Model-Based Classification with Dissimilarities : a Maximum Likelihood Approach

E.-P. Ndong Nguéma[1] Guillaume Saint-Pierre[2] [3]

September 4, 2007

[1]Laboratoire de Mathématiques et Analyse des Systèmes, Ecole Polytechnique, P.O. Box 8390, Yaoundé, Cameroon. Tel: +237 22 45 47; Fax: +237 22 45 47. e-mail: ep_nd_ng@yahoo.com. *Corresponding Author.*

[2]LIVIC, Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs, Unité mixte INRETS-LCPC, Bâtiment 824, 14, route de la Minière, Satory, 78000 Versailles, France. Tel : +33 (0)1 40 43 29 33; e-mail: saintpierre@lcpc.fr.

[3]This work was started when the authors where at the Université de Paris-Sud at Orsay, the first author as a visiting researcher through a scholarship of the French Cooperation, and the second in a postdoctoral position funded by the Institut National de la Recherche en Informatique et en Automatique (INRIA).

Abstract

Most of classification problems concern applications with objects lying in an Euclidean space, but, in some situations, only dissimilarities between objects are known. We are concerned with supervised classification analysis from an observed dissimilarity table, which task is classifying new unobserved or *implicit* objects (only known through their dissimilarity measures with previously classified ones forming the *training data set*) into predefined classes.

This work concentrates on developing *model-based classifiers* for dissimilarities which take into account the measurement error w.r.t. Euclidean distance. Basically, it is assumed that the unobserved objects are unknown parameters to estimate in an Euclidean space, and the observed dissimilarity table is a random perturbation of their Euclidean distances of gaussian type. Allowing the distribution of these perturbations to vary across pairs of classes in the population leads to more flexible classification methods than usual algorithms. Model parameters are estimated from the training data set via the Maximum Likelihood (ML) method, and allocation is done by assigning a new implicit object to the group in the population and positioning in the Euclidean space maximizing the conditional group likelihood with the estimated parameters. This point of view can be expected to be useful in classifying dissimilarity tables that are no longer Euclidean due to measurement error or instabilities of various types. Two possible structures are postulated for the error, resulting in two different model-based classifiers. First results on real or simulated data sets show interesting behavior of the 2 proposed algorithms, ant the respective effects of the dissimilarity type and of the data intrinsic dimension are investigated. For these latter two aspects, one of the constructed classifiers appears to be very promising. Interestingly, the data intrinsic dimension seems to have a much less adverse effect on our classifiers than initially feared, at least for small to moderate dimensions.
Keywords: *dissimilarity data, model-based classifier, maximum likelihood estimate, intrinsic data dimension, success classification rate, multidimensional scaling.*

1 Contribution and Originality

Traditional classification problems concern applications with objects lying in an Euclidean space, but, in some situations, only some type of pairwise dissimilarity measure between objects is available. Today, practical applications are numerous (e.g. [3, 8, 11, 29, 30, 31, 32, 41, 42]). Moreover, using dissimilarity measures can be of much interest to analyze proximity between curves or objects in high dimensional spaces, or, more generally, between objects of complicated intrinsic structure.

None of the existing algorithms for dissimilarity data classification is based on standard principles of statistical inference. Moreover, apart from [38], they do not take into account measurement error in the dissimilarities. Therefore, they are more suited to classify dissimilarity tables that result from exactly computed pairwise distances between objects in a data set that were originally given through attributes in an Euclidean space. They are seldom recommended for coarser or noisy dissimilarity types, whereas real world dissimilarity data quite often fall in that category.

In the work presented here, we develop a new approach based on the purely statistical viewpoint of *Multidimensional Scaling (MDS)* [40]. Although interesting alternatives exist, MDS remains today the leading mathematical methodology for handling dissimilarity data. We are concerned with classification analysis from a table of such observed data from an otherwise non observable population of objects. The main goal is to assign a new object to one of *a priori* groups in the population using as only information its dissimilarities with previously classified ones which thus form the *Training Data Set*. Basically, our approach to solving this problem assumes a probability model in which the observed dissimilarities are Euclidean distances perturbed with random gaussian errors. Actually, two possible probability models are investigated, differing by the structure they postulate for the perturbation of the Euclidean distances which led to the observed dissimilarities. In each of these models, the unobserved objects are regarded as unknown parameters lying in an Euclidean space. Estimating these parameters in the statistical sense is thus equivalent to positioning the unobserved objects in the Euclidean space given their respective pairwise dissimilarities, which is the traditional MDS main concern. Such a model-based approach then has the advantage to simultaneously estimate objects positioning and group labeling. Since it is unknown, the dimension p of the Euclidean space shall serve as a tuning parameter to be estimated from the data by cross validation and the "within one standard error of the minimum error towards model parsimony" rule.

Each of the two probabilistic models so postulated for dissimilarity data allows us to derive a general purpose classifier for such data. The two constructed classifiers are nicknamed M1.BC and M2.BC respectively. The latter (M2.BC) exhibits high flexibility to adapt to the dissimilarity type at hand in the data. Its classification performance on some classical data sets appears comparable to that of some of the already available best classifiers on dissimilarity data.

2 Introduction

Traditional classification problems concern applications with objects lying in an Euclidean space, but, in some situations, only some type of pairwise dissimilarity measure between objects is available. Today, practical applications are numerous. Without attempting to be exhaustive, the following domains often make use of dissimilarity tables in analyzing or classifying collected data: psychology, sociology, signal processing, pattern recognition, document databases, detection of biomedical patterns [3, 30], natural language processing [29], classification of spectra [32], detection of abnormal events in computer networks [8], image indexing and retrieval [11], designing of recognition systems for vocally impaired persons [42], face authentication systems [41], or classification of seismic signals [31]. Moreover, using dissimilarity measures can be of much interest to analyze proximity between curves or objects in high dimensional spaces, or, more generally, between objects of complicated intrinsic structure. In those situations, specification of features to properly represent objects can be quite problematic, and moving from poor given features to dissimilarities may improve the performance in the classification task [10, 30, 35, 36].

Paradoxically, more work seems to have been done for (true) dissimilarity data in the area of *clustering* than in that of *classification*, contrary to the case of classical data known through attributes in an Euclidean space. Indeed, in the by now reference book [20], dissimilarity data only appear in the lone chapter (among 14) on clustering. One reason for that rather strange situation might stem from the fact that, in clustering, suffice it to be "heuristically good", whereas in classification one has the *error rate benchmark* hanging over one's head. The former goal appears to be conceivably achievable with dissimilarity data since it is intrinsically linked to their very nature. On the other hand, successfully passing the error rate benchmark in classification appears to be *a priori* more problematic for dissimilarity data due to their quite often unstable or unprecise nature. As such, even the *k nearest neighbors* (k-NN) method (see Fukunaga [12]), the traditional classification method most apparently suited for this type of data, can run into trouble in terms of error rate performance. Its well known drawbacks and advantages are preserved in this context: it is rather slow and non suited for non-spherical class shapes, neither for noisy data, but efficient with non connected classes. This latter fact is directly related to its very intuitive construction. The difficulty for the k-NN method to handle noisy data is particularly to be highlighted here, since real world dissimilarity data quite often fall in that category.

Other heuristic arguments lead to various algorithms for classifying objects on the basis of their dissimilarities with already classified ones. Guérin-Dugué and Celeux [15] propose a classification technique for dissimilarity data which leads to a quadratic-like classifier based on a pseudo Mahalanobis distance. Its advantages are rapidity, adaptation to incomplete dissimilarity data and apparent insensitivity to the (unknown) intrinsic data dimension, a major difficult issue in handling this type of data. Another distinctive approach is to treat the dissimilarities of objects with the n ones in the training set as coordinates of those objects in an n-dimensional Euclidean space. To classify the objects, one can then resort to any of the traditional classification algorithms applied in that n-dimensional space (albeit with appropriate adaptations for some of these algorithms). This approach is studied in [33, 34, 36]. In [19, 38], the same authors (with others) also examine a third approach in which the objects are first embedded in a low dimensional Euclidean or pseudo-Euclidean space where they are then classified. Another general purpose approach (see [18]) is to use the given pairwise dissimilarities as substitutes to Euclidean distances in kernels-based classification methods such as the popular Support Vector Machines (SVM). More complicated and specific distance classifiers are also needed when dealing with more involved classification issues such as subspace classification [1].

However, none of the afore-mentioned techniques for dissimilarity data classification is based on standard principles of statistical inference. Moreover, apart from [38], they do not take into account measurement error in the dissimilarities.

In the work presented here, we develop a new approach based on the purely statistical viewpoint

of *Multidimensional Scaling (MDS)* [40]. Although interesting alternatives exist, MDS remains today the leading mathematical methodology for handling dissimilarity data. We are concerned with classification analysis from a table of such observed data from an otherwise non observable population of objects. The main goal is to assign a new object to one of *a priori* groups in the population using as only information its dissimilarities with previously classified ones which thus form the *Training Data Set*. Basically, our approach to solving this problem assumes a probability model in which the observed dissimilarities are Euclidean distances perturbed with random gaussian errors. Actually, two possible probability models are investigated, differing by the structure they postulate for the perturbation of the Euclidean distances which led to the observed dissimilarities. In each of these models, the unobserved objects are regarded as unknown parameters lying in an Euclidean space. Estimating these parameters in the statistical sense is thus equivalent to positioning the unobserved objects in the Euclidean space given their respective pairwise dissimilarities, which is the traditional MDS main concern. Such a model-based approach then has the advantage to simultaneously estimate objects positioning and group labeling. Since it is unknown, the dimension p of the Euclidean space shall serve as a tuning parameter to be estimated from the data by cross validation and the "within one standard error of the minimum error towards model parsimony" rule.

The paper is organized as follows. Section 3 presents our global framework, from the data and the statistical models to the shape of our proposed solutions to the *dissimilarity data classification problem*. Section 4 details the learning phase in each solution, while Section 5 explains how a new (implicit) observation is classified. Section 6 is devoted to some miscellaneous considerations, including an empirical discussion about the respective *a priori* powers of our constructed prediction rules and the handling of the dissimilarity data intrinsic dimension problem in these rules. Section 7 then presents some numerical experiments on both simulated and real data. Finally, Section 8 draws some concluding remarks.

3 Framework

3.1 The data

Let Ω be a population divided into G disjoint groups labeled $k = 1, \ldots, G$, in the respective proportions (known or not):

$$\pi_1, \ldots, \pi_G > 0.$$

Our situation of interest here is that in which the objects in Ω cannot be concretely observed. Nevertheless, it is assumed that each of them can be identified through a coordinate system in an Euclidean space $\mathbb{R}^p$:

$$X = (x_1, \ldots, x_p) \in \mathbb{R}^p,$$

but with X being, thus, unobservable. Henceforth, we shall use Ω and $\mathbb{R}^p$ interchangeably. Initially, in our work, the dimension p is assumed known. Later on, it will appear that it needs to be estimated from the data so as to maximize the classification success rate of the derived prediction rules.

For the forthcoming classification problem, our learning situation is that in which there exist n objects in Ω,

$$X_1, \ldots, X_n \in \mathbb{R}^p, \tag{1}$$

as unobservable as other objects in Ω, but for which:

1. their respective group labelings are known:

$$g_1, \ldots, g_n \in \{1, \ldots, G\}; \tag{2}$$

2. there exists a dissimilarity measure $d(X, Y)$ between objects in Ω such that, for each object U in Ω, one can measure independently its respective dissimilarities with $X_1, \ldots, X_n$:

$$d(U, X_1), \ldots, d(U, X_n). \tag{3}$$

3.2 The statistical models

There does not seem to be a universally accepted probability model for dissimilarity data, maybe because such a model should somehow vary with the type of dissimilarity at hand. And, indeed, one encounters quite a variety of these in the literature. Nevertheless, Ramsay [40] assumes one such model for this type of data. For the sake of our classification problem, we postulate two (hopefully plausible) probability models for the dissimilarities between objects in Ω:

- **Model 1.**
  Here, it is assumed that there exists a symmetric square matrix $\Sigma = (\sigma_{kl}^2) \in \mathcal{M}_G(\mathbb{R}_+^*)$ such that:

$$\text{for all } X, Y \in \Omega, \quad d(X, Y) \sim \mathcal{N}(\| X - Y \|, \sigma^2_{g(X), g(Y)}), \tag{4}$$

  where

  - $\mathcal{M}_G(K)$ is the subset of square matrices of order G with all elements in K,
  - $\|.\|$ is the Euclidean norm in $\mathbb{R}^p$,
  - $g(X)$ is the group label of object $X \in \Omega$,
  - $\mathcal{N}(m, \sigma^2)$ is the univariate normal distribution with mean m and variance σ^2.

- **Model 2.**
  Here, it is assumed that there exist two symmetric square matrices $\mathbf{A} = (a_{kl})$, $\Sigma = (\sigma_{kl}^2) \in \mathcal{M}_G(\mathbb{R}_+^*)$ such that:

$$\text{for all } X, Y \in \Omega, \quad d(X, Y) \sim \mathcal{N}(a_{g(X), g(Y)} \| X - Y \|, \sigma^2_{g(X), g(Y)}). \tag{5}$$

Hypotheses (4) and (5) essentially mean that, in both postulated models, the dissimilarity measure $d(X, Y)$ between objects X and Y is a perturbation of their Euclidean distance in $\mathbb{R}^p$ with an error having a gaussian distribution. It shall turn out that the key assumption for our classification problem is that this gaussian distribution has a mean and/or variance completely determined by their Euclidean distance in $\mathbb{R}^p$ and their pair of groups in Ω. Indeed, this strongly classification oriented structure of our models is intended to provide more flexibility for them in trying to adapt to each dissimilarity data set. One should note that Model 1 is a submodel of Model 2. The former has just a *scale parameter* for the distribution of dissimilarities corresponding to the same pair of groups, while the latter adds a parameter which affects the *location* of the distribution. One should thus expect a better generalization performance for a classification rule based on Model 2.

It is important to outline that these are *conditional models*: they each hypothesize a distribution for the dissimilarity between two objects in Ω given the coordinates of these objects in $\mathbb{R}^p$. Although unobserved, no distribution is assumed for the objects themselves (somewhat in the spirit of *logistic prediction*). Rather, we shall treat these objects as unknown parameters to estimate in $\mathbb{R}^p$ when needed.

3.3 The dissimilarity data classification problem

Let $d_{ij} = d(X_i, X_j)$, for $i, j = 1, \ldots, n$. Using the available data, i.e.

1. the symmetric matrix $\mathbf{D} = (d_{ij}) \in \mathcal{M}_n(\mathbb{R}_+)$,

2. the group labels $g_1, \ldots, g_n \in \{1, \ldots, G\}$ (of the unobserved learning objects $X_1, \ldots, X_n$),

we intend to design a prediction rule $U \mapsto \widehat{g}(U)$, from Ω into $\{1, \ldots, G\}$, such that $\widehat{g}(U) = g(U)$ with a high probability in Ω.

The important practical requirement is, however, that for any $U \in \Omega$, the computation of its group label prediction $\widehat{g}(U)$ can use the only information which can be gathered about U, namely (3), i.e.

$$\widehat{g}(U) = \widehat{g}(d(U, X_1), \ldots, d(U, X_n)). \tag{6}$$

3.4 Some useful terminology

Since its concrete observation is impossible, any object $U \in \Omega$ can only be an *implicit observation* or a *virtual observation* or a *pseudo observation*. What can really be observed and manipulated about U is the vector:

$$d(U, \underline{X}) = (d(U, X_1), \ldots, d(U, X_n)) \in (\mathbb{R}_+)^n,$$

where $\underline{X} = (X_1, \ldots, X_n)$. Thus, $d(U, \underline{X})$ is an *explicit observation* or a *real observation* or a *true observation*.

Note, however, that it is truly the implicit observation or a certain type of information about it (here its *group label*) which is of interest for us. The explicit observation, hardly directly interpretable, only provides us with a convenient mean for trying to reach that implicit information hidden to our eyes. In that context:

- $\underline{X} = (X_1, \ldots, X_n)$ is our implicit or virtual *training (or learning) data set*;
- the vector lines of the matrix $\mathbf{D}$ constitute our *explicit* or *real training data set*;

whereas

- the objects $U \in \Omega \setminus \{X_1, \ldots, X_n\}$ are potential *implicit* or *virtual test data*,
- the vectors $d(U, \underline{X})$ being the corresponding *true test data*.

Remark. In general, the matrix $\mathbf{D}$ does not determine the configuration $\underline{X}$ uniquely. But this degree of freedom shall not matter for our classification problem. What is going to be important is, once $X_1, \ldots, X_n$ are estimated, the relative position of any object U in Ω w.r.t. these anchor points.

In the language of the *Multidimensional Scaling* community (see [2] for a comprehensive presentation of the subject), $\underline{X} = (X_1, \ldots, X_n)$ is called a *configuration of points* for the *dissimilarity matrix* $\mathbf{D}$. Indeed, in that scientific community, a chief concern is to devise operational methods to compute such a configuration given $\mathbf{D}$, or, at least, one good enough w.r.t. a reasonably chosen criterion (generally of least squares type). It shall turn out, in what follows, that this shall also be a key point in our classification procedures. In our context, $\underline{X}$ therefore deserves to be called the *implicit learning configuration* (of points or objects in Ω).

3.5 Our methodology

Let $\underline{\pi} = (\pi_1, \ldots, \pi_G)$ be the vector of groups proportions in Ω. To simplify notations, let also, for any $U \in \Omega$:

$$g = g(U) \in \{1, \ldots, G\}, \quad d_i = d(U, X_i) \ (i = 1, \ldots, n), \quad \underline{d} = d(U, \underline{X}) = (d_1, \ldots, d_n). \tag{7}$$

Similarly, we shall denote $\mathcal{P}$, the set of parameters in each of our models. Hence,

- in Model 1, $\mathcal{P} = \Sigma$, while in Model 2, $\mathcal{P} = (\mathbf{A}, \Sigma)$.

Note that in Model μ, $\mathcal{P} \in \mathcal{M}_G(\mathbb{R}_+^*)^\mu$.

As already stated, in our approach basically we regard objects in Ω as fixed unknown parameters which need estimation whenever requested. They are to be distinguished from those in $\mathcal{P}$ which are the intrinsic parameters in each model. With this in mind, we shall then proceed as follows in each of our models:

- *Learning Phase.* Using the explicit learning data, i.e. the dissimilarity matrix $\mathbf{D}$ and the group labels $g_1, \ldots, g_n$, we estimate the learning configuration $\underline{X} \in (\mathbb{R}^p)^n$, the intrinsic parameters set $\mathcal{P}$ and, if needed, the vector of groups proportions $\underline{\pi}$. This is done by a ML procedure, yielding respective estimates $\widehat{\underline{X}}$, $\widehat{\mathcal{P}}$ and $\widehat{\underline{\pi}}$. If the groups proportions are known, of course one simply takes $\widehat{\underline{\pi}} = \underline{\pi}$. This trivial case is excluded henceforth.

- *Prediction Phase.* Here, we wish to predict the group label g of a new implicit observation $U \in \Omega$ "landmarked" by the explicitly observed vector $\underline{d}$ in (7) of its respective dissimilarities with the points in the implicit learning configuration $\underline{X}$. To conform to our designed methodology in which objects in Ω are regarded as parameters, it then comes that with this new implicit observation enters a new unknown parameter, i.e. U itself. Whence, the joint distribution of the couple $(\underline{d}\,,\, g)$ is parameterized by U, $\underline{X}$, $\mathcal{P}$ and $\underline{\pi}$. Now, using golden standards in classification methodology (see [6], [20]), one should predict the group membership of U as that group $g = k$ among $1, \ldots, G$ which maximizes the *a posteriori* group probability given the explicit observation $\underline{d}$:

$$\Pr\left(g = k \,|\, \underline{d}\,, U, \underline{X}, \mathcal{P}, \underline{\pi}\right), \tag{8}$$

  were U, $\underline{X}$, $\mathcal{P}$ and $\underline{\pi}$ available. This is the same as maximizing, w.r.t. g, the joint likelihood of $(\underline{d}\,,\, g)$:

$$f(\underline{d}\,, g \,|\, U, \underline{X}, \mathcal{P}, \underline{\pi}) \;=\; \Pr\left(g \,|\, \underline{\pi}\right) \cdot f(\underline{d} \,|\, g, U, \underline{X}, \mathcal{P}) \;=\; \pi_g \cdot f(\underline{d} \,|\, g, U, \underline{X}, \mathcal{P}). \tag{9}$$

  But since the values of the parameters U, $\underline{X}$, $\mathcal{P}$ and $\underline{\pi}$ are unknown, the idea is to first estimate them as best as one can, and then maximize, w.r.t. g, the so obtained estimated version of (9). For $\underline{X}$, $\mathcal{P}$ and $\underline{\pi}$, the learning phase will have done the job with the respective ML estimates $\widehat{\underline{X}}$, $\widehat{\mathcal{P}}$ and $\widehat{\underline{\pi}}$. However, there clearly remains a *nuisance parameter* in (9) which was not estimated in the learning phase: U itself. Our strategy is then to *simultaneously* estimate U and predict g by maximizing, w.r.t. the couple (U, g), the estimated version of (9):

$$f(\underline{d}\,, g \,|\, U, \widehat{\underline{X}}, \widehat{\mathcal{P}}, \widehat{\underline{\pi}}) \;=\; \widehat{\pi}_{\,g} \cdot f(\underline{d} \,|\, g, U, \widehat{\underline{X}}, \widehat{\mathcal{P}}). \tag{10}$$

  This is achieved through the following two steps:

  1. For each $k \in \{1, \ldots, G\}$, the object U is estimated as best as possible under the hypothesis that U is in group k. This is obtained by maximizing, w.r.t. U, the estimated conditional likelihood of $\underline{d}$ given $g = k$:

$$f(\underline{d} \,|\, g = k, U, \widehat{\underline{X}}, \widehat{\mathcal{P}}). \tag{11}$$

     We denote $\widehat{U}_k$, the estimation so computed for U.

  2. We then predict the group of U in Ω by $\widehat{g} \in \{1, \ldots, G\}$ given by:

$$\widehat{g} \;=\; \arg \max_{k \,\in\, \{1,\ldots,G\}} \left[\widehat{\underline{\pi}}_k \cdot f(\underline{d} \,|\, g = k, \widehat{U}_k, \widehat{\underline{X}}, \widehat{\mathcal{P}})\right]. \tag{12}$$

In Sections 4 and 5 to follow, we detail these two phases for our models.

4 The learning procedures

Recall that we are given the learning data:

1. the dissimilarity matrix $\mathbf{D} = (d_{ij}) \in \mathcal{M}_n(\mathbb{R}_+)$, where $d_{ij} = d(X_i, X_j)$ for $i, j = 1, \ldots, n$;
2. the group labels $g_1, \ldots, g_n \in \{1, \ldots, G\}$ of the points in the unobserved learning configuration $\underline{X} = (X_1, \ldots, X_n)$;

and one wishes to use these learning data to estimate, in each of our postulated models, the unknown learning configuration $\underline{X} \in (\mathbb{R}^p)^n$, the set of intrinsic parameters $\mathcal{P}$, and $\underline{\pi} = (\pi_1, \ldots, \pi_G)$, the vector of groups proportions. This will be done by ML estimation.

But before moving any further, it is important to keep in mind that as a (proper) dissimilarity matrix, $\mathbf{D}$ is symmetric with a zero diagonal. Hence, the useful data it contains can be found entirely, for instance, in its strict upper triangular portion. We denote this upper triangular array hereafter by $\mathbf{D}_{\text{up}}$, which can thus be substituted to $\mathbf{D}$ in the whole model based classification analysis.

To proceed in our ML estimation procedures, we assume the following hypotheses:

H1. The coefficients d_{ij} in $\mathbf{D}_{\text{up}}$ have been observed independently.

H2. The group labels $g_1, \ldots, g_n$ have also been observed independently (and independently of the d_{ij}'s) and their common distribution has $\underline{\pi}$ as lone parameter.

H3. Given $\underline{g}$, $\underline{X}$ and $\mathcal{P}$, the distribution of $\mathbf{D}_{\text{up}}$ does not depend on $\underline{\pi}$.

Before we proceed, a brief remark is in order here. The number of independent parameters in our models ($np + (G^2 + 3G - 2)/2$ for Model 1, and $np + G^2 + 2G - 1$ for Model 2) grows unboundedly with the number of independent observations ($n(n+1)/2$). This shouldn't hurt, however, orthodox statistical methodology since the ratio of the latter over the former grows unboundedly in parallel; so when $n \to +\infty$, the number of independent observations more and more outweighs the number of parameters to estimate.

4.1 The structure of the learning data likelihood

The full learning data is the (hyper)couple $(\mathbf{D}_{\text{up}}, \underline{g})$, which, under **H2**-**H3**, has likelihood:

$$f(\mathbf{D}_{\text{up}}, \underline{g} \,|\, \underline{X}, \mathcal{P}, \underline{\pi}) \;=\; \Pr(\underline{g} \,|\, \underline{\pi}) \cdot f(\mathbf{D}_{\text{up}} \,|\, \underline{g}, \underline{X}, \mathcal{P}). \tag{13}$$

It thus comes that to maximize $f(\mathbf{D}_{\text{up}}, \underline{g} \,|\, \underline{X}, \mathcal{P}, \underline{\pi})$ w.r.t. $(\underline{X}, \mathcal{P}, \underline{\pi})$, one needs to maximize separately:

1. $\Pr(\underline{g} \,|\, \underline{\pi})$ w.r.t. $\underline{\pi} = (\pi_1, \ldots, \pi_G)$;
2. $f(\mathbf{D}_{\text{up}} \,|\, \underline{g}, \underline{X}, \mathcal{P})$ w.r.t. $(\underline{X}, \mathcal{P})$.

We now proceed to solve these two separate maximization problems in each of our models.

4.2 ML estimation of groups proportions

By **H2**, one has in all our models:

$$\Pr(\underline{g} \,|\, \underline{\pi}) \;=\; \prod_{i=1}^{n} \Pr(g_i \,|\, \underline{\pi}) \;=\; \prod_{i=1}^{n} \pi_{g_i} \;=\; \prod_{k=1}^{G} \pi_k^{n_k}, \tag{14}$$

where n_k is the number of objects X_i in the learning configuration $\underline{X}$ with group label $g_i = k$. A well known result has it that among all probability vectors $\underline{\pi} = (\pi_1, \ldots, \pi_G)$ (i.e. satisfying: $\pi_1, \ldots, \pi_G \geq 0$ and $\pi_1 + \cdots + \pi_G = 1$), (14) achieves maximal value at

$$\widehat{\underline{\pi}} \;=\; (\widehat{\pi}_1, \ldots, \widehat{\pi}_G), \quad \text{with } \widehat{\pi}_k = n_k/n, \text{ for } k = 1, \ldots, G. \tag{15}$$

4.3 ML estimation of $\underline{X}$ and $\mathcal{P}$: How it proceeds

As outlined in Section 4.1, in each of our models one needs to maximize the conditional likelihood $f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathcal{P})$ w.r.t. $(\underline{X},\,\mathcal{P})$. Now, by **H1**,

$$f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathcal{P})\;=\;\prod_{i=1}^{n}\prod_{j>i}f(d_{ij}\,|\,\underline{g}\,,\underline{X},\mathcal{P}),\tag{16}$$

where the usual convention that a product with an empty set of indices equals 1 is assumed throughout.

As is obvious from (16), $f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathcal{P})$ is quite a complicated function of $(\underline{X},\mathcal{P})$ to maximize, even numerically. Nevertheless, we are going to follow this latter path by constructing, in each Model μ, a sequence $(\underline{\widehat{X}}^{\,\nu},\widehat{\mathcal{P}}^{\,\nu})\,\in\,(\mathbb{R}^p)^n\times\mathcal{M}_G(\mathbb{R}_+^*)^\mu$ in the hope that it shall converge towards

$$(\underline{\widehat{X}},\,\widehat{\mathcal{P}})\;=\;\arg\max_{(\underline{X},\,\mathcal{P})}f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathcal{P}),\tag{17}$$

or, at least, towards a local maximum, which we shall still denote $(\underline{\widehat{X}},\,\widehat{\mathcal{P}})$ and use accordingly. To achieve this, we proceed iteratively by alternating maximization w.r.t. $\underline{X}$ and maximization w.r.t. $\mathcal{P}$. Notice then that:

- in Model 1, $\widehat{\mathcal{P}}^{\,\nu}=\widehat{\Sigma}^{\,\nu}$, while in Model 2, $\widehat{\mathcal{P}}^{\,\nu}=(\widehat{\mathbf{A}}^{\,\nu},\widehat{\Sigma}^{\,\nu})$.

With this in mind, the alternating maximization process goes as follows:

- *Initialization Step*: $\widehat{\Sigma}^0\;=\;(\widehat{\sigma}^2_{kl,0})$, with $\widehat{\sigma}_{kl,0}=1$, for $k,l\;=\;1,\ldots,G$;
  and, in Model 2, $\widehat{\mathbf{A}}^{\,0}=\widehat{\Sigma}^0$.
- At step $\nu\geq 0$, with $\widehat{\mathcal{P}}^{\,\nu}$ at hand:

  (M.$\underline{X}$) Computation of $\underline{\widehat{X}}^{\,\nu}$ by maximizing $f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\widehat{\mathcal{P}}^{\,\nu})$ w.r.t. $\underline{X}$.

  (M.$\mathcal{P}$) Computation of $\widehat{\mathcal{P}}^{\,\nu+1}$ by maximizing $f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{\widehat{X}}^{\,\nu},\mathcal{P})$ w.r.t. $\mathcal{P}$.

4.4 Maximization (M.$\underline{X}$) for computing $\underline{\widehat{X}}^{\,\nu}$ given $\widehat{\mathcal{P}}^{\,\nu}$.

We first detail the case of Model 1 before giving the short adaptation needed for Model 2.

4.4.1 Maximization (M.$\underline{X}$) in Model 1.

Here,

$$f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathcal{P})=f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\Sigma)\;=\;\prod_{i=1}^{n}\prod_{j>i}\left[\frac{1}{\sigma_{g_i\,g_j}}\cdot\varphi\left(\frac{d_{ij}\;-\delta_{ij}\,(\underline{X})}{\sigma_{g_i\,g_j}}\right)\right],\tag{18}$$

where $\delta_{ij}\,(\underline{X})\;=\;\|\,X_i\,-\,X_j\,\|$, for $i,j\;=\;1,\ldots,n$; and $\varphi(x)\;=\;\exp(-x^2/2)/\sqrt{2\pi}$ is the pdf of the standard normal distribution.

Given $\Sigma=\widehat{\Sigma}^{\,\nu}=(\widehat{\sigma}^2_{kl,\nu})\in\mathcal{M}_G(\mathbb{R}_+^*)$, taking the logarithm on both sides of (18) yields:

$$\ln f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\widehat{\mathcal{P}}^{\,\nu})\;=\;\widehat{C}^{\,\nu}\;-\;\frac{1}{2}\cdot\widehat{\rm STRESS}_{\,\nu}(\underline{X}),\tag{19}$$

where $\widehat{C}^{\,\nu}$ is a constant w.r.t. $\underline{X}$ and one defines

$$\widehat{\rm STRESS}_{\,\nu}(\underline{X})\;=\;\sum_{i=1}^{n}\sum_{j>i}\widehat{w}_{ij,\nu}\cdot(\widehat{d}_{ij,\nu}-\delta_{ij}\,(\underline{X}))^2,\tag{20}$$

with $\widehat{d}_{ij,\nu} = d_{ij}$ and $\widehat{w}_{ij,\nu} = \widehat{\sigma}^{-2}_{g_i g_j,\nu}$, for $i, j = 1, \ldots, n$.

So, from (19), one deduces that maximizing $f(\mathbf{D}_{\rm up} \,|\, \underline{g}\,, \underline{X}, \widehat{\mathcal{P}}^{\,\nu})$ w.r.t. $\underline{X}$ is equivalent to minimizing $\widehat{\rm STRESS}_{\,\nu}(\underline{X})$ w.r.t. that same (np)-dimensional variable. Now, this latter minimization problem is a common *Multidimensional Scaling (MDS)* situation where one searches a configuration of points $\underline{X}$ in $\mathbb{R}^p$ which inter-points (Euclidean) distances $\delta_{ij}\,(\underline{X})$ closely approximate given dissimilarities d_{ij} between the X_i's. Such a configuration can be obtained by minimizing an expression like (20) called a *weighted STRESS criterion* with the given weights $\widehat{w}_{ij,\nu}$ on the observed dissimilarities d_{ij}'s.

We solve this rather involved minimization problem by one of the leading algorithms in the MDS area: the SMACOF algorithm of de Leeuw (see [21, 25]) which is essentially an implementation of [17]. This iterative algorithm requiring an initial configuration $\underline{\widehat{X}}^{\,\nu,0}$ to start with, it is provided as follows:

- Case $\nu = 0$: $\underline{\widehat{X}}^{\,\nu,0} = \text{O.d.}(\underline{X}^T)$.

  Here, $\underline{X}^T$ is taken to be the configuration computed from the d_{ij}'s by Torgerson's *classical MDS* (finite) algorithm [43, 44], and O.d.$(\underline{X})$ is the *optimal dilation transformation* of Malone et al. [28, 27], pushing further initial ideas of [13]:

$$\text{O.d.}(\underline{X}) \;=\; \frac{<\Delta_{\rm up}(\underline{X}),\, \mathbf{D}_{\rm up}\,>}{\|\,\Delta_{\rm up}(\underline{X})\,\|_F^2} \cdot \underline{X}, \tag{21}$$

  where $\Delta_{\rm up}(\underline{X})$ is the strictly upper triangular part of $\Delta(\underline{X})$, the $n \times n$ distance matrix with coefficients $\delta_{ij}\,(\underline{X})$; $\|\cdot\|_F$ is the Frobenius norm for such triangular arrays, with weights $\widehat{w}_{ij,\nu}$ on the coefficients, and $<\cdot,\cdot>$ is the corresponding inner product. Malone et al. [28] gave a theoretical justification and some numerical evidence in the unweighted (or equal weights) case showing that the transformation (21) can significantly improve a poor configuration $\underline{X}$ as far as minimization of the STRESS criterion is concerned. However, their derivation extends steadily in the weighted case as well since it uses only the Euclidean structure intrinsic in that least squares criterion. In any event, it always decreases the STRESS criterion, albeit insignificantly after the first application.

- Case $\nu \geq 1$: $\underline{\widehat{X}}^{\,\nu,0} = \text{O.d.}(\underline{\widehat{X}}^{\,\nu-1})$.

This last systematic choice of initialization considerably accelerates the convergence both in the SMACOF algorithm here, and in the whole iterative maximization process **(M.$\underline{X}$)-(M.$\mathcal{P}$)**.

The stopping criterion in the SMACOF algorithm shall be that the $\widehat{\rm STRESS}_{\,\nu}$ values of two consecutive configurations $\underline{\widehat{X}}^{\,\nu,q-1}$ and $\underline{\widehat{X}}^{\,\nu,q}$ differ, in relative value, by less than, say, 0.01, or, otherwise, that $q = 25$.

Remark. The main advantage of the SMACOF algorithm is that, although quite slow and with no guarantee to converge to a global minimum of the $\widehat{\rm STRESS}_{\,\nu}$ function, it has the virtue to decrease it at each iteration and so, in our context, to increase the likelihood. For that reason, we are likely to converge to a (local) maximum of $f(\mathbf{D}_{\rm up} \,|\, \underline{g}\,, \underline{X}, \widehat{\mathcal{P}}^{\,\nu})$ w.r.t. $\underline{X}$.

So, to stop at iteration q in the SMACOF algorithm, we require that:

$$(0 <)\; \widehat{\rm STRESS}_{\,\nu}(\underline{\widehat{X}}^{\,\nu,q-1}) - \widehat{\rm STRESS}_{\,\nu}(\underline{\widehat{X}}^{\,\nu,q}) \;<\; 0.01 \cdot \widehat{\rm STRESS}_{\,\nu}(\underline{\widehat{X}}^{\,\nu,q-1}). \tag{22}$$

If this is realized or $q = 25$, we stop the iteration and set: $\underline{\widehat{X}}^{\,\nu} = \underline{\widehat{X}}^{\,\nu,q}$.

We stress that [23] could have been a possible alternative to the use of the SMACOF algorithm here to minimize the STRESS criterion.

4.4.2 Maximization (M.$\underline{X}$) in Model 2.

Here, (16) becomes:

$$f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathcal{P}) = f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{X},\mathbf{A},\Sigma) \;=\; \prod_{i=1}^{n}\prod_{j>i}\left[\frac{1}{\sigma_{g_i\,g_j}}\cdot\varphi\left(\frac{d_{ij}\;-a_{g_i\,g_j}\,\delta_{ij}\,(\underline{X})}{\sigma_{g_i\,g_j}}\right)\right]. \tag{23}$$

So (19)-(20) still holds if one redefines $\widehat{d}_{ij,\nu}$ and $\widehat{w}_{ij,\nu}$ respectively as:

$$\widehat{d}_{ij,\nu} = d_{ij}\,/\widehat{a}_{g_i\,g_j\,,\nu}, \quad \widehat{w}_{ij,\nu} \;=\; (\widehat{a}_{g_i\,g_j\,,\nu}/\widehat{\sigma}_{g_i\,g_j\,,\nu})^{\,2}. \tag{24}$$

The maximization of $\widehat{\rm STRESS}_{\,\nu}(\underline{X})$ w.r.t. $\underline{X}$ then proceeds exactly as in Model 1 as detailed in Section 4.4.1 above.

4.5 Maximization (M.$\mathcal{P}$) for computing $\widehat{\mathcal{P}}^{\,\nu+1}$ given $\underline{\widehat{X}}^{\,\nu}$.

In what follows, $\varphi(x|\,m,\sigma^2)$ denotes the pdf of the univariate gaussian distribution with mean m and variance σ^2.

4.5.1 Maximization (M.$\mathcal{P}$) in Model 1.

Setting $\underline{X} = \underline{\widehat{X}}^{\,\nu}$ in (18) yields in Model 1:

$$f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{\widehat{X}}^{\,\nu},\mathcal{P}) \;=\; f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{\widehat{X}}^{\,\nu},\Sigma) \;=\; \prod_{i=1}^{n}\prod_{j>i}\varphi(d_{ij}\,|\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2_{g_i\,g_j})$$

$$= \prod_{k=1}^{G}\prod_{\substack{i=1\\ g_i=k}}^{n}\prod_{l=1}^{G}\prod_{\substack{j>i\\ g_j=l}}\varphi(d_{ij}\,|\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2_{kl}) \;=\; \prod_{k=1}^{G}\prod_{l=1}^{G}\left[\prod_{\substack{i=1\\ g_i=k}}^{n}\prod_{\substack{j>i\\ g_j=l}}\varphi(d_{ij}\,|\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2_{kl})\right]$$

$$= \left[\prod_{k=1}^{G}\widehat{\Psi}_{kk,\nu}(\sigma^2_{kk})\right]\cdot\left[\prod_{1\le k<l\le G}\widehat{\Psi}_{kl,\nu}(\sigma^2_{kl})\right], \tag{25}$$

where we define:

$$\widehat{\Psi}_{kk,\nu}(\sigma^2) \;=\; \prod_{\substack{i=1\\ g_i=k}}^{n}\prod_{\substack{j>i\\ g_j=k}}\varphi(d_{ij}\,|\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2), \quad \text{for } k=1,\ldots,G\,; \tag{26}$$

$$\widehat{\Psi}_{kl,\nu}(\sigma^2) \;=\; \prod_{\substack{i=1\\ g_i=k}}^{n}\prod_{\substack{j=1\\ g_j=l}}^{n}\varphi(d_{ij}\,|\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2), \quad \text{for } 1\le k<l\le G. \tag{27}$$

To see that (25) holds with definitions (26)-(27), one should recall that $\sigma_{kl} = \sigma_{lk}$ since Σ is a symmetric matrix.

From (25)-(27), it comes that maximizing $f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{\widehat{X}}^{\,\nu},\Sigma)$ w.r.t. $\Sigma = (\sigma^2_{kl}) \in \mathcal{M}_G(\mathbb{R}^*_+)$ is equivalent to doing so separately for each of the $G(G+1)/2$ functions of one positive real variable $\widehat{\Psi}_{kl,\nu}(\sigma^2)$ $(1\le k\le l\le G)$. In this respect, it is important to emphasize that in our models, except for symmetry, no other a priori relationship is assumed between the coefficients of the matrix Σ, contrary to those of a covariance matrix for instance.

Now, by their very respective definitions (26)-(27), each of the functions $\widehat{\Psi}_{kl,\nu}(\sigma^2)$ can be recast as:

$$\Psi(\sigma^2) \,=\, \prod_{i=1}^{r} \varphi(x_i|\, m_i\,, \sigma^2) \,=\, \prod_{i=1}^{r} \varphi(x_i \,-\, m_i\,|\, 0, \sigma^2),$$

where $x_1\,, \ldots,\, x_r$ are independent observations from the respective gaussian distributions $\mathcal{N}(m_1\,, \sigma^2\,)$, $\ldots\,, \mathcal{N}(m_r\,, \sigma^2\,)$, with given means $m_1\,, \ldots,\, m_r$ and common unknown variance σ^2. Therefore, $x_1 - m_1$, $\ldots$, $x_r - m_r$ are r i.i.d. observations from $\mathcal{N}(0, \sigma^2\,)$. Given these latter, estimating σ^2 by ML, i.e. maximizing the likelihood $\Psi(\sigma^2)$, is a standard exercise in Statistics Textbooks and yields:

$$\widehat{\sigma}^2 \,=\, \frac{1}{r} \sum_{i=1}^{r} (x_i \,-\, m_i\,)^2.$$

Let then

$$N_k \,= \text{number of couples of indices } (i, j) \text{ among } 1, \ldots, n \text{ satisfying: } i < j \text{ and } g_i \,= g_j \,= k; \quad (28)$$

$$N_{kl} \,= \text{number of couples of indices } (i, j) \text{ among } 1, \ldots, n \text{ satisfying: } g_i \,= k \text{ and } g_j \,= l. \quad (29)$$

Based on the preceding analysis and given expressions (25) to (27), it comes that, in Model 1, $f(\mathbf{D}_{\rm up}\,|\underline{g}\,, \underline{\widehat{X}}^{\,\nu}, \mathcal{P})$ is maximal w.r.t. $\mathcal{P}$ at $\mathcal{P} \,=\, \widehat{\mathcal{P}}^{\,\nu+1} \,=\, \widehat{\Sigma}^{\,\nu+1}$ which diagonal elements are given by:

$$\widehat{\sigma}^2_{kk,\,\nu+1} \;\;=\;\; \arg\max_{\sigma^2} \widehat{\Psi}_{kk,\nu}(\sigma^2) \,=\, \frac{1}{N_k} \sum_{\substack{i<j \\ g_i \,= g_j \,= k}} \Big[d_{ij} \,-\, \delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}) \Big]^2, \quad (30)$$

whereas its off-diagonal elements are:

$$\widehat{\sigma}^2_{kl,\,\nu+1} \;\;=\;\; \arg\max_{\sigma^2} \widehat{\Psi}_{kl,\nu}(\sigma^2) \,=\, \frac{1}{N_{kl}} \sum_{\substack{g_i \,= k \\ g_j \,= l}} \Big[d_{ij} \,-\, \delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}) \Big]^2. \quad (31)$$

These explicit expressions highlight the fact that, in Model 1, the maximization problem **(M.$\mathcal{P}$)** has a unique global solution $\widehat{\mathcal{P}}^{\,\nu+1}$ which can easily be computed in closed form. No iterative numerical process is requested here.

4.5.2 Maximization (M.$\mathcal{P}$) in Model 2.

Analogously to the case of Model 1 above, setting $\underline{X} = \underline{\widehat{X}}^{\,\nu}$ in (23) yields:

$$f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,, \underline{\widehat{X}}^{\,\nu}, \mathcal{P}) \,=\, f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,, \underline{\widehat{X}}^{\,\nu}, \mathbf{A}, \Sigma) \,=\, \prod_{i=1}^{n} \prod_{j>i} \varphi(d_{ij}\,|\, a_{g_i\,g_j}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}), \sigma^2_{g_i\,g_j}\,)$$

$$= \;\; \prod_{k=1}^{G} \prod_{\substack{i=1 \\ g_i \,= k}}^{n} \prod_{l=1}^{G} \prod_{\substack{j>i \\ g_j \,= l}} \varphi(d_{ij}\,|\, a_{kl}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}), \sigma^2_{kl}\,) \,=\, \prod_{k=1}^{G} \prod_{l=1}^{G} \left[\prod_{\substack{i=1 \\ g_i \,= k}}^{n} \prod_{\substack{j>i \\ g_j \,= l}} \varphi(d_{ij}\,|\, a_{kl}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}), \sigma^2_{kl}\,) \right]$$

$$= \;\; \left[\prod_{k=1}^{G} \widehat{\Psi}_{kk,\nu}(a_{kk}\,, \sigma^2_{kk}\,) \right] \cdot \left[\prod_{1 \le k < l \le G} \widehat{\Psi}_{kl,\nu}(a_{kl}\,, \sigma^2_{kl}\,) \right], \quad (32)$$

where now:

$$\widehat{\Psi}_{kk,\nu}(a,\sigma^2) \;=\; \prod_{\substack{i=1\\ g_i=k}}^{n}\prod_{\substack{j>i\\ g_j=k}}\varphi(d_{ij}\,|\,a\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2), \quad \text{for } k=1,\ldots,G\,; \tag{33}$$

$$\widehat{\Psi}_{kl,\nu}(a,\sigma^2) \;=\; \prod_{\substack{i=1\\ g_i=k}}^{n}\prod_{\substack{j=1\\ g_j=l}}^{n}\varphi(d_{ij}\,|\,a\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}),\sigma^2), \quad \text{for } 1\le k<l\le G, \tag{34}$$

and again the symmetry of the intrinsic parameter matrices $\mathbf{A}$ and Σ played a key role.

Here, by their respective definitions (33)-(34), each of the functions $\widehat{\Psi}_{kl,\nu}(a,\sigma^2)$ can be recast as:

$$\Psi(a,\sigma^2) \,=\, \prod_{i=1}^{r}\varphi(y_i|\,a\,x_i\,,\sigma^2) \,=\, \prod_{i=1}^{r}\varphi(y_i\,-\,a\,x_i\,|\,0,\sigma^2). \tag{35}$$

Now, (35) is the likelihood of the linear least squares model

$$y_i = a\,x_i\,+\varepsilon_i\,, \tag{36}$$

where the x_i's are given real numbers, the y_i's are observed random quantities and $\varepsilon_1\,,\cdots,\varepsilon_r$ are i.i.d. centered gaussian errors with common variance σ^2. Standard computations yield ML estimates of a and σ^2 in (36), i.e. the values maximizing $\Psi(a,\sigma^2)$:

$$\widehat{a} = \sum_{i=1}^{r} x_i\,y_i \,\Big/\, \sum_{i=1}^{r} x_i^2, \qquad \widehat{\sigma}^2 = \frac{1}{r}\sum_{i=1}^{r}(y_i\,-\widehat{a}x_i)^2.$$

Applying this to (33)-(34), one sees that, in Model 2, $f(\mathbf{D}_{\rm up}\,|\,\underline{g}\,,\underline{\widehat{X}}^{\,\nu},\mathcal{P})$ is maximal w.r.t. $\mathcal{P}$ at $\mathcal{P}=\widehat{\mathcal{P}}^{\,\nu+1}=(\widehat{\mathbf{A}}^{\,\nu+1},\widehat{\Sigma}^{\,\nu+1})$ with the diagonal coefficients of the matrices $\widehat{\mathbf{A}}^{\,\nu+1}$ and $\widehat{\Sigma}^{\,\nu+1}$ given by:

$$\widehat{a}_{kk,\,\nu+1} \;=\; \sum_{\substack{i<j\\ g_i=g_j=k}} d_{ij}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}) \,\Big/\, \sum_{\substack{i<j\\ g_i=g_j=k}} \Big[\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu})\Big]^2, \tag{37}$$

$$\widehat{\sigma}^2_{kk,\,\nu+1} \;=\; \frac{1}{N_k}\sum_{\substack{i<j\\ g_i=g_j=k}} \Big[d_{ij}\,-\,\widehat{a}_{kk,\,\nu+1}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu})\Big]^2, \tag{38}$$

whereas the respective off-diagonal elements are:

$$\widehat{a}_{kl,\,\nu+1} \;=\; \sum_{\substack{g_i=k\\ g_j=l}} d_{ij}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu}) \,\Big/\, \sum_{\substack{g_i=k\\ g_j=l}} \Big[\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu})\Big]^2, \tag{39}$$

$$\widehat{\sigma}^2_{kl,\,\nu+1} \;=\; \frac{1}{N_{kl}}\sum_{\substack{g_i=k\\ g_j=l}} \Big[d_{ij}\,-\,\widehat{a}_{kl,\,\nu+1}\,\delta_{ij}\,(\underline{\widehat{X}}^{\,\nu})\Big]^2. \tag{40}$$

One concludes that, in Model 2 also, the maximization problem **(M.$\mathcal{P}$)** has a unique global solution $\widehat{\mathcal{P}}^{\,\nu+1}=(\widehat{\mathbf{A}}^{\,\nu+1},\widehat{\Sigma}^{\,\nu+1})$ which can easily be computed in closed form, with no iterative process required.

4.6 The stopping criterion in the iterative maximization process (M.$\underline{X}$)-(M.$\mathcal{P}$)

Given how we initialized the SMACOF algorithm in Section 4.4, we stop the alternating maximization process **(M.$\underline{X}$)-(M.$\mathcal{P}$)** when the following criterion is satisfied:

$$(0 <) \widehat{\mathrm{STRESS}}_{\nu-1}(\underline{X}^{\nu-1}) - \widehat{\mathrm{STRESS}}_{\nu}(\underline{X}^{\nu}) < 0.01 \cdot \widehat{\mathrm{STRESS}}_{\nu-1}(\underline{X}^{\nu-1}), \tag{41}$$

or $\widehat{\mathrm{STRESS}}_{\nu-1}(\underline{X}^{\nu-1}) \leq \widehat{\mathrm{STRESS}}_{\nu}(\underline{X}^{\nu})$ (i.e. no STRESS decrease between steps $\nu-1$ and ν). Thus, if this stopping criterion is realized, or $\nu = 50$ (say), we set $(\underline{X}, \widehat{\mathcal{P}}) = (\underline{\widehat{X}}^{\nu}, \widehat{\mathcal{P}}^{\nu})$, and this ends the learning phase of our proposed solutions to the dissimilarity data classification problem.

5 The classification procedures

5.1 The prediction rule for a new implicit observation $U \in \Omega$

Recall that U is only "observed" through its dissimilarities $d_i = d(U, X_i)$ with each of the learning implicit observations $X_1, \ldots, X_n$ (themselves unobserved). From these dissimilarities, one wishes to predict appropriately the group label $g \in \{1, \ldots, G\}$ of U in Ω.

With $(\underline{X}, \mathcal{P}, \underline{\pi})$ estimated in the learning phase by MLE yielding $(\underline{\widehat{X}}, \widehat{\mathcal{P}}, \underline{\widehat{\pi}})$ in Section 4, to predict g, hopefully with a high probability, we proceed through the two successive steps described in the *Prediction Phase* of Section 3.5, and thus obtain a prediction $\widehat{g} \in \{1, \ldots, G\}$ for the group of the implicit observation U in Ω. Note that this procedure also computes, as a byproduct, a ML type estimate for U once $X_1, \ldots, X_n$ have been positioned at $\widehat{X}_1, \ldots, \widehat{X}_n$ and $(\mathcal{P}, \underline{\pi})$ estimated by $(\widehat{\mathcal{P}}, \underline{\widehat{\pi}})$. In the notations of (11)-(12), this ML like estimate of U is $\widehat{U}_k$ with $k = \widehat{g}$.

5.2 About the sub-maximization requested in the prediction rule

The Step 1 of the *Prediction Phase* (see Section 3.5) consists, for each group label $k \in \{1, \ldots, G\}$, in maximizing, w.r.t. U, the function in $\mathbb{R}^p$:

$$\widehat{F}_k(U) = f(\underline{d} \,|\, g = k, U, \underline{\widehat{X}}, \widehat{\mathcal{P}}), \tag{42}$$

where we recall that $\underline{d} = (d_1, \ldots, d_n) = (d(U, X_1), \ldots, d(U, X_n))$. Now, by the hypothesis that dissimilarities are observed independently,

$$\widehat{F}_k(U) = \prod_{i=1}^{n} f(d_i \,|\, g = k, U, \underline{\widehat{X}}, \widehat{\mathcal{P}}). \tag{43}$$

Again, we detail the case of Model 1 before outlining the simple adjustment needed to handle the prediction in Model 2.

5.2.1 The sub-maximization for prediction in Model 1

In Model 1, (43) becomes:

$$\widehat{F}_k(U) = \prod_{i=1}^{n} \left[\frac{1}{\widehat{\sigma}_{k,\,g_i}\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(\frac{d_i - \| U - \widehat{X}_i \|}{\widehat{\sigma}_{k,\,g_i}} \right)^2 \right\} \right]. \tag{44}$$

Clearly, maximizing $\widehat{F}_k(U)$ w.r.t. $U \in \mathbb{R}^p$, the solution of which was denoted $\widehat{U}_k$ in Section 3.5, is the same as minimizing w.r.t. U:

$$\widehat{H}_k(U) = \sum_{i=1}^{n} \widehat{\omega}_{k,\,g_i} \left(\widehat{d}_{ik} - \| U - \widehat{X}_i \| \right)^2, \tag{45}$$

where

$$\widehat{\omega}_{k,\,g_i} = 1/\widehat{\sigma}^2_{k,\,g_i}\,, \qquad \widehat{d}_{ik} = d_i\,. \tag{46}$$

The minimization of $\widehat{H}_k(U)$ is performed iteratively, starting from a chosen initial estimate $\widehat{U}_{k,0}$ of $\widehat{U}_k$, through a numerical nonlinear optimization algorithm. In our case, since all of the programming was done in the R statistical computing system [39], we used the `nlm` function. However, in order to significantly speed up iterations, it appears wise to equip the R function evaluating $\widehat{H}_k(U)$ for any given U with a "gradient" attribute consisting of the value of its exact gradient evaluated at U. This is readily seen to be:

$$\nabla\widehat{H}_k(U) \;=\; 2\sum_{i=1}^{n}\widehat{\omega}_{k,\,g_i}\cdot(\,\|\,U-\widehat{X}_i\,\|-d_i\,)\cdot\frac{U-\widehat{X}_i}{\|\,U-\widehat{X}_i\,\|}\,. \tag{47}$$

This manœuvre prevents the `nlm` function from using finite differences to approximate the p coordinates of $\nabla\widehat{H}_k(U)$, thus nearly dividing the computing time roughly by p. The gain is enormous as soon as $p \geq 2$.

The computation of an initial approximation $\widehat{U}_{k,0}$ of $\widehat{U}_k$ is discussed in the Appendix.

5.2.2 The sub-maximization for prediction in Model 2

In Model 2, (43) becomes:

$$\widehat{F}_k(U) \;=\; \prod_{i=1}^{n}\left[\,\frac{1}{\widehat{\sigma}_{k,\,g_i}\sqrt{2\pi}}\,\exp\left\{\,-\frac{1}{2}\left(\frac{d_i\,-\widehat{a}_{k,\,g_i}\,\|\,U-\widehat{X}_i\,\|}{\widehat{\sigma}_{k,\,g_i}}\right)^2\right\}\right]. \tag{48}$$

So (45) still holds, provided one takes:

$$\widehat{\omega}_{k,\,g_i} = (\widehat{a}_{k,\,g_i}/\widehat{\sigma}_{k,\,g_i})^2, \qquad \widehat{d}_{ik} = d_i/\widehat{a}_{k,\,g_i}\,. \tag{49}$$

One can then proceed as in Section 5.2.1 above for Model 1 to numerically minimize $\widehat{F}_k(U)$.

6 Miscellaneous aspects

6.1 Acronyms for our 2 model based classifiers

As reference acronyms for the two classifiers so constructed, we shall use, from now on:

- "M1.BC" for the pairwise dissimilarities Model 1 based classifier;
- "M2.BC" for the pairwise dissimilarities Model 2 based classifier;

where "pw.d.M1.BC" and "pw.d.M2.BC" would have been more appropriate but too long.

6.2 About the *a priori* predictive power of M1.BC/M2.BC

Considering the way our prediction rules are designed (see Sections 3.5 and 5), the structure of the function $\widehat{H}_k(U)$ given by (45)-(46) suggests that classes k and l are well separated by Model 1 when lines k and l of the matrix Σ are appreciably different vectors in $\mathbb{R}^G$. However, how the magnitude of that vector difference affects the separability of the two classes is still to be properly investigated. Nevertheless, preliminary empirical evidence seems to suggest that the difference needs not be that so big in order for the model to correctly discriminate between the two classes.

A similar empirical analysis holds for Model 2 by simultaneously considering this time the lines of the matrices $\mathbf{A}$ and Σ. However, as already explained, the coefficients in $\mathbf{A}$ affect the location of dissimilarities pairwise distributions while those of Σ are scale factors. As such, it is the former that appear to play the leading discriminatory role in the classification.

6.3 About the dissimilarity data intrinsic dimension problem

The first quite obvious problem which appears when trying to implement our classification methodology for dissimilarity data, as described in the previous Sections, is that of the data intrinsic dimension p which is seldom known. The strategy chosen to cope around this apparent hurdle is to regard p simply as an arbitrary embedding dimension acting as a tuning parameter to be estimated from the data. For a range of p values, chosen to be 1 to 12 in our experiments, the "good" one is considered to be the one yielding the highest classification rate estimated through 5-blocks Cross Validation [6, 20]. However, the "within one standard error of the minimum error towards model parsimony" rule. is also included as a complement in that choice, meaning that the final estimate p_e of p is chosen to be the smallest among those with a 5-blocks Cross Validation success rate estimate within one standard error of the highest rate.

7 Numerical experiments

To evaluate and compare our two proposed classification procedures for dissimilarity data, two types of experiments were conducted:

1. the first ones to weigh the effect of the dissimilarity type and of the data intrinsic dimension on the performance of our classifiers (Section 7.1);

2. the second ones to test these classifiers on some "true" dissimilarity data, i.e. given through a table of pre-computed pairwise dissimilarities (Section 7.2).

In each type of experiments, after being investigated internally, our classifiers were also compared to some reference existing classifiers suited for dissimilarity data. These comprise:

- 1-NN: *Nearest Neighbor* classifier.

- RLDA-D: *Regularized Linear Discriminant Analysis for Dissimilarities* classifier. This is the usual LDA classifier [20] applied to dissimilarity data viewed as vectors in an n dimensional Euclidean space, where the n coordinates equal, for each object, its respective dissimilarities with the n objects of the Training Data Set. The regularization is needed in this context because the size of the Training Data Set does not exceed the space dimension n, rendering the common class covariance matrix singular and, thus, impossible to invert as requested in the LDA classifying engine. The regularization is obtained by adding a positive constant λ to the diagonal elements of the estimated common class covariance matrix which then becomes invertible. More precisely, let $\widehat{s}$ be the biggest of those elements, and $\widehat{\lambda} = \widehat{s}/200$. Then, in our experiments, the results presented for RLDA-D are those obtained with the value of λ yielding the highest success rate among the set of values $\{\widehat{\lambda}, 2\widehat{\lambda}, \ldots, 15\widehat{\lambda}\}$.

- RdQDA-D: *Regularized diagonal Quadratic Discriminant Analysis for Dissimilarities* classifier. When one constrains the covariance matrices in the different classes to be diagonal, the usual QDA classifier [20] becomes dQDA. And RdQDA-D is the version of dQDA applied to dissimilarity data as with RLDA-D above, except that here $\widehat{s}$ is the biggest coefficient among the diagonal elements of all estimated class covariance matrices, and $\widehat{\lambda} = \widehat{s}/500$. However, it would have been natural to consider here, instead, the RQDA-D version of QDA. This was not done because results in [33, 34] seems to suggest that RLDA-D nearly always significantly outperforms RQDA-D while being far cheaper, probably due to a high overparameterization of RQDA-D in this context. On the other hand, RdQDA-D seems a good compromise here to handle the case of possible different class covariance matrices, with much less coefficients to estimate.

- SVM-D: the *Support Vector Machine for Dissimilarities* acting on the same n-dimensional space as RLDA-D and RdQDA-D above. The implementation of the support vector machinery used here was that of the R package `e1071` [9], which is based on [7]. We used the function `svm` of that package with default settings.

7.1 M1.BC/M2.BC vs. dissimilarity type/intrinsic data dimension

7.1.1 The experimental framework

Our first type of numerical experiments uses 4 data sets (described in Section 7.1.2), both real and simulated ones, each given through their coordinates in an Euclidean space $\mathbb{R}^d$. Such a data set is first converted to a dissimilarity matrix by means of a chosen distance in $\mathbb{R}^d$ computed between pairs of objects in the set. The objects are then classified by applying M1.BC, M2.BC and the aforementioned competing classifiers to the derived dissimilarity matrix. For each classifier, the success rate is estimated through 5-blocks Cross Validation, alongside the standard deviation of that estimation. Moreover, M1.BC and M2.BC were applied for values of the embedding dimension p ranging from 1 to 12. This allows to weigh the effect of the data intrinsic dimension in the classification performance of these two new classifiers, since this dimension is known for these data sets, at least the simulated ones.

On each data set, 4 distances have been put in competition, in order to assess the effect of the dissimilarity type on our 2 classifiers performance. The competing distances in our experiments are (where $X = (x_i)$ and $Y = (y_i) \in \mathbb{R}^d$):

1. the Euclidean or L^2-distance: $d_2(X,Y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$;
2. the absolute deviation or L^1-distance: $d_1(X,Y) = \sum_{i=1}^d \mid x_i - y_i \mid$;
3. the maximum norm or L^∞-distance: $d_\infty(X,Y) = \max_{i=1}^d \mid x_i - y_i \mid$;
4. a kind of normalized L^1-distance: $d_{1,st}(X,Y) = \sum_{i=1}^d \frac{\mid x_i - y_i \mid}{1 + \mid x_i - y_i \mid}$.

The interest in this last distance lies in the fact that it somehow tends to avoid that bigger coordinates outweighs the others in the classification process. Such a manoeuvre is also likely to augment chances that the computed distances come closer to be normally distributed, which is a pairwise assumption in our models. This is a consequence of the Central Limit Theorem (CLT) of Probability Theory which asserts that the sum of d independent random variables tends to be normally distributed when d is (moderately) large and none of the random variables variances dominates the other ones in magnitude. An argument already advocated in [33, 34] where, however, standardization is achieved using the more classical statistical approach of subtracting the mean of each coordinate and dividing by its standard deviation, both estimated from the training sample. One then uses d_1 or d_2 to compute the pairwise distances. Another alternative to $d_{1,st}$ could have been the Canberra distance, but this is rather suited for data with positive attributes.

7.1.2 The data sets

The data sets considered were:

Data set S1: A simulated data set [20, page 301] consisting of two classes of 100 observations each obtained as follows:

- For class 1, the 100 observations are iid vectors in $\mathbb{R}^{10}$ which coordinates are iid univariate standard Gaussian constrained so that these vectors have each a squared Euclidean norm $> \chi^2_{10}(0.5) \simeq 9.34$.

- For class 2, the 100 observations are iid vectors in $\mathbb{R}^{10}$ which coordinates are iid univariate standard Gaussian.

Data set S2: Another simulated data set of 225 observations consisted in 5 Gaussian classes in $\mathbb{R}^4$ with the following characteristics:

- Class 1: mean $= (10, 12, 10, 12)$, variance $= \mathbf{I}_4$, $\widehat{\pi}_1 = 0.12$;
- Class 2: mean $= (8.5, 10.5, 8.5, 10.5)$, variance $= \mathbf{I}_4$, $\widehat{\pi}_2 = 0.16$;
- Class 3: mean $= (12, 14, 12, 14)$, variance $= \mathbf{I}_4$, $\widehat{\pi}_3 = 0.2$;
- Class 4: mean $= (13, 15, 7, 9)$, variance $= 4\mathbf{I}_4$, $\widehat{\pi}_4 = 0.24$;
- Class 5: mean $= (7, 9, 13, 15)$, variance $= 9\mathbf{I}_4$, $\widehat{\pi}_5 = 0.28$.

This gaussian mixture is attributed to Bozdogan [4] who used it to investigate his ICOMP criterion and pointed out that the resulting mixture contains 5 highly overlapping classes.

Data set S3: The training sample of the `waveform` data found on the accompanying website of the book [20], and credited to Breiman et al. [5]. The sample is of size 300 split in 3 classes of respectively 106, 94 and 100 observations, given in dimension $d = 21$.

Data set S4: The test sample of the `ZIP Code` data found on the accompanying website of the book [20]. These are deslanted and normalized 16×16 grayscale images of handwritten digits, automatically scanned from envelopes by the U.S. Postal Service, and due to the neural network group at AT&T research labs [24]. The data are thus given in dimension $d = 256$. The sample is of size 2007. Since this was too huge for the memory requirements of our PC on which all the computations were performed, we extracted a random subsample of size 400 while respecting the proportions of the 10 decimal digits (which constitute the classes to recognize) in the test sample.

We point out that, to be consistent, for data sets for which both a training and a test samples were available, we used only either the training sample or the test sample (or a randomly extracted part of one of these), estimating the success rate through 5-blocks Cross validation on this sample as for the other data sets, alongside a standard error estimate of that success rate estimate.

It needs also to be emphasized that, for real world data, the dimension d at which a data set was sampled needs not be the true intrinsic dimension of those data. To distinguish, we shall call the former the *sampling dimension*. The intrinsic dimension might be much smaller.

The results are summarized in Table 1.

7.1.3 Comment on the results: M1.BC/M2.BC on the Euclidean tables

The Euclidean distance needs to be singled out because each of Models 1 and 2, on which our classifiers are based, essentially approximates any given dissimilarity table by an Euclidean one, assuming an error of a particular pairwise gaussian type. One would then expect a more predictable behavior of our classifiers on Euclidean tables. Now, two striking unexpected surprises appear in Table 1 (except for the `waveform` and `zip Code` data for which the arbitrary upper bound 12 imposed on the embedding dimension somewhat hinders the observation) and in our other experiments not mentioned here:

Fact 1. *The best success rate is not achieved at the data intrinsic dimension.*

Fact 2. *The success rate at that dimension is shockingly low, and often the lowest (or close to so)!*

These two facts are not unrelated, but not exactly identical. The easier to explain is the second one. For M1.BC, it stems from formulas (30)-(31) and the well known phenomenon in computer

Table 1: Success rates (%), with standard errors, for various dissimilarities classifiers and different distances used on Data Sets S1 to S4.

		M1.BC	M2.BC	1-NN	RLDA-D	RdQDA-D	SVM-D
S1 ($d = 10$)	d_2	73 ± 6.1 (1) (54 ± 3.6)	90 ± 1.4 (3) (51.5 ± 4.3)	61.5 ± 4.8	92 ± 1.2	93.5 ± 1	93 ± 2.3
	d_1	77 ± 1.2 (3) (73.5 ± 2.3)	87 ± 1.5 (2) (78.5 ± 1.5)	67 ± 5	88.5 ± 1.3	92 ± 0.9	91.5 ± 2
	d_∞	76.5 ± 4.2 (9) (77 ± 3.2)	85 ± 1.4 (2) (76.5 ± 4.1)	58.5 ± 4.3	86 ± 1.3	89 ± 2.6	88 ± 2.7
	$d_{1,st}$	72 ± 3.3 (1) (61.5 ± 3.6)	80.5 ± 2.2 (3) (67 ± 3.5)	63.5 ± 5.6	83.5 ± 1.9	83.5 ± 3.8	89.5 ± 2.2
S2 ($d = 4$)	d_2	93.8 ± 1.6 (1) (22.2 ± 2.6)	93.3 ± 2.4 (1) (36.9 ± 4.4)	91.6 ± 2.4	94.7 ± 1.7	93.3 ± 1.8	95.6 ± 2
	d_1	93.8 ± 1.1 (2) (76.4 ± 3.6)	93.8 ± 2.3 (1) (90.2 ± 1.5)	90.7 ± 2.3	90.7 ± 1.6	93.3 ± 1.8	95.1 ± 2
	d_∞	86.2 ± 2.4 (1) (71.1 ± 2.6)	91.6 ± 1.9 (1) (87.1 ± 0.8)	88.9 ± 2.7	92.4 ± 0.5	94.7 ± 1.5	95.1 ± 1.8
	$d_{1,st}$	89.3 ± 1.6 (3) (86.7 ± 2.5)	91.1 ± 2.2 (4) (91.1 ± 2.2)	85.3 ± 2.7	86.2 ± 1.8	94.2 ± 2.1	94.2 ± 1.8
S3 ($d = 21$)	d_2	81.7 ± 2.9 (4)	82.3 ± 0.7 (3)	73.3 ± 1.3	83 ± 1.3	77 ± 1.6	81.7 ± 1.6
	d_1	79 ± 2.4 (2)	80 ± 2 (2)	73.7 ± 0.6	78.7 ± 1.3	74.7 ± 1.4	82 ± 2.1
	d_∞	82.7 ± 0.9 (2)	82.3 ± 2.5 (2)	71 ± 3	79.3 ± 1.2	76.7 ± 0.5	77.7 ± 1.2
	$d_{1,st}$	74.3 ± 2.4 (2)	76.7 ± 1.7 (2)	72.7 ± 1.7	78 ± 1.5	75.7 ± 0.8	78.7 ± 2.4
S4 ($d = 256$)	d_2	63.8 ± 3.9 (11)	70.2 ± 1.2 (8)	83.2 ± 0.8	87.8 ± 1.3	71.2 ± 1	83.8 ± 1.7
	d_1	57 ± 0.6 (4)	70 ± 1.4 (6)	82.8 ± 0.8	83.8 ± 3.1	64.8 ± 2.3	81 ± 2.2
	d_∞	——	——	56 ± 0.7	40.8 ± 1.5	58.2 ± 0.5	22 ± 2.8
	$d_{1,st}$	65.2 ± 3.5 (11)	70.5 ± 2.5 (8)	80.5 ± 0.9	83 ± 2.6	63.8 ± 2.2	80.5 ± 2.2

S1: data simulated in [20, page 301]; S2: `Bozdogan's mixture`; S3: `waveform`; S4: `zipCode`;
d: data sampling dimension; —— : numerical instabilities.

Please NOTE: For M1.BC and M2.BC on data sets S1 and S2, the first line gives the highest success rate using the "within one standard error of the maximum rate towards model parsimony" rule. and the corresponding embedding dimension, while the second line gives the success rate at the sampling dimension d.

Table 2: Success rates (%), with standard errors, for various dissimilarities classifiers on Data Sets `Jeffreys` and `proteins`.

	M1.BC	M2.BC	1-NN	RLDA-D	RdQDA-D	SVM-D
`Jeffreys`	67.9 ± 0.9 (2)	79.5 ± 2.7 (1)	72.1 ± 2.9	78.7 ± 2	71.3 ± 2.1	79.1 ± 2.8
`proteins (5 cl.)`	71 ± 2.2 (5)	75.6 ± 2.3 (3)	77 ± 2.4	84.4 ± 0.7	72.2 ± 2.3	84.7 ± 1
`proteins (2 cl.)`	92 ± 0.6 (4)	93.5 ± 0.9 (5)	92.4 ± 1	94.7 ± 1.5	92 ± 1.2	94.9 ± 1

arithmetic called *numerical cancellation*. Indeed, starting with dissimilarities d_{ij} which are exact Euclidean distances, when they are projected at the right dimension, the MDS algorithms provide a close to exact configuration of points for those distances. It then comes that the computed Euclidean distances $\delta_{ij}\left(\underline{\widehat{X}}\right)$ between points in the obtained configuration $\underline{\widehat{X}}$ can result, for certain data sets, in close to machine precision approximations to the starting distances d_{ij}'s. Whence the differences $d_{ij} - \delta_{ij}\left(\underline{\widehat{X}}\right)$ are numerically meaningless quantities, just being an accumulation of rounding errors (which, as is well documented [22], seldom exhibit any probabilistic random pattern). So, in that situation, the coefficients σ_{kl}^2 are estimated by (30)-(31) with no correct significant digit, and any information about the classes is thus lost during the Learning Phase. This results in a classification nearly always done to the widest class, if any. When this happens, it remains so at higher dimensions. The same reasoning applies to Model 2 as well because, in that model, when projecting exact Euclidean distances at the right dimension, the estimated coefficients $\widehat{a}_{kl}$ quite often come very close to 1, as is obvious from formulas (37)-(39), which brings us back to the case of Model 1 just discussed.

To summarize, by their very structures (4) and (5), our models actually classify according to dissimilarities error measurement w.r.t. Euclidean distances. Now, when the error measurement w.r.t. Euclidean distances is negligible, our models are lost in their classification endeavors, having no concrete information about the classes to rely upon. At present, our solution to this somewhat disturbing phenomenon is the Cross validation dimension selection procedure, which gives interesting results in the above examples. However, for the data sets in these examples, since they were initially given through their attributes in an Euclidean space, an obvious alternative exists to try to cope around the identified problem. It is to use a distance other than the Euclidean in computing their pairwise dissimilarity matrix to which one then applies our classification procedures, which is discussed hereafter. For the future, other solutions of wider applicability are being investigated for this problem.

Fact 1 is more interesting because it exhibits that a data set may be better separated, and thus classifiable, when projected at a dimension much lower than its intrinsic dimension. Indeed, in many of our numerical experiments, the best classification rate was recorded at dimension 1, especially for M1.BC. One can regard this as a serious compensation for the deficiency posed by **Fact 2**.

7.1.4 Comment on the results: M1.BC/M2.BC for the other distances

The most obvious general pattern discernable in the results for non Euclidean tables is the clear superiority of M2.BC over M1.BC, except in some few cases. This is observed in terms of the highest rate across dimensions as well as the overall dimensionwise comparison of success rates. Which confirms the *a priori* feeling that M2.BC has an intrinsically greater flexibility to adapt to the type of dissimilarity computed on the data. This classifier thus appears much less adversely affected by the departure of the dissimilarity from Euclideanicity. The farther the dissimilarity is to the Euclidean distance, the positively higher is the difference between M2.BC success rate and that of M1.BC. This can be seen through the fact that this difference is generally smallest for Euclidean tables, followed by L^1-distance tables.

Another general pattern is that the effect of the intrinsic data dimension seems completely blurred and hardly interpretable when applying M1.BC and M2.BC to classify non Euclidean tables. This somewhat unexpected phenomenon is probably worth an investigation in the future.

On the other hand, no distance seems to exhibit a discernable superiority or inferiority over the others in terms of classification performance. Nevertheless, the two ones less closest to the Euclidean (i.e. d_∞ and $d_{1,st}$) appear to present the most dimensionwise stable classification results (i.e. smaller success rate variability across varying values of the embedding dimension p). And, when it does not cause instabilities in the MDS algorithms, d_∞ comes the closest to rank first in terms of overall classification performance.

A first explanation to the complete numerical instability recorded by M1.BC/M2.BC in the MDS algorithms on the d_∞ distance table for the `zipCode` data (for all embedding dimensions) seems to stem

from the fact this table contains only a small proportion of distinct dissimilarities (a few hundreds out of several tens of thousands). Indeed, in many instances, the d_∞ distance behaves more like a discrete distance, which appears to be an advantage in the M1.BC/M2.BC classifying methodology since the error with respect to the Euclidean distance is then significant (see the Comment in Section 7.1.3) and explains the generally good performance. However, when the discreteness character is too severe for a given data set, the SMACOF algorithm we used to solve our MDS problems is in irrecoverable trouble. A solution here would be to use other MDS algorithms, which is currently under investigation.

Finally, the $d_{1,st}$ distance is to be singled out here for two reasons:

- For yet to be clarified reasons, the success rate for both M1.BC and M2.BC at the embedding dimension $p = 1$ is very often abnormally low for dissimilarity tables computed with that distance.
- On the other hand, for other embedding dimensions, M1.BC success rate generally exhibits very small variability for $d_{1,st}$, and comparable to the best success rates for other distances on each data set. The same for M2.BC success rate.

It thus looks as if, except when the embedding is at dimension 1, this distance significantly smoothes away the effect of the intrinsic data dimension. This is quite an interesting phenomenon which needs exploration and confirmation in the future. This might suggest that classifying dissimilarities and finding their intrinsic data dimension are not inherently related issues.

7.1.5 Comment on the results: M1.BC/M2.BC vs. other dissimilarities classifiers

Since M2.BC nearly always outperforms M1.BC, we restrict the comparison here to a confrontation between the former and already existing classifiers. The keys issues here are: classification performance and sensitivity to the dissimilarity data type. Nevertheless, before proceeding, one should not overlook, in comparing them with other classifiers for dissimilarity data, the distinctive feature that M1.BC and M2.BC in a sense try to optimize the MDS embedding of these data with regard to classification purposes.

For the success rate, on the considered data sets, M2.BC is nearly always better than 1-NN (only exception here: the `zipCode` data, see hereafter), overall slightly worse than RLDA-D and SVM-D, and close to a dead heat with RdQDA-D. It turns out that M2.BC is disfavored mainly here by its performance on the `zipCode` data. Now, this data set was mostly included in our experiments to have a first idea of our classifiers behavior on a data set gathered at a high sampling dimension. The results show that there is still work to do in order to better handle the intrinsic data dimension problem in our methodology aimed at efficiently combining the classification and the embedding tasks for dissimilarity data.

On the ground of the sensitivity to the dissimilarity data type, for these data sets and the distances considered, M2-BC is neck to neck with RLDA-D, RdQDA-D and SVM-D. But, probably, more extensive experiments are needed here to deliver a conclusive statement.

7.2 Type 2 experiments: true dissimilarity data sets

Here, our classifiers are applied to a small set of "true" dissimilarity data matrices, i.e. with no pre-given coordinates in an Euclidean space.

We considered two dissimilarity data sets named "Jeffreys" and "proteins" of unknown intrinsic data dimension. The `Jeffreys` data (distances between images from the Corel Database and described in [16]) consist of 473 observations spread in 4 classes of 102, 110, 140 and 121 observations respectively, and were already used in [15]. The `proteins` data consist of 569 observations spread in 4 classes of 298, 151, 23 and 97 observations respectively. For these latter data, we tested our classifiers both on the whole data set and on the subset consisting of the 2 most probable classes. This last subset of

449 observations is the one considered in [15]. The motivation behind such a restriction is that, quite often, for proteins data, a premier preoccupation is to find whether there are not too many classes, and if, indeed, some classes would not better be merged.

The results are gathered in Table 2. The main feature is that M2.BC exhibits a clear superior discriminatory power over M1.BC, although more striking for the `Jeffreys` data, and somewhat less for the `proteins` data. Compared to other classifiers, M2.BC outperforms RdQDA-D on these data, slightly dominates also 1-NN, and lags behind RLDA-D and SVM-D on the full proteins data set (with 5 classes).

In the whole "proteins" data set, two classes nearly overlap. That is why they are harder to classify. They should probably better be merged.

8 Conclusion

In this work, we developed two new methods for classifying objects on the basis of their pairwise dissimilarities. Each method is derived from a postulated probability model for such data. These methods thus yield two model-based classifiers constructed on purely statistical grounds (mainly ML estimation), avoiding heuristics of any kind. A key assumption in that construction is that the unobserved objects are regarded as parameters lying in an Euclidean space which are estimated during the learning phase through an iterated MDS algorithm, alongside other model parameters. Using the unknown dimension of the Euclidean space as a tuning parameter in the classification allows to choose the dimension with the highest success rate for each method by Cross Validation.

The small number of experiments performed on real or simulated data sets suggest that this new approach for dissimilarity data classification can be regarded as a promising step in attempts aimed at devising a general purpose classifier for (arbitrary) dissimilarity data. And, generally, as expected, the Model 2 based classifier exhibits a better generalization performance than the one based on Model 1. Nevertheless, further experiments with more varied data sets are needed to better assess the two proposed methods for the classification of such data. Finally, while missing dissimilarities can be handled by our classifiers as usual in MDS methodology by giving them zero weights, an important aspect of dissimilarities-based classification is not yet addressed in our work: that of prototypes selection (see, e.g., [26, 37]). This shall be part of our planned future research.

Acknowledgements. The authors wishes to thank Gilles Celeux and Jean-Michel Marin who introduced them in this research theme, and provided an invaluable insight into it all along, with many fruitful discussions resulting in useful critiques which helped in the shaping of the provided solutions. Thanks also go to Elizabeth Gassiat who took patience to read and comment an early draft of this work. The same to Anne Guérin-Dugué who kindly provided the Jeffreys and proteins data. And, not the least, the first author wishes to express warm thanks to Prs. D. Dacunha-Castelle and H. Gwét for having encouraged him to engage in this research and Pr. Jean Coursol for having initiated him in the Classification and Data Mining areas and the use of the R Statistical Computing System through a course he gave at the Ecole Polytechnique of Yaoundé in June 2004. He is also indebted to Patrick Jakubowicz and Yves Misiti for assistance with computing facilities at Orsay.

References

[1] Balachander, T. and Kothari, R. (1999). "Introducing Locality and Softness in Subspace Classification", *Pattern Analysis & Applications*, 2(1):53-58.

[2] Borg, I. and Groenen, P.J.F. (1997). "Modern Multidimensional Scaling. Theory and Applications", *Springer Series in Statistics*, Springer, New York, NY, USA.

[3] Bottigli, U., Golosio, B., Masala, G.L., Oliva, P., Stumbo, S., Cascio, D., Fauci, F., Magro, R., Raso, G., Vasile, M., Bellotti, R., De Carlo, F., Tangaro, S., De Mitri, I., De Nunzio, G., Quarta, M., Preite Martinez, A., Tata, A., Cerello, P., Cheran, S.C. and Lopez Torres, E. (2006). "Dissimilarity Application in Digitized Mammographic Images Classification", *Journal of Systemics, Cybernetics & Informatics*, 4(3):18-22.

[4] Bozdogan, H. (1993). "Choosing the Number of Component Clusters in the Mixture-Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix", in O. Opitz, B.Lausen, and R.Klar (eds.), *Studies in Classification, Data Analysis, and Knowledge Organization*, 40-54, Springer-Verlag, Heidelberg, Germany.

[5] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). "Classification and Regression Trees", Wadsworth.

[6] Celeux, G. (2003). "Analyse discriminante", in G. Govaert (Ed.), *Analyse des données*, 201-234, Lavoisier, Paris, France.

[7] Chang, C.-C. and Lin, C.-J. (2003). "LIBSVM: a library for Support Vector Machines", *Technical Report*, National Taiwan University, Taipei, Taiwan [http://www.csie.ntu.edu.tw/~cjlin/libsvm].

[8] Dickinson, P.J., Bunke, H., Dadej, A. and Kraetzl, M. (2004). "Object-Based Image Content Characterisation for Semantic-Level Image Similarity Calculation", *Pattern Analysis & Applications*, 7(3):243-254.

[9] Dimitriadou, E., Hornik, K., Leisch, F. and Meyer, D. (2006). "e1071: Misc Functions of the Department of Statistics (e1071)", R *package*, version 1.5-16, TU Wien, Vienna, Austria.

[10] Duin, R.P.W., Pękalska, E., Paclík, P. and Tax, D.M.J. (2004). "The dissimilarity representation, a basis for domain based pattern recognition?", *Representations in Pattern Recognition*, IAPR Workshop, Cambridge, 43:56.

[11] Fournier, J., Cordi, M., and Philipp-Foliguet, S. (2001). "RETIN: A Content-Based Image Indexing and Retrieval System", *Pattern Analysis & Applications*, 4(2-3):153-173.

[12] Fukunaga, K. (1990). "Introduction to statistical pattern recognition", *2nd ed.*, *Computer Science and Scientific Computing Series*, Academic Press, Inc, Boston, MA, USA.

[13] Glunt, W., Hayden, T.L. and Liu, W.-M. (1991). "The embedding problem for predistance matrices", *Bulletin of Mathematical Biology*, 53:769-796.

[14] Gower, J.C. (1966). "Some distance properties of latent root and vector methods in multivariate analysis", *Biometrika*, 53:315-328.

[15] Guérin-Dugué, A. and Celeux, G. (2001). "Discriminant Analysis on Dissimilarity Data: A New FastGaussian like Algorithm", *AISTAT 2001*, Florida, USA.

[16] Guérin-Dugué, A. and Oliva, A. (2000). "Classification of Scene Photographs from Local Orientation features", *Preprint.*

[17] Guttman, L. (1968). "A general nonmetric technique for finding the smallest coordinate space for a configuration of points", *Psychometrika*, 33:469-506.

[18] Haasdonk, B. and Bahlmann, C. (2004). "Learning with Distance Substitution Kernels", *Proc. 26th DAGM Symposium (Tübingen, Germany)*, 220-227, Springer, Berlin, Germany.

[19] Harol, A., Pękalska, E., Verzakov, S. and Duin, R.P.W. (2006). "Augmented embedding of dissimilarity data into (pseudo-)Euclidean spaces", *Joint IAPR International Workshops on Statistical and Structural Pattern Recognition (Honk Kong, China)*, Lecture Notes in Computer Science, 4109:613-621.

[20] Hastie, T., Tibshirani, R. and Friedman, J. (2001). "The Elements of Statistical Learning. Data Mining, Inference and Prediction", *Springer Series in Statistics*, Springer, New York, NY, USA [http://www-stat.stanford.edu/~tibs/ElemStatLearn].

[21] Heiser, W.J. and de Leeuw, J. (1986). "SMACOF-I", *Technical Report UG-86-02*, Department of Data Theory, University of Leiden, Leiden, The Netherlands.

[22] Higham, N.I. (2002). "Accuracy and Stability of Numerical algorithms", 2nd ed., *Society for Industrial and Applied Mathematics*, Philadelphia, PA, USA.

[23] Kearsley, A.J., Tapia, R.A. and Trosset, M.W. (1998). "The Solution of the Metric STRESS and SSTRESS Problems in Multidimensional scaling Using Newton's Method", *Computational Statistics*, 13(3):369-396.

[24] Le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L. (1990). "Handwritten digit recognition with a back-propagation network", in D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, Vol. 2, Morgan Kaufman, Denver, CO, USA.

[25] de Leeuw, J. (1988). "Convergence of the majorization method for multidimensional scaling", *Journal of Classification*, 5:163-180.

[26] Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pękalska, E. and Duin, R.P.W. (2006). "Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces", *Pattern Recognition*, 39:1827-1838.

[27] Malone, S.W., Tarazaga, P. and Trosset, M.W. (2002). "Better Initial configurations for Metric Multidimensional Scaling", *Computational Statistics and Data Analysis*, 41:143-156.

[28] Malone, S.W. and Trosset, M.W. (2000). "Optimal dilations for metric multidimensional Scaling", In *2000 Proceedings of the Statistical Computing Section and Section on Statistical Graphics*, American Statistical Association, Alexandria, VA, USA.

[29] Martins, A., Figueiredo, M. and Aguiar, P. (2007). "Kernels and similarity measures for text classification", *6th Conference on Telecommunications - ConfTele2007*, Peniche, Portugal.

[30] Masala, G.L. (2006). "Pattern Recognition Techniques Applied to Biomedical Patterns", *International Journal of Biomedical Sciences*, 1(1):47-55.

[31] Orozco, M., García, M.E., Duin, R.P.W. and Castellanos, C.G. (2006). "Dissimilarity-Based Classification of Seismic Signals at Nevado del Ruiz Volcano", *Earth Sciences Research Journal*, 10(2):57-65.

[32] Paclík, P. and Duin, R.P.W. (2003). "Dissimilarity-based classification of spectra: computational issues", *Real-Time Imaging*, 9:237-244.

[33] Pękalska, E. and Duin, R.P.W. (2000). "Classifiers for dissimilarity-based pattern recognition", in A. Sanfeliu, J.J. Villanueva, M. Vanrell, R. Alquezar, A.K. Jain (eds.), *Proc. 15th Int. Conference on Pattern Recognition (Barcelona, Spain), vol. 2, 12-16*, Pattern Recognition and Neutral Networks, IEEE Computer Society Press, Los Alamitos, CA, USA.

[34] Pękalska, E. and Duin, R.P.W. (2000). "Classification on dissimilarity data : A first look", in L.J. Van Vliet, J.W.J. Heinjnsdijk, T. Kielman, P.M.W. Knijnenburg (eds.), *Proc. Annual Conference of the Advanced School for Computing and Imaging (Lommel, Belgium), 221-228*, Pattern Recognition and Neutral Networks, IEEE Computer Society Press, Los Alamitos.

[35] Pękalska, E. and Duin, R.P.W. (2002). "Dissimilarity representations allow for building good classifiers", *Pattern Recognition Letters*, 23(8):943-956.

[36] Pękalska, E. and Duin, R.P.W. (2006). "Dissimilarity-based classification with vectorial representations", *International Conference on Pattern Recognition*, vol. 3:137-140, Hong Kong.

[37] Pękalska, E., Duin, R.P.W. and Paclík, P. (2006). "Prototype selection for dissimilarity-based classifiers", *Pattern Recognition*, 39(2):189-208.

[38] Pękalska, E., Paclík, P. and Duin, R.P.W. (2002). "A Generalized Kernel Approach to Dissimilarity-Based Classification", *Journal of Machine Learning Research, Special Issue on Kernel Methods*, 2(2):175-211.

[39] The R Development Core Team (2007). "R: A Language and Environment for Statistical Computing. Reference Index", Version 2.5.0, R Foundation for Statistical Science.

[40] Ramsay, J.O. (1982). "Some Statistical Approaches to Multidimensional Scaling Data", *Journal of the Royal Statistical Society*, Ser. A, 145:285-312.

[41] Srisuk, S., Petrou, M., Kurutach, W. and Kadyrov, A. (2005). "A face authentication system using the trace transform", *Pattern Analysis & Applications*, 8(1-2):50-61.

[42] Tolba1, A.S., and Abu-Rezq, A.N. (1998). "Arabic glove-talk (AGT): A communication aid for vocally impaired", *Pattern Analysis & Applications*, 1(4):218-230.

[43] Torgerson, W.S. (1952). "Multidimensional scaling: I. Theory and method", *Psychometrika*, 17:401-419.

[44] Young, G. and Householder, A.S. (1938). "Discussion of a set of points in terms of their mutual distances", *Psychometrika*, 3:19-22.

APPENDIX:
The computation of an initial approximation $\widehat{U}_{k,0}$ to minimize $\widehat{H}_k(U)$

Since we intend to compute $U = \widehat{U}_k$, the point at which the function $\widehat{H}_k(U)$ defined by (45) reaches its minimum value, intuitively this minimum should satisfy, as closely as possible, the approximate equalities:

$$\| U - \widehat{X}_i \| \approx d_i, \text{ for } i = 1, \ldots, n,$$

which implies, by squaring and expanding the squared Euclidean norms of the differences:

$$2 < \widehat{X}_i, U > - \| U \|^2 \approx \| \widehat{X}_i \|^2 - d_i^2, \text{ for } i = 1, \ldots, n,$$

or

$$2\,\widehat{x}_{i1}u_1 + \cdots + 2\,\widehat{x}_{ip}u_p - u_{p+1} \approx \| \widehat{X}_i \|^2 - d_i^2, \text{ for } i = 1, \ldots, n, \tag{50}$$

where $U = (u_1, \ldots, u_p)^T$, $u_{p+1} = \| U \|^2$ and $\widehat{X}_i = (\widehat{x}_{i1}, \ldots, \widehat{x}_{ip})$, for $i = 1, \ldots, n$. Now, (50) can be regarded as an approximate linear system of n equations in the $p+1$ scalar unknowns $u_1, \ldots, u_{p+1}$. The unknown u_{p+1} introduces a nonlinear constraint: $u_{p+1} = u_1^2 + \cdots + u_p^2$. This constraint can be eliminated by suppressing an arbitrarily chosen equation from (50) while subtracting it from all the

remaining $n-1$ ones, therefore obtaining an approximate linear system of $n-1$ equations in the p unknowns $u_1, \ldots, u_p$, which one then solves by Least Squares. Instead we chose the simpler strategy to directly solve (50) by Least Squares and merely discard the estimation obtained for u_{p+1}.

In any event, whatever the strategy chosen between the two outlined above to approximately solve (50), it is important to note that the matrix of the solved Least Squares system is entirely determined by the estimated learning configuration $\underline{\widehat{X}}$. Thus, its QR factorization can be computed once for all after the Learning Phase for use throughout the Prediction Phase to quickly solve (50) for each new explicit observation $\underline{d}$ to classify.

However, since $n \gg p$, there are obvious far more economical ways to approximately solve (50). For instance (see [23, 27]), one could retain just $p+1$ equations among the n in (50) and solve a (hopefully) Cramer system for $u_1, \ldots, u_{p+1}$ (or for $u_1, \ldots, u_p$ by first suppressing one of the $p+1$ equations and subtracting it from the p others). The reason we chose not to follow these cheap paths is twofold:

1. One would have to devise a cheap selecting criterion for the $p+1$ equations among the initial n in (50). Now, the two simplest choices are that of the first $p+1$ ones or a random choice, but this may result in undesirable effects for the numerical stability of the resulting system in certain cases and/or the convergence of the numerical minimization algorithm starting from the obtained initial approximation of $\widehat{U}_k$. On the other hand, more sophisticated selecting criteria may imply a significant overhead in computation with no appreciable gain over the use of all the equations. Whereas, since $n \gg p$, using all the equations almost certainly guarantees that we least squarely solve a linear system of full rank p.

2. By using all the equations in (50) for computing an initial approximation $\widehat{U}_{k,0}$ to $\widehat{U}_k$, we increase our chances of obtaining a good one in order to guarantee fast convergence in the numerical nonlinear minimizer used to minimize the function $\widehat{H}_k(U)$ defined by (45).

However, pushing this latter argument further, one may legitimately argue that, considering the structure of the function $\widehat{H}_k(U)$, weighted least squares should be the right strategy in solving our suggested linear systems approximately to obtain $\widehat{U}_{k,0}$. This is right, but would entail a serious increase in computational cost, since the weights would then vary according to each implicit object U and each possible group label in $\{1, \ldots, G\}$.