21st IEEE International Conference on Machine Learning and Applications (ICMLA), Nassau, Bahamas, 2022

DOI: https://doi.org/10.1109/ICMLA55696.2022.00143

Classification of Functional Data: A Comparative Study

Carlos Ramos-Carreño
Department of Computer Science
Universidad Autónoma de Madrid
Madrid, Spain
0000-0003-2566-7058

José Luis Torrecilla
Department of Mathematics
Universidad Autónoma de Madrid
Madrid, Spain
0000-0003-3719-5190

Alberto Suárez
Department of Computer Science
Universidad Autónoma de Madrid
Madrid, Spain
0000-0003-4534-0909

***Abstract*—In functional classification problems the data available for learning are characterized by functions, rather than vectors of attributes. In consequence, multivariate classifiers need to be adapted, and new types of classifiers designed to take into account the special characteristics of these types of data. In this work, an empirical evaluation of different classification methods is carried out using a variety of functional classification problems from different areas of application. The classifiers considered include nearest centroids with functional means as class prototypes and functional distances, standard multivariate classifiers used in combination with a variable selection method, classifiers based on the notion of functional depth, a functional version of k-nearest neighbors (k-NN), and random forest. From the results of this comparative study one concludes that random forest is among the best off-the-shelf classifiers not only for multivariate but also for functional classification problems. The variable selection method used in combination with a quadratic discriminant has fairly good overall accuracy using only a small set of impact points. This dimensionality reduction leads to improvements both in efficiency and interpretability. Finally, a functional version of k-NN that uses the α-Mahalanobis distance exhibits consistently good predictive performance in all the problems considered. This robustness makes k-NN a good benchmark for functional classification.**

***Index Terms*—functional data analysis, classification, Mahalanobis distance, functional k-NN**

I. Introduction

In many machine learning problems, the data available for induction are characterized by continuous functions. Examples include learning from time signals (e.g., the diagnosis of heart disease from electrocardiograms [1]), spatial data (e.g., the estimation of ore concentration in mineral deposits [2]), or spatio-temporal data (e.g., causal discovery in climate science [3]). For the purpose of learning, the curves that characterize the training instances can be modeled as random functions, which are the object of study of a branch of statistics known as Functional Data Analysis (FDA). Given that these functions depend on a continuous parameter, they are infinite-dimensional and possess statistical properties that are markedly different from random vectors [4], [5]. As a result, standard multivariate methods such as logistic regression or discriminant analysis cannot be applied directly [4], [5]. Furthermore, some classification problems are singular, which means that one can, in principle, achieve zero Bayes error [6].

A functional dataset consists of a collection of N instances $\mathcal{D} = \{(x_n(t), y_n)\,; n = 1, \ldots, N\}$, where $x_n(t)$, with $t \in \mathcal{T} \subset \mathbb{R}$, is the real-valued function that characterizes the n example, and $y_n \in \{1, \ldots, K\}$ is the corresponding class label. Typically, the functions are available as measurements of the process at a grid of M discrete times $\mathbf{t} = (t_1, \ldots, t_M)$

$$x(\mathbf{t}) = (x(t_1), \ldots, x(t_M)) \tag{1}$$

The goal of this study is to compare the predictive performance of different classifiers for these types of data. Most off-the-shelf methods, such as the ones implemented in the Python library scikit-learn [7], assume that the instances to be classified are characterized by vectors of attributes. The direct application of these methods to functional data in discretized form does not take advantage of their continuous structure, and can be problematic from a theoretical and practical perspective. A possible approach is to use a dimensionality reduction method and then apply a standard multivariate classifier [8]. Alternatively, the classification method can be adapted to the functional setting. For instance, a nearest-centroid classifier can be built using the per-class functional means as prototypes and a functional metric to compute distances [6]. It is also possible to utilize functional measures of centrality in a sample to design depth-based classifiers [9], [10]. In k-NN, a functional distance can be used to identify the nearest neighbors [11]. Random forest [12] can be directly employed with functional data by using the values of the functions as attributes.

The performance of these different types of classifiers has been tested using a wide range of functional classification problems from different areas of application. A total of 5 multiclass datasets (arrowheads, fish, phoneme, plane and symbols) and 13 binary classification problems are used in this comparison. A summary of the characteristics of these datasets is presented in Table I. In this table, N is the sample size, M the size of the grid, and K the number of classes. For the Phoneme dataset the data has been smoothed by applying a Nadaraya-Watson smoother to the original curves. A binary version of this dataset truncated to the first 50 features has also been included in the study [6]. We also include the second derivatives of Tecator, as for this dataset they contain most of the information [13]. To compute the derivatives, the original curves have been approximated using B-splines.

TABLE I
CHARACTERISTICS OF THE DATASETS CONSIDERED.

	N	M	K	Majority class (%)	Reference
ArrowHead	211	251	3	38.39	[14]
Australian	190	365	2	77.37	[15]
Cell	90	18	2	51.11	[16]
Coffee	56	286	2	51.79	[14]
ECG	2026	85	2	74.33	[11]
Fish	350	463	7	14.29	[14]
Growth	93	31	2	58.06	[11]
GunPoint	200	150	2	50.00	[14]
MCO	89	360	2	50.56	[11]
Medflies	534	30	2	52.06	[11]
NOx	115	24	2	66.09	[17]
Phoneme	4509	256	5	25.79	[11]
Phoneme (bin)	1717	50	2	59.52	[11]
Plane	210	144	7	14.29	[14]
Symbols	1020	398	6	17.75	[14]
Tecator	215	100	2	64.19	[13]
Tecator (2^{nd} der)	215	100	2	64.19	[13]
Yoga	3300	426	2	53.64	[14]

II. EMPIRICAL EVALUATION

In this study, four different families of classifiers are considered: nearest centroid classifiers [6], a functional variable selection method [8] used in combination with different multivariate classifiers, classifiers based on the notion of depth [9], [10], and k-NN classifiers that employ different functional distances [11]. In what follows, we first compare classifiers within each of these families. From each of these families the classifier that has the best overall predictive performance in the problems considered is selected. Finally, the selected predictors are compared among each other and with random forest, which is one of the best off-the-shelf classifiers [18].

The infrastructure for the empirical evaluation is provided by the scikit-datasets Python package [19]. The classifiers are built using scikit-fda [20], a Python package that offers a comprehensive set of tools for statistical analysis and machine learning for functional data, in combination with scikit-learn [7]. For each of the classification problems, stratified sampling is used to partition the data into a training set, which includes 70% of the instances available for learning, and a test set with the remaining ones. When necessary, 5-fold cross-validation within the training set is used to determine the values of the hyperparameters of the different classifiers.

The results reported consist of the mean accuracy and standard deviation over 100 random partitions for each particular classifier-dataset combination. To account for sample variability, the classifiers are trained and tested using the same partitions. For each dataset, the scores of the best and second best predictors are highlighted in boldface and underlined, respectively. An asterisk is used to indicate statistically significant differences at the 5% level using a paired t-test.

An overall comparison of the different classification methods is made in terms of their average accuracy and rank. A Friedman test is used to determine whether the overall differences among average ranks are statistically significant. If significant differences are detected by this test a pairwise post-hoc Nemenyi test is used [21]. The results of these tests are summarized in critical distance (CD) diagrams generated with the autorank Python package [22].

A. Nearest centroid classifiers

Nearest centroid classifiers (NC) are simple methods that assign to a given observation $x(t)$ the label of the class whose centroid (the mean) c_k, $k = 1, \ldots, K$ is closest to it:

$$\hat{y} = \underset{k=1,\ldots,K}{\arg\min}\, d(x(t), c_k), \tag{2}$$

where $d(\cdot)$ is a functional distance. In spite of their simplicity, they can achieve optimal results for some classification problems [6]. Four different metrics are considered:

- NC-L1 uses the L_1-distance.
- NC-L2 with the L_2-distance.
- NC-Mah uses a kind of Mahalanobis distance. Since the covariance operator is non-invertible in the functional setting, this distance cannot be directly translated. Here we use the α-Mahalanobis distance proposed in [23], an adaptation of the original measure, in which the covariance operator is regularized so that it becomes invertible. The regularization parameter α is fixed by cross validation in the range $\{10^{-i}, i = 0, \ldots, 6\}$.
- NC-Ang utilizes the angular distance defined as

$$d_{\text{angular}}(x_1, x_2) = \frac{1}{\pi} \arccos\left(\frac{\langle x_1, x_2\rangle}{\|x_1\|\|x_2\|}\right). \tag{3}$$

The results of these experiments are presented in Table II and summarized in a CD diagram (Figure 1). NC with the α-Mahalanobis distance clearly outperforms the other competitors. The average rank of NC-Mah is significantly better than NC-L1 and NC-L2. It also obtains significant victories in a number of datasets.

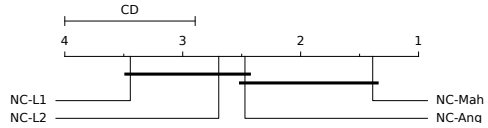

Fig. 1. CD diagram of the nearest centroid classifiers. Classifiers are grouped with a thick dark line if the differences among their average ranks are not statistically significant.

B. Functional variable selection

An usual strategy to deal with functional data consists in applying a dimensionality reduction method to transform the original trajectories into vectors. Then, any standard multivariate classifier can be used. Variable selection methodology provides interpretable reductions by replacing the original functions by their values at several well chosen points. In this work, we have chosen the Reproducing Kernel-based Variable Selection (RKVS) method proposed by one of us in [8] as a representative of this family of techniques. The name RKVS refers to the role of the reproducing kernel Hilbert spaces

TABLE II
COMPARISON AMONG NEAREST CENTROIDS CLASSIFIERS.

	NC-L1	**NC-L2**	**NC-Mah**	**NC-Ang**
ArrowHead	0.734 ± 0.045 (4)	0.765 ± 0.054 (3)	**0.782 ± 0.043 (1)***	0.765 ± 0.054 (2)
Australian	0.864 ± 0.038 (3)	0.852 ± 0.041 (4)	0.903 ± 0.040 (2)	**0.906 ± 0.034 (1)**
Cell	0.861 ± 0.069 (3)	0.861 ± 0.065 (2)	**0.869 ± 0.063 (1)**	0.818 ± 0.067 (4)
Coffee	0.955 ± 0.049 (4)	0.959 ± 0.048 (2)	**1.000 ± 0.000 (1)***	0.959 ± 0.048 (2)
ECG	0.780 ± 0.025 (4)	0.840 ± 0.016 (2)	**0.971 ± 0.006 (1)***	0.790 ± 0.028 (3)
Fish	0.616 ± 0.041 (4)	0.635 ± 0.045 (3)	**0.716 ± 0.041 (1)***	0.640 ± 0.044 (2)
Growth	0.742 ± 0.102 (4)	0.777 ± 0.093 (3)	**0.958 ± 0.032 (1)***	0.934 ± 0.039 (2)
GunPoint	0.712 ± 0.060 (2)	0.706 ± 0.058 (3)	**0.770 ± 0.051 (1)***	0.684 ± 0.055 (4)
MCO	0.635 ± 0.079 (4)	0.637 ± 0.078 (3)	**0.986 ± 0.026 (1)***	0.839 ± 0.068 (2)
Medflies	0.551 ± 0.031 (3)	**0.556 ± 0.030 (1)**	0.548 ± 0.034 (4)	0.553 ± 0.032 (2)
NOx	0.726 ± 0.077 (4)	0.782 ± 0.079 (3)	**0.898 ± 0.052 (1)***	0.851 ± 0.057 (2)
Phoneme	0.851 ± 0.008 (4)	0.867 ± 0.007 (3)	**0.910 ± 0.006 (1)***	0.869 ± 0.008 (2)
Phoneme (binary)	0.750 ± 0.017 (4)	0.762 ± 0.016 (3)	**0.812 ± 0.015 (1)***	0.798 ± 0.015 (2)
Plane	0.954 ± 0.028 (2)	0.953 ± 0.027 (3)	**0.969 ± 0.020 (1)***	0.949 ± 0.028 (4)
Symbols	0.887 ± 0.015 (3)	0.891 ± 0.015 (2)	**0.920 ± 0.016 (1)***	0.878 ± 0.015 (4)
Tecator	0.681 ± 0.046 (4)	0.685 ± 0.045 (3)	**0.953 ± 0.021 (1)***	0.860 ± 0.036 (2)
Tecator (2nd derivative)	0.961 ± 0.020 (2)	0.960 ± 0.022 (3)	0.959 ± 0.021 (4)	**0.971 ± 0.017 (1)***
Yoga	0.519 ± 0.035 (4)	0.528 ± 0.038 (2)	**0.578 ± 0.016 (1)***	0.527 ± 0.032 (3)
Average accuracy	*0.766*	*0.779*	***0.861***	*0.811*
Average rank	*3.444*	*2.667*	***1.389***	*2.444*

theory in deriving properties of the method, such as optimality results for some Gaussian models. RKVS aims at selecting the variables to maximize the (multivariate) Mahalanobis distance between the class means in the reduced space.

Here, we use the greedy implementation of RKVS available in scikit-fda with four standard multivariate classifiers included in scikit-learn: linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbors with the Euclidean distance (k-NN) and random forest (RF). The number of variables selected and the k of k-NN are chosen by cross-validation between 1 and 10. The number of trees in RF y fixed to 100. Finally, as RKVS is defined for binary problems, we follow a one-versus-rest strategy for the multiclass datasets.

Accuracy results and ranking for each dataset are shown in Table III. In general terms, there are no such big differences as in the previous section. This is illustrated in the associated CD diagram shown in Figure 2. However, RKVS+QDA and RKVS+k-NN obtain better results with significant victories in three datasets each. In particular, the application of QDA exhibit better global results as indicated by the average rank. This difference it is not appreciated in the average accuracy by the effect of the Yoga problem where the performance of RKVS+QDA is bad, maybe because of a lack of Gaussianity.

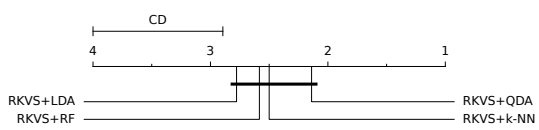

Fig. 2. Critical distance diagram of the multivariate classifiers applied after RKVS variable selection.

C. Depth-based classifiers

Another classification strategy is labelling observations according with their centrality in the populations: the more central an observation is in a sample, the more probable it is to belong to that sample. The notion of centrality is quite natural in one-dimensional data, but it becomes fuzzy in higher dimensions. This effect is particularly critical in the infinite dimensional spaces of function where nothing similar to a natural order exists. Fortunately, there is a bunch of statistical depth functions that allow us to quantify this centrality in order to define the median, detect outliers or classify.

In this context, the simpler proposals for classifying are the so called maximum depth methods (MD) [9]. They assign a new observation into the class in which it is deeper. That is, the class label assigned to $x(t)$ is

$$\hat{y} = \underset{k=1,\ldots,K}{\arg\max} D_k(x(t)), \tag{4}$$

where D_k denotes the depth measure in the k-th class.

A different approach to depth-based classification is followed by the generalized depth-depth classifier (DD$^{\mathrm{G}}$) [10]. This methodology consists in projecting the functional data into a lower-dimensional *space of depths* and then, use any multivariate classification rule. The features in this reduced space are the depth values of the trajectories in each class, for one or more depth measures. In other words, given a functional observation $x(t)$ and a set of depth functions $D^1, \ldots, D^G$, the DD$^{\mathrm{G}}$ classifier performs the transformation $x(t) \to (D^i_k(x(t)))$, $i = 1, \ldots, G$, $k = 1, \ldots, K$, before classifying. Here, we consider the simplest case of $G = 1$.

We have tested these two techniques with two different depth measures: Modified Band Depth (MBD) [24], and a depth measure based on the previously commented α-Mahalanobis distance (Mah) [23]. DD$^{\mathrm{G}}$ is combined with a simple k-NN with the Euclidean distance. The number of neighbors is set by cross-validation between 1 and 10. Accuracy and rank positions by dataset are shown in Table IV. The associated CD diagram is shown in Figure 3.

These results are quite straightforward. On the one hand, DD$^{\mathrm{G}}$ methods clearly beat their same depth MD counterparts.

TABLE III
COMPARISON AMONG RKVS-BASED CLASSIFIERS.

	RKVS+LDA	**RKVS+QDA**	**RKVS+k-NN**	**RKVS+RF**
ArrowHead	0.787 ± 0.044 (4)	0.832 ± 0.043 (3)	**0.855 ± 0.045 (1)***	0.834 ± 0.043 (2)
Australian	0.889 ± 0.035 (4)	0.890 ± 0.039 (3)	0.936 ± 0.029 (2)	**0.937 ± 0.030 (1)**
Cell	0.842 ± 0.059 (3)	0.841 ± 0.062 (4)	0.883 ± 0.062 (2)	**0.891 ± 0.061 (1)**
Coffee	**0.974 ± 0.044 (1)**	0.970 ± 0.045 (2)	0.968 ± 0.048 (3)	0.951 ± 0.052 (4)
ECG	0.982 ± 0.005 (4)	0.987 ± 0.006 (3)	**0.997 ± 0.002 (1)***	0.993 ± 0.005 (2)
Fish	0.804 ± 0.039 (2)	**0.824 ± 0.037 (1)***	0.786 ± 0.038 (3)	0.770 ± 0.041 (4)
Growth	0.951 ± 0.034 (2)	**0.952 ± 0.038 (1)**	0.944 ± 0.040 (3)	0.916 ± 0.054 (4)
GunPoint	0.881 ± 0.038 (4)	0.909 ± 0.042 (2)	0.892 ± 0.041 (3)	**0.928 ± 0.039 (1)***
MCO	**0.961 ± 0.044 (1)**	0.959 ± 0.040 (2)	0.881 ± 0.060 (3)	0.855 ± 0.072 (4)
Medflies	0.591 ± 0.033 (2)	0.589 ± 0.033 (3)	0.568 ± 0.033 (4)	**0.599 ± 0.032 (1)**
NOx	0.914 ± 0.048 (2)	**0.919 ± 0.037 (1)**	0.890 ± 0.050 (3)	0.877 ± 0.049 (4)
Phoneme	0.922 ± 0.006 (4)	**0.927 ± 0.005 (1)***	0.923 ± 0.005 (3)	0.924 ± 0.006 (2)
Phoneme (binary)	**0.820 ± 0.015 (1)***	0.815 ± 0.015 (2)	0.806 ± 0.013 (3)	0.806 ± 0.014 (4)
Plane	0.971 ± 0.019 (2)	**0.974 ± 0.020 (1)**	0.960 ± 0.020 (4)	0.971 ± 0.021 (3)
Symbols	0.870 ± 0.015 (4)	0.950 ± 0.016 (3)	0.959 ± 0.010 (2)	**0.961 ± 0.010 (1)**
Tecator	0.938 ± 0.026 (2)	**0.974 ± 0.019 (1)***	0.879 ± 0.035 (3)	0.833 ± 0.045 (4)
Tecator (2nd derivative)	0.936 ± 0.026 (4)	0.978 ± 0.015 (2)	**0.980 ± 0.015 (1)**	0.978 ± 0.017 (2)
Yoga	0.686 ± 0.019 (4)	0.752 ± 0.027 (3)	**0.902 ± 0.011 (1)***	0.899 ± 0.010 (2)
Average accuracy	*0.873*	***0.891***	*0.889*	*0.884*
Average rank	*2.778*	***2.111***	*2.500*	*2.556*

On the other hand, given a classifier, the α-Mahalanobis depth versions outperform those using MBD. In summary, DD^G-Mah is the undoubted winner in this family with better performance in average and significant victories in most datasets.

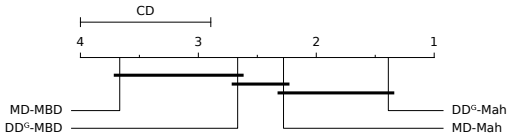

Fig. 3. Critical distance diagram of the depth based classifiers.

D. Functional k-NN classifiers

The k-nearest neighbors classifier (k-NN) can be extended to the functional case by using a distance between functions to find the neighbors. It can be considered a sort of reference method for supervised classification in FDA [11]. Some of the reasons for it are its simplicity, ease of motivation and the fact that it typically does not lead to gross classification errors.

We have tested the k-NN classifier implemented in scikit-fda with the same metrics used for the NC-Classifiers (Subsection II-A): L_1, L_2, α-Mahalanobis and Angular. Parameters k and α are selected by cross-validation as before. The results by classifier-dataset can be seen in Table V, and the corresponding critical distance diagram is shown in Figure 4.

Rank results show a certain equality between all the proposals, with no significant differences (see Figure 4). However, versions with Mahalanobis and angular distances obtain better results in terms of accuracy. These differences are mostly motivated by the bad performances of L1 and L2 proposals in MCO and Tecator datasets, which probably need a more global approach. Finally, choose k-NN-Mah over k-NN-Ang because of the number of significant victories.

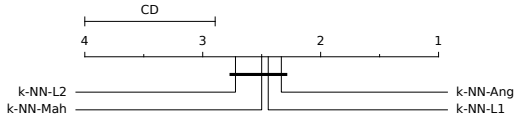

Fig. 4. Critical distance diagram of the functional k-NN classifiers.

E. Final comparison

In this section, a final comparison is made among the best classifiers of each family and a random forest composed of 100 trees. The results are summarized in Table VI and in Figure 5.

A first conclusion of this study is that the α-Mahalanobis distance, recently proposed in [23], has very good properties for classification when used by a distance-based functional classifier, such as nearest centroids or k-NN. Furthermore, it has also proven to be useful to design depth-based classifiers.

From the results, it is apparent that the best overall accuracy is achieved by the functional version of k-NN that employs the α-Mahalanobis distance, random forest, and the classifier that uses RKVS-variable selection followed by QDA.

The depth-based classifier has the poorest performance among the methods considered. This means that depth, while being a useful statistical concept, does not capture all the information necessary for classification. It should therefore be used in combination with additional features.

The performance of the nearest centroid classifier is affected by the poor accuracy obtained when the mean is not a good prototype of the curves of a given class; for example, if the distribution of the curves is asymmetric, multimodal or the curves are not aligned, as in ArrowHead, GunPoint and Yoga.

Random forest (RF) achieves the highest average rank in the problems considered. This means that it is one of the best classifiers not only for multivariate, but also for functional classification problems. A possible explanation is that RF takes advantage of the high dimensionality of the data to build

TABLE IV
COMPARISON AMONG DEPTH-BASED CLASSIFIERS.

	MD-MBD	**MD-Mah**	**DD^G-MBD**	**DD^G-Mah**
ArrowHead	0.722 ± 0.051 (4)	0.813 ± 0.048 (2)	0.739 ± 0.045 (3)	**0.825 ± 0.043 (1)***
Australian	0.747 ± 0.055 (4)	0.870 ± 0.047 (3)	0.875 ± 0.040 (2)	**0.898 ± 0.043 (1)***
Cell	0.764 ± 0.081 (4)	0.869 ± 0.063 (3)	**0.878 ± 0.054 (1)**	0.875 ± 0.058 (2)
Coffee	0.895 ± 0.068 (3)	**0.978 ± 0.039 (1)**	0.893 ± 0.065 (4)	0.972 ± 0.049 (2)
ECG	0.740 ± 0.024 (4)	0.909 ± 0.021 (2)	0.874 ± 0.010 (3)	**0.942 ± 0.013 (1)***
Fish	0.453 ± 0.057 (4)	0.619 ± 0.042 (3)	0.633 ± 0.041 (2)	**0.687 ± 0.044 (1)***
Growth	0.739 ± 0.100 (3)	**0.941 ± 0.040 (1)***	0.718 ± 0.080 (4)	0.927 ± 0.044 (2)
GunPoint	0.536 ± 0.026 (4)	0.793 ± 0.058 (3)	0.829 ± 0.049 (2)	**0.835 ± 0.053 (1)**
MCO	0.634 ± 0.077 (4)	0.877 ± 0.070 (2)	0.636 ± 0.088 (3)	**0.904 ± 0.053 (1)***
Medflies	0.529 ± 0.025 (2)	0.489 ± 0.037 (4)	**0.552 ± 0.039 (1)***	0.515 ± 0.038 (3)
NOx	0.700 ± 0.072 (4)	0.809 ± 0.055 (2)	0.778 ± 0.052 (3)	**0.863 ± 0.053 (1)***
Phoneme	0.813 ± 0.009 (4)	0.901 ± 0.006 (2)	0.847 ± 0.008 (3)	**0.905 ± 0.006 (1)***
Phoneme (binary)	0.745 ± 0.018 (3)	0.797 ± 0.015 (2)	0.721 ± 0.022 (4)	**0.804 ± 0.016 (1)***
Plane	0.901 ± 0.044 (4)	0.948 ± 0.027 (2)	**0.963 ± 0.028 (1)***	0.937 ± 0.029 (3)
Symbols	0.722 ± 0.022 (4)	0.926 ± 0.016 (2)	0.922 ± 0.013 (3)	**0.957 ± 0.011 (1)***
Tecator	0.684 ± 0.043 (3)	0.947 ± 0.028 (2)	0.621 ± 0.047 (4)	**0.967 ± 0.024 (1)***
Tecator (2nd derivative)	0.964 ± 0.023 (4)	0.970 ± 0.018 (2)	0.967 ± 0.017 (3)	**0.978 ± 0.017 (1)***
Yoga	0.588 ± 0.020 (4)	0.640 ± 0.016 (3)	0.659 ± 0.014 (2)	**0.698 ± 0.015 (1)***
Average accuracy	*0.715*	*0.839*	*0.784*	***0.860***
Average rank	*3.667*	*2.278*	*2.667*	***1.389***

TABLE V
COMPARISON AMONG FUNCTIONAL k-NN CLASSIFIERS.

	k-NN-L1	**k-NN-L2**	**k-NN-Mah**	**k-NN-Ang**
ArrowHead	0.883 ± 0.038 (4)	0.892 ± 0.034 (3)	**0.903 ± 0.031 (1)***	0.896 ± 0.034 (2)
Australian	0.945 ± 0.029 (3)	0.944 ± 0.024 (4)	0.951 ± 0.026 (2)	**0.958 ± 0.028 (1)***
Cell	0.918 ± 0.050 (3)	**0.935 ± 0.047 (1)***	0.924 ± 0.051 (2)	0.877 ± 0.055 (4)
Coffee	0.971 ± 0.041 (3)	0.978 ± 0.033 (2)	0.962 ± 0.049 (4)	**0.979 ± 0.033 (1)**
ECG	0.996 ± 0.002 (4)	0.998 ± 0.002 (2)	0.998 ± 0.002 (3)	**0.999 ± 0.002 (1)***
Fish	0.796 ± 0.037 (3)	0.816 ± 0.032 (2)	0.776 ± 0.033 (4)	**0.818 ± 0.032 (1)**
Growth	**0.957 ± 0.033 (1)**	0.956 ± 0.036 (2)	0.921 ± 0.052 (3)	0.918 ± 0.042 (4)
GunPoint	**0.951 ± 0.028 (1)***	0.938 ± 0.026 (3)	0.916 ± 0.031 (4)	0.938 ± 0.026 (2)
MCO	0.774 ± 0.079 (4)	0.801 ± 0.072 (3)	**0.986 ± 0.025 (1)***	0.941 ± 0.036 (2)
Medflies	0.541 ± 0.036 (3)	0.540 ± 0.034 (4)	0.544 ± 0.035 (2)	**0.550 ± 0.035 (1)**
NOx	**0.894 ± 0.046 (1)***	0.877 ± 0.046 (2)	0.871 ± 0.054 (3)	0.858 ± 0.049 (4)
Phoneme	0.909 ± 0.007 (2)	0.908 ± 0.006 (3)	**0.911 ± 0.006 (1)***	0.894 ± 0.007 (4)
Phoneme (binary)	**0.809 ± 0.014 (1)**	0.808 ± 0.015 (2)	0.803 ± 0.015 (4)	0.806 ± 0.015 (3)
Plane	0.968 ± 0.021 (2)	0.964 ± 0.022 (4)	**0.983 ± 0.014 (1)***	0.964 ± 0.021 (3)
Symbols	**0.966 ± 0.008 (1)**	0.962 ± 0.009 (4)	0.966 ± 0.009 (2)	0.962 ± 0.009 (3)
Tecator	0.798 ± 0.051 (4)	0.825 ± 0.048 (3)	**0.952 ± 0.024 (1)***	0.939 ± 0.028 (2)
Tecator (2nd derivative)	**0.980 ± 0.018 (1)**	0.973 ± 0.019 (4)	0.980 ± 0.019 (3)	0.980 ± 0.017 (2)
Yoga	0.932 ± 0.008 (3)	**0.933 ± 0.008 (1)**	0.916 ± 0.009 (4)	0.933 ± 0.008 (2)
Average accuracy	*0.888*	*0.892*	***0.903***	*0.901*
Average rank	*2.444*	*2.722*	*2.500*	***2.333***

decision trees whose predictions are complementary, in the sense that their individual errors tend to be independent and are therefore averaged out in the final forest prediction. An interesting avenue of exploration is to enhance the design of RF to take advantage of the functional nature of the data.

The multivariate classifiers LDA, QDA, k-NN, and RF have an excellent overall performance when used in combination with RKVS. This is probably a consequence of the functional nature of this variable selection procedure. A further advantage is that the data, in principle infinite dimensional, is represented in a low dimensional space. This dimensionality reduction entails significant gains in efficiency, simplifies the analysis, and improves the interpretability of the classifiers learned.

The functional k-NN classifier is robust and performs consistently well in the classification problems considered, as evidenced by the fact that it has the highest average accuracy. This provides further empirical support to the proposal made in [11] for k-NN to be used as a benchmark method for comparison in functional classification problems.

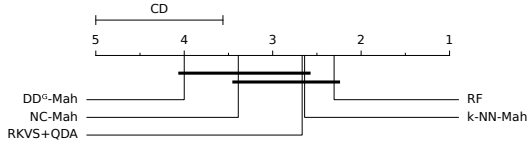

Fig. 5. Critical distance diagram of the classifiers.

TABLE VI
FINAL COMPARISON AMONG THE CLASSIFIERS.

	NC-Mah	**RKVS+QDA**	**DD^G-Mah**	**k-NN-Mah**	**RF**
ArrowHead	0.782 ± 0.043 (5)	0.832 ± 0.043 (3)	0.825 ± 0.043 (4)	**0.903 ± 0.031 (1)***	0.852 ± 0.041 (2)
Australian	0.903 ± 0.040 (3)	0.890 ± 0.039 (5)	0.898 ± 0.043 (4)	0.951 ± 0.026 (2)	**0.961 ± 0.026 (1)***
Cell	0.869 ± 0.063 (4)	0.841 ± 0.062 (5)	0.875 ± 0.058 (3)	**0.924 ± 0.051 (1)***	0.911 ± 0.048 (2)
Coffee	**1.000 ± 0.000 (1)***	0.970 ± 0.045 (4)	0.972 ± 0.049 (3)	0.962 ± 0.049 (5)	0.979 ± 0.029 (2)
ECG	0.971 ± 0.006 (4)	0.987 ± 0.006 (3)	0.942 ± 0.013 (5)	**0.998 ± 0.002 (1)***	0.992 ± 0.003 (2)
Fish	0.716 ± 0.041 (4)	**0.824 ± 0.037 (1)***	0.687 ± 0.044 (5)	0.776 ± 0.033 (3)	0.806 ± 0.037 (2)
Growth	**0.958 ± 0.032 (1)**	0.952 ± 0.038 (2)	0.927 ± 0.044 (3)	0.921 ± 0.052 (4)	0.917 ± 0.059 (5)
GunPoint	0.770 ± 0.051 (5)	0.909 ± 0.042 (3)	0.835 ± 0.053 (4)	0.916 ± 0.031 (2)	**0.964 ± 0.022 (1)***
MCO	**0.986 ± 0.026 (1)**	0.959 ± 0.040 (3)	0.904 ± 0.053 (4)	0.986 ± 0.025 (2)	0.783 ± 0.081 (5)
Medflies	0.548 ± 0.034 (3)	0.589 ± 0.033 (2)	0.515 ± 0.038 (5)	0.544 ± 0.035 (4)	**0.621 ± 0.027 (1)***
NOx	0.898 ± 0.052 (2)	**0.919 ± 0.037 (1)***	0.863 ± 0.053 (5)	0.871 ± 0.054 (3)	0.869 ± 0.054 (4)
Phoneme	0.910 ± 0.006 (4)	**0.927 ± 0.005 (1)***	0.905 ± 0.006 (5)	0.911 ± 0.006 (3)	0.924 ± 0.006 (2)
Phoneme (binary)	0.812 ± 0.015 (2)	**0.815 ± 0.015 (1)***	0.804 ± 0.016 (4)	0.803 ± 0.015 (5)	0.810 ± 0.015 (3)
Plane	0.969 ± 0.020 (4)	0.974 ± 0.020 (3)	0.937 ± 0.029 (5)	**0.983 ± 0.014 (1)**	**0.983 ± 0.017 (1)**
Symbols	0.920 ± 0.016 (5)	0.950 ± 0.016 (4)	0.957 ± 0.011 (3)	0.966 ± 0.009 (2)	**0.967 ± 0.011 (1)**
Tecator	0.953 ± 0.021 (3)	**0.974 ± 0.019 (1)***	0.967 ± 0.024 (2)	0.952 ± 0.024 (4)	0.810 ± 0.048 (5)
Tecator (2nd derivative)	0.959 ± 0.021 (5)	0.978 ± 0.015 (3)	0.978 ± 0.017 (4)	0.980 ± 0.019 (2)	**0.990 ± 0.012 (1)***
Yoga	0.578 ± 0.016 (5)	0.752 ± 0.027 (3)	0.698 ± 0.015 (4)	0.916 ± 0.009 (2)	**0.932 ± 0.008 (1)***
Average accuracy	*0.861*	*0.891*	*0.860*	***0.903***	*0.893*
Average rank	*3.389*	*2.667*	*4.000*	*2.611*	***2.278***

ACKNOWLEDGMENT

The authors acknowledge financial support from the Spanish Ministry of Education and Innovation, projects PID2019-106827GB-I00 / AEI / 10.13039/501100011033 and PID2019-109387GB-I00, and from grant FPU18/00047. We acknowledge the computer resources provided by CCC-UAM.

REFERENCES

[1] A. Mincholé, J. Camps, A. Lyon, and B. Rodríguez, "Machine learning in the electrocardiogram," *Journal of Electrocardiology*, vol. 57, pp. S61–S64, 2019.

[2] B. Jafrasteh, N. Fathianpour, and A. Suárez, "Comparison of machine learning methods for copper ore grade estimation," *Computational Geosciences*, vol. 22, pp. 1371–1388, 2018.

[3] I. Ebert-Uphoff and Y. Deng, "Causal discovery from spatio-temporal data with applications to climate science," in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 606–613.

[4] A. Cuevas, "A partial overview of the theory of statistics with functional data," *Journal of Statistical Planning and Inference*, vol. 147, pp. 1–23, 2014.

[5] J.-L. Wang, J.-M. Chiou, and H.-G. Müller, "Functional data analysis," *Annual Review of Statistics and Its Application*, vol. 3, no. 1, pp. 257–295, 2016.

[6] A. Delaigle and P. Hall, "Achieving near perfect classification for functional data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 267–286, Mar. 2012.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.

[8] J. R. Berrendero, A. Cuevas, and J. L. Torrecilla, "On the Use of Reproducing Kernel Hilbert Spaces in Functional Classification," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1210–1218, Jul. 2018.

[9] A. Cuevas, M. Febrero, and R. Fraiman, "Robust estimation and classification for functional data via projection-based depth notions," *Computational Statistics*, vol. 22, no. 3, pp. 481–496, Sep. 2007.

[10] J. A. Cuesta-Albertos, M. Febrero-Bande, and M. Oviedo de la Fuente, "The DD^G-classifier in the functional setting," *TEST*, vol. 26, no. 1, pp. 119–142, Mar. 2017.

[11] A. Baillo, A. Cuevas, and R. Fraiman, "Classification methods for functional data," *The Oxford Handbook of Functional Data Analysis*, Nov. 2010.

[12] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, p. 5–32, oct 2001.

[13] F. Ferraty and P. Vieu, *Nonparametric Functional Data Analysis: Theory and Practice*, ser. Springer Series in Statistics. New York: Springer-Verlag, 2006.

[14] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, Nov. 2019.

[15] Bureau of Meteorology, Australia, "Long-term historical rainfall data for australia," Boulder CO, 1992. [Online]. Available: https://doi.org/10.5065/7V14-A428

[16] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273–3297, Dec. 1998.

[17] M. Febrero, P. Galeano, and W. González-Manteiga, "Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels," *Environmetrics*, vol. 19, no. 4, pp. 331–345, Jun. 2008.

[18] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, Jan. 2014.

[19] D. Díaz-Vico and C. Ramos-Carreño, "scikit-datasets: Scikit-learn-compatible datasets," Mar. 2022. [Online]. Available: https://github.com/daviddiazvico/scikit-datasets

[20] C. Ramos-Carreño, A. Suárez, J. L. Torrecilla, M. Carbajo Berrocal, and M. Manchón, "GAA-UAM/scikit-fda: Functional Data Analysis in Python," 10 2019. [Online]. Available: https://github.com/GAA-UAM/scikit-fda

[21] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, no. Jan, pp. 1–30, 2006.

[22] S. Herbold, "Autorank: A Python package for automated ranking of classifiers," *Journal of Open Source Software*, vol. 5, no. 48, p. 2173, Apr. 2020.

[23] J. R. Berrendero, B. Bueno-Larraz, and A. Cuevas, "On Mahalanobis Distance in Functional Settings," *Journal of Machine Learning Research*, vol. 21, no. 9, pp. 1–33, 2020.

[24] S. López-Pintado and J. Romo, "On the Concept of Depth for Functional Data," *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 718–734, Jun. 2009.