Classification, 5*, 113-136.

GOWER, J. C. (1966), "Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis," *Biometrika*., *53*, 325-338.

GOWER, J. C. (1971), "A General Coefficient of Similarity and Some of Its Properties," *Biometrika*., *27*, 857-871.

GREEN, P. E., CARMONE, F. J., and KIM, J. (1990), "Variable Selection in Clustering," *Journal of Classification, 7*, 271-285.

HARTIGAN, J. A. (1967), "Representation of Similarity Matrices by Trees," *Journal of the American Statistical Association, 62,* 1140-1158.

HUBERT, L., and ARABIE, P. (1985), "Comparing Partitions," *Journal of Classification, 2*, 193-218.

HUBERT, L. and ARABIE, P. (1995), "Iterative Projection Strategies for the Least-Squares Fitting of Tree Structures to Proximity Data," *British Journal of Mathematical and Statistical Psychology*, *48*, 281-317.

LEGENDRE, P., and GALLAGHER, E. "Ecologically Meaningful Transformations for Ordination of Species Data," *Oecologia*, (in press).

LEGENDRE, P., and LEGENDRE, L. (1998), *Numerical Ecology, 2nd English ed.*, Amsterdam: Elsevier.

LEGENDRE, P. (2000), *Program K-means. User's guide*, Département de sciences biologiques, Université de Montréal. Available from the WWWeb site <http://www.fas.umontreal.ca/biol/legendre/>.

MacQUEEN, J. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," *in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Eds., L. M. Le Cam & J. Neyman, Berkeley: University of California Press, Vol. 1, 281-297.

MAKARENKOV, V. (2001), "T-REX – Reconstructing and Visualizing Phylogenetic Trees and Reticulation Networks", *Bioinformatics* (in press).

MAKARENKOV, V., and CASGRAIN, P. (2000), *T-Rex Package of Application Programs for Tree and Reticulogram Reconstruction*, Département de sciences biologiques, Université de Montréal. Available from the WWWeb site <http://www.fas.umontreal.ca/biol/legendre/>.

MAKARENKOV, V., and LECLERC, B. (1999), "An Algorithm for the Fitting of a Tree Metric According to a Weighted Least-Squares Criterion," *Journal of Classification, 16,* 3-27.

MAKARENKOV, V., and LEGENDRE, P. (1999), *OVW: Optimal Variable Weighting for Ultrametric and Additive Tree Clustering and K-Means Partitioning,* Département de sciences biologiques, Université de Montréal. Available from the WWWeb site <http://www.fas.umontreal.ca/biol/legendre/>.

MILLIGAN, G. W. (1985), "An Algorithm for Generating Artificial Test Clusters," *Psychometrika, 44,* 343-346.

MILLIGAN, G. W. (1989), "A Validation Study of a Variable Weighting Algorithm for Cluster Analysis," *Journal of Classification, 6,* 53-71.

MILLIGAN, G. W., and COOPER, M. C. (1985), "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika, 50,* 159-179.

POLAK, E. (1971), *Computational Methods in Optimization,* New York: Academic Press.

PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., and VETTERLING, W. T. (1986), *Numerical Recipes, The Art of Scientific Computing,* Cambridge, England: Cambridge University Press.

PRUZANSKY, S., TVERSKY, A., and CARROLL, J. D. (1982), "Spatial Versus Tree Representations of Proximity Data," *Psychometrika, 47,* 3-19.

ROBINSON, D.R., and FOULDS, L.R. (1981), "Comparison of Phylogenetic Trees", *Mathematical Biosciences, 53,* 131-147.

SAITOU, N., and NEI, M. (1987), "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees," *Molecular Biology and Evolution, 4,* 406-425.

SATTATH, S., and TVERSKY, A. (1977), “Additive Similarity Trees,” *Psychometrika, 42*, 319-345.

SOKAL, R. R., and ROHLF, F. J. (1962), “The Comparison of Dendrograms by Objective Methods,” *Taxon, 11*, 33-40.

Table 1. Synthetic data used by De Soete (1986, Table 1) illustrating the application of his optimal variable weighting procedure for ultrametric trees.

Objects	Variables			
	1	2	3	4
1	0.4082	0.000	0.0564	–0.0188
2	0.4082	0.000	0.7104	0.8879
3	0.4082	0.000	–0.5435	0.4931
4	0.4082	0.000	–0.0227	–0.6123
5	–0.2041	0.3536	0.6128	0.9475
6	–0.2041	0.3536	–0.7937	–0.7604
7	–0.2041	0.3536	–0.2072	–0.0368
8	–0.2041	0.3536	0.3818	0.1197
9	–0.2041	–0.3536	0.9152	0.3362
10	–0.2041	–0.3536	–0.6031	–0.9367
11	–0.2041	–0.3536	0.4861	0.2143
12	–0.2041	–0.3536	–0.3770	–0.0060

Table 2. Mean recovery values for error-free, with outliers, and error-perturbed data for additive tree reconstruction. For each case, the mean values (over 100 simulated data sets) of the cophenetic correlation (Cor) and the Robinson and Foulds (RF) topological distance are given; Cor = 1 and RF = 0 when there is perfect recovery. Trees obtained using all equal weights (Equal) for the variables are compared to trees obtained using the optimal weights found by our algorithm (Weighted).

		Error-free				Outliers				Error-perturbed			
No. of objects *n*	No. of variables *m*	Equal Cor	Equal RF	Weighted Cor	Weighted RF	Equal Cor	Equal RF	Weighted Cor	Weighted RF	Equal Cor	Equal RF	Weighted Cor	Weighted RF
8	4	1.000	0.010	0.975	0.043	0.844	0.095	0.837	0.126	0.759	0.105	0.768	0.154
8	6	1.000	0.015	0.915	0.121	0.847	0.118	0.780	0.181	0.748	0.094	0.739	0.218
8	8	1.000	0.005	0.918	0.151	0.881	0.127	0.790	0.228	0.705	0.105	0.680	0.249
16	4	1.000	0.011	0.992	0.038	0.940	0.040	0.938	0.063	0.871	0.113	0.876	0.125
16	10	1.000	0.003	0.940	0.105	0.906	0.024	0.845	0.134	0.757	0.092	0.681	0.180
16	16	1.000	0.002	0.928	0.105	0.918	0.017	0.824	0.146	0.751	0.093	0.685	0.173
24	4	1.000	0.004	0.995	0.031	0.970	0.026	0.968	0.049	0.913	0.091	0.905	0.105
24	14	1.000	0.002	0.952	0.105	0.970	0.011	0.910	0.128	0.750	0.105	0.689	0.166
24	24	1.000	0.000	0.959	0.120	0.976	0.007	0.922	0.125	0.707	0.121	0.651	0.180

Table 3. Mean recovery values for error-free data with 1, 2, or 3 added random noise dimensions for additive tree reconstruction. See Table 2 for the meanings of Equal, Weighted, Cor, and RF.

		1 noise dimension				2 noise dimensions				3 noise dimensions			
No. of objects *n*	No. of variables *m*	Equal Cor	Equal RF	Weighted Cor	Weighted RF	Equal Cor	Equal RF	Weighted Cor	Weighted RF	Equal Cor	Equal RF	Weighted Cor	Weighted RF
8	4	0.877	0.464	0.920	0.224	0.801	0.544	0.911	0.304	0.755	0.570	0.919	0.265
8	6	0.921	0.466	0.927	0.213	0.859	0.499	0.920	0.240	0.810	0.533	0.923	0.240
8	8	0.935	0.444	0.932	0.252	0.894	0.498	0.917	0.301	0.870	0.531	0.936	0.284
16	4	0.864	0.751	0.921	0.476	0.779	0.799	0.891	0.560	0.741	0.793	0.826	0.629
16	10	0.950	0.742	0.951	0.355	0.922	0.766	0.953	0.452	0.903	0.753	0.949	0.458
16	16	0.964	0.706	0.940	0.459	0.948	0.735	0.944	0.516	0.930	0.744	0.945	0.509
24	4	0.855	0.761	0.930	0.475	0.781	0.772	0.883	0.541	0.742	0.771	0.850	0.595
24	14	0.972	0.713	0.958	0.433	0.957	0.730	0.964	0.506	0.942	0.745	0.968	0.540
24	24	0.981	0.689	0.965	0.478	0.967	0.705	0.968	0.532	0.961	0.712	0.966	0.555

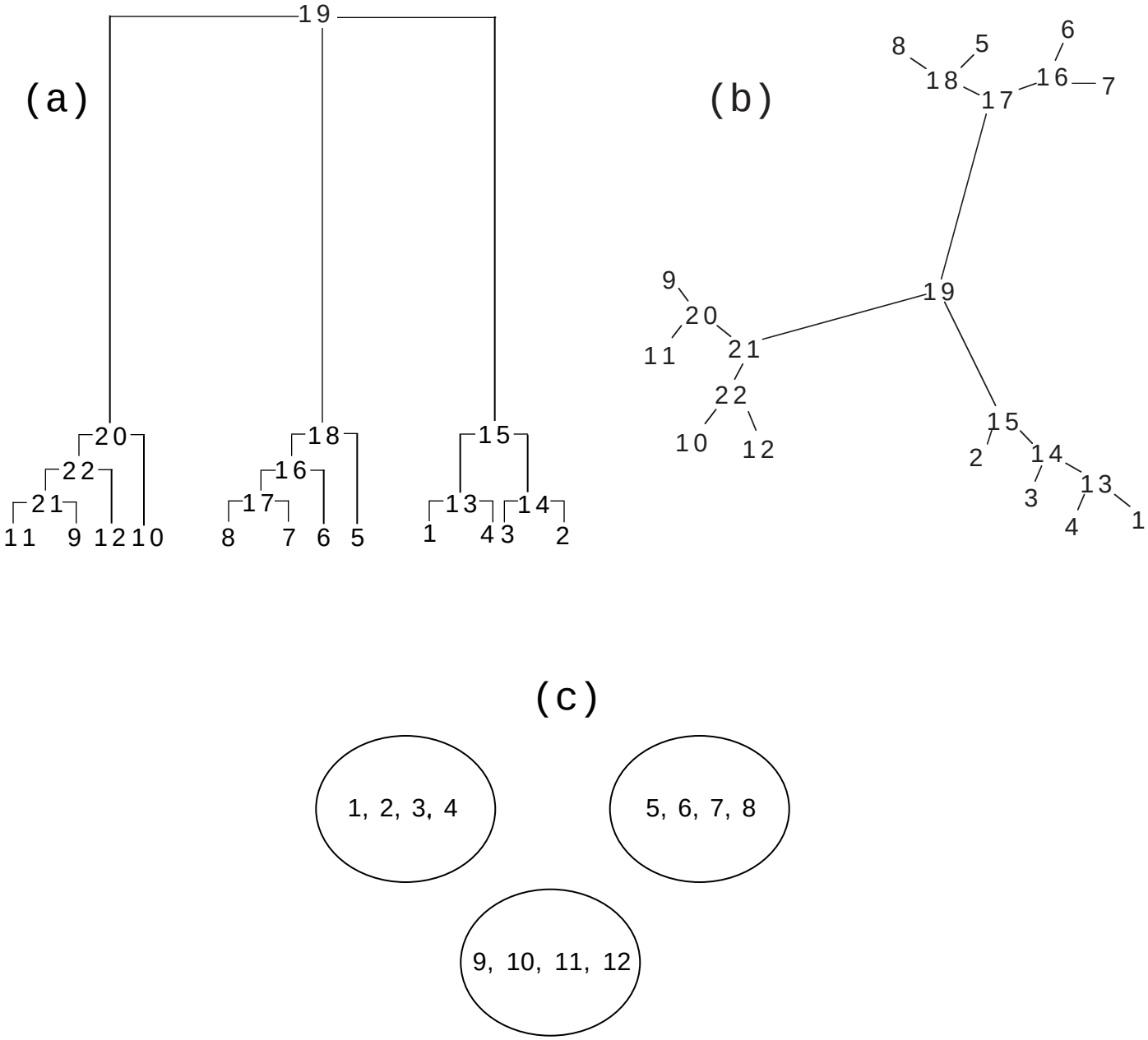
Table 4. Mean recovery values expressed using the corrected Rand index by Hubert and Arabie (1985) for *K*-means partitioning. The computations were carried out for error-free, with outlier, and error-perturbed conditions, and error-free data with 1, 2, and 3 added random noise dimensions. Partitions obtained using all equal weights (Eq.) for the variables are compared to partitions obtained using the optimal weights found by the OVW algorithm (We.).

		Error-free		Outliers		Error-perturbed		1 noise dim.		2 noise dim.		3 noise dim.	
No. of objects *n*	No. of variables *m*	Eq.	We.	Eq.	We.	Eq.	We.	Eq.	We.	Eq.	We.	Eq.	We.
10	2	0.839	0.842	0.776	0.774	0.653	0.652	0.562	0.728	0.443	0.619	0.394	0.534
10	4	0.926	0.904	0.879	0.858	0.772	0.726	0.658	0.774	0.575	0.738	0.518	0.688
10	6	0.962	0.952	0.912	0.873	0.817	0.820	0.723	0.814	0.661	0.750	0.669	0.778
25	2	0.817	0.787	0.835	0.821	0.670	0.648	0.486	0.676	0.391	0.596	0.409	0.573
25	4	0.947	0.895	0.887	0.824	0.801	0.767	0.662	0.829	0.597	0.787	0.452	0.666
25	6	0.937	0.933	0.888	0.790	0.864	0.753	0.752	0.878	0.696	0.803	0.579	0.790
50	2	0.916	0.914	0.850	0.839	0.695	0.689	0.504	0.753	0.432	0.666	0.424	0.653
50	4	0.960	0.937	0.877	0.894	0.837	0.774	0.631	0.871	0.607	0.819	0.542	0.767
50	6	0.979	0.935	0.926	0.853	0.892	0.770	0.759	0.899	0.740	0.843	0.731	0.841
100	10	0.960	0.975	0.934	0.827	0.942	0.823	0.868	0.935	0.900	0.941	0.806	0.861
200	10	0.979	0.963	0.937	0.886	0.926	0.891	0.851	0.940	0.923	0.935	0.798	0.806

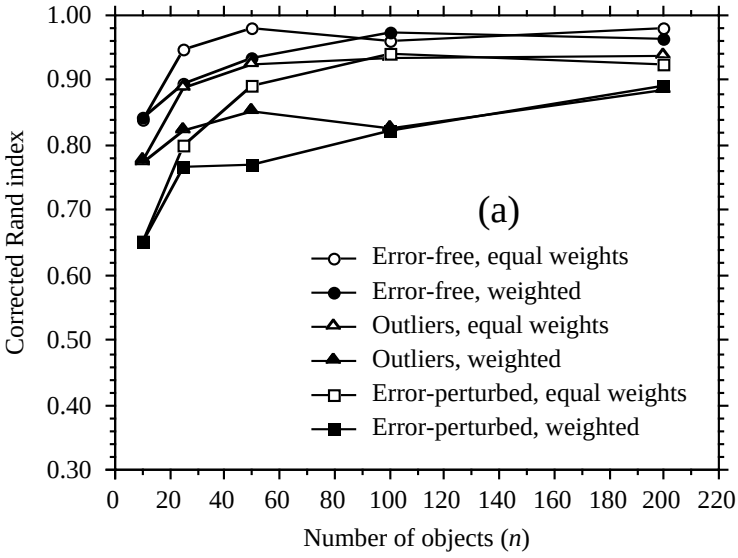
Figure caption

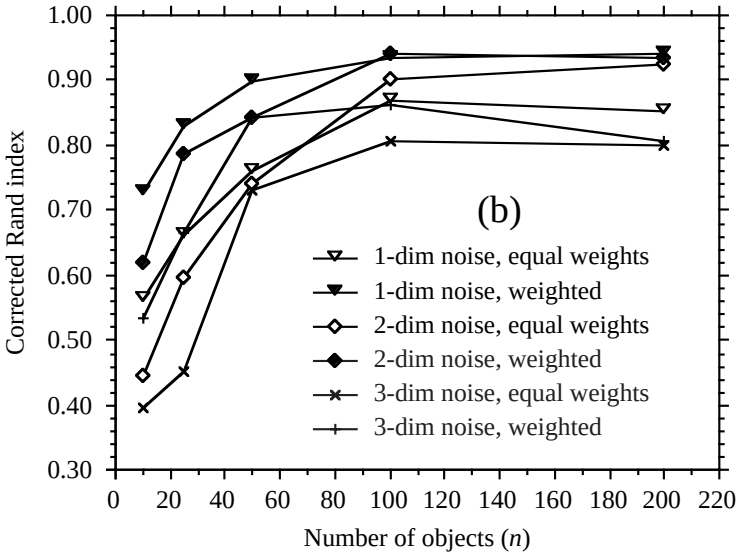
Figure 1. Classification structures obtained for the Table 1 data using optimal weights computed by OVW. (a) Dendrogram from ultrametric clustering; (b) additive tree; (c) *K*-means partition.

Figure 2. Comparison of results using equal weights to those using the optimal weights found by OVW: partition recovery, measured by the corrected Rand index, as a function of the number of objects. The following lines from Table 4, selected because $n \geq 5m$, are plotted: (n = 10, m = 2), (n = 25, m = 4), (n = 50, m = 6), (n = 100, m = 10), and (n = 200, m = 10), where n is the number of objects and m is the number of variables.



Makarenkov & Legendre, Figure 1





Makarenkov & Legendre, Figure 2