



The DD^G -classifier in the functional setting

J. A. Cuesta-Albertos¹  · M. Febrero-Bande²  ·
M. Oviedo de la Fuente² 

Received: 18 February 2015 / Accepted: 26 July 2016 / Published online: 6 August 2016
© Sociedad de Estadística e Investigación Operativa 2016

Abstract The maximum depth classifier was the first attempt to use data depths instead of multivariate raw data in classification problems. Recently, the DD-classifier has addressed some of the serious limitations of this classifier but issues still remain. This paper aims to extend the DD-classifier as follows: first, by enabling it to handle more than two groups; second, by applying regular classification methods (such as k NN, linear or quadratic classifiers, recursive partitioning, etc) to DD-plots, which is particularly useful, because it gives insights based on the diagnostics of these methods; and third, by integrating various sources of information (data depths, multivariate functional data, etc) in the classification procedure in a unified way. This paper also proposes an enhanced revision of several functional data depths and it provides a simulation study and applications to some real data sets.

Keywords DD-classifier · Functional depths · Functional data analysis

Electronic supplementary material The online version of this article (doi:[10.1007/s11749-016-0502-6](https://doi.org/10.1007/s11749-016-0502-6)) contains supplementary material, which is available to authorized users.

✉ M. Oviedo de la Fuente
manuel.oviedo@usc.es

J. A. Cuesta-Albertos
juan.cuesta@unican.es

M. Febrero-Bande
manuel.febrero@usc.es

¹ Department of Mathematics, Statistics and Computation, University of Cantabria, Santander, Spain

² Department of Statistics and Operations Research, University of Santiago de Compostela, Santiago de Compostela, Spain

Mathematics Subject Classification 62-09 · 62G99 · 62H30

1 Introduction

In this paper, we explore the possibilities of depths in classification problems in multi-dimensional or functional spaces. Depths are relatively simple tools that order points in a space depending on how deep they are with respect to a probability distribution, \mathbf{P} .

In the one-dimensional case, it is easy to order points with respect to \mathbf{P} . The median is the innermost point, and the extreme percentiles are the outermost points. Moreover, if F_P denotes the distribution function of \mathbf{P} , then

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

is an index which measures how deep $x \in \mathbb{R}$ is with respect to \mathbf{P} . This index can also be applied to samples by replacing F_P with the empirical distribution function. Other available possibilities for defining $D_P(x)$ (see, for instance, Sect. 2.1.1) include those in which $D_P(x)$ decreases with the distance between x and the mean of \mathbf{P} , in this case, the deepest point. Most of them are positive and bounded; the bigger the index is, the deeper the point is.

In the multidimensional case, no natural order is present; thus, it is hard to order the points from the inner to outer parts of a distribution or sample. Different authors have proposed several depths to overcome this difficulty. Liu et al. (1999) extensively review multivariate depths.

To the best of our knowledge, Liu (1990) paper was the first to use depths for classification and it did so by proposing the maximum depth classifier (MD classifier): given two probability measures (or classes, or groups) \mathbf{P} and \mathbf{Q} , and a depth, D , we classify the point x as produced by \mathbf{P} if $D_P(x) > D_Q(x)$. Ghosh and Chaudhuri (2005) fully develop this procedure.

The MD classifier seems quite reasonable, but it presents some drawbacks, which can be better understood with the help of the DD-plots, introduced in Liu et al. (1999) to graphically compare two multivariate distributions or samples (see also Li and Liu 2004). Given two probability distributions, \mathbf{P} and \mathbf{Q} on \mathbb{R}^p , a DD-plot is a two-dimensional graph (regardless of p) in which, for every $x \in \mathbb{R}^p$, the pair $(D_P(x), D_Q(x)) \in \mathbb{R}^2$ is represented. Examples of DD-plots appear in Figs. 1 and 2. Thus, the MD classifier gives to \mathbf{Q} (resp. to \mathbf{P}) the points represented in the DD-plot above (below) the main diagonal. Figure 1 contains two DD-plots corresponding to samples from bidimensional normal distributions in which \mathbf{P} is standard in both the cases. The mean of \mathbf{Q} in the first DD-plot is $(2, 2)^t$, and its covariance is the identity. In the other case, \mathbf{Q} is centered, but its covariance is twice the identity. In both graphs, points in black come from \mathbf{P} and points in gray come from \mathbf{Q} . We employed the Half-space Depth (HS) (see Liu et al. 1999). All sample sizes are 500. Both graphs show the main diagonal.

The MD classifier is optimal in the first case, but it is plainly wrong in the second case, because it classifies almost all the points as produced by \mathbf{Q} . The idea developed

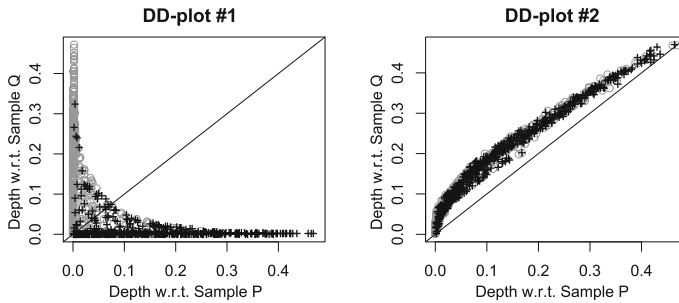


Fig. 1 DD-plots of two samples drawn from two-dimensional normal distributions. In both the cases, \mathbf{P} is a standard two-dimensional distribution. \mathbf{Q} differs from \mathbf{P} in the mean in the first DD-plot and in the covariance matrix in the second

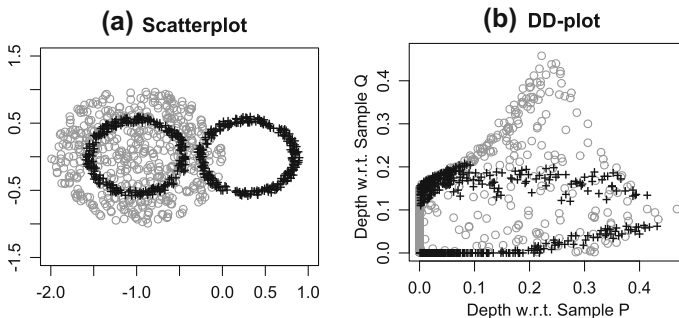


Fig. 2 Scatterplot of two uniform samples and associated DD-plot

in Li et al. (2012) is that the DD-plot contains information that helps obtain a good classifier. For instance, in the second DD-plot in Fig. 1, the proportion of gray points is very high in an area close to the vertical axis. Li et al. (2012) propose replacing the main diagonal by a function, whose graph splits the DD-plot into two zones with the lowest misclassification rate (the paper only fully develops the polynomial case). This is termed the DD-classifier.

The DD-classifier is a big improvement over the MD classifier and, in the problem cited above according to Li et al. (2012), the classification of the DD-classifier is very close to the optimal classification. However, an important limitation of the DD-classifier is that it is incapable of dealing efficiently with more than two groups. Li et al. (2012) solved this problem for g groups by applying a majority voting scheme that increased the computational complexity with the need to solve $\binom{g}{2}$ two-group problems.

Moreover, in some two-group cases, a function cannot split the points in the DD-plot correctly. Let us consider the situation presented in Fig. 2. The points in the scatterplot come from two samples with 2000 points each. The gray points come from a uniform distribution, \mathbf{Q} , on the unit ball centered on $(-1, 0)^t$. The black 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 0.5 (resp. 0.6) and a ring of the same size centered at $(0.3, 0)^t$. The optimal classifier assigns points in both rings to \mathbf{P} and the

rest to \mathbf{Q} . Figure 2 also shows the associated DD-plot. Obviously, no function can split the DD-plot into two zones giving the optimal classifier, since this would require a function separating the points in areas with black points from the rest of the DD-plot, which is impossible. Interchanging the axis could fix this particular problem. Yet, a situation in which this rotation is insufficient is seemingly possible.

There are also several valid depths in functional spaces (Sect. 2.1 presents some of them). By making use of the DD-classifier, we may also apply these depths in classification problems. However, the same problems mentioned in the multidimensional case will arise. Moreover, another limitation of the DD-plot is its incapacity to take information coming from different sources into account. This fault is more important in the functional setting in which we could classify using some transformations of the original curves (such as derivatives) simultaneously with the original trajectories.

In this paper, we present the DD^G -classifier as a way to address all the mentioned shortcomings of the DD-classifier in the functional setting, even though we may also apply this procedure to multivariate data or to the cartesian product of functional spaces with multivariate data. In fact, the DD^G -classifier allows for handling more than two groups and incorporating information from different sources. The price we pay for this is an increment in the dimension which goes from 2 in the DD-plot to the number of groups times the number of different sources of information handled. The DD^G -classifier can also handle more than one depth simultaneously (once again increasing the dimension). The letter G in the name of the procedure refers to this incremented dimension. Finally, it allows the use of regular classifiers (such as k NN, SVM, etc). Since it is no longer compulsory to use functions to separate groups, it is then possible, for instance, to identify “islands” inside a DD^G plot, thus avoiding the need to use rotations.

Concerning the combination of information, it is worth mentioning that, on one hand, Sect. 2.1 includes some extensions of well-known depths that let us construct new depths taking into account pieces of information from several sources; and, on the other hand, we can use some of the diagnostic tools of the classification procedures employed inside of the DD^G -classifier to assess the relevance of the available information. For the sake of brevity, we only show this idea in the second example in Sect. 3 in which we conclude that the second derivative of the curves contains the relevant information.

The paper is organized as follows: Sect. 2 presents the basic ideas behind the proposed classifier. Section 2.1 presents some functional depths and analyzes some modifications that may improve them. Section 3 shows two examples of several classifiers applied to DD-plots. Section 4 contains the results of some simulations as well as applications to a few real data sets. The paper ends with a discussion about the proposed method.

2 DD^G -classifier

Li et al. (2012) define the DD-plot in the case involving only two groups, as a two-dimensional graph that plots the pairs $(D_1(x), D_2(x))$. Here, $D_i(x)$ is the depth of the point x with 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} defining the data and \mathbb{R}^2 :

$$x \rightarrow (D_1(x), D_2(x)) \in \mathbb{R}^2.$$

The DD-classifier tries to identify the two groups using the information provided by the DD-plot. Since we have transformed our data to include them in \mathbb{R}^2 , the framework for separating classes is much simpler granted that the depths contain relevant information on how to separate the groups. Thus, the choice of a depth is now a crucial step. In Li et al. (2012), the classification rule is a polynomial function (up to a selected order k), ensuring that the point $(0, 0)^t$ belongs to it. This rule has three 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 is the number of possible ways to select k points from N , and each of the selections has an associated order k polynomial that interpolates between these k points and $(0, 0)^t$. Clearly, as N increases, the complexity of the estimation process grows at the rate N^k . Second, the problem of classifying more than two groups was solved in Li et al. (2012) using majority voting that requires repeating the procedure for every combination of the groups. This means that the optimization must be solved $\binom{g}{2}$ times, where g is the number of groups. In addition, to avoid the dependence of the classification rule on the pre-specified order of the groups, we must interchange the axes of the DD-plot and repeat the optimization procedure. Therefore, the number of polynomial models computed to create the classification rule is $2\binom{g}{2}\binom{N}{k}$, which can be extremely large. Finally, polynomials always give borders between groups which do not permit the construction of zones assigned to one group included in a zone assigned to the other, such as the horizontal black band between the gray zones in the DD-plot in Fig. 2.

The DD^G -classifier proposed herein attempts to offer a unified solution to these drawbacks. Suppose that we have a process in the product space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$, multivariate (functional) data, in which we have g groups (classes or distributions). Let us begin by assuming that $p = 1$. The DD^G -classifier begins by selecting a depth D and computing the following map:

$$x \rightarrow \mathbf{d} = (D_1(x), \dots, D_g(x)) \in \mathbb{R}^g.$$

We can now apply any available classifier that works in a g -dimensional space to separate the g groups. Lange et al. (2014) apply the same idea. The main differences between the proposal of Lange et al. and ours are that the former only considers finite-dimensional data, and this map is a preliminary step to constructing what is called the feature space. Then, the authors only use a special kind of linear classifier on this feature space. This requires making pairwise comparisons, thus classifying points using a majority vote scheme. Mosler and Mozharovskiy (2015) apply this classifier to functional data, but they do so only after performing a dimension-reduction technique on the data. An application of the k NN classifier to the classical DD-plot can be found in Vencálek (2011).

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

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

where $\mathbf{D}^i(x_i)$ is the g -dimensional vector giving the depths of the point $x_i \in \mathcal{X}_i$ with respect to the groups $1, \dots, g$ and $G = g \times p$.

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 select the right depth by cross-validation. In principle, an obvious solution could be to include all the depths at the same time and, from the diagnostics of the classification method, select which depths are useful. This approach, however, produces an increase of the dimension of vector \mathbf{d} up to $G = g \sum_{i=1}^p l_i$, where $l_i \geq 1$ is the number of depths used in the i th component. Clearly, the advantage of this approach depends on how the classification method handles the information provided by the depths. Instead of that, we propose selecting the useful depths while attempting to maintain the dimension G low. We can do this using the distance correlation \mathcal{R} , see Székely et al. (2007), which characterizes independence between vectors of arbitrary finite dimensions. Recently, Székely and Rizzo (2013) proposed a bias-corrected version. Here, our recommendation is to compute the bias-corrected distance correlation between the multivariate vector of depths (\mathbf{d}) and the indicator of the classes ($Y = (\mathbb{1}_{\{x \in C_1\}}, \mathbb{1}_{\{x \in C_2\}}, \dots, \mathbb{1}_{\{x \in C_g\}})$), and select the depth that maximizes the distance correlation among the available ones. In subsequent steps, we may add other depths having a low distance correlation with the ones selected in the previous steps. This tool could also be useful in assessing how much of the relation between the functional data and the indicator of the groups we can collect using the recent extension of the distance correlation to functional spaces provided by Lyons (2013). Indeed, the computation of this measure is quite easy, because it only depends on the distances among data (see Definition 4 in Székely et al. (2007)). In Sect. 3, we provide an example of the application of these ideas.

2.1 Data depths for functional data

As previously mentioned, the DD-classifier is especially interesting in the functional context, because it enables us to decrease the dimension of the classification problem from infinite to G . This section reviews several functional data depths that are later used with the DD^G -classifier and provides some extensions to cover multivariate functional data.

2.1.1 Fraiman and Muniz depth (FM)

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

$$\text{FM}_i = \int_0^T D_i(t) dt. \quad (2)$$

An obvious generalization of the FM depth is to consider integrating different univariate depths, such as the *Half-Space depth* (HS, which is defined in (1)), the *Simplicial depth* (SD), or the *Mahalanobis depth* (MhD):

$$D_i^{\text{SD}}(t) = 2F_{N,t}(x_i(t)) (1 - F_{N,t}(x_i(t)^-)),$$

$$D_i^{\text{MhD}}(t) = \left[1 + (x_i(t) - \hat{\mu}(t))^2 / \hat{\sigma}^2(t) \right]^{-1},$$

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. For instance, the deepest curve may vary depending on this selection.

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 is an appealing idea, because it can keep the dimension of our classification problem low. This would, however, come at a risk of losing information. We can do this in two ways:

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

$$\text{FM}_i^w = \sum_{j=1}^p w_j \text{FM}(x_i^j),$$

where $\mathbf{w} = (w_1, \dots, w_p)$ is a suitable vector of weights. In the choice of \mathbf{w} , we must take into account the differences in the scales of the depths (for instance, the FM depth using SD as the univariate depth takes values in $[0, 1]$, whereas the half-space 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 their derivatives). In this case, we can define a p -summarized version of FM depth (FM^p) as

$$\text{FM}_i^p = \int_0^T D_i^p(t) dt,$$

where $D_i^p(t)$ is a p -variate depth of the vector $(x_i^1(t), \dots, x_i^p(t))$ with respect to S_t .

2.1.2 h -mode depth (hM)

Cuevas et al. (2007) proposed the hM depth as a functional generalization of the likelihood depth to measure how surrounded one curve is with respect to the others. The population hM depth of a datum x_0 is given by

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

where X is a random element describing the population, m is a suitable metric or semi-metric, $K(\cdot)$ is a kernel, and h is the bandwidth parameter. Given a random sample

x_1, \dots, x_N of X , the empirical h -mode depth is defined as

$$\hat{f}_h(x_0) = N^{-1} \sum_{i=1}^N K(m(x_0, x_i)/h). \quad (3)$$

Equation (3) is similar to the usual nonparametric kernel density estimator, with a main difference: our interest is focused on what happens in a neighbourhood of each point, so the bandwidth is not intended to converge to zero when $N \rightarrow \infty$, 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 $K(0)/N$. Our default choice for h is the quantile 15 % of the distances among different points in the sample using the standard Gaussian density as K .

We can apply a weighted depth of the components and use it with multivariate functional data. Another alternative is to construct a new metric that combines the ones defined in the components of the product space using a p -dimensional metric, such as, for example, the Euclidean; i.e., take

$$m\left((x_0^1, \dots, x_0^p), (x_i^1, \dots, x_i^p)\right) := \sqrt{m_1(x_0^1, x_i^1)^2 + \dots + m_p(x_0^p, x_i^p)^2}, \quad (4)$$

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 prevent one single component from dominating the overall distance.

2.1.3 Random projection methods

Several depths based on random projections basically use the same scheme. Given a sample x_1, \dots, x_N of functions in a Hilbert space with scalar product $\langle \cdot, \cdot \rangle$, we randomly select a unit vector a in this space (independently of x_i) and project the data onto the one-dimensional subspace generated by a . The sample depth of a datum x is the univariate depth of the projection $\langle a, x \rangle$ with respect to the projected sample $\{\langle a, x_i \rangle\}_{i=1}^N$. Although a single projection is theoretically enough (see Cuesta-Albertos et al. 2007), a random projection method usually generates several directions, a_1, \dots, a_R , $R > 1$ and summarizes them in different ways. Here, we will use:

- *Random projection* (RP): proposed in Cuevas et al. (2007), it uses univariate HS depth and summarizes the depths of the projections through the mean (using $R = 50$ as a default choice). Therefore, if $D_{a_r}(x)$ is the depth associated with the r th projection, then

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

The extensions to multivariate functional data are similar to those proposed for the FM depth, except for the fact that using a p -variate depth with the projections here does not require a common support for all components. The RPD depth proposed in Cuevas et al. (2007) is an example of this extension using the original curves and their

derivatives as components of multivariate functional data, which are two dimension in this case.

2.1.4 Other depth measures

Some authors have proposed other functional depth measures over the last years, but they are all closely related to the three above-mentioned depths. For instance, the *modified band depth* (MBD) proposed in López-Pintado and Romo (2009) can be seen as a particular FM depth case using the simplicial depth as a univariate depth. The works by Ieva and Paganoni (2013) and Claeskens et al. (2014) are in line with the extension of the FM depth to multivariate functional data with common support. The first paper provides a generalization of the MBD that uses the simplicial depth as p -variate depth, and the second uses the multidimensional half-space depth.

The two proposals in Sguera et al. (2014) are the extension to functional data of the multivariate spatial depth (see Serfling 2004). The two depths, called functional spatial depth (FSD) and Kernelized functional spatial depth (KFSD), differ in meaning. The first one is a global depth, whereas the KFSD has a clear local pattern. We tried them and obtained that FSD gives results very much akin to FM or RP, while KFSD behaves like the hM depth. This is why we included none of them in the simulations and real-case studies.

2.2 Classification methods

The last step in the DD^G -classifier procedure is to select a suitable classification rule. Fortunately, we now have a purely multivariate classification problem in dimension G , which many procedures successfully handle based on either discriminant or regression ideas (see Ripley 1996).

We have chosen to use the following multivariate classification procedures given their simplicity and/or the ease with which inferences may be drawn from them:

1. *Based on discriminant analysis* The linear discriminant analysis (LDA) is the most classical discriminant procedure. Introduced by Fisher, it is a particular application of Bayes' rule classifier under the assumption that all groups in the population have a normal distribution with different means but the same covariance matrix. The quadratic discriminant analysis (QDA) is an extension relaxing the assumption of equality among covariance matrices.
2. *Based on logistic regression models* Here, the classifiers employ the logistic transformation to compute the posterior probability of belonging to a certain group using the information of the covariates. The generalized linear models (GLM) linearly combine the information of vector \mathbf{d} , whereas the generalized additive models (GAM) (see Wood 2004) relax the linearity assumption in GLMs allowing the use of a sum of general smooth functions of each variate.
3. *Nonparametric classification methods* are based on nonparametric estimates of the group densities. The classical and simplest method is the so-called k -Nearest Neighbour (k NN) in which, given $k \in \mathbb{N}$, point \mathbf{d} is assigned to the majority class of the k nearest data points in the training sample. Another possibility is

to use a common bandwidth for all data and the Nadaraya–Watson estimator to assess the probability of belonging to each group. NP denotes this method. A k NN method could be considered an NP method using the uniform kernel and a locally selected bandwidth. These two methods are quite flexible and powerful. However, as opposed to the previous methods, with these it is not easy to diagnose which part of vector \mathbf{d} is important for the final result.

Many other classifiers could be employed here, for instance: classification trees, artificial neural networks (ANN), support vector machines (SVM) or multivariate adaptive regression splines, among others, but applying any of these methods usually requires choosing several auxiliary parameters or designs tailored for each and every specific application. In addition, as in the case of nonparametric classification methods, the tradeoff between the interpretability and the predictability of these methods is biased towards the latter.

Theoretical properties and/or the ease with which one may draw inferences could influence one's choice of classifiers. For example, from the theoretical point of view, the k NN 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 the standard rule. Yet, better inferences may be drawn from other classifiers, such as LDA, GLM, or GAM models.

3 Illustration of regular classification methods in DD-plots

3.1 Multivariate example

This section explores the different classifiers that can be applied to DD-plots as an alternative to what [Li et al. \(2012\)](#) propose: given $k_0 = 1, 2, \dots$, the classifier is the polynomial f , with degree at most k_0 , such that $f(0) = 0$, which gives the lowest misclassification error in the training sample. We denote this classifier by DDk_0 . We select k_0 points of the sample and take the polynomial going through these points and the origin to construct the candidate polynomials.

In our implementation, we ignored the step in the implementation of [Li et al. \(2012\)](#) in which they select the order k_0 by cross-validation. Instead of that we provide the best result for $k_0 = 1, 2, 3$ using M initial combinations ($M = 10,000$ by default) in each case and optimizing the best m ones ($m = 1$ by default). Notice that we may consider the MD classifier as a particular case of DD1 simply fixing the slope with a value of 1.

Figure 3 shows the results for DD1, DD2, and DD3 classifiers and plots the application, for example, in Fig. 2b. The titles of the subplots are in the general form $DD\text{-plot}(\text{depth}, \text{classif})$, where *depth* is the depth employed (HS denotes the multidimensional half-space depth) and *classif* denotes the classification method. The gray or black colors of the sample points indicate the group that they belong to. The background image is light gray and dark gray to indicate the areas, where a new data point would be assigned to the gray or black groups, respectively.

The misclassification error rates are, respectively, (0.262, 0.215, and 0.201). There is a clear superiority of DD3 over the other classifiers, but in some areas (see, for exam-

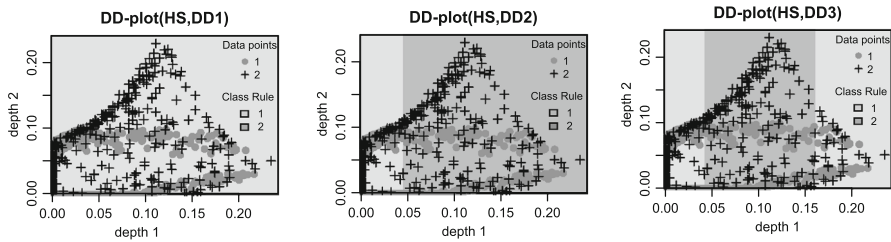


Fig. 3 From left to right DD-plot using DD1, DD2 and DD3 classifiers applied for the DD-plot in Fig. 2b. The depth in all the cases is the HS

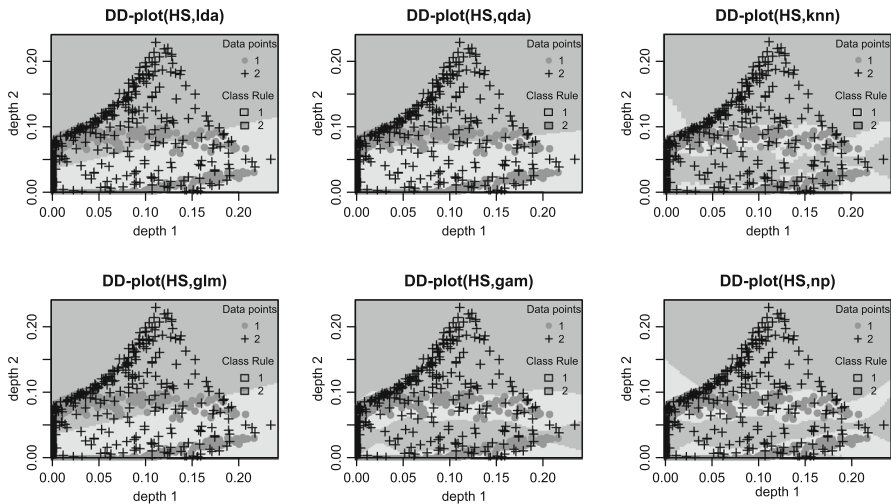


Fig. 4 From left to right, top to bottom DD-plot using LDA, QDA, k NN, GLM, GAM, and NP classifiers applied for the DD-plot in Fig. 2b. The depth in all the cases is the HS

ple, the rectangle $[0.0, 0.2] \times [0.0, 0.1]$, a polynomial cannot satisfactorily classify the data.

Figure 4 shows the result applying LDA, QDA, k NN, GLM, GAM, and NP to the same data. The misclassification rates are, respectively, (0.472, 0.51, 0.136, 0.472, 0.152 and 0.152). LDA, QDA, and GLM methods do not achieve the result obtained by DD3 which is outperformed by k NN, GAM, and NP. Notice that the optimal classifier gives an expected misclassification rate of 0.138, very close to the result obtained with k NN (the choice of k is done by cross-validation). The key to this improvement over the DD-classifier is the flexibility of k NN, GAM, and NP, which can model complicated situations like this one.

3.2 Functional example: tecator

In this section, we use the Tecator data set to illustrate our procedure. Section 4.1 revisits this data set to compare the performance of the DD^G -classifier from the prediction point of view with other proposals (Fig. 5).

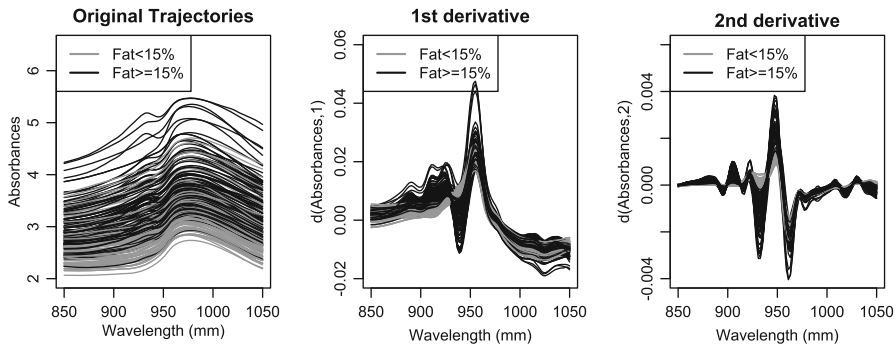


Fig. 5 Spectrometric curves of the Tecator data set and their first two derivatives

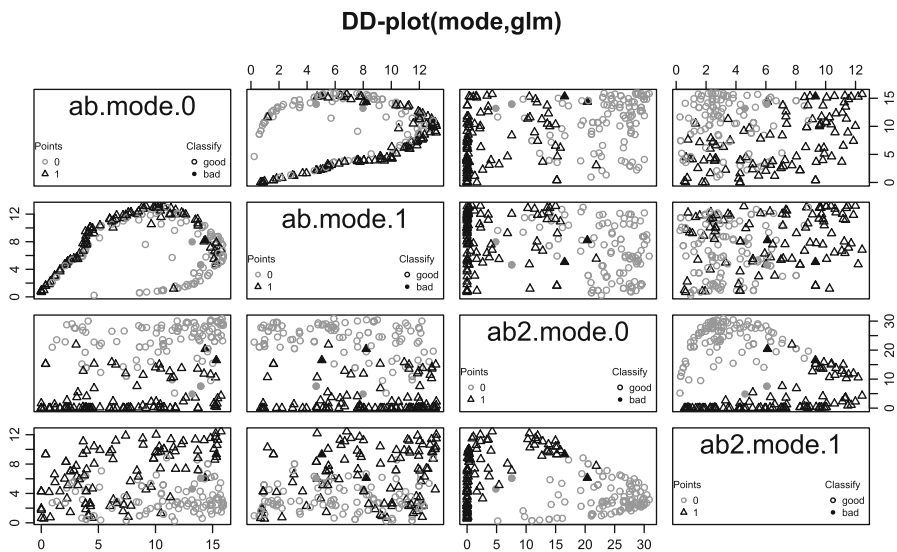
The data, treated herein as multivariate functional, were drawn for a spectrometric study to predict the fat content of meat slices using the absorbance curves provided by the *tecator infrared food analyzer* device. Many papers have treated the data from the point of view of the regression or classification (e.g., [Ferraty and Vieu 2009](#); [Febrero-Bande and González-Manteiga 2013](#) and references therein), and they have concluded that the relevant information for these goals is located in the second derivative. Figure 5 shows the original data jointly with their first two derivatives. The color depends on the fat content of the slices. Let us suppose here that we are interested in identifying the samples with percentage of fat above 15 % ($\text{ifat} = \mathbb{1}_{\{\text{Fat} \geq 0.15\}}$) using the absorbance curves (ab) and their second derivatives ($ab2$) with the DD^G -classifier. First, concerning the depth, we use FM, RP, and hM, where we have employed the univariate Mahalanobis depth to compute the first two and the usual L_2 -distance between functions in hM with the default choice of h equal to the quantile 0.15 of the set $\{d(x_i, x_j), i \neq j\}$. Then, we can explore at least five possibilities identified through the different suffixes for each depth:

- .0: The depth only uses original trajectories, $\mathbf{d} = (D_0^0(x), D_1^0(x))$.
- .2: The depth only uses the second derivatives, $\mathbf{d} = (D_0^2(x), D_1^2(x))$.
- .w: Use a weighted sum of the depth of the original trajectories and the depth of the second derivatives, $\mathbf{d} = (D_0^w(x), D_1^w(x))$ with $D_i^w = 0.5D_i^0 + 0.5D_i^2$.
- .m: Use all combinations depth/group, $\mathbf{d} = (D_0^0(x), D_1^0(x), D_0^2(x), D_1^2(x))$.
- .p: Combine the depths of the trajectories and their derivatives (a two-dimensional functional data set) within the depth procedure. With FM and RP depths, we use the two-dimensional Mahalanobis depth. The hM method uses an Euclidean metric as in (4), $\mathbf{d} = (D_1^p(x), D_2^p(x))$.

As mentioned above, the distance correlation proposed in [Székely et al. \(2007\)](#) can help detect the depth that best summarizes the variate ifat . Table 1 shows the distance correlation between the group variate (ifat) and the different depths. Since this metric only uses the distance among data, we can also compute it with respect to the functional covariates: $\mathcal{R}(\text{ifat}, ab) = 0.14$ and $\mathcal{R}(\text{ifat}, ab2) = 0.77$, supporting the idea that the second derivative contains the important information for classification. In Table 1, we also see that the depths based on the second derivative explain at least the same amount

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

	FM.0	FM.2	FM.w	FM.m	FM.p
$\mathcal{R}(\text{ifat}, \mathbf{d})$	0.058	0.771	0.393	0.365	0.058
	RP.0	RP.2	RP.w	RP.m	RP.p
$\mathcal{R}(\text{ifat}, \mathbf{d})$	0.065	0.774	0.407	0.396	0.065
	hM.0	hM.2	hM.w	hM.m	hM.p
$\mathcal{R}(\text{ifat}, \mathbf{d})$	0.114	0.789	0.706	0.762	0.114

**Fig. 6** Example of pairs of hM depths used by the GLM classifier with the spectrometric curves of the Tecator data set and its second derivative

of information as the functional covariate does. In particular, FM.2, RP.2, hM.2, hM.w, and hM.m have values over 0.7. The first derivative ($ab1$) was not considered here, because its distance correlation with ifat ($\mathcal{R}(\text{ifat}, ab1) = 0.63$) is lower than the second one and both are quite related among themselves ($\mathcal{R}(ab1, ab2) = 0.86$). Therefore, if we must select just one depth, we must chose hM.2. In a second step, if we want to add more information, it is preferable to include the original trajectories because of their lower distance correlation with $ab2$ ($\mathcal{R}(ab, ab2) = 0.23$).

The next step is to select a classifier that takes advantage of the dependence the distance correlation measure found. The k NN could be a good choice, because it is quite simple to implement. From the diagnosis point of view, however, a classifier, such as the GLM, may be preferable. Using the hM.m depth (second best choice), we have four variates: $ab.mode.0$, $ab.mode.1$, $ab2.mode.0$, and $ab2.mode.1$, where the notation $var.depth.group$ stands for the $depth$ computed for variate var with respect to the points in the group $group$.

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

	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

Figure 6 shows the result of using a GLM classifier with the combinations of the four variates, which clearly reveals that the ones associated with the second derivative separate the two groups more efficiently. More interestingly, the contribution of each component can be assessed through the diagnosis of the GLM. Table 2 shows the classical diagnosis of the estimates of a GLM model in which the variates associated with the depths of the second derivative are both clearly significant; this, however, is not true for the original curves.

4 A simulation study and the analysis of some real data sets

We simulated four models (inspired by the ones in Cuevas et al. (2007)) to check the performance of the proposed classifier. In all the cases, we obtained the curves 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 $e_{\cdot j}$ is a Gaussian process with zero mean and $\text{Cov}(e_{\cdot j}(s), e_{\cdot j}(t)) = \theta_j \exp(-|s - t|/0.3)$. In all the models, $\theta_1 = 0.5$ and $\theta_2 = 0.25$, which gives the second group half the error of the first. The mean functions include an additional parameter k , which is fixed at $k = 1.1$. Note that Cuevas et al. (2007) take $k = 1.2$, which makes the classification task easier given the bigger separation of the groups. The functions were generated in the interval $[0, 1]$ using an equispaced grid of 51 points. We chose these models in an attempt to preserve a high similarity jointly between groups in the original trajectories and in their derivatives.

- *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 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, with means $m_{1,I=0} = 22(1 - t)t^k$ and $m_{1,I=1} = 30(1 - t)t^k$, in the first population and $m_{2,I=0} = 26(1 - t)^k t$ and $m_{2,I=1} = 34(1 - t)^k t$ in the second one.
- *Model 4* This uses the same subgroups defined in Model 3 but considers each subgroup as a group in itself. Therefore, this is an example with four groups.

Thus, Models 1 and 4 are unimodal, while Models 2 and 3 contain at least one multimodal group. In the last two models, the hM depth (which is local) should do better than the others.

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 we generated a

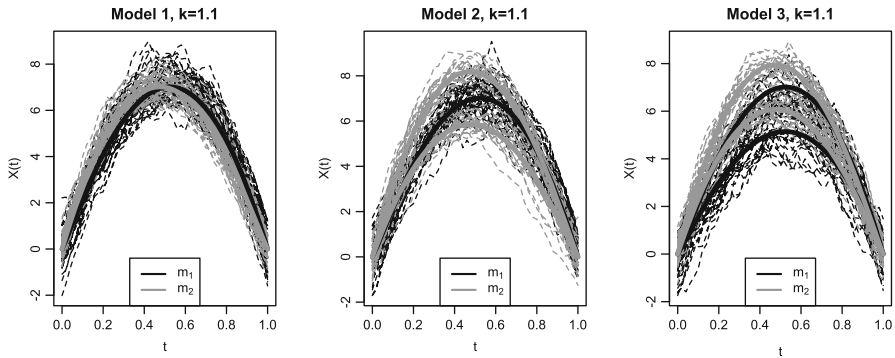


Fig. 7 Sample of 20 functions for every simulation model along with the means of each subgroup (m_1 's (black lines) and m_2 's (gray lines))

test sample of 50 observations from each group. For Models 3 and 4, we generated $N = 400$ training observations (100 for each subgroup). Tables 3, 4, 5, 6 show the misclassification rates for the test samples. Figure 7 presents some of the curves obtained with each model.

To compare, we employed the FM, RP, and hM depths (computed with the default choices explained in Sect. 2) using the original trajectories and/or derivatives of every curve, computed using splines. We denoted the different depth options, such as in Sect. 3, but we now used the first derivative ($.I$) instead of the second one.

We computed distance \mathcal{R} to select the best option among the different depths (first row of Tables 3, 4, 5, 6). The overall winner is hM.w (closely followed by hM.p and hM.m); this suggests that the combined information of the curves and the first derivatives is better than using only one of the two. We can deduce from the relatively small distance correlations obtained that this example is quite a difficult one for a classification task. As a reference, we have employed the Functional k NN (FkNN) with the usual cross-validation choice for k for all examples.

The list of classifiers includes DD1, DD2, and DD3 as classical classifiers and also LDA, QDA, k NN, NP, GLM, and GAM. Note that the procedures DDi , $i = 1, 2, 3$ cannot be used with the $.m$ option.

Table 3 summarizes the complete results for Model 1, where the results of the distance correlation are, broadly speaking, confirmed: hM.m renders the best results, closely followed by hM.w and hM.p. In these columns, the linear classifiers (LDA and GLM) seem to work slightly better than the others. This means that the simplest linear models can perform the classification task successfully. We computed the FkNN in its three versions: $.0$, $.I$, and $.p$; the last of which uses the Euclidean distance combining the first two. The results were 23.04, 18.93, and 19.02%, respectively.

Model 2 (Table 4) is a difficult scenario for methods based on RP and FM depths, as we may deduce from the low values of the distance correlation. These methods work well in homogeneous groups rather than in groups comprised of subgroups, as is the case. The least misclassification error is obtained with the combinations hM.p-QDA, hM.p-GLM, and hM.p-GAM (9.3%), although many classifiers based on hM.w, hM.p, or hM.m have misclassification rates under 10%. The results for FkNN were 13.63, 13.31, and 12.57%.

Table 3 Distance correlation and misclassification rates for Model 1

	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	27.2	21.5	20.3	21.2		24.5	16.7	17.6	17.8		20.3	15.9	15.6	15.7	
DD2	24.9	20.4	18.6	19.4		24.0	15.4	17.4	17.6		18.7	13.7	12.6	12.8	
DD3	25.1	20.4	18.7	19.5		24.4	15.8	17.1	17.3		19.0	14.0	13.1	13.3	
LDA	24.2	20.3	18.4	19.4	17.9	24.6	15.6	17.5	17.5	15.5	18.4	13.1	12.2	12.3	11.7
QDA	24.5	20.3	18.4	19.5	18.1	25.1	15.8	17.5	17.6	16.4	18.5	13.2	12.1	12.3	11.9
kNN	28.2	23.0	20.8	21.9	20.7	27.4	18.0	19.3	19.3	17.7	20.4	15.1	13.6	13.9	13.3
NP	28.9	24.0	21.6	22.3	18.7	29.0	18.7	20.2	20.2	16.2	21.5	15.8	14.6	14.7	12.5
GLM	24.1	19.9	18.3	19.2	17.8	24.3	15.6	17.2	17.2	15.4	18.4	13.1	12.1	12.3	11.6
GAM	24.2	20.0	18.0	19.0	17.8	23.8	15.2	16.7	16.9	15.2	18.2	13.1	12.1	12.2	11.7

Mean of 200 runs

Table 4 Distance correlation and misclassification rates for Model 2

	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	32.5	26.6	25.5	16.1		31.1	22.3	22.0	21.6		14.0	15.8	10.6	10.2	
DD2	22.8	27.1	21.0	16.0		20.8	21.1	16.6	16.3		11.5	16.0	10.6	9.9	
DD3	23.0	27.4	21.3	16.1		20.8	20.9	16.6	16.3		11.8	16.2	10.8	10.2	
LDA	22.0	26.4	20.3	16.1	17.9	21.1	20.3	16.4	17.0	15.5	12.3	15.1	10.0	9.7	10.1
QDA	22.3	26.7	20.6	15.9	18.7	20.8	20.5	16.1	16.5	16.0	11.9	14.9	9.8	9.3	9.8
kNN	25.8	30.7	24.0	18.2	21.1	22.0	23.6	17.9	18.1	17.5	12.7	17.3	11.5	10.8	11.5
NP	26.7	31.5	25.0	18.9	18.8	22.9	24.4	18.9	19.1	16.5	13.3	18.1	12.3	11.6	10.7
GLM	22.1	26.3	20.2	15.6	17.9	19.7	20.1	15.4	15.9	15.0	11.7	15.1	9.7	9.3	9.7
GAM	22.4	26.4	20.5	15.5	18.1	19.5	20.1	15.2	15.7	15.2	11.0	15.3	9.9	9.3	9.6

Mean of 200 runs

Model 3 (Table 5) is even more difficult for the RP and FM methods. In both the cases, using the first derivative is better than using the original curves or a weighted version of them. For these depths, the combined information '[FM.p-GAM (20.6 %) and RP.m-NP (21.7 %)] results in the best misclassification errors. This is also true for the hM method, but it consistently yields lower misclassification errors. The best combinations are hM.w-GAM and hM.m-GAM (16.2 %), which give better results than those obtained using FkNN: 23, 21.4, 21.21 %.

Model 4 results (Table 6) are better than Model 3 results. This supports the idea that homogeneous groups are easier to classify with RP and FM depths. In all the cases, the weighted version improves the classification of every individual component. This hints that the two components have complementary pieces of the information required for classifying. The best combinations for each depth are: FM.w-GLM (15.9 %), RP.m-GLM (14 %), hM.w-GAM, hM.p-GLM, and hM.m-GLM (11.3 %). The FkNN gives quite disappointing results: 19.8, 21.45, and 20.16 %; which is probably due to the difficulty of the scenario.

4.1 Application to real data sets

We have applied our proposal to several popular data sets in the functional data analysis literature. [Bafllo et al. \(2010\)](#) extensively review functional classification. In what follows, we will briefly describe the data sets, the results found in the literature, and our best results using the DD^G -classifier.

- *Tecator* We observe several scheme differences in the literature upon the use of the Tecator data set for classification. This includes the cutoff for groups, the size of the training, and testing samples and even the number of runs. In [Febrero-Bande and Oviedo de la Fuente \(2012\)](#), the scheme cutoff=15%/train=165/test=50/runs=500 is employed obtaining with an FKGAM model a best misclassification rate of 2.1 %. Here, using depths, the best result is 1.3 % with the hM.2-DD2 model. The classical FkNN using the second derivative obtains 1.9 %.

In [Galeano et al. \(2015\)](#), a misclassification error of 1 % is reported using a centroid method with the functional Mahalanobis semidistance and with the scheme cutoff=20%/train=162/test=53/runs=500. Following the same scheme but with 200 runs, the hM.2-DD2 (error rate: 1.3 %) performs quite well and slightly better than does the classifier using kNN (2.5 %). In fact, all the classifiers using the second derivative show misclassification rates in the interval [1.3, 3.3 %]. This is comparable to the FkNN classifier, which obtains 1.92 %.

- *Berkeley growth study* This data set contains 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, [Bafllo and Cuevas \(2008\)](#) treated this data set and obtained a best cross-validation misclassification rate of 3.23 % using an FkNN procedure. In our application, we obtain a better result using the combinations hM.0-LDA and hM.0-QDA (2.2 %).

Table 5 Distance correlation and misclassification rates for Model 3

	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	30.9	27.9	29.9	28.4		31.2	27.9	29.7	29.5		27.4	23.3	24.8	25.5	
DD2	29.5	24.5	26.1	22.0		29.8	24.9	27.2	27.0		19.4	17.9	16.3	16.9	
DD3	25.5	24.3	21.8	21.4		28.3	23.2	23.4	23.3		19.4	18.1	16.5	17.0	
LDA	32.0	25.2	29.8	26.4	25.1	32.0	27.0	30.3	30.6	27.0	24.0	18.4	19.1	20.2	18.2
QDA	28.4	23.9	24.9	22.3	21.5	30.3	24.7	26.6	26.6	23.4	21.9	17.7	17.7	18.6	16.9
kNN	25.9	25.0	22.2	21.3	21.6	27.3	23.6	22.9	22.9	22.4	20.3	18.2	16.6	17.2	16.7
NP	25.5	24.3	22.1	21.3	21.3	27.7	23.3	22.5	22.8	21.7	19.9	17.9	16.4	17.0	16.4
GLM	32.0	25.1	29.8	26.4	25.2	32.3	27.2	30.4	30.6	27.3	23.8	18.3	18.7	20.0	18.0
GAM	24.8	23.8	21.2	20.6	21.6	26.1	22.9	22.1	21.8	21.8	19.4	17.6	16.2	16.8	16.2

Mean of 200 runs

Table 6 Distance correlation and misclassification rates for Model 4

	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
DD1	21.6	29.1	17.8	18.1		23.9	19.5	16.9	16.8		19.5	17.9	14.6	14.4	
DD2	21.7	28.9	17.2	17.7		23.9	18.9	16.7	16.4		17.9	16.6	12.7	12.5	
DD3	23.0	29.6	18.9	19.2		25.2	20.3	18.3	18.1		19.4	18.0	14.6	14.4	
LDA	21.0	27.4	16.4	16.9	16.8	23.2	18.3	15.9	16.0	14.3	17.6	15.9	12.5	12.1	12.0
QDA	21.0	28.0	17.0	18.4	17.4	23.8	18.9	16.6	16.5	16.3	17.9	16.2	11.8	12.2	12.7
kNN	21.5	30.2	17.1	17.6	17.4	23.0	19.2	16.1	16.0	15.3	17.3	16.5	12.0	12.2	12.4
NP	20.7	28.2	16.6	17.1	17.0	22.3	18.6	15.8	15.6	15.3	17.0	16.1	11.9	12.0	12.4
GLM	20.9	27.7	15.9	16.4	16.1	23.0	18.0	15.5	15.4	14.0	16.6	15.8	11.4	11.3	11.3
GAM	20.6	27.5	16.0	16.5	16.9	21.8	17.9	15.1	15.1	14.5	15.9	15.8	11.3	11.5	12.1

Mean of 200 runs

- *Phoneme*: The *phoneme* data set is also quite popular in the FDA community, although it has its origins in the area of Statistical Learning (see [Hastie et al. 1995](#)). The data set has 2000 log-periodograms of 32 ms duration corresponding to five different phonemes (*sh*, *dcl*, *iy*, *aa*, *ao*). This appeared as a functional classification problem in [Ferraty and Vieu \(2003\)](#). The authors randomly split the data into training and test samples with 250 cases, 50 per class, in each sample and repeat the procedure 200 times. Their best result was an 8.5 % misclassification rate. With our proposals, the combination hM.m–LDA misclassifies 7.5 %.
- *Delaigle and Hall (2012)* also used this data set, but restricted it to the use of the first 50 discretization points and to the binary case using the two most difficult phonemes, (*aa*, *ao*) and they obtained 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 s for an hour in the isolated cardiac cells of a mouse. [Cuevas et al. \(2004\)](#) used the data (two groups: control and treatment) as functional data for ANOVA testing; the data set is available in the `fda.usc` package. Baíllo and Cuevas (2008) considered it an FDA classification problem, where a cross-validation procedure yielded a best error rate of 11.23 %. Our best results are the combinations hM.l–DD1, hM.m–LDA, hM.m–QDA, and hM.m–NP with an error rate of 2.2 %.
- *Cell cycle* This data set contains temporal gene expression measured every 7 min (18 observations per curve) of 90 genes involved in the yeast cell cycle. [Spellman et al. \(1998\)](#) originally obtained the data and [Leng and Müller \(2006\)](#) and [Rincón Hidalgo and Ruiz Medina \(2012\)](#) used it to classify these genes into two groups. The first group has 44 elements related to 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. This work imputed several missing observations in the data set using a B-spline basis of 21 elements. Both papers cited above obtain a misclassification rate of 10 % (nine misclassified genes) with a different number of errors for each group. Our proposal achieves a 6.7 % rate with the combinations hM.l–DD1, hM.w–kNN, hM.w–NP, hM.p–DD1, hM.m–kNN, and hM.m–NP, but almost all procedures based on hM.l or hM.w yield a misclassification rate of 8.9 % at the most.
- *Kalivas* This example comes from [Kalivas \(1997\)](#). [Delaigle and Hall \(2012\)](#) used it for classification. It contains near-infrared spectra of 100 wheat samples from 1100 to 2500 nm in 2 nm intervals. With the protein content of each sample, it constructs two groups using a binary threshold of 15 % that places 41 data 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 far beyond the best in [Delaigle and Hall \(2012\)](#) (where, using a centroid classifier, they got $CENT_{PC1} = 0.22$ %), but the latter requires projecting in a specific direction that corresponds to small variations on the subinterval [1100, 1500] in this case. Notice that any depth procedure based on the whole interval cannot achieve a

better result than a technique focused on the small interval containing the relevant information for the discrimination process.

5 Conclusions

This paper presents a procedure that extends the DD-classifier procedure proposed in Li et al. (2012) and adapts it to the functional context in several ways:

- Due to the flexibility of the new classifier, the proposal can deal with several depths or with more than two groups within the same integrated framework. In fact, the DD^G -classifier converts the data into a multivariate data set with columns constructed using depths. Then, it uses a suitable classifier among the classical ones based on discrimination (LDA, QDA) or regression procedures (k NN, NP, GLM, and GAM). Further research could consider more classifiers (such as SVM or ANN) without changing the procedure too much. The choice of a classifier must be based on the weakness and the strengths of each one of them. For instance, the use of classifiers, such as LDA or GLM, is recommended for the diagnostic part, because it is easier to interpret the rule for separating groups, even if this comes at a cost to the predictive performance.
- The DD^G -classifier is especially interesting within a high-dimensional or functional 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 study and the number of depths that the statistician may decide to employ times the number of sources of information used. For instance, if we have three groups in the data and the method uses two different depths, the multivariate dimension of the DD^G -classifier is 6. This dimension is clearly more tractable for the problem, but we may reduce this number in some other ways too. This paper reviews functional data depths by including modifications to summarize multivariate functional data (the data are made up of vectorial 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 that could sometimes be greater than the dimension of the original space. For instance, the classical example of Fisher Iris data has four variables and three groups, so we can work in dimension three using the DD^G -classifier map in its simplest. However, we can also consider a univariate depth for each variable. Dimension G then grows to 12.

- The execution time for each method, measured in CPU seconds, depends on the complexity of the combination depth/classifier. Taking the Model 1 with the original curves as a reference, we obtain the fastest time with the combination FM–LDA (0.05 s). The QDA and GLM methods render similar times. Using the GAM classifier adds 0.07 s. The nonparametric classifiers (NP and k NN) typically add 0.35–0.40 s to the time due to the computation of the distance matrix among points in the DD-plot. The use of random projections increases the time 0.01 s per combination and the computation of the hM depth takes 1.05 s, the time employed by the FkNN. The use of a combined depth option ($.w$, $.p$, $.m$) doubles the execution time. The DDk choices obtain 0.07, 13.77, and 39.77 s, respectively,

with the default choice ($M = 10000$ and $m = 1$), even though we can achieve better execution times using $M = 500$ and $m = 50$ while maintaining similar misclassification rates.

- 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) in versions higher than 1.2.2. `classif.DD` is the principal function; it contains all the options shown in this paper related to depths and classifiers. Most of the figures we present are regular outputs of this function.

Acknowledgements This research was partially supported by the Spanish Ministerio de Ciencia y Tecnología, Grants MTM2011-28657-C02-02, MTM2014-56235-C2-2-P (J.A. Cuesta-Albertos) and MTM2013-41383-P (M. Febrero-Bande and M. Oviedo de la Fuente).

References

- Baíllo A, Cuevas A (2008) Supervised functional classification: a theoretical remark and some comparisons. [arXiv:0806.2831](https://arxiv.org/abs/0806.2831) (arXiv preprint)
- Baíllo A, Cuevas A, Fraiman R (2010) The Oxford handbook of functional data analysis, chap Classification methods for functional data. Oxford University Press, Oxford, pp 259–297
- Claeskens G, Hubert M, Slaets L, Vakili K (2014) Multivariate functional halfspace depth. *J Am Stat Assoc* 109(505):411–423
- Cuesta-Albertos JA, Fraiman R, Ransford T (2007) A sharp form of the Cramer–Wold theorem. *J Theor Probab* 20(2):201–209
- Cuevas A, Febrero M, Fraiman R (2004) An ANOVA test for functional data. *Comput Stat Data Anal* 47(1):111–122
- Cuevas A, Febrero M, Fraiman R (2007) Robust estimation and classification for functional data via projection-based depth notions. *Comput Stat* 22(3):481–496
- Delaigle A, Hall P (2012) Achieving near perfect classification for functional data. *J R Stat Soc Ser B* 74(2):267–286
- Febrero-Bande M, Oviedo de la Fuente M (2012) Statistical computing in functional data analysis: the R package `fda.usc`. *J Stat Softw* 51(4):1–28
- Febrero-Bande M, González-Manteiga W (2013) Generalized additive models for functional data. *TEST* 22(2):278–292
- Ferraty F, Vieu P (2003) Curves discrimination: a nonparametric functional approach. *Comput Stat Data Anal* 44(1):161–173
- Ferraty F, Vieu P (2009) Additive prediction and boosting for functional data. *Comput Stat Data Anal* 53(4):1400–1413
- Fraiman R, Muniz G (2001) Trimmed means for functional data. *TEST* 10(2):419–440
- Galeano P, Esdras J, Lillo RE (2015) The mahalanobis distance for functional data with applications to classification. *Technometrics* 57(2):281–291
- Ghosh AK, Chaudhuri P (2005) On maximum depth and related classifiers. *Scand J Stat* 32(2):327–350
- Hall P, Park BU, Samworth RJ (2008) Choice of neighbor order in nearest-neighbor classification. *Ann Stat* 36(5):2135–2152
- Hastie T, Buja A, Tibshirani R (1995) Penalized discriminant analysis. *Ann Stat* 23(1):73–102
- Ieva F, Paganoni AM (2013) Depth measures for multivariate functional data. *Comm Stat Theory Methods* 42(7):1265–1276
- Kalivas JH (1997) Two data sets of near infrared spectra. *Chemom Intell Lab Syst* 37(2):255–259
- Lange T, Mosler K, Mozharovskiy P (2014) Fast nonparametric classification based on data depth. *Stat Pap* 55(1):49–69
- Leng X, Müller HG (2006) Classification using functional data analysis for temporal gene expression data. *Bioinformatics* 22(1):68–76
- Li J, Liu R (2004) New nonparametric tests of multivariate locations and scales using data depth. *Stat Sci* 19(4):686–696

- Li J, Cuesta-Albertos JA, Liu RY (2012) *DD*-Classifier: nonparametric classification procedure based on *DD*-plot. *J Am Stat Assoc* 107(498):737–753
- Liu RY (1990) On a notion of data depth based on random simplices. *Ann Stat* 18(1):405–414
- Liu RY, Parelius JM, Singh K (1999) Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Ann Stat* 27(3):783–858
- López-Pintado S, Romo J (2009) On the concept of depth for functional data. *J Am Stat Assoc* 104(486):718–734
- Lyons R (2013) Distance covariance in metric spaces. *Ann Probab* 41(5):3284–3305
- Mosler K, Mozharovskiy P (2015) Fast *DD*-classification of functional data. *Stat Pap* 1–35. doi:[10.1007/s00362-015-0738-3](https://doi.org/10.1007/s00362-015-0738-3)
- Ramsay J, Silverman B (2005) *Functional data analysis*. Springer, Berlin
- Rincón Hidalgo MM, Ruiz Medina MD (2012) Local wavelet-vaguelette-based functional classification of gene expression data. *Biom J* 54(1):75–93
- Ripley B (1996) *Pattern recognition and neural networks*. Cambridge University Press, Cambridge
- Serfling R (2004) Nonparametric multivariate descriptive measures based on spatial quantiles. *J Stat Plann Inference* 123(2):259–278
- Sguera C, Galeano P, Lillo R (2014) Spatial depth-based classification for functional data. *TEST* 23(4):725–750
- Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, 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 GJ, Rizzo ML (2013) The distance correlation *t*-test of independence in high dimension. *J Multivar Anal* 117:193–213
- Székely GJ, Rizzo ML, Bakirov NK (2007) Measuring and testing dependence by correlation of distances. *Ann Stat* 35(6):2769–2794
- Vencálek O (2011) *Weighted data depth and depth based discrimination*. Doctoral Thesis. Charles University. Prague
- Wood SN (2004) Stable and efficient multiple smoothing parameter estimation for generalized additive models. *J Am Stat Assoc* 99(467):673–686