The DD^G-classifier in the functional setting

J.A. Cuesta–Albertos*
Dpt. of Mathematics, Statistics and Computation, Univ. of Cantabria, Spain
M. Febrero–Bande† and M. Oviedo de la Fuente†
Dpt. of Statistics and Op. Res. Univ. of Santiago de Compostela, Spain

June 4, 2014

Abstract

The Maximum Depth was the first attempt to use data depths instead of multivariate raw data to construct a classification rule. Recently, the DD–classifier has solved some of the several serious limitations of the Maximum Depth classifier but some issues still remain. This paper is devoted to extend the DD–classifier in the following ways: first, to surpass the limitation of the DD–classifier when more than two groups are involved. Second to apply regular classification methods (like kNN, linear or quadratic classifiers, recursive partitioning,...) to DD–plots to obtain useful insights through the diagnostics of these methods. And third, to integrate different sources of information (data depths or multivariate functional data) in a unified way in the classification procedure. Aside, as the DD–classifier trick is specially useful in the functional framework, an enhanced revision of several functional data depths is done in the paper. A simulation study and the application to some classical real datasets are also provided showing the power of the new proposal.

Keywords: DD–Classifier, Functional Depths, Functional Data Analysis

1 Introduction

In the one-dimensional case, it is easy to use the natural order to order the points in the full space with respect to a probability distribution $\mathbf{P}$, with the median being the innermost point and the

*Research partially supported by the Spanish Ministerio de Ciencia y Tecnología, grants MTM2011-28657-C02-02

†Research partially supported by the Spanish Ministerio de Ciencia e Innovación, grants MTM2008-03010

extreme percentiles the outermost points. Moreover, if F_P denotes the distribution function of $\mathbf{P}$, then the index

$$D_P(x) = \min\{F_P(x), 1 - F_P(x)\} \tag{1}$$

measures how deep is a point $x \in \mathbb{R}$ with respect to $\mathbf{P}$. This index can also be applied to samples replacing F_P by the empirical distribution function. Other possibilities to define $D_P(x)$ are available (see, for instance, Subsection 3.1), including those satisfying that $D_P(x)$ decreases with the distance between x and the mean of $\mathbf{P}$, which, in turn, is the deepest point. Most of them are positive, bounded, and the bigger the index the deeper the point.

In the multidimensional case, there exists no natural order and, consequently, the ordering of the points from the inner to the outer part of a distribution or sample is not so easy. To overcome this difficulty, several indices have been proposed under the common name of "depths": Given a probability measure $\mathbf{P}$ defined on $\mathbb{R}^p$, $p \geq 1$, and $x \in \mathbb{R}^p$ a depth of x with respect to $\mathbf{P}$, $D_P(x)$, is an index telling how depth is x with respecto to $\mathbf{P}$. A nice review on multivariate depths can be found in Liu et al. (1999).

Those problems are more difficult to be solved in the functional case. However, there also exist several depths valid in this kind of space. We present some of them in Section 3 along with some extensions that allow to create new depths taking into account pieces of information from several sources.

Several applications of depths in multivariate and functional statistics have been proposed so far in the literature including exploratory and descriptive statistics, statistical tests of hypothesis,... In this paper, we are mostly interested in the possibilities that they offer in the supervised learning problem from a nonparametric point of view.

To the best of our acknowledgment, the first paper in which the depths were used for this goal was Liu (1990) where the maximum depth classifier was proposed. The idea is the following: Given two probabilities (or classes, or groups) $\mathbf{P}$ and $\mathbf{Q}$, and a depth, D, we classify the point x as produced by $\mathbf{P}$ if $D_P(x) > D_Q(x)$. This procedure was fully developed in Ghosh and Chaudhuri (2005).

A main tool in this paper are the DD–plots. They were introduced in Liu et al. (1999) for graphical comparisons of two multivariate distributions or samples based on data depth. A DD–plot is a two-dimensional graph (regardless the dimension of the space supporting the samples) in which, given

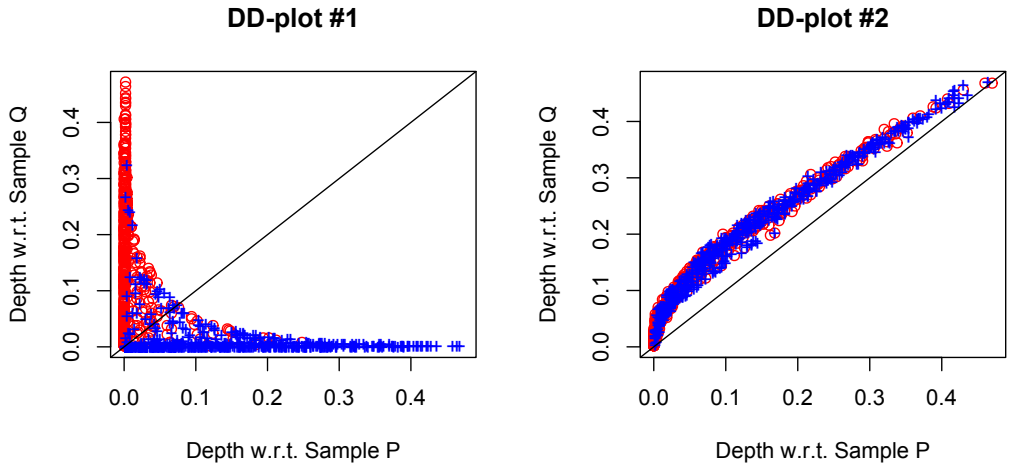

Figure 1: DD–plots of two samples coming from two-dimensional normal distributions. In both cases **P** is a standard 2-dimensional distribution. **Q** differs from **P** in the mean in the first DD–plot and in the covariance matrix in the second one.

two probability distributions, **P** and **Q** on $\mathbb{R}^p$, for every point $x \in \mathbb{R}^p$, the pair $(D_P(x), D_Q(x))$ is represented. We show examples of DD–plots in Figures 1 and 2.

It was shown in Liu et al. (1999) that some distributional differences, such as location, scale, skewness or kurtosis differences, are associated with characteristic patterns in the DD–plot. Therefore, DD–plots can provide simple diagnostic tools for visual comparisons of two samples of any dimension. For example, Li and Liu (2004) derives several nonparametric tests of multivariate locations and scales by detecting possible departures from the expected patterns of graphs in DD–plots.

In this paper we are interested in the application of DD–plots to classify points. Let us pay some attention to Figure 1. Both DD–plots correspond to samples coming from bidimensional normal distributions, where **P**, in both cases, is standard normal. The mean of **Q** in the first DD–plot is $(2, 2)^t$ and its covariance is the identity. In the other case **Q** is centered but its covariance is twice the identity. In both graphs, points in blue come from **P** while points in red were produced by **Q**. Both sample sizes were 500. In both graphs the main diagonal is also drawn. According to the maximum depth principle, points above (resp. below) this line are classified as produced by **Q** (resp. by **P**), i.e. the main diagonal splits the DD–plot in two classes which give the maximum depth classifier.

This classification procedure is optimal in the first case, but, obviously it is plainly wrong in the

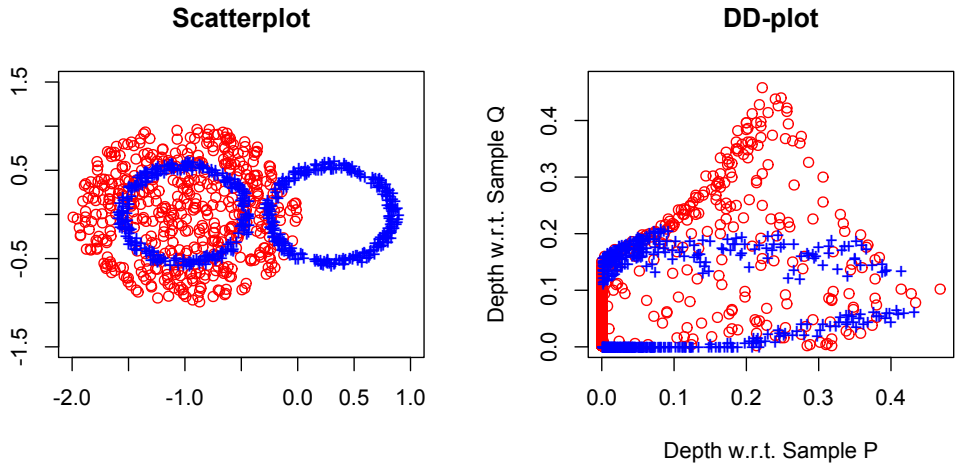

Figure 2: Scatterplot of two uniform samples and associated DD–plot.

second since it classifies almost all points as produced by distribution $\mathbf{Q}$. An interesting idea in Li et al. (2012) was to notice that, in spite of the maximum depth classifier fails, the DD–plot could contain information allowing to obtain a good classifier. For instance, in the second DD–plot in Figure 1 the proportion of red points is very high in the area close to the vertical axis. The authors of Li et al. (2012) proposed to replace the main diagonal by a function (in that paper only the polynomial case is fully developed) whose graph splits the DD-plot in two zones with the lowest missclassification rate. This is the so-called *DD-classifier*.

Obviously, the DD–classifier is a big step forward to solve the limitations of the maximum depth classifier and, in fact, in the problem we comment, according to Li et al. (2012), the DD–classifier gives a classification very close to the optimal one. However, there are some situations in which a function can not classify correctly the points. Let us consider the situation we present in Figure 2. The scatterplot represents two samples with sizes 2,000 each. Red points were taken from a uniform distribution, denoted by $\mathbf{Q}$, on the unit ball centered on $(-1, 0)^t$. Blue ones come from a mixture of two uniform probabilities with probability .5 each which is denoted by $\mathbf{P}$. First one is supported by the ring centered at $(-1, 0)^t$ and inner (resp. outer) radius equal to .5 (resp. .6) and the second one by the same ring but with center at $(0.3, 0)^t$. The optimal classifier classifies points in both two rings to $\mathbf{P}$ and the rest to $\mathbf{Q}$.

The associated DD–plot is also shown in Figure 2. It is obvious that no function can split the

DD–plot in two zones giving the optimal classifier since this would require a function separating the points in areas with blue points from the rest of the DD–plot and this is impossible. However, this goal can be achieved with a classifier like the kNN applied to the DD–plot.

Having those ideas in mind, the first objective of this paper is explore the possibility of applying regular classification methods (like kNN, linear or quadratic classifiers, recursive partitioning,...) to DD–plots and to obtain useful insights through the diagnostics of these methods. Additionally, we will to surpass the limitation of DD–classifier when more than two distributions are involved (the solution for this problem in Li et al. (2012) was to apply a majority voting scheme). Finally, we will be able to integrate different sources of information in the same classification procedure. The last goal is specially interesting in the functional context where some transformations of the original curves (as for example derivatives) could be computed and used for classification. By this, this paper is mainly focused in the functional setting although the procedure can also be directly applied to the multivariate setting. In particular, we want to remark that some of the diagnostic tools of the employed classification procedures can be used to asses the relevance of the available functional information. In order to avoid to enlarge too much the paper, we only carry out this idea in the second example in Section 4 where we conclude that the relevant information is contained in the second derivative of the involved curves.

This paper is organized as follows: In Section 2 we present the basic ideas below the proposed classifier. Section 3 is devoted to present some functional depths as well as to analyze some modifications which could improve them. In Section 4 we present two examples of several classification procedures applied to DD–plots. Section 5 contains the results of some simulations as well as the application to some real data sets. The paper ends with a discussion on the proposed method.

2 DD^G-Classifier

In Li et al. (2012), the DD–plot is defined, in the case in which only two groups are involved, as a two dimensional graph where the pair $(D_1(x), D_2(x))$ is plotted. Here, $D_i(x)$ is the depth of the point x respect to the data in the i-th group. With this notation, the DD–plot is, simply, a map

between the (functional) space $\mathcal{X}$ where the data are defined, and $\mathbb{R}^2$:

$$\begin{aligned} \mathcal{X} &\rightarrow \mathbb{R}^2 \\ x &\rightarrow (D_1(x), D_2(x)). \end{aligned}$$

The aim of the DD–classifier is to identify both groups using the information provided by the DD–plot. Since, we have transformed our data to live in $\mathbb{R}^2$, it happens that the task of separating classes can be made in a much more simple framework, assuming that the depths contain relevant information about how to separate the groups. So, the choice of a depth becomes now a crucial step. On the other hand, in Li et al. (2012) the classification rule was established as a polynomial function (up to a previously fixed order k) ensuring that the point $(0, 0)$ belongs to it. This rule has two main drawbacks. First, the number of different polynomials of order k that can conform a classification rule is $\binom{N}{k}$ where N is the sample size. This number is the possible ways of selecting k points from N and each one of those selections has associated the order k polynomial which interpolates those k points and $(0, 0)^t$. Clearly, as N increases, the complexity of the estimation process grows at a rate of order N^k. Second, a polynomial always gives a connected border between groups and does not allow the construction of, let us say, islands (zones assigned to one group completely surrounded by the others, like the horizontal red band between the blue zones in the DD-plot in Figure 2) which can appeared in the map. Finally, the problem of classifying more than two groups was not completely solved in the original paper.

The DD^G–classifier which we propose here tries to solve the previous drawbacks in a unified way. Suppose that we have realizations of a process in the product space $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_p$ (multivariate (functional) data) where we have g groups (classes or distributions) to be separated using data–depths. To describe the DD^G–classifier, let us begin assuming that $p = 1$. In this case, the DD^G–classifier begins selecting a depth D and computing the following map:

$$\begin{aligned} \mathcal{X} &\rightarrow \mathbb{R}^g \\ x &\rightarrow \mathbf{d} = (D_1(x), \ldots, D_g(x)). \end{aligned}$$

Now we can apply any available classification procedure working in an g–dimensional space to separate the g groups.

The extension of the procedure to the case $p > 1$ is simple: we only need to select an appropriate

depth D^j for each subspace $\mathcal{X}_j$ and consider the map

$$\begin{aligned} \mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_p &\rightarrow \mathbb{R}^{g \times p} \\ x = (x_1, \ldots, x_p) &\rightarrow \mathbf{d} = (\vec{D}^1(x_1), \ldots, \vec{D}^p(x_p)), \end{aligned}$$

where $\vec{D}^i(x_i)$ is the g-dimensional vector giving the depths of the point $x_i \in \mathcal{X}_i$ with respect to the groups $1, \ldots, g$.

Our last consideration is related to the selection of the depth. As we stated before, the chosen depth may have influence in the result. The solution in Li et al. (2012) was to choose the right one by Cross-Validation. Our proposal here is to use several depths at the same time. To be precise, for the i-th component of the product space, we can try $k_i \geq 1$ different data–depth functions. The DD^G–classifier constructs the following map:

$$\begin{aligned} \mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_p &\rightarrow \mathbb{R}^G \\ x = (x_1, \ldots, x_p) &\rightarrow \mathbf{d} = (\vec{D}^1(x_1), \ldots, \vec{D}^{k_1}(x_1), \ldots, \vec{D}^1(x_p) \ldots, \vec{D}^{k_p}(x_p)). \end{aligned}$$

where the notation $\vec{D}^k(x_i)$ denotes the g–dimensional vector of the k-th depth applied to the i-th component of x (i.e., the depths of x_i with respect to the g groups computed using the depth D^k) and $G = g \sum_{i=1}^{p} k_i$. Notice that, with an obvious abuse of notation, we have employed the same notation $\vec{D}^k$ for all the depths applied in the marginal spaces in spite of they can be different.

With this map, the DD–classifier translates the information of $\mathcal{X}$ into a vector of dimension G. This feature is specially interesting in the functional context or when dealing with high-dimensional spaces. The last step in the DD^G–classifier procedure is to select an appropriate classification rule. But now, we have to deal with a purely multivariate classification problem in dimension G and many procedures are known to handle this problem (see, for example, Ripley (1996)). The following multivariate classification procedures will be considered here for simulation and applications attending to their simplicity and/or easiness to draw inferences:

1. *Linear Discriminant Analysis* (LDA): This is the most classical discriminant procedure. It is due to Fisher and is the particular application of the Bayes' Rule Classifier under the assumption that all the groups or classes in the population have a normal distribution with different means ($\left\{\bar{\mathbf{d}}_i\right\}_{i=1}^{g}$) but the same covariance matrix (Σ). The LDA classification rule assigns an element $\mathbf{d}$ to the class i which satisfies that $L_{ij}(\mathbf{d}) > 0, j = 1, \ldots g, j \neq i$ where

$$L_{ij}\left(\mathbf{d}\right) = -\frac{1}{2}\left\{\bar{\mathbf{d}}_i^T \Sigma^{-1} \bar{\mathbf{d}}_i - \bar{\mathbf{d}}_j^T \Sigma^{-1} \bar{\mathbf{d}}_j\right\} + \log\left(\frac{\pi_i}{\pi_j}\right) + \left(\bar{\mathbf{d}}_i - \bar{\mathbf{d}}_j\right)^T \Sigma^{-1} \mathbf{d},$$

being π_i the probability a priori for class i.

2. *Quadratic Discriminant Analysis* (QDA): It is similar to LDA, excepting for the fact that QDA does not assume equality between covariance matrices. The QDA classification rule assigns an element $\mathbf{d}$ to the class i satisfying $Q_{ij}(\mathbf{d}) > 0, j = 1, \ldots g, j \neq i$, where

$$\begin{aligned} Q_{ij}(\mathbf{d}) &= -\frac{1}{2}\left\{\log\frac{|\Sigma_i|}{|\Sigma_j|} + \bar{\mathbf{d}}_i^T\Sigma_i^{-1}\bar{\mathbf{d}}_i - \bar{\mathbf{d}}_j^T\Sigma_i^{-1}\bar{\mathbf{d}}_j\right\} + \log\left(\frac{\pi_i}{\pi_j}\right) + \\ &\quad + \bar{\mathbf{d}}_i^T\Sigma_i^{-1}\mathbf{d} - \bar{\mathbf{d}}_j^T\Sigma_j^{-1}\mathbf{d} - \frac{1}{2}\mathbf{d}^T\left(\Sigma_i^{-1} - \Sigma_j^{-1}\right)\mathbf{d}. \end{aligned}$$

3. *Logistic Regression Model* (GLM): As a particular case of the Generalized Linear Models, the logistic regression model modelizes the posterior probability given $\mathbf{d}$ through

$$\text{logit } p(i|\mathbf{d}) = \log\left(\frac{p(i|\mathbf{d})}{1-p(i|\mathbf{d})}\right) = \alpha_i + \boldsymbol{\beta}_i^T\mathbf{d},$$

and the point $\mathbf{d}$ is assigned to the class i where $p(i|\mathbf{d}) > p(j|\mathbf{d}), j = 1, \ldots, g, j \neq i$.

4. *Generalized Additive Models* (GAM): The Generalized Additive Models (see Wood (2004)) relax the linearity assumption in GLMs allowing the use of a sum of general smooth functions f_j for the posterior probability, i.e.

$$\text{logit } p(i|\mathbf{d}) = \log\left(\frac{p(i|\mathbf{d})}{1-p(i|\mathbf{d})}\right) = \alpha_i + \sum_{k=1}^{g} f_{k,i}(d_k),$$

where d_k is the k-th component of $\mathbf{d}$. The functions $f_{k,i}$ may belong to a known parametric family (polynomials, for instance) or they may even be functions to be estimated non-parametrically.

5. *Nonparametric classification methods*: These methods are based on non-parametric estimates of the densities of the groups. The most simple (and classical) one is the k–nearest neighbour (kNN) where, given $k \in \mathbb{N}$, the point $\mathbf{d}$ is assigned to the class containing a majority of the k nearest data points in the training sample.

   Another possibility is to estimate $p(i|\mathbf{d})$ through the Nadaraya–Watson estimator:

$$p(i|\mathbf{d}) = \frac{\sum_{n=1}^{N}\mathbf{1}_{G_n=i}K\left(m(\mathbf{d},\mathbf{d}_n)/h\right)}{\sum_{n=1}^{N}K\left(m(\mathbf{d},\mathbf{d}_n)/h\right)},$$

where N is the size of the training sample, G_n is the class of n-th data point in the training sample, K is a kernel and $m(\mathbf{d}, \mathbf{d}_n)$ is a suitable distance in $\mathbb{R}^G$ between $\mathbf{d}$ and $\mathbf{d}_n$ which is re-scaled by the bandwidth parameter h. This method will be denoted by NP.

A kNN method could be considered a NP method using the uniform kernel and a locally selected bandwidth. These two methods are quite flexible and powerful but unlike the previous ones, it is not easy to obtain diagnosis about which part of the vector $\mathbf{d}$ is important for the final result.

6. *Classification trees* (tree): A classification tree is a ordered sequence of questions that split the data set until finishing with a prediction of the class. The questions (nodes) are formulated in terms of variables shortcuts dividing the data set into two branches and proceeding iteratively until no more splits can be made (with a reasonable criterium). A class is assigned to every terminal node and, given a point, the classification rule is as simple as to answer the questions until a terminal node is reached. Then, the assigned group is the class of the terminal node. So, the classification trees are simple and easy to interpret but they split the space into hyperrectangles that sometimes are not too close to the optimal solution.

7. Finally, there are some other classification methods that could be employed here like, for example, Artificial Neural Networks (ANN), Support Vector Machine (SVM) or Multivariate Adaptive Regression Splines (MARS), among others, but the application of these methods needs the choice of several auxiliary parameters or designs and the inference from them is not so straightforward as the methods mentioned earlier. Nevertheless, in some cases, it could be an option to explore when only the prediction task is required.

3 Data depths for functional data

As mentioned earlier, the DD–classifier is specially interesting in the functional context because it allows to dramatically decrease the dimension of the classification problem from infinite to G. In this section, several functional data depths that will be used later with the DD^G–classifier will be revised. Some extensions to cover multivariate functional data are provided.

3.1 Fraiman and Muniz Depth (FM)

The FM depth (Fraiman and Muniz (2001)) was the first one proposed in the literature in a functional context. It is also known as integrated depth by its definition. Given a sample $x_1, \ldots, x_N$ of functions defined on the interval $[0, T]$, let $S_t = (x_1(t), \ldots, x_N(t))$ be the values of those functions on a given $t \in [0, T]$. Denote by $F_{N,t}$, the empirical distribution of the sample S_t and, by $Z_i(t)$, an univariate depth of $x_i(t)$ in this sample (in the original paper, $Z_i(t) = 1 - |1/2 - F_{N,t}(x_i(t))|$). Then, the FM depth for the i-th datum is defined as:

$$FM_i = \int_0^T Z_i(t)dt. \tag{2}$$

The first obvious generalization of the FM depth is to consider different univariate depths to be integrated. A list of possible depths can be found in Liu et al. (1999), from where we select the following ones for the univariate case:

- Tukey depth, $Z_i^{TD}(t) = \min\left(F_{N,t}\left(x_i(t)\right), 1 - F_{N,t}\left(x_i(t)^-\right)\right)$.
- Simplicial depth, $Z_i^{SD}(t) = 2F_{N,t}\left(x_i(t)\right)\left(1 - F_{N,t}\left(x_i(t)^-\right)\right)$.
- Likelihood depth, $Z_i^{LD}(t) = \hat{f}(x_i(t))$, where $\hat{f}$ is a consistent estimate of the density function at t.
- Mahalanobis depth, $Z_i^{MhD}(t) = \left[1 + \left(x_i(t) - \hat{\mu}(t)\right)^2 / \hat{\sigma}^2(t)\right]^{-1}$, where $\hat{\mu}(t), \hat{\sigma}^2(t)$ are estimates of the mean and variance at point t.

The choice of a particular univariate depth modifies the behavior of the FM depth and, for instance, the deepest curve may have a different meaning depending on this selection.
A generalization of the FM-depth for multivariate functional data (when the elements belong to a product space $\mathcal{X} = (\mathcal{X}^1 \times \ldots \times \mathcal{X}^p)$, where the $\mathcal{X}^i$'s are functional spaces) without increasing the dimension of our classification problem can be done in the following two ways:

- *Weighted depth*: Compute the depths for every component $(FM_{i,1}, \ldots, FM_{i,p})$ and then define a weighted version of the FM–depth (FM^w) as:

$$FM_i^w = \sum_{i=1}^{p} w_i FM_{i,p}$$

being $\vec{w} = (w_1, \ldots, w_p)$ a suitable vector of weights.

In the choice of the weights it must be taken into account the possibly different scale of the depths (for instance, the FM depth using SD as the univariate depth takes values in $[0, 1]$, whereas the Tukey depth always belongs to the interval $[0, 1/2]$).

This extension is also useful when we are interested in mixing different depths to obtain a new one reducing the dimension of the classification problem.

- *Common support*: Suppose that all $\mathcal{X}^i$ have the same support $[0, T]$ (this happens, for instance, when using the curves and their derivatives). In this case, we can define a p-summarize version of FM–depth (FM^p) depth as:

$$FM_i^p = \int_0^T Z_i^p(t)dt,$$

where $Z_i^p(t)$ is a p-variate depth of the vector $(x_i^1(t), \ldots, x_i^p(t))$ with respect to the sample at point t.

3.2 h–mode depth (hM)

The h–mode depth was proposed in Cuevas et al. (2007) as a functional generalization of the likelihood depth with the idea of measuring how surrounded is one curve respect to the others. The population h–mode depth of a datum x_0 is given by:

$$f_h(x_0) = \mathbf{E}\left[K(m\left(x_0, X\right)/h)\right], \tag{3}$$

where X is a random element describing the population, m is a suitable metric or semi-metric, $K(t)$ is an asymmetric kernel and h is the bandwidth parameter. Given a random sample $x_1, \ldots, x_N$ of X, the empirical h–mode depth is defined as:

$$\hat{f}_h(x_0) = N^{-1} \sum_{i=1}^{N} K(m\left(x_0, x_i\right)/h). \tag{4}$$

Equation (4) is similar to the usual nonparametric kernel density estimator but with a main difference: as our interest is focused in what happens in a neighborhood of each point, the bandwidth is not intended to converge to zero when $N \to \infty$, and the only constraint is that the bandwidth

should be large enough to avoid pathological situations (for instance, the bandwidth should not be as small as every point in the sample has the same depth equal to $K(0)/N$).

With respect to the extension of this depth when we have multivariate functional data, the use of a weighted depth of the components can be applied. Another possibility in this case, is to construct a new metric from those existing in each component of the product space combining them using a p-dimensional metric like, for example, the euclidean, i.e. $m\left(\left(x_0^1,\ldots,x_0^p\right),\left(x_i^1,\ldots,x_i^p\right)\right) := \sqrt{m_1\left(x_0^1,x_i^1\right)^2+\cdots+m_p\left(x_0^p,x_i^p\right)^2}$ where m_i is denoting the metric in the i-component of the product space.

3.3 Random Projection methods

There are several depths based on random projections using, basically, the same scheme. Given a sample $x_1,\ldots,x_N$ of functions in a Hilbert space, an unit vector a in this space is selected (independently from the x_i's) and the data are projected onto the one-dimensional subspace generated by a. Then, the sample depth of a datum x is defined as the univariate depth of the corresponding projection, $\langle a,x\rangle$, with respect to the projected sample $\{\langle a,x_i\rangle\}_{i=1}^N$. Although, theoretically, only one projection is enough (see Cuesta-Albertos et al. (2007)), the random projection methods usually generate several directions, $a_1,\ldots,a_R, R>1$, to increase the power. Those methods differ on the univariate depth they employ and, how the depths obtained from those projections are summarized. Here, we will handle

- Random Projection (RP): Proposed in Cuevas et al. (2007), it uses the Tukey univariate depth and summarizes the depths of the projections through the mean. So, if $D_{a_r}(x)$ is the depth associated with the r–th projection, then,

$$RP(x) = R^{-1}\sum_{r=1}^{R} D_{a_r}(x).$$

- Random Tukey (RT): Proposed in Cuesta-Albertos and Nieto-Reyes (2008), it uses the Tukey depth in both steps: as the univariate depth to compute the depth for every projection and to summarize the projections. Then,

$$RT(x) = \min\left\{D_{a_r}(x):\ r=1,\ldots,R\right\}.$$

The extensions to multivariate functional data are similar to those proposed with the FM depth. The first option is to construct a weighted depth from the depth of each component. The second possibility is to use a p-variate depth with the projections in the same way as it is done with FM depth, although in this case it is not needed to assume that all components have a common support. The RPD depth proposed in Cuevas et al. (2007) is an example of this extension using the original curves and their derivatives as the components of the multivariate functional data, which, in this case, are two-dimensional.

3.4 Band depth methods

The band methods (proposed in López-Pintado and Romo (2009)) treat the trajectories of the functions as graphs and create bands with them. More specifically, let $G(x) = \{(t, x(t)) : t \in [0, T]\}$ the graph of the function x, and let

$$B\left(x_{i_1}, \ldots, x_{i_k}\right) = \left\{(t, y) : t \in [0, T], \min_{r=1,\ldots,k} x_{i_r}(t) \leq y \leq \max_{r=1,\ldots,k} x_{i_r}(t)\right\}$$

the band delimited by $\{x_{i_1}, \ldots, x_{i_k}\}$. For every $j \in \{2, \ldots, N\}$, the Band Depth (BD) is defined as:

$$BD^{(j)}(x) = \binom{N}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq N} I\left\{G(x) \subseteq B\left(x_{i_1}, \ldots, x_{i_j}\right)\right\},$$

which is the proportion of bands created with j curves containing the whole graph of x.
A modification of this depth is to consider a different function than the indicator in the previous equation. In particular, if λ, denotes the uniform distribution on $[0, T]$, then the Modified Band Depth (MBD) is defined as:

$$MBD^{(j)}(x) = \binom{N}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq N} \lambda\left\{t : (t, G(x(t))) \in B\left(x_{i_1}(t), \ldots, x_{i_j}(t)\right)\right\}.$$

This depth has a close relationship with FM because, as pointed out by the authors, in the finite dimensional framework (i.e., in the case in which $x = (x(t_1), \ldots, x(t_d))$), $MBD^{(2)}(x) = \frac{1}{d}\sum_{j=1}^{d} SD(x(t_j))$ where $SD(x(t))$ is the univariate simplicial depth of $x(t)$, which is a finite dimensional version of the FM depth using the simplicial depth as univariate one.

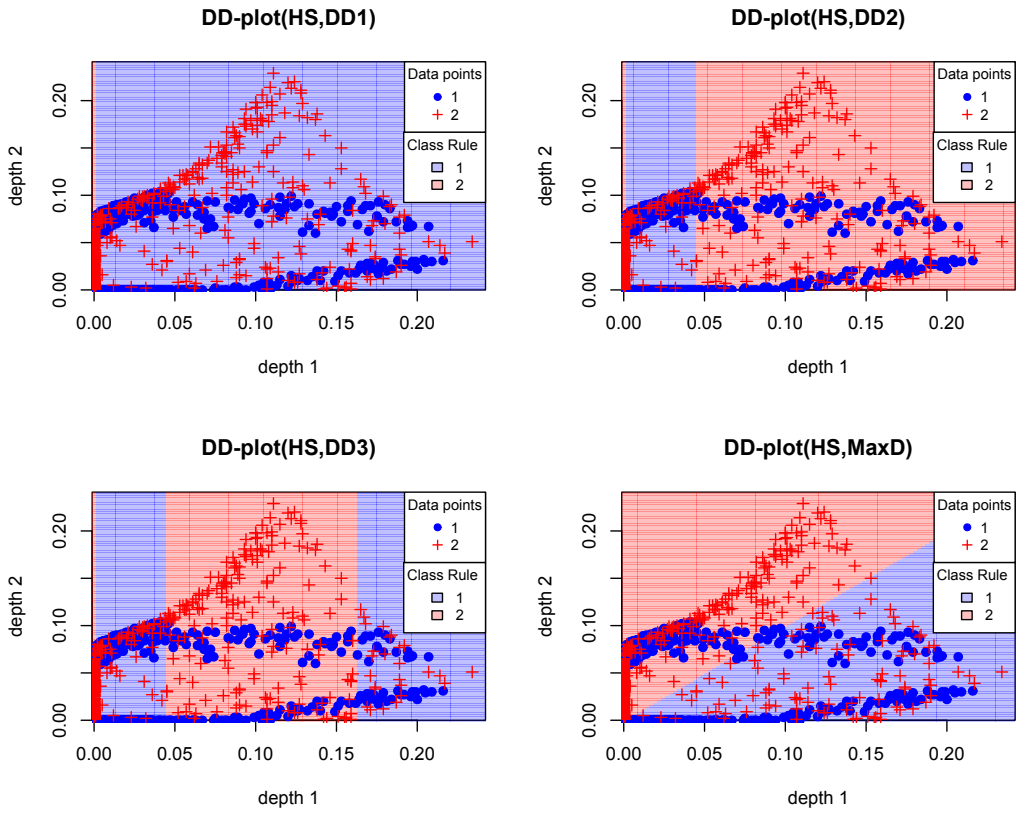

Figure 3: From left to right, top to bottom DD–plot using DD1, DD2, DD3 and Maximum Depth classifiers to the DD-plot in Figure 2.b. The one-dimensional depth in all cases is the Tukey depth.

4 Illustration of regular classification methods in DD-plots

This section is devoted to explore the different classifiers that can be applied to DD–plots as an alternative to the proposal in Li et al. (2012). In that paper, given $k_0 = 1, 2, \ldots$, the classifier is the polynomial, p, with degree up most k_0, such that $p(0) = 0$, that gives the lower misclassification error in the training sample. We denote this classifier by DDk_0. This polynomial is constructed selecting k_0 points of the sample and taking the polynomial going through them and the origin. As previously stated, this construction has two main drawbacks: first, the number of different possible polynomials depends on the sample size and increases polynomially with order k_0. Second, the region given by the polynomial classifier cannot have "islands" and this assumption sometimes fails. Finally, the Maximum depth classifier is a particular case of DD1 fixing the slope equal to 1.

The application to the example in Figure 2.b is plotted in Figure 3 which shows the results from DD1, DD2, DD3 and the Maximum depth classifier. The titles of the subplots are in the general form DD–plot(*depth*,*classif*) where *depth* is the depth employed (HS in this example denotes the Tukey depth) and *classif* denotes the classification method. In all cases, colours red and blue are employed to differentiate the group at which the points in the sample belong to. In the same way, the background image is colored in light red and light blue marking the areas where a new data point would be assigned to red and blue groups respectively. The misclassification erros are, respectively, (0.262, 0.215, 0.201, 0.292). There is a clear improvement of DD3 method over Maximum Depth but there are some areas (see for example, the rectangle $[0.0, 0.2] \times [0.0, 0.1]$) where a polynomial cannot satisfactorily classify the data.

In Figure 4, the rest of the classifiers for the same example are considered: LDA, QDA, kNN, GLM, GAM and tree. The first two methods are classical tools in multivariate analysis whereas the last four are based on regression models with different types of complexity. The misclassification erros are, respectively,: (0.472, 0.51, 0.136, 0.472, 0.152, 0.201). Clearly, the LDA, QDA and GLM methods cannot achieve the result obtained by DD3 which is outperformed by kNN and GAM. It is important to take into account that the optimal classifier gives a theoretical misclassification error of 0.138, very close to the result obtained with the kNN procedure. The key of this improvement over the DD-clasifier is the flexibility of both methods that can model such complicated situations. The next example contains multivariate functional data. The Tecator data set is a reference

among the functional data practitioners and corresponds to a spectrometric problem where the goal is to predict the fat content of meat slices using absorbance curves provided by an automatic device called Tecator Infrared Food Analyzer. Many papers have treated this data set from the regression or the classification point of view (see for example, Ferraty and Vieu (2009) or Febrero-Bande and González-Manteiga (2013) and references therein) with the conclusion that the relevant information for the prediction or the classification is mainly located in the second derivative. For this illustration, let us suppose that we are interested in classifying those samples with percentage of fat above 15% using the absorbances curves (ab) and their second derivative ($ab2$) using the DD^G classifier. To this end, four variates are computed using the hM depth: $ab.mode.0$, $ab.mode.1$, $ab2.mode.0$, $ab2.mode.1$ where the notation $var.depth.group$ stands for the $depth$ computed for variate var with respect to the points in the group $group$.

The result using a GLM as classifier appears in Figure 5 with the combinations of the four variates, showing clearly that those associated with the second derivative are able to separate both groups more efficiently. The prediction misclassification error obtained from 200 runs splitting the sample among a training component (165) and a test one (50) is quite good (3.8%) although in the training, the mean of misclassification error is lower (2.9%) pointing out a tendency to overestimate.

But, more interesting in this illustration is that the contribution of each component can be assessed through the diagnosis of the GLM. The classical diagnosis of the estimates of a GLM model is shown in Table 1 where the variates associated with the depths of the second derivative are both significant whereas for the original curves, the variates associated with the depth is only significant for the first group. If the contribution of the original curves is removed from the model, the misclassification error is slightly better (3.5%) whereas the DD^G classifier with only the original curves presents a misclassification error of 27.2% far away from the optimal. Our best result was obtained with the NP method (2.1%) using the first and the second derivative. This result is similar to that one obtained in Febrero-Bande and González-Manteiga (2013) with a purely functional regression model (not using depths).

	Estimate	Std. Error	z value	$\mathbb{P}(>\|z\|)$
(Intercept)	1.697	1.421	1.194	0.232
ab.mode.0	−0.492	0.16	−3.066	0.002
ab.mode.1	0.074	0.125	0.594	0.552
ab2.mode.0	−0.559	0.108	−5.161	0
ab2.mode.1	0.62	0.156	3.966	0

Table 1: Output for the GLM classifier in the Tecator data set

5 A simulation study and the analysis of some real data sets.

Four models (inspired in those in Cuevas et al. (2007)) have been simulated in order to check the performance of the proposed classifier. In all cases, the curves are obtained from the process $X_{\cdot j}(t) = m_j(t) + e_{\cdot j}(t)$, where m_j is the mean function of group $j = 1, 2$ and e^j is a gaussian process with zero mean and $Cov(e_{\cdot j}(s), e_{\cdot j}(t)) = \theta_j \exp(-|s-t|/0.3)$. In all models, $\theta_2 = \theta_1/2$ giving to the second group half the error than to the first one. The functions also include an additional parameter k which is fixed at $k = 1.1$ in all cases; thus this parameter has no influence in the results. The functions were generated in the interval $[0, 1]$ using an equispaced grid of 51 points.

- *Model 1*: The population P_1 has mean $m_1 = 30(1-t)t^k$ and the error process has $\theta_1 = 0.50$. The mean for P_2 is given by $m_2 = 30(1-t)^k t$.

- *Model 2*: The population P_1 is identical to Model 1 but the second is bimodal since it is composed by two subgroups as a function of a binomial variate I with $\mathbb{P}(I = 1) = 0.5$. Here, $m_{2,I=0} = 35(1-t)^k t$ and $m_{2,I=1} = 25(1-t)^k t$.

- *Model 3*: Both populations are composed by two subgroups, being the subgroup means of the first group $m_{1,I=0} = 40(1-t)^k t$ and $m_{1,I=1} = 20(1-t)^k t$, and the subgroup means for the second group $m_{2,I=0} = 30(1-t)^k t$ and $m_{2,I=1} = 10(1-t)^k t$.

- *Model 4*: This is the same as Model 3 but considering now every subgroup as a group itself. So, this is an example with four groups.

The simulation results are based on 200 independent runs. In every run, $N = 200$ training observations for Models 1 and 2 (100 for each group), and a test sample of 100 observations (50 from each group) were generated. For Models 3 and 4, $N = 400$ training observations are generated (100 for each subgroup). Tables 2 to 5 show the proportion of misclassification errors for the test sample. Some curves obtained with each model are presented in Figure 6.

For the comparison, the FM, RP and hM depths were employed using the original trajectories and/or the derivatives of every curve that were computed using splines. The MBD depth was not considered here because of its similarity with the FM depth. The different depth options are denoted by a suffix in the following way:

- *.0*: The depth only uses original trajectories.
- *.1*: The depth only uses the first derivative.
- *.w*: The depth is computed using a balanced weighted sum of the depth of the original trajectories and the depth of the derivatives, i.e. $d_w = 0.5d_0 + 0.5d_1$.
- *.p*: The depths of the trajectories and its derivatives (a two dimensional functional data) are combined inside the depth procedure. With FM and RP depths, a bivariate depth is employed in the computations. The hM method uses an euclidean metric as the combination of the metrics in the subspaces.
- *.m*: For the DD^G–classifier, the vector of all combinations of depth/group are considered. This is the problem with the largest dimension.

With respect to the classifiers, the list includes DD1, DD2 and DD3 as classical classifiers and also LDA, QDA, kNN, NP, GLM and GAM. The other possible methods were not considered here due to their complexity (ANN, SVM) or because they were outperformed by other methods (tree). Note that the procedures DDi, i=1,2 and 3 can not be used with the *.m* option.

The results for Model 1 are summarized in Table 2. This is a quite hard example due to the choice of the parameter $k = 1.1$ (note that in Cuevas et al. (2007) the parameter was $k = 1.2$ that makes easier the classification task). The first conclusion is that the first derivative gives better classification results than the original curves with a gain of about 2.5–4% between a column using the original curves (.0) or using the derivative (.1). The weighted depth (.w) improves another

	FM.0	FM.1	FM.w	FM.p	FM.m	RP.0	RP.1	RP.w	RP.p	RP.m	hM.0	hM.1	hM.w	hM.p	hM.m
DD1	32.9	28.4	26.6	27.8		27.9	26.8	23.6	22.6		25.7	22.9	22.0	22.8	
DD2	32.6	28.9	27.0	28.0		28.1	26.8	24.0	23.0		25.6	23.1	21.9	22.8	
DD3	32.9	29.2	27.1	28.6		28.2	26.4	24.0	23.5		26.4	23.9	22.7	23.4	
LDA	32.5	28.1	26.4	27.3	26.4	27.9	25.4	23.1	22.2	23.2	25.3	22.2	21.4	22.0	21.4
QDA	32.3	28.3	26.3	27.5	26.3	27.8	25.6	23.5	22.5	23.0	25.4	22.3	21.4	22.1	21.4
kNN	34.1	30.3	28.3	28.9	28.3	29.0	27.0	23.8	23.8	24.3	27.0	23.8	22.7	23.7	22.7
NP	33.2	29.9	27.7	28.6	27.7	28.7	27.1	24.1	23.4	24.1	26.7	23.5	22.3	23.1	22.3
GLM	32.5	28.1	26.4	27.2	26.4	27.5	25.6	23.1	22.4	23.3	25.2	22.1	21.4	22.0	21.4
GAM	32.8	28.6	26.7	28.0	26.7	27.8	26.2	23.5	22.9	23.6	25.7	22.6	21.7	22.4	21.7

Table 2: Misclassification prediction errors for Model 1 (mean of 200 runs)

additional 2–3%. The best combination for this model is hM.w–LDA, hM.w–QDA or hM.w–GLM with a 21.4% of classification error, and, in general, the classifiers based on modal depth have a 22–23% of misclassification error, better than using RP or FM depths (the minimums for those depths are 22.2% and 26.3%, respectively). On the other hand, in terms of classifiers, the linear classifiers (LDA, GLM, DD1) seem to work slightly better than the others (the minimum of all columns are obtained in one of these three rows). This means that the simplest linear models are able to do the task successfully.

	FM.0	FM.1	FM.w	FM.p	FM.m	RP.0	RP.1	RP.w	RP.p	RP.m	hM.0	hM.1	hM.w	hM.p	hM.m
DD1	36.7	26.5	27.1	16.1		21.7	22.7	18.1	13.4		13.3	15.6	10.3	15.0	
DD2	22.7	27.3	21.1	16.0		20.0	23.1	17.8	13.4		10.9	15.8	10.3	15.3	
DD3	22.7	27.0	20.9	15.9		19.9	22.3	18.2	13.6		11.8	16.4	11.2	16.1	
LDA	22.0	26.4	20.3	16.1	17.9	20.3	21.9	16.9	13.3	15.6	12.0	15.0	9.8	14.5	9.8
QDA	22.3	26.7	20.6	15.9	18.7	19.2	21.9	17.1	13.2	15.3	11.4	14.8	9.5	14.4	9.4
kNN	23.7	27.8	22.1	16.2	19.2	20.5	22.9	17.4	13.6	16.5	12.0	15.9	10.3	15.3	10.6
NP	23.5	27.4	21.6	16.5	18.9	19.6	22.8	17.8	14.0	16.3	11.8	15.8	10.3	15.3	10.5
GLM	22.1	26.3	20.2	15.6	17.9	19.6	22.1	16.8	13.1	16.0	11.4	15.0	9.5	14.5	9.4
GAM	22.9	26.4	20.9	15.7	18.7	19.8	22.9	17.1	13.3	16.2	10.8	15.2	9.8	14.7	9.7

Table 3: Misclassification errors for Model 2 (mean of 200 runs)

The second model (Table 3) is a difficult scenario for methods based on RP or FM depths. These methods work fine when the groups are homogeneous rather than being constituted by subgroups as in this case. The best misclassification error is obtained by the combination hM.m–QDA or hM.m–GLM (9.4%), although almost all classifiers based on hM.w or hM.m have misclassification errors under 10.5%. The methods based on the original curves performs slightly better than those using first derivatives although both are always improved by the weighted version.

	FM.0	FM.1	FM.w	FM.p	FM.m	RP.0	RP.1	RP.w	RP.p	RP.m	hM.0	hM.1	hM.w	hM.p	hM.m
DD1	34.5	24.2	29.9	25.6		29.6	21.2	25.3	22.2		21.0	16.9	17.2	16.9	
DD2	33.9	22.6	28.0	21.4		27.9	20.1	21.8	18.0		17.6	15.9	14.8	15.7	
DD3	29.1	22.0	24.1	21.6		24.2	20.0	20.6	18.1		18.3	16.6	15.5	16.4	
lda	36.2	22.5	28.7	23.8	22.7	29.9	20.5	23.0	20.1	19.9	18.8	15.6	14.9	15.6	14.7
qda	32.7	22.0	26.0	21.8	20.4	28.1	20.4	21.5	18.2	18.8	17.9	15.6	14.4	15.4	14.3
knn	30.1	23.0	24.7	21.8	21.3	25.4	20.8	20.9	18.1	18.9	18.2	16.4	14.9	16.3	14.5
np	30.0	22.3	24.9	21.7	21.1	24.8	20.1	20.5	18.1	18.2	17.8	15.9	14.7	15.7	14.3
glm	36.3	22.5	28.8	23.6	22.7	29.7	20.0	22.9	19.4	19.5	18.8	15.8	14.9	15.6	14.6
gam	29.1	21.7	24.3	21.0	20.3	25.1	20.1	20.7	17.7	17.8	17.4	15.6	14.5	15.4	13.8

Table 4: Misclassification errors for Model 3 (mean of 200 runs)

The third model (Table 4) is even harder for RP and FM methods. In both cases, the use of the first derivative is preferable than the use of the original curves or a weighted version of them. For these methods, the best misclassification errors are obtained using the full model (FM.m–GAM (20.3%) and RP.p–GAM (17.7%)). This is also true for the hM method but obtaining consistently lower misclassification errors. The best combination is hM.m–GAM with only a 13.8%.

Finally, the results for the fourth model (Table 5) are better than those for Model 3, supporting the idea that the homogeneous groups are easier to classify for RP and FM methods. Indeed, in all cases, the weighted version improves the classification of each component alone. In our view, this means that the two components have complementary pieces of the information for classifying. The best combinations for each depth are: FM.w–GLM (16.2%), RP.p–GLM (13.4%) and hM.w–GLM, hM.m–GLM (12.1%). Note that, since we have four groups, the DDi, i=1,2 and 3, procedures can not be used here.

	FM.0	FM.1	FM.w	FM.p	FM.m	RP.0	RP.1	RP.w	RP.p	RP.m	hM.0	hM.1	hM.w	hM.p	hM.m
LDA	21.0	27.3	16.3	17.1	16.3	20.1	24.9	15.3	14.0	15.0	17.1	17.2	12.7	16.1	12.7
QDA	21.0	28.2	17.5	18.9	17.5	19.9	24.5	15.2	14.9	15.3	17.7	17.8	12.6	16.6	12.6
kNN	21.4	29.9	17.1	17.7	17.1	19.9	25.6	15.2	14.3	15.4	17.0	18.4	12.6	17.1	12.6
NP	20.7	27.8	16.5	17.2	16.5	19.8	24.8	14.9	14.0	15.1	16.9	18.0	12.6	16.6	12.6
GLM	20.9	27.9	16.2	16.7	16.2	19.8	24.7	14.9	13.4	15.1	16.3	17.3	12.1	16.1	12.1
GAM	20.9	28.1	16.5	17.1	16.5	19.8	25.2	15.5	14.0	15.2	16.2	17.8	12.5	16.5	12.5

Table 5: Misclassification errors for Model 4 (mean of 200 runs)

5.1 Application to real data sets.

We have applied our proposal to several well-known data sets found in the functional data analysis literature. A nice review on functional classification can be seen in Baillo et al. (2010). In the following, we will briefly describe the data sets, the results found in the literature and our best results using depths.

- *Berkeley Growth Study Data*: The curves in this dataset correspond to the heights of 39 boys and 54 girls from age 1 to 18. It constitutes a classical example included in Ramsay and Silverman (2005) and in the `fda` R-package.

  As a classification problem, this dataset was treated in Baillo and Cuevas (2008), where using a functional kNN procedure, a best cross-validation classification rate of 96.77% was obtained. In our application, the best result is obtained by hM.p– LDA with 96.5%.

- *Phoneme*: The *phoneme* dataset is also quite popular in the FDA community although it was originated in the area of Statistical Learning (see Hastie et al. (1995)). The dataset has 2000 log-periodograms of 32ms duration corresponding to five diferent phonemes (*sh, dcl, iy, aa, ao*).

  As a functional classification problem, the data set appeared in Ferraty and Vieu (2003). Randomly splitting the data into training and test sample (750, 150 per class – 1250, 250 per class, respectively) and repeating the procedure 50 times, the best result achieved by those authors was 8.5% of misclassification rate. Applying the same scheme and our proposals, the combination of hM.m–LDA misclassifies 7.6%.

This dataset was also used in Delaigle and Hall (2012) but restricted to the use of the first 50 discretization points and to the binary case using the two most difficult phonemes: (*aa, ao*) obtaining a misclassification rate of 20% when $N = 100$. Our best result is 19.0% obtained by hM.m–GLM although most hM procedures are below 20%.

- *MCO Data*: These curves correspond to mitochondrial calcium overload (MCO), measured every 10 seconds during an hour in isolated mouse cardiac cells. The data (two groups: control and treatment) were used as functional data in Cuevas et al. (2004) for ANOVA testing and it is available in the `fda.usc` package.

  As a classification FDA problem, this example was considered in Baillo and Cuevas (2008) where using a Cross Validation procedure, a best error rate of 11.23% was obtained. Our best result is the combination hM.1–GLM with a error rate of 10.1%.

- *Cell Cycle*: The curves in this dataset are temporal gene expression measured every 7 minutes (18 observations per curve) of 90 genes involved in the yeast cell cycle. The data were originally obtained by Spellman et al. (1998) and used in Leng and Müller (2006) and Rincón Hidalgo and Ruiz-Medina (2012) with the goal of classifying these genes into two groups. The first group has 44 elements related with the G1 phase regulation. The other 46 genes conform the second group and are related with S, S/G2, G2/M and M/G1 phases. The dataset has several missing observations which have been reconstructed in this work using a B-spline basis of 21 elements.

  Both papers mentioned above obtain the same misclassification rate: 10% (9 misclassified genes) but varying the number of errors for each group. Our proposal achieves a 6.67% (6 misclassified genes) with the combination FM.m–kNN but almost all procedures based on hM.1, hM.w or hM.m obtain a misclassification rate of 8.9% or lower.

- *Kalivas*: This example comes from Kalivas (1997) and was used for a functional classification problem in Delaigle and Hall (2012). The curves are near infrared spectra of 100 wheat samples from 1100nm to 2500nm in 2nm intervals. Two groups are constructed using the protein content of each sample, which is known in advance, using a binary threshold of 15% that leaves 41 elements in the first group and 59 in the second.

Our best result for 200 random samples of size 50 was the combination FM.m–QDA with a 3.7% of misclassification error. This rate is quite far from the best in Delaigle and Hall (2012) ($CENT_{PC1} = 0.22\%$) using the centroid classifier but the latter requires to project into a specific direction that in this case corresponds to a very small variations in the interval $[1100, 1500]$. So, any depth procedure cannot achieve a better result because the depth is computed using the whole interval and not focusing in the small interval that contains the relevant information for the discrimination process.

6 Conclusions

In this paper we have presented a procedure that extends the DD-classifier proposed in Li et al. (2012) and adapted to functional context in several ways:

- The first improvement is the ability of the new proposal to handle more than two groups due to the flexibility of the new classifiers considered. In fact, the DD^G classifier is converting the data in a multivariate dataset whose columns are constructed using depths and the new classifiers are classical multivariate classifiers based on discrimination procedures (LDA, QDA) or regression ones (kNN, NP, GLM, GAM). More classifiers could be considered here (SVM, neural networks,...) but, in our experience, there is no gain in prediction using more complex classifiers. On the contrary, as the complexity of the classifier increases, there is a tendency to overestimate in the estimation stage and, consequently, a loss of predictive power appears. For instance, in our simulations, not always the GAM procedure is better than the GLM although the last one is a particular case of the former.

- As an aside of the change in the classifiers, several depth procedures can be taken into account at the same time in order to improve the classification or in order to diagnose whether a depth contains or not useful information for the classification process. For the diagnosing part, it is recommended that the classifier be simple to analyze like, for instance, LDA or GLM procedures.

- The DD^G–classifier "trick" is specially interesting in a functional or high dimensional framework because it changes the dimension of the classification problem from infinite or large

dimension to G, where G depends only on the number of groups under consideration and the number of depths that the statistician decides to employ. For instance, if we have 3 groups in the data and the method is using 2 different depths, the multivariate dimension of the DD^G–classifier is 6. Clearly, this is a more tractable dimension for the problem, but, there are also some options to reduce this number. In this paper, a revision of functional data depths is done including modifications to summarize multivariate functional data (the data are conformed with vectors of functions) without increasing or even reducing the dimension of the problem at hand. In a multivariate setting, this could be not so advantageous because the dimension G is a multiple of the number of groups and, sometimes, it could be greater than the dimension of the original space. For instance, in the classical example of Fisher iris data, there are four variables and three groups that using the DD^G–classifier map, in its most simple case, can be comprised in dimension three. But, we can also consider a univariate depth for each variable, and, then, the dimension G grows up to twelve.

- The functions needed to perform this procedure are freely available at CRAN in the `fda.usc` package (Febrero-Bande and Oviedo de la Fuente (2012)) for versions higher than 1.2.0. The principal function is called `classif.DD` and contains all the options shown in this paper related to depths and classifiers. The figures in this paper are also regular outputs of this function.

SUPPLEMENTAL MATERIALS

Supplemental Code Rar compressed file containing the code for generating the plots and results in the paper (paper.code.R) along with the code for the simulation study (file simul.xxx.R) and the code for applications to real data sets (classif.xxxx.R). The file also contains a directory with the data that cannot be obtained from R-packages. (rar file)

R-package fda.usc R-package fda.usc (1.2.0) used to perform the procedures shown in the paper. (GNU zipped tar file)

References

Baillo, A. and Cuevas, A. (2008). Supervised functional classification: A theoretical remark and some comparisons. *arXiv preprint arXiv:0806.2831*.

Baillo, A., Cuevas, A., and Fraiman, R. (2010). *The Oxford Handbook of Functional Data Analysis*, chapter Classification methods for functional data, pages 259–297. Oxford University Press.

Cuesta-Albertos, J. and Nieto-Reyes, A. (2008). The random Tukey depth. *Computational Statistics & Data Analysis*, 52(11):4979–4988.

Cuesta-Albertos, J. A., Fraiman, R., and Ransford, T. (2007). A sharp form of the Cramer–Wold theorem. *Journal of Theoretical Probability*, 20(2):201–209.

Cuevas, A., Febrero, M., and Fraiman, R. (2004). An ANOVA test for functional data. *Computational Statistics & Data Analysis*, 47(1):111–122.

Cuevas, A., Febrero, M., and Fraiman, R. (2007). Robust estimation and classification for functional data via projection–based depth notions. *Computational Statistics*, 22(3):481–496.

Delaigle, A. and Hall, P. (2012). Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):267–286.

Febrero-Bande, M. and González-Manteiga, W. (2013). Generalized additive models for functional data. *TEST*, 22(2):278–292.

Febrero-Bande, M. and Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, 51(4):1–28.

Ferraty, F. and Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44(1):161–173.

Ferraty, F. and Vieu, P. (2009). Additive prediction and boosting for functional data. *Computational Statistics & Data Analysis*, 53(4):1400–1413.

Fraiman, R. and Muniz, G. (2001). Trimmed means for functional data. *Test*, 10(2):419–440.

Ghosh, A. K. and Chaudhuri, P. (2005). On maximum depth and related classifiers. *Scandinavian Journal of Statistics*, 32(2):327–350.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *The Annals of Statistics*, 23(1):73–102.

Kalivas, J. H. (1997). Two data sets of near infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, 37(2):255–259.

Leng, X. and Müller, H.-G. (2006). Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1):68–76.

Li, J., Cuesta-Albertos, J. A., and Liu, R. Y. (2012). DD–classifier: Nonparametric classification procedure based on DD–plot. *Journal of the American Statistical Association*, 107(498):737–753.

Li, J. and Liu, R. (2004). New nonparametric tests of multivariate locations and scales using data depth. *Statistical Science*, 19(4):686–696.

Liu, R. Y. (1990). On a notion of data depth based on random simplices. *The Annals of Statistics*, 18(1):405–414.

Liu, R. Y., Parelius, J. M., Singh, K., et al. (1999). Multivariate analysis by data depth: Descriptive statistics, graphics and inference, (with discussion and a rejoinder by Liu and Singh). *The Annals of Statistics*, 27(3):783–858.

López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718–734.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. Springer.

Rincón Hidalgo, M. M. and Ruiz-Medina, M. D. (2012). Local wavelet-vaguelette-based functional classification of gene expression data. *Biometrical Journal*, 54(1):75–93.

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge Uni. Press, Cambridge.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686.

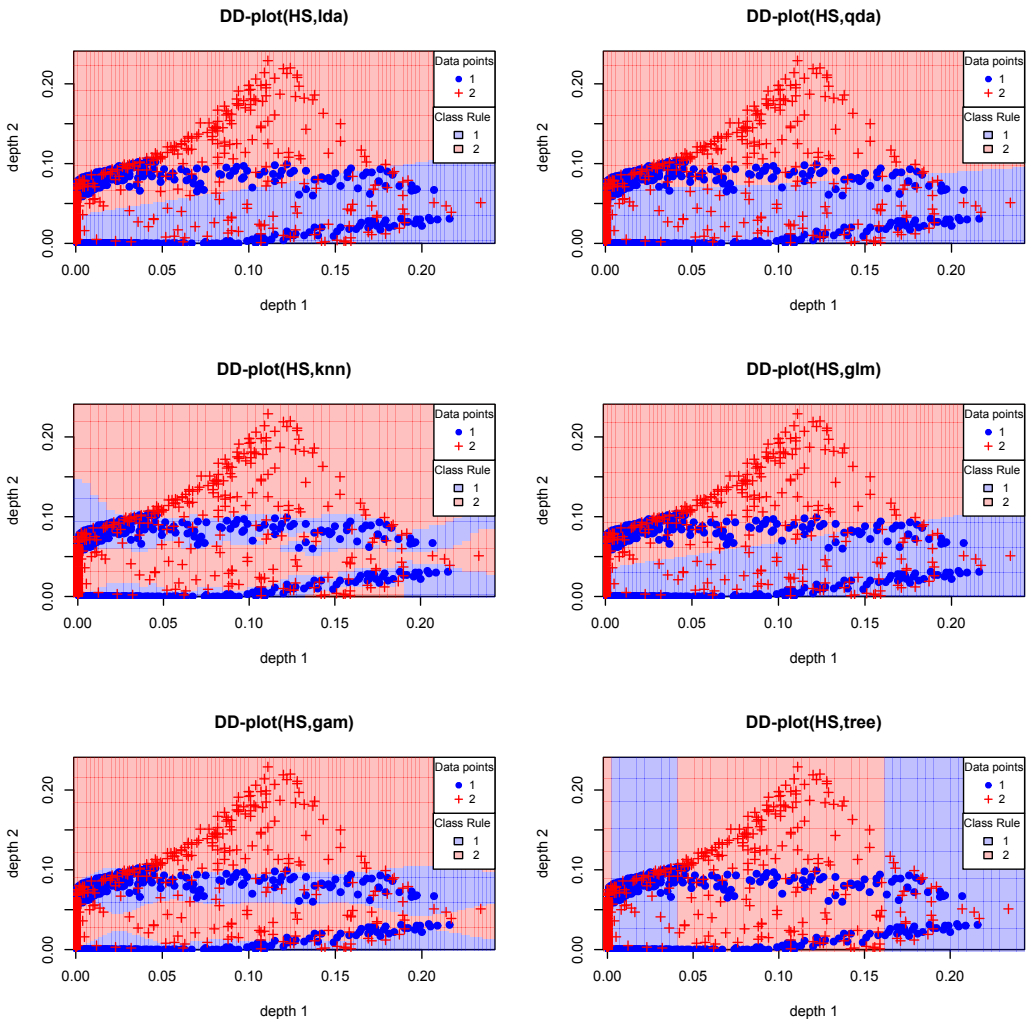

Figure 4: From left to right, top to bottom DD–plot using LDA, QDA, kNN, GLM, GAM and tree classifiers to the DD-plot in Figure 2.b. The one-dimensional depth in all cases is the Tukey depth.

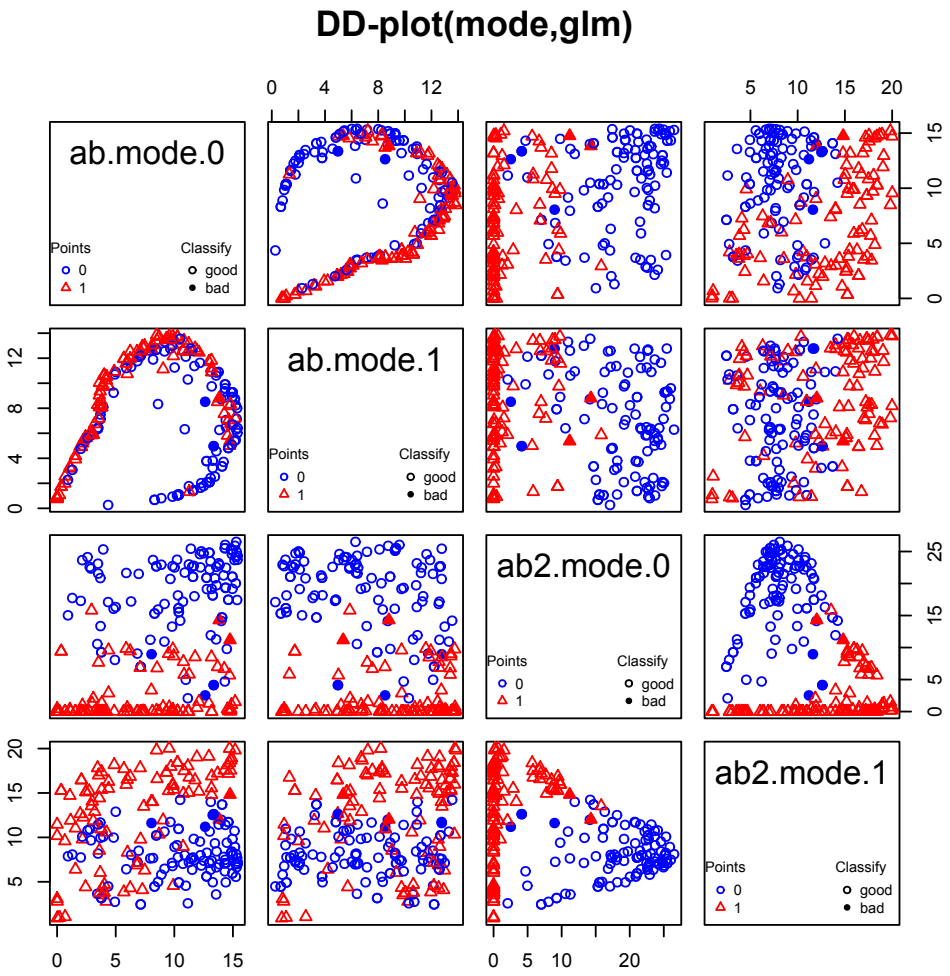

Figure 5: Example of GLM classifier using the spectrometric curves of the Tecator data set and its second derivative.

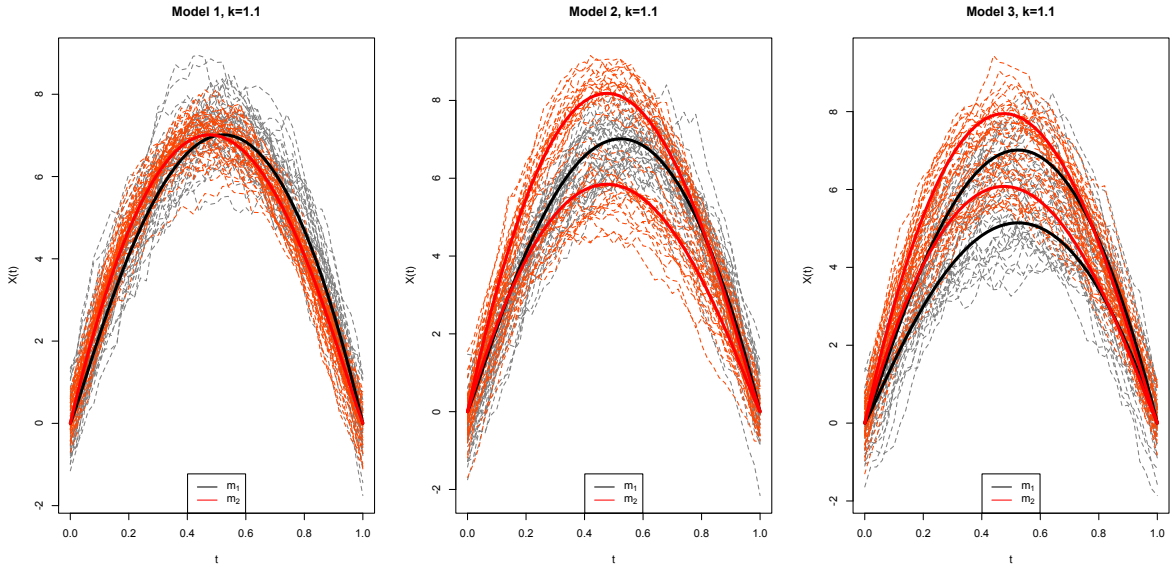

Figure 6: A sample of 40 functions for every simulation model along with the means of each sub-group (m_1's (black lines) and m_2's (red line)).