The DD^G-classifier in the functional setting

J.A. Cuesta-Albertos, M. Febrero-Bande and M. Oviedo de la Fuente

HOW TO CITE

Cuesta-Albertos, J.A., Febrero-Bande, M. and Oviedo de la Fuente, M. (2017) The DD^G-classifier in the functional setting. *TEST*, 26. 119-142.

FUNDING

Research has been funded by project INNPAR2D (reference code MTM2016-76969-P), from the Ministry of Economy and Competitiveness and the European Regional Development Fund (ERDF).

The DD^G-classifier in the functional setting

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

February 13, 2015

Abstract

The maximum depth classifier was the first attempt to use data depths instead of multivariate raw data to construct a classification rule. Recently, the DD–classifier has resolved several serious limitations of the maximum depth classifier but some issues still remain. This paper is devoted to extending 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 (such as kNN, linear or quadratic classifiers, recursive partitioning, and others) to DD–plots to obtain useful insights through the diagnostics of these methods; and third, to integrate various sources of information (data depths, multivariate functional data) in the classification procedure in the unified way. Moreover, as the DD–classifier trick is especially useful in the functional framework, an enhanced revision of several functional data depths is proposed in the paper. A simulation study and applications 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 natural order to order points in the full space with respect to a probability distribution $\mathbf{P}$, with the median being the innermost point and the extreme

*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

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 $x \in \mathbb{R}$ is with respect to $\mathbf{P}$. This index can also be applied to samples replacing F_P by the empirical distribution function. Other possibilities for defining $D_P(x)$ 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 and bounded, and the index is bigger the deeper the point.

In the multidimensional case there exists no natural order and consequently ordering 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 of how deep x is with respect to $\mathbf{P}$. A nice review of multivariate depths can be found in Liu et al. (1999).

Those problems are more difficult to solve in the functional case. However, there are also several depths that are valid in this kind of space. We present some of them in Section 3 along with some extensions that allow us 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 to date in the literature including exploratory and descriptive statistics and statistical tests of hypothesis, among others. 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 knowledge, the first paper in which depths were used for this purpose 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 is the DD–plots. These plots 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 of the dimension of the space supporting the samples) in which, given two probability distributions, $\mathbf{P}$ and $\mathbf{Q}$ on $\mathbb{R}^p$, for every point $x \in \mathbb{R}^p$, the pair

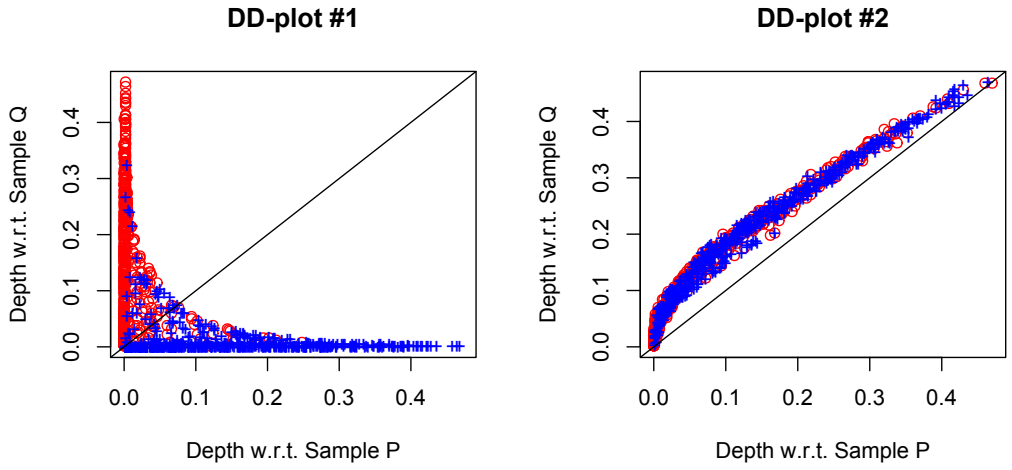

Figure 1: DD–plots of two samples drawn from two-dimensional normal distributions. In both cases **P** is a standard 2-dimensional distribution. **Q** differs from **P** in the mean in the first DD–plot and in the covariance matrix in the second.

$(D_P(x), D_Q(x))$ 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 and 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) develop 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 classifying points. In the example of Figure 1, both DD–plots correspond to samples 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** and points in red from **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 into 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 second since it classifies almost all points as produced by distribution **Q**. An interesting idea in Li

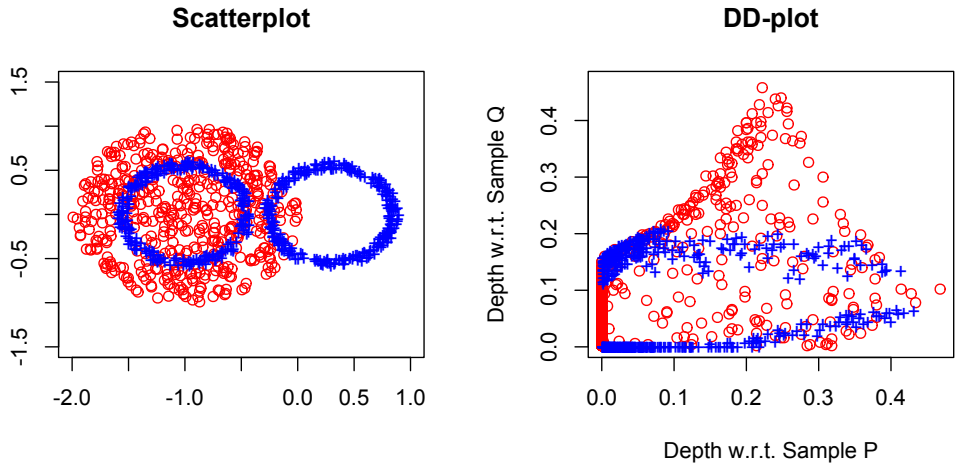

Figure 2: Scatterplot of two uniform samples and associated DD–plot.

et al. (2012) was to notice that in spite of failures of the maximum depth classifier, the DD–plot could contain information enabling a good classifier to be obtained.. For instance, in the second DD–plot in Figure 1 the proportion of red points is very high in an area close to the vertical axis. The authors of Li et al. (2012) proposed replacing the main diagonal by a function whose graph splits the DD-plot into two zones with the lowest misclassification rate (in that paper only the polynomial case is fully developed). This is termed the *DD-classifier*.

Obviously, the DD–classifier is a big step forward in solving the limitations of the maximum depth classifier and in fact, in the problem cited above, according to Li et al. (2012), the DD–classifier gives a classification very close to the optimal classifier. However, there are some situations in which a function can not classify the points correctly. Let us consider the situation presented in Figure 2. The points in the scatterplot are from two samples, each with sizes 2,000 each. The red points were taken from a uniform distribution, denoted by $\mathbf{Q}$, on the unit ball centered on $(-1, 0)^t$. The blue points are from distribution $\mathbf{P}$ which is uniform on the union of two rings: a ring centered at $(-1, 0)^t$ with inner (resp. outer) radius of .5 (resp. .6), and a ring of the same size centered at $(0.3, 0)^t$. The optimal classifier classifies points in both 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, which is impossible. However,

this goal can be achieved with a classifier such as the kNN applied to the DD–plot, as proposed in Vencálek (2011).

Having these ideas in mind, the first objective of this paper is to explore the possibility of applying regular classification methods (such as kNN, linear or quadratic classifiers, recursive partitioning ,or others) to DD–plots and obtaining useful insights through the diagnostics of these methods. Additionally, we wish to overcome the limitation of the 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). Lastly, we want to 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 (such as derivatives, for example) could be computed and used for classification. This paper is mainly focused on the functional setting, although the procedure can also be applied to multivariate data. In particular, we want to remark that some of the diagnostic tools of the classification procedures employed here can be used to assess the relevance of the available functional information. In order to avoid making the paper too long, we only show this idea in the second example in Section 4 where we conclude that the relevant information is contained in the second derivative of the curves.

The paper is organized as follows: In Section 2 we present the basic ideas behind the proposed classifier. Section 3 is devoted to presenting some functional depths and to analyzing 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 applications to some real data sets. The paper ends with a discussion of 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 pairs $(D_1(x), D_2(x))$ are 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, to put it 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 the tow groups using the information provided by the DD–plot. Since we have transformed our data to be in $\mathbb{R}^2$, it happens that the task of separating classes can be made in a much simpler framework, assuming that the depths contain relevant information about how to separate the groups. Thus, the choice of a depth has now become a crucial step. In Li et al. (2012) the classification rule was 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 serve as 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 such that each of the selections has an associated order k polynomial which interpolates between these 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, say, islands (zones assigned to one group completely surrounded by others, like the horizontal red band between the blue zones in the DD-plot in Figure 2) on 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 offer a unified solution to these drawbacks. Suppose that we have implementations of a process in the product space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$ (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 by assuming that $p = 1$. In this case, the DD^G–classifier begins by selecting a depth D and computing the following map:

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

We can now apply any available classification procedure that works in a g–dimensional space to separate the g groups. The same idea is applied in Lange et al. (2014). The main differences between Lange et al.'s and our proposal are that in the former only finite-dimensional data are considered, whereas in this paper, this map is a preliminary step to constructing what is called the feature space. Also, that the authors only use a special kind of linear classifier on this feature space which requires making pairwise comparisons, thus classifying points using a majority vote scheme. Mosler and Mozharovskyi (2014) applies this classifier to functional data, but only after applying a dimension-reduction technique to the data.

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

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

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 influence the result. The solution in Li et al. (2012) was to choose the right depth 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:

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

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 the fact that they can be different.

With this map, the DD^G–classifier translates the information in $\mathcal{X}$ into a vector of dimension G. This feature is especially interesting in the functional context or when dealing with high-dimensional spaces. But new important questions arise, related to how representative the previous maps are. Moreover, if a practitioner considers G too high, how should a useful depth for the classification problem be selected from among all the possibilities? The answer to the latter question can be assessed using the distance correlation $\mathcal{R}$, proposed in Székely et al. (2007), which characterizes independence between vectors of arbitrary finite dimensions. Recently, in Székely and Rizzo (2013), a bias-corrected version is considered and a test of independence developed. Here, our recommendation is to compute the bias-corrected distance correlation between the multivariate vector of depths ($\mathbf{d}$) and the indicator of the groups $\left(Y = \left(\mathbb{1}_{\{x \in g_1\}}, \mathbb{1}_{\{x \in g_2\}}, \ldots, \mathbb{1}_{\{x \in g_g\}}\right)\right)$, and select

the depth that maximizes the distance correlation from among the available depths. In subsequent steps, other depths can be added that have a low distance correlation between the new depth and those selected in previous steps. Also, using the recent extension of the distance correlation to functional spaces provided by Lyons (2013), this tool could be useful for assessing how much of the relation between the functional data and the indicator of the groups can be collected. Later, in Section 4, we will provide an example of the application of these ideas to the selection of the depth.

The last step in the DD^G–classifier procedure is to select an appropriate classification rule. Fortunately, we now have 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 in the simulations and real data analysis 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 a 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

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

   where π_i is the a priori probability for class i.

2. *Quadratic Discriminant Analysis* (QDA): This is similar to LDA, except 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

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

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

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

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

4. *Generalized Additive Models* (GAM): 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.,

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

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 k–nearest neighbour (kNN) in which, 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 point in the training sample, K is a kernel and $m(\cdot, \cdot)$ is a suitable distance in $\mathbb{R}^G$ which is re-scaled by the bandwidth parameter h. This method will be denoted by NP.

   A kNN method could be considered an 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 diagnose 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 a prediction of the class is obtained. The questions (nodes) are formulated in terms of variable shortcuts dividing the data set into two branches and proceeding iteratively until no more splits can be made (with a reasonable criterion). A class is associated with every terminal node and, given a point, the classification rule is as simple as answering

the questions until a terminal node is reached. The group assigned is then is the class of this terminal node. Classification trees are simple and easy to interpret but they split the space into hyperrectangles that may not be very close to the optimal solution.

7. Finally, there are some other classification methods that could be employed here such as, for example, artificial neural networks (ANN), support vector machine (SVM) or multivariate adaptive regression splines (MARS), among others, but the application of any of these methods needs the choice of several auxiliary parameters or designs and the inference from them is not as straightforward as from the above methods. Nevertheless, in some cases, they could be options to explore when only prediction is required.

The choice among the different classifiers could be influenced by their theoretical properties and/or how easy it is to draw inferences. For example, from the theoretical point of view, the kNN classifier can achieve optimal rates close to Bayes' risk (a complete review on this classifier can be found in Hall et al. (2008)) and it could be considered as the standard rule. But better inferences can be drawn from more simple classifiers such as LDA, Logistic or GAM models. Moreover, as discussed in Hand (2006), such simple classifiers are usually hard to beat in real life scenarios, as they add interpretability to the classification rule, which is sometimes more important than predictability.

3 Data Depths for Functional Data

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

3.1 Fraiman and Muniz Depth (FM)

The FM depth (Fraiman and Muniz (2001)) was the first to be 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 $Z_i(t)$

a 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 potential depths can be found in Liu et al. (1999), from which we select the following for the univariate case:

- Tukey depth, $Z_i^{TD}(t) = \min\left(F_{N,t}(x_i(t)), 1 - F_{N,t}(x_i(t)^-)\right)$.
- Simplicial depth, $Z_i^{SD}(t) = 2F_{N,t}(x_i(t))\left(1 - F_{N,t}(x_i(t)^-)\right)$.
- 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, $Z_i^{MhD}(t) = \left[1 + (x_i(t) - \hat{\mu}(t))^2 / \hat{\sigma}^2(t)\right]^{-1}$, 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 the deepest curve, for instance, may vary depending on this selection. In particular, the choice of the SD inside the FM depth is a particular case of the modified band depth (López-Pintado and Romo (2009)). An interesting scenario arises when we are faced with multivariate functional data; i.e., when the elements belong to a product space of functional spaces: $\mathcal{X} = (\mathcal{X}^1 \times \cdots \times \mathcal{X}^p)$. A depth combining the information of all components seems an appealing idea because it will maintain the dimension of our classification problem low, but it does so at the risk of losing some information. This can be done in the following two ways:

- *Weighted depth*: given $x_i = (x_i^1, \ldots, x_i^p) \in \mathcal{X}$, compute the depth of every component, obtaining the values $FM(x_i^j), j = 1, \ldots, p$, and then define a weighted version of the FM–depth (FM^w) as:

  $$FM_i^w = \sum_{i=j}^{p} w_j FM(x_i^j)$$

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

In the choice of the weights, the potential difference in the scale of the depths must be taken into account (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 $[0, 1/2]$).

- *Common support*: Suppose that all $\mathcal{X}^i$ have the same support $[0, T]$. (This happens, for instance, when using the curves and the derivatives). In this case, we can define a p-summarized 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. Particular applications of this idea can be found in Ieva and Paganoni (2013) and Claeskens et al. (2014). The first paper provides a generalization of the MBD that uses simplicial depth as p-variate depth, and the second uses the Tukey halfspace depth.

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 one curve is with 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 on 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 so small that every point in the sample has the same depth equal to $K(0)/N$).

A weighted depth of the components can always be applied to extend this depth when we have multivariate functional data. Another possibility in this case is to construct a new metric from those existing in each component of the product space and combining them using a p-dimensional metric such as, 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 denotes the metric in the i-component of the product space. It is important here to ensure that the different metrics of the spaces have similar scales to avoid one single component dominating the overall distance.

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 with scalar product $\langle\cdot,\cdot\rangle$, a unit vector a in this space is selected (independently of the x_i's) and the data are projected onto the one-dimensional subspace generated by a. 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 a single projection is enough (see Cuesta-Albertos et al. (2007)), random projection methods usually generate several directions, $a_1,\ldots,a_R, R>1$, to increase the power. These methods differ in the univariate depth they employ and in how the depths obtained from those projections are summarized. Here, we will discuss:

- Random Projection (RP): Proposed in Cuevas et al. (2007), it uses univariate Tukey 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

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

The extensions to multivariate functional data are similar to those proposed for 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 we proposed for the FM depth, although in this case it is not necessary 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.

4 Illustration of Regular Classification Methods in DD-Plots

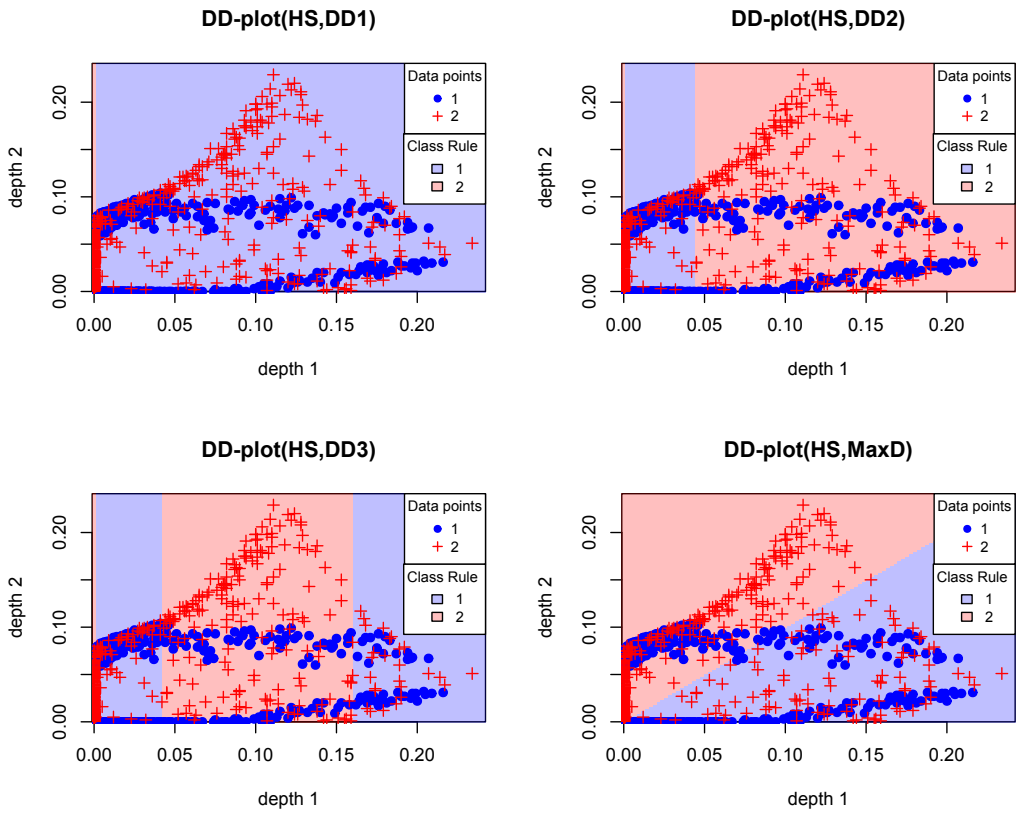

Figure 3: From left to right, top to bottom DD–plot using DD1, DD2, DD3 and Maximum Depth classifiers to the DD-plot in Figure 2.b. The one-dimensional depth in all cases is the Tukey depth.

This section is devoted to exploring 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 g, with degree at most k_0 such that $g(0) = 0$, that gives the lowest misclassification error in the training sample. We denote this classifier by DDk_0. This polynomial is constructed by selecting k_0 points of the sample and taking the polynomial going through these points and the origin. As previously stated, this construction has two main drawbacks: first, the number of possible different 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 with a value of 1.

The application to the example in Figure 2.b is plotted in Figure 3, which shows the results for 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. The sample points are colored red or blue to indicate which group the belong to. The background image is colored light red and light blue to indicae the areas where a new data point would be assigned to red and blue groups respectively. The misclassification error rates are, respectively, (0.262, 0.215, 0.201, 0.292). There is a clear superiority of the 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 shown: LDA, QDA, kNN, GLM, GAM and tree. The misclassification error rates 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 in turn 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 to this improvement over the DD-classifier is the flexibility of two methods that can model such complicated situations.

The next example contains multivariate functional data. The Tecator data set is a standard reference among the functional data practitioners. The data are drawn from a spectrometric problem where the goal is to predict the fat content of meat slices using absorbance curves provided by an automatic device called the 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), Febrero-Bande and González-Manteiga (2013) and references therein) with the conclusion that the relevant information for the prediction or the classification is located mainly in the second derivative. For this illustration, let us suppose that we are interested in classifying those samples with percentage of fat above 15% ($ifat{=}\mathbb{1}_{\{\text{Fat}\geq 0.15\}}$) using the absorbance curves (ab) and their second derivative ($ab2$) using the DD^G classifier. First, there are many possibilities for selecting the depth. For every depth mentioned in the previous section (FM, RP, hM), at least

five possibilities can be explored. These are denoted by the following suffices:

.0: The depth uses only original trajectories. $\mathbf{d} = (D_0^0(x), D_1^0(x))$

.2: The depth only uses the second derivative. $\mathbf{d} = (D_0^2(x), D_1^2(x))$

.w: The depth is computed using a weighted sum of the depth of the original trajectories and the depth of the second derivative. $\mathbf{d} = (D_0^w(x), D_1^w(x))$ with $D_i^w = 0.5D_i^0 + 0.5D_i^2$.

.p: The depths of the trajectories and their derivatives (a two dimensional functional data sets) are combined within 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. $\mathbf{d} = (D_1^p(x), D_2^p(x))$

.m: For the DD^G–classifier, the vector of all depth/group is used. $\mathbf{d} = (D_0^0(x), D_1^0(x), D_0^2(x), D_1^2(x))$.

	FM.0	FM.2	FM.w	FM.m	FM.p
$\mathcal{R}(\text{ifat}, \mathbf{d})$	0.081	0.711	0.4	0.356	0.081
	RP.0	RP.2	RP.w	RP.m	RP.p
$\mathcal{R}(\text{ifat}, \mathbf{d})$	0.079	0.734	0.339	0.248	0.079
	hM.0	hM.2	hM.w	hM.m	hM.p
$\mathcal{R}(\text{ifat}, \mathbf{d})$	0.161	0.749	0.702	0.727	0.161

Table 1: Distance correlation between ifat and the different options for depths for the Tecator data set

As mentioned above, the distance correlation proposed in Székely et al. (2007) can help detect the depth that best summarizes the classification variate. The distance correlation between the group variate (ifat) and the different depths are shown in Table 1. This correlation can also be computed with respect to the functional covariates ($\mathcal{R}(\text{ifat}, ab)$=0.188, $\mathcal{R}(\text{ifat}, ab2)$=0.673) supporting the idea that the important information for classification is contained in the second derivative and that the depths based on the second derivative explain at least the same amount of information as the functional covariate does. In particular, only FM.2, RP.2, hM.2, hM.w, hM.m have values above 0.7.

The next step is to select a classifier that takes advantage of the dependence found by the distance correlation measures. Among all the possibilities, the kNN method seems a good choice from the predictive point of view because it is quite simple to implement. But from the diagnosis point of view, a classifier like the GLM may be preferable. Using the hM.m depth, we have four variates: *ab.mode*.0, *ab.mode*.1, *ab*2.*mode*.0, *ab*2.*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 is shown in Figure 5 with the combinations of the four variates, showing clearly that those associated with the second derivative are able to separate the tow groups more efficiently. More interesting 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 2 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 are only significant for the first group.

	Estimate	Std. Error	z value	$\mathbb{P}(>\|z\|)$
(Intercept)	3.538	2.161	1.637	0.102
ab.mode.0	−0.473	0.166	−2.841	0.004
ab.mode.1	0.054	0.155	0.347	0.729
ab2.mode.0	−0.471	0.103	−4.585	0
ab2.mode.1	1.09	0.301	3.624	0

Table 2: Output for the GLM classifier in the Tecator data set

For the remaining classifiers, 200 runs were done, randomly splitting the sample between a training component (165) and a test component(50). The GLM (3.5%) performs quite well,l and slightly better than the the classifier using kNN (3.9%). The best result among the selected depths, however, is obtained by the combination hM.m–NP with 2.7%. So, in conclusion, the important decision to be made in this case is the selection of the depth (here depending on the second derivative), although almost all classifiers provide similar results.

5 A Simulation Study and the Analysis of Some Real Data Sets.

Four models (inspired by those in Cuevas et al. (2007)) were 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 the models, $\theta_1 = 0.5$ and $\theta_2 = 0.25$, giving the second group half the error of the first. The functions also include an additional parameter k which is fixed at $k = 1.1$; thus this parameter has no influence on 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 $m_1 = 30(1-t)t^k$. The mean for P_2 is $m_2 = 30(1-t)^k t$.
- *Model 2*: The population P_1 is the same as in Model 1 but P_2 is bimodal since it is composed of two subgroups as a function of a binomial variate I with $\mathbb{P}(I = 1) = 0.5$. Here, $m_{2,I=0} = 25(1-t)^k t$ and $m_{2,I=1} = 35(1-t)^k t$.
- *Model 3*: Both populations are composed of two subgroups, in which the subgroup means of the first group are $m_{1,I=0} = 22(1-t)^k t$ and $m_{1,I=1} = 30(1-t)^k t$, and the subgroup means for the second group are $m_{2,I=0} = 26(1-t)^k t$ and $m_{2,I=1} = 34(1-t)^k t$.
- *Model 4*: This uses the same subgroups defined in Model 3 but considers each 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 50 observations 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 misclassification rates for the test samples. 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, which were computed using splines. The MBD depth was not considered here because of its similarity with FM depth. The different depth options are denoted as in Section 4 except that the first derivative (*.1*) is used instead of the second derivative.

Proceeding as in Section 4, the $\mathcal{R}$ is computed to select the best option from among the different depths (first row of Table 3). The overall winner is hM.w suggesting that the combined information of the curves and the first derivative is better than using only one of the components. This is a quite difficult example for a classification task as can be deduced from the relative small distance correlations obtained. Note that here $k = 1.1$ whereas in Cuevas et al. (2007) it was $k = 1.2$ which makes the classification task easier due to a clearer separation of the two groups.

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

	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
R(Y,d)	0.23	0.28	0.34	0.32	0.32	0.34	0.25	0.36	0.36	0.38	0.38	0.42	0.50	0.49	0.47
DD1	27.20	22.20	20.40	21.30		24.70	18.20	18.50	18.30		20.30	17.40	16.40	16.30	
DD2	25.00	21.10	18.80	19.60		24.50	16.90	17.80	17.70		18.80	15.70	13.60	13.80	
DD3	25.20	21.10	19.10	19.60		24.00	17.10	18.20	17.80		18.90	16.10	13.90	14.10	
LDA	24.20	20.70	18.70	19.70	18.70	24.40	17.20	17.90	17.90	16.50	18.40	15.00	12.80	13.00	13.00
QDA	24.50	21.00	18.60	19.70	18.80	25.00	16.80	18.10	17.90	17.20	18.50	15.20	12.70	12.80	13.30
kNN	25.20	22.00	19.20	20.30	19.20	24.60	17.40	18.40	17.90	17.20	18.80	15.90	13.70	13.80	13.70
NP	24.80	21.70	19.30	20.00	18.60	24.90	17.30	18.20	17.80	17.20	18.40	15.80	13.80	13.60	13.40
GLM	24.10	20.60	18.60	19.40	18.70	24.20	16.60	17.50	17.50	16.20	18.40	15.10	12.80	13.00	13.00
GAM	24.70	20.90	18.20	19.40	19.00	23.80	16.80	17.40	17.80	16.60	18.40	15.40	13.30	13.30	13.70

Table 3: Distance correlation and misclassification prediction errors for Model 1 (mean of 200 runs)

The complete results for Model 1 are summarized in Table 3 where the conjectures observed with the distance correlation are confirmed. The best results are obtained for hM.w with small differences between classifiers, except for DD1. In this case, the linear classifiers (QDA, LDA, GLM) seem to work slightly better than the others (the minima of all columns are obtained in one of these three rows). This means that the simplest linear models are able to perform the classification 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
R(Y,d)	0.24	0.16	0.22	0.32	0.26	0.28	0.13	0.24	0.32	0.24	0.48	0.34	0.50	0.44	0.48
DD1	32.50	26.90	27.30	16.70		32.30	21.20	22.00	21.50		14.00	14.50	10.30	11.10	
DD2	22.70	27.40	22.80	16.70		20.10	20.10	16.30	16.00		11.30	14.80	9.60	10.80	
DD3	22.90	27.40	22.90	16.90		20.50	20.20	16.80	15.90		11.60	14.80	9.70	11.00	
LDA	22.00	26.30	21.90	16.40	17.60	21.20	19.70	16.70	16.50	14.80	12.30	13.90	9.30	10.40	9.40
QDA	22.30	26.40	21.90	16.30	18.20	20.60	19.50	16.10	16.20	15.20	11.90	13.60	8.90	10.10	9.10
kNN	23.60	28.00	23.50	17.30	19.60	19.90	20.50	16.60	16.30	16.70	12.10	14.80	9.40	10.60	10.20
NP	23.50	27.50	23.30	17.50	19.20	20.50	20.80	17.00	16.80	16.20	12.00	14.70	9.40	10.90	10.10
GLM	22.10	26.20	22.20	16.00	17.80	19.30	19.40	15.60	15.50	14.50	11.70	13.60	8.80	10.20	9.00
GAM	22.90	26.50	22.40	16.60	18.90	19.30	19.90	16.10	15.50	15.00	11.00	13.90	9.10	10.30	9.50

Table 4: Distance correlation and misclassification errors for Model 2 (mean of 200 runs)

The second model (Table 4) is a difficult scenario for methods based on RP and FM depths as can be deduced from the low values of the distance correlation. These methods work well when the groups are homogeneous rather than being constituted of subgroups as in this case. The least misclassification error is obtained by the hM.w–GLM (8.8%), although all classifiers based on hM.w or hM.m have misclassification error rates under 10.2%.

The third model (Table 5) is even harder for RP and FM methods. In both cases, the use of the first derivative is preferable to 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.p–GAM (21%) and RP.1–GAM (23.2%)). This is also true for the hM method but it consistently yields lower misclassification errors. The best combinations are hM.w–QDA, hM.w–GAM, hM.m–QDA and hM.m–GAM with only 15.1%.

Finally, the results for the fourth model (Table 6) are better than those for Model 3, supporting the idea that homogeneous groups are easier to classify with 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 needed for classification. The best combinations for each depth are: FM.w–GLM, FM.m–GLM (16.2%), RP.p–GLM (15.2%) and hM.p–GLM (12.1%). Note that, since we have four groups, the DDi, $i = 1, 2, 3$, 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
R(Y,d)	0.08	0.24	0.18	0.22	0.16	0.16	0.27	0.23	0.30	0.32	0.32	0.38	0.41	0.38	0.40
DD1	33.90	23.90	29.80	25.50		33.90	26.00	30.20	30.60		23.60	19.00	19.60	20.70	
DD2	33.40	22.20	26.50	21.30		33.10	23.90	28.70	28.80		19.00	16.50	15.50	16.40	
DD3	29.10	22.50	24.30	21.60		29.70	23.70	24.80	25.00		19.40	16.80	15.70	16.80	
LDA	35.60	22.60	28.70	23.60	28.70	36.90	24.60	31.70	31.50	31.60	20.50	16.10	15.90	17.00	15.90
QDA	32.50	22.20	25.90	21.70	25.90	34.20	24.00	28.40	28.50	28.90	19.30	16.10	15.10	16.10	15.10
kNN	30.50	23.10	24.50	21.80	24.50	30.20	24.20	25.50	25.50	25.40	19.70	16.80	15.70	16.70	15.70
NP	30.60	22.60	24.80	21.50	24.80	30.40	23.50	25.80	25.60	25.60	19.50	16.50	15.40	16.50	15.40
GLM	35.70	22.60	28.70	23.40	28.70	36.90	25.00	31.70	31.80	31.90	20.30	16.30	15.60	16.70	15.60
GAM	29.60	21.90	24.20	21.00	24.20	30.50	23.20	25.20	25.40	25.30	18.90	16.30	15.10	16.10	15.10

Table 5: Distance correlation and misclassification errors for Model 3 (mean of 200 runs)

	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
R(Y,d)	0.60	0.47	0.65	0.65	0.63	0.60	0.56	0.67	0.69	0.66	0.64	0.58	0.69	0.68	0.68
LDA	21.00	27.30	16.30	17.10	16.30	23.30	19.30	15.90	15.90	16.00	17.60	17.60	13.00	12.70	13.00
QDA	21.00	28.20	17.50	18.90	17.50	24.10	20.10	16.60	16.50	16.80	17.90	17.80	12.80	12.90	12.80
kNN	23.60	32.80	19.30	19.80	19.30	24.90	22.20	17.90	18.00	18.20	18.50	20.50	14.10	14.20	14.10
NP	20.70	27.80	16.50	17.20	16.50	22.50	19.10	16.00	15.90	16.00	17.00	18.10	12.70	12.80	12.70
GLM	20.90	27.90	16.20	16.70	16.20	23.10	18.70	15.60	15.20	15.70	16.60	17.20	12.20	12.10	12.20
GAM	20.90	28.20	16.50	17.10	16.50	21.80	19.20	15.80	15.70	16.00	16.20	17.90	12.60	12.60	12.60

Table 6: Distance correlation and 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 Baíllo 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 mimicking the original procedure.

- *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 Baíllo 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 the combinations hM.0–LDA, hM.0–QDA, hM.0–kNN with 97.8%.

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

  It appeared as a functional classification problem in Ferraty and Vieu (2003). Randomly splitting the data into training and test sample (250, 50 per class – 250, 50 per class, respectively) and repeating the procedure 200 times, the best result achieved by the authors was an 8.5% misclassification rate. Applying our proposals, the combination hM.m–LDA misclassifies 7.2%.

  This dataset was also used in Delaigle and Hall (2012) but was 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 18.6% obtained by hM.w–QDA although most hM procedures yield errors below 20%.

- *MCO Data*: These curves correspond to mitochondrial calcium overload (MCO), measured every 10 seconds for 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 the dataset is available in the `fda.usc` package.

  This was analyzed as an FDA classification problem, was considered in Baíllo and Cuevas (2008) where using a cross validation procedure, a best error rate of 11.23% was obtained. Our best results are the combinations hM.1–DD1, hM.m–LDA, hM.m–QDA, hM.m–NP with an error rate of 2.2%.

- *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 G1 phase regulation. The remaining 46 genes make up the second group and are related to the S, S/G2, G2/M and M/G1 phases. The dataset has several missing observations which were imputed 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 with different numbers of errors for each group. Our proposal achieves a 5.6% rate with the combination hM.p–NP but almost all procedures based on hM.1 or hM.w yield 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 1100 nm to 2500 nm in 2 nm intervals. Two groups are constructed using the protein content of each sample, which is known in advance, using a binary threshold of 15% that places 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% 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 projecting in a specific direction that in this case corresponds to small variations on the subinterval $[1100, 1500]$. So, any depth procedure based on the whole interval cannot achieve a better result than a technique focused 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 it to the 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. In fact, the DD^G classifier converts the data into 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 procedures (kNN, NP, GLM, GAM). More classifiers could be considered here (such as SVM or 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. For instance, in our simulations, the GAM procedure is not always better than the GLM although the latter is a particular case of the former.

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

- The DD^G–classifier is especially interesting in a functional or high-dimensional framework because it changes the dimension of the classification problem from large or infinite 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, in addition, some options ways to reduce this number. In this paper, a review of functional data depths is made by including modifications to summarize multivariate functional data (the data are made up of vectors of functions) without increasing or even reducing the dimension of the problem at hand.

  In a multivariate setting, this might not be so advantageous because the dimension G is a multiple of the number of groups and it could sometimes 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, so that using the DD^G–classifier map in its most simple case can be worked 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. Most 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 (1.2.1) used to perform the procedures shown in the paper. (GNU zipped tar file)

References

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

Baíllo, 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.

Claeskens, G., Hubert, M., Slaets, L., and Vakili, K. (2014). Multivariate functional halfspace depth. *J. Amer. Statist. Assoc.*, 109(505):411–423.

Cuesta-Albertos, J. A., Fraiman, R., and Ransford, T. (2007). A sharp form of the cramer–wold theorem. *J. Theoret. Probab.*, 20(2):201–209.

Cuevas, A., Febrero, M., and Fraiman, R. (2004). An anova test for functional data. *Comput. Statist. Data Anal.*, 47(1):111–122.

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

Delaigle, A. and Hall, P. (2012). Achieving near perfect classification for functional data. *J. Royal Statist. Soc.: Series B*, 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. *J. Statist. Software*, 51(4):1–28.

Ferraty, F. and Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. *Comput. Statist. Data Anal.*, 44(1):161–173.

Ferraty, F. and Vieu, P. (2009). Additive prediction and boosting for functional data. *Comput. Statist. Data Anal.*, 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. *Scand. J. Statist.*, 32(2):327–350.

Hall, P., Park, B. U., and Samworth, R. J. (2008). Choice of neighbor order in nearest-neighbor classification. *Ann. Statist.*, pages 2135–2152.

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statist. Sci.*, 21(1):1–14.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *Ann. Statist.*, 23(1):73–102.

Ieva, F. and Paganoni, A. M. (2013). Depth measures for multivariate functional data. *Comm. Statist.-Theory Methods*, 42(7):1265–1276.

Kalivas, J. H. (1997). Two data sets of near infrared spectra. *Chemom. Intell. Lab. Syst.*, 37(2):255–259.

Lange, T., Mosler, K., and Mozharovskyi, P. (2014). Fast nonparametric classification based on data depth. *Statist. Papers*, 55(1):49–69.

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. *J. Amer. Statist. Assoc.*, 107(498):737–753.

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

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

Liu, R. Y., Parelius, J. M., and Singh, K. (1999). Multivariate analysis by data depth: descriptive statistics, graphics and inference, (with discussion and a rejoinder by Liu and Singh). *Ann. Statist.*, 27(3):783–858.

López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *J. Amer. Statist. Assoc.*, 104(486):718–734.

Lyons, R. (2013). Distance covariance in metric spaces. *Ann. Probab.*, 41(5):3284–3305.

Mosler, K. and Mozharovskyi, P. (2014). Fast dd-classification of functional data. *arXiv preprint arXiv:1403.1158*.

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 University 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. *Mol. Biol. Cell*, 9(12):3273–3297.

Székely, G. J. and Rizzo, M. L. (2013). The distance correlation t-test of independence in high dimension. *J. Multivariate Anal.*, 117:193–213.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *Ann. Statist.*, 35(6):2769–2794.

Vencálek, O. (2011). *Weighted data depth and depth based discrimination. Doctoral Thesis.* Charles University. Prague.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *J. Amer. Statist. Assoc.*, 99(467):673–686.

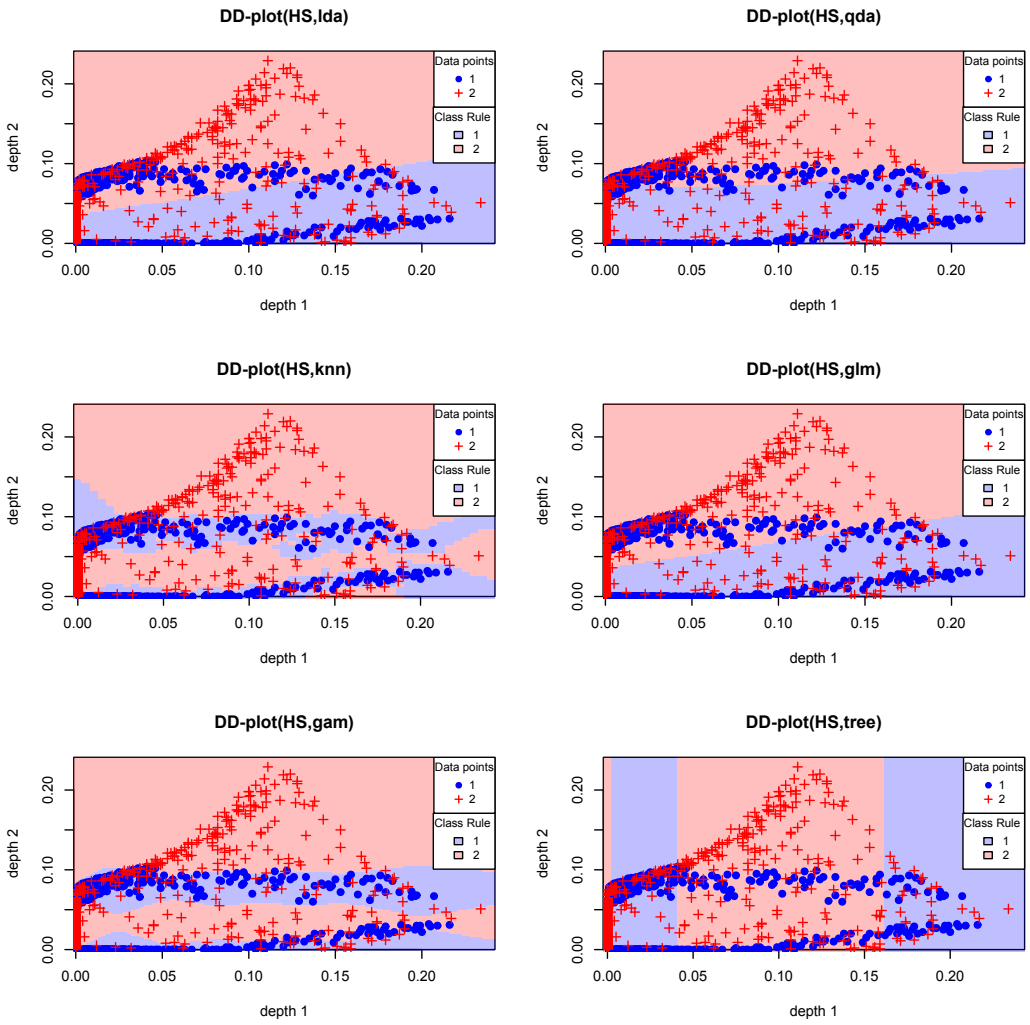

Figure 4: From left to right, top to bottom DD–plot using LDA, QDA, kNN, GLM, GAM and tree classifiers to the DD-plot in Figure 2.b. The one-dimensional depth in all cases is the Tukey depth.

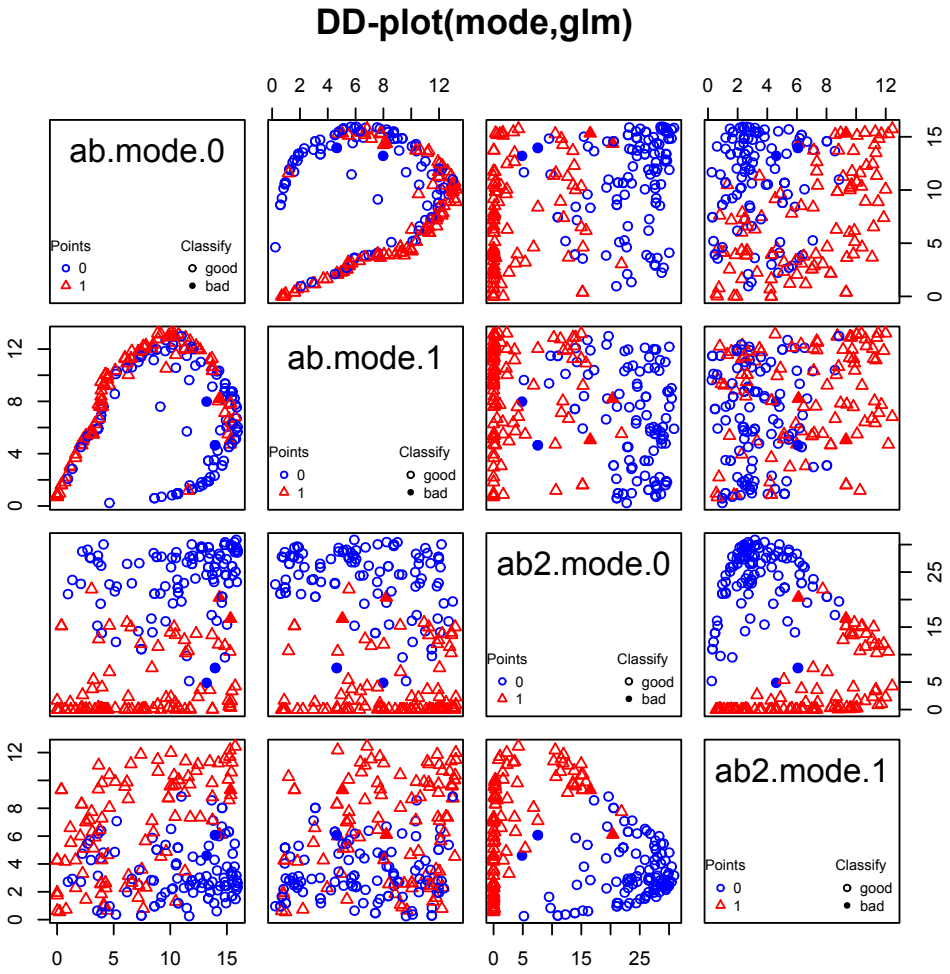

Figure 5: Example of GLM classifier using the spectrometric curves of the Tecator data set and its second derivative.

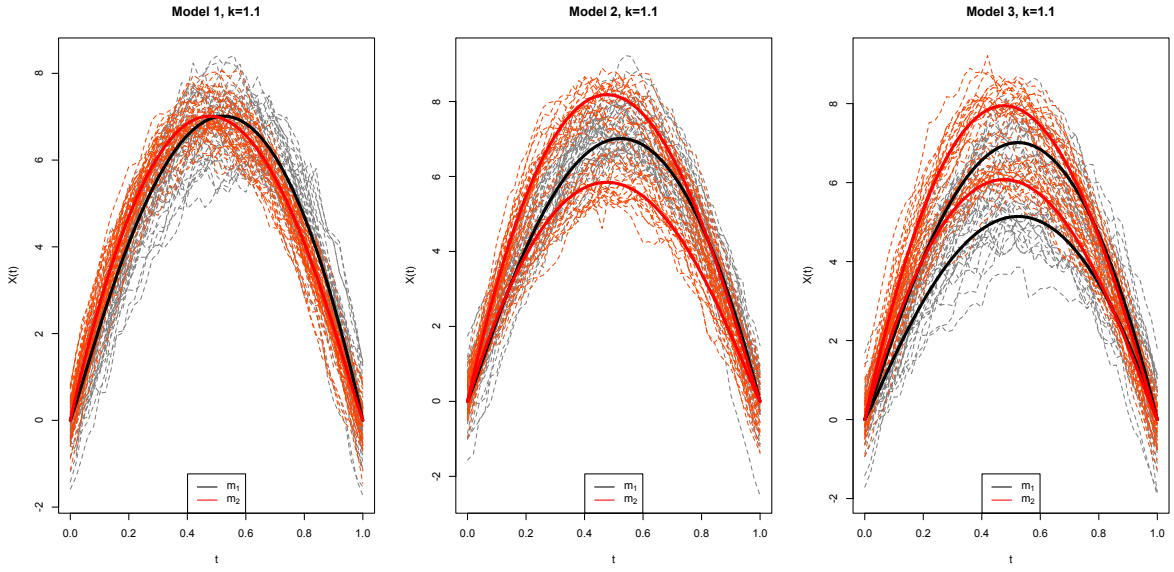

Figure 6: A sample of 40 functions for every simulation model along with the means of each sub-group (m_1's (black lines) and m_2's (red line)).