A New Model for Counterfactual Analysis for Functional Data

Carrizosa, Emilio; Ramírez-Ayerbe, Jasone; Romero Morales, Dolores

Document Version
Final published version

Published in:
Advances in Data Analysis and Classification

DOI:
10.1007/s11634-023-00563-5

Publication date:
2024

Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-023-00563-5

REGULAR ARTICLE

A new model for counterfactual analysis for functional data

Emilio Carrizosa[1] · **Jasone Ramírez-Ayerbe**[1] · **Dolores Romero Morales**[2]

Received: 13 March 2023 / Accepted: 25 September 2023

Abstract
Counterfactual explanations have become a very popular interpretability tool to understand and explain how complex machine learning models make decisions for individual instances. Most of the research on counterfactual explainability focuses on tabular and image data and much less on models dealing with functional data. In this paper, a counterfactual analysis for functional data is addressed, in which the goal is to identify the samples of the dataset from which the counterfactual explanation is made of, as well as how they are combined so that the individual instance and its counterfactual are as close as possible. Our methodology can be used with different distance measures for multivariate functional data and is applicable to any score-based classifier. We illustrate our methodology using two different real-world datasets, one univariate and another multivariate.

Keywords Counterfactual explanations · Mathematical optimization · Functional data · Prototypes · Random forests

Mathematical Subject Classification 90C90 · 62H30

1 Introduction

Machine learning models are increasingly being used for high stakes decision-making settings such as healthcare, law or finance. Many of these machine learning models are black-boxes and therefore they do not explain how they arrive to decisions in a way

✉ Jasone Ramírez-Ayerbe
mrayerbe@us.es

Emilio Carrizosa
ecarrizosa@us.es

Dolores Romero Morales
drm.eco@cbs.dk

[1] Instituto de Matemáticas de la Universidad de Sevilla, Seville, Spain

[2] Department of Economics, Copenhagen Business School, Frederiksberg, Denmark

Published online: 25 October 2023

that humans can understand. Nowadays, there is an increasing number of laws and regulations (Goodman and Flaxman 2017) coming into place to enforce the decisions of algorithms to be interpretable (a.k.a. transparent) (Du et al. 2019; Eiras-Franco et al. 2019; Fu et al. 2022; Miller 2019; Zhdanov et al. 2022). Interpretability is enhanced by selecting the features that have the greatest impact on the model as a whole (Benítez-Peña et al. 2021; Bertsimas et al. 2016; Zheng et al. 2021), but also knowing these locally for the decision made for each individual (Lundberg and Lee 2017; Lundberg et al. 2020; Ribeiro et al. 2016).

In this paper, we specifically address the problem of interpretability when data are functions. This type of data arises in important domains such as econometrics, energy, marketing (Jank and Shmueli 2006; Sood et al. 2009; Sunar and Swaminathan 2021). There is rather extensive literature on the use of machine learning to analyse functional data, e.g., adapting Support Vector Machine models to functional data (Blanquero et al. 2019; Chaovalitwongse et al. 2008), using regression trees to detect critical intervals (Blanquero et al. 2023) or novel forms of intepretability when dealing with functional data (Carrizosa et al. 2022; Martín-Barragán et al. 2014). See also Aneiros et al. (2022), Ramsay (2006) for an overview of methods for functional data analysis.

A specific type of interpretability tools is the counterfactual explanation (Carrizosa et al. 2023; Martens and Provost 2014; Wachter et al. 2017) where one seeks the minimum cost changes that can be made to an instance such that the given machine learning model would have classified it in a different class. For instance, in a credit score application one may be interested in knowing how the debt history of a person should have been to change the prediction to *loan should be granted*. See Guidotti (2022), Karimi et al. (2022), Verma et al. (2020) for recent surveys on Counterfactual Analysis.

Apart from the advantages mentioned above to finding counterfactuals for a given instance, in terms of guidance on how to change the predicted class to desired one, there are others to the stakeholder. First, it allows us to know how robust the prediction is, i.e., how much should the record be perturbed to make the classifier label it in a different class. Second, imposing some sort of sparsity in the process of building counterfactuals allow us to identify the most relevant features, i.e., those that, for this particular instance, are forcing the classifier to classify it in the desired class.

While the literature on machine learning to analyse functional data is extensive, this is not the case for counterfactual analysis. Most of the work on counterfactual explanations focuses on tabular, image data or text data (Karimi et al. 2022; Ramon et al. 2020; Tolkachev et al. 2022), and much less on functional data. In principle, one could apply the methods developed for tabular data also to functional data, just by considering that each feature is the measurement of the function at a time instant. However, doing so, fundamental information such as the autocorrelation structure along consecutive time instants would be lost. For this reason, some works on counterfactual explanations exploiting the functional nature of data have been suggested, e.g., Ates et al. (2021), Delaney et al. (2021), but, as far as the authors know, none of them uses the structure and properties of the machine learning model.Moreover, when working with functional data, other types of distance measures may appear, such as the Dynamic Time Warping distance (Xing et al. 2010), which, with our methodology, we can consider.

An instance $\boldsymbol{x} \in \mathcal{X} \subset \mathcal{F}^J$ is defined as a vector of J functional features. The counterfactual explanation $\boldsymbol{x} \in \mathcal{X}^0 \subset \mathcal{X}$ of the instance $\boldsymbol{x}^0$ is a hypothetical instance generated by combining existing instances in the dataset, heareafter *prototypes*, so that the cost $C(\boldsymbol{x}, \boldsymbol{x}^0)$ of perturbing the features in $\boldsymbol{x}^0$ to yield $\boldsymbol{x}$ is minimal. With this, we achieve certain interpretability goals. First, we can deal with multivariate functional data, i.e., our data are functions taking values in some $\mathbb{R}^J$. Second, we are able to identify the instances from the dataset that generate the counterfactual for each instance, controlling how sparse the counterfactual explanation is, in terms of both the number of prototypes used to create the counterfactual $\boldsymbol{x}$ and the number of functional features changed to move from $\boldsymbol{x}^0$ to $\boldsymbol{x}$. Third, we can model the cost function C by means of different distance measures, including popular distances in functional analysis such as the Dynamic Time Warping distance. We will show that, under mild assumptions, obtaining counterfactual explanations reduces to solving a Mixed Integer Convex Quadratic Model with linear constraints, which can be solved with standard optimization packages.

The remainder of the paper is organized as follows. In Sect. 2, we model the problem of finding counterfactual explanations when data are functions through an optimization problem. In Sect. 3, we focus on counterfactual analysis for additive tree models. In Sect. 4, a numerical illustration using real-world datasets is provided. Finally, conclusions and possible lines of future research are provided in Sect. 5.

2 Counterfactual analysis for functional data

In this section, we will detail the mathematical optimization formulation for generating counterfactual explanations when dealing with functional data. To do this we need to model the structure of the counterfactual instances, the constraints associated with them, as well as the cost function C. This will be done in what follows. We postpone to the next section the analysis of the case in which a state-of the-art score-based classifier, namely, an additive classification tree, is used as classifier.

Recall that an instance $\boldsymbol{x} \in \mathcal{X} \subset \mathcal{F}^J$ is defined as a vector of J functional features. Hence, $\boldsymbol{x} = (x_1(t), \ldots, x_J(t))$, where $x_j : [0, T] \to \mathbb{R}$, $j = 1, \ldots, J$, are Riemann integrable functions defined in interval $[0, T]$. Notice that $x_j(t)$ may be a static feature, e.g., birth date, defined then as a constant function. Note also that, for a given time instant $t \in [0, T]$, $\boldsymbol{x}(t)$ is a vector in $\mathbb{R}^J$ components, which may represent J measurements of independent attributes, or they may be related, e.g., one can include an attribute x_j, some of its derivatives to provide information also on e.g., the growth speed or the convexity of function x_j.

As mentioned in the introduction, the construction of counterfactual solutions depends on the (multiclass) classifier used. Assuming a score-based classifier, we are given a function $f : \mathcal{X} \subset \mathcal{F}^J \to \{1, \ldots, K\}$ based on score functions $(f_1, \ldots, f_K)$, where K is the number of classes. Given an instance $\boldsymbol{x}^0 \in \mathcal{X}$, let $f(\boldsymbol{x}^0) \in \arg\max_k f_k(\boldsymbol{x}^0)$ denote the class assigned by the classifier to $\boldsymbol{x}^0$. For a fixed class k^*, the counterfactual instance of $\boldsymbol{x}^0$ is defined in this paper as the feasible $\boldsymbol{x}$ obtained with a minimal cost of perturbation of $\boldsymbol{x}^0$ and classified by the score-based classifier in class k^*. This yields the following optimization problem:

$$\begin{cases} \min_{\boldsymbol{x}} & C(\boldsymbol{x}, \boldsymbol{x}^0) \\ \text{s.t.} & f_{k^*}(\boldsymbol{x}) \geq f_k(\boldsymbol{x}) \quad \forall k = 1, \ldots, K \quad k \neq k^* \\ & \boldsymbol{x} \in \mathcal{X}^0. \end{cases} \tag{1}$$

The objective function $C(\boldsymbol{x}, \boldsymbol{x}^0)$ is a cost function that measures the dissimilarity between the given instance $\boldsymbol{x}^0$ and the counterfactual instance $\boldsymbol{x}$. In the feasible region, we have two types of constraints. In the first one, we ensure that the counterfactual $\boldsymbol{x}$ is classified in class k^* by imposing that the score $f_{k^*}(\boldsymbol{x})$ is the maximum across all k. In the second type of constraint, we ensure that the counterfactual is in $\mathcal{X}^0 \subset \mathcal{F}^J$, the set defined through the actionability and plausibility constraints (Mohammadi et al. 2021; Wachter et al. 2017), i.e., constraints ensuring that a counterfactual does not change immovable features, and that guarantee that counterfactual explanations are realistic.

2.1 Counterfactual instances and constraints

Let us discuss the constraints on $\boldsymbol{x}$ in Problem (1). First, we need to ensure that the counterfactual explanation $\boldsymbol{x}$ is realistic. In the case of functional data, this yields an infinite-dimensional optimization problem. To enhance the tractability of this requirement, we propose the use of instances of the dataset, i.e., prototypes, to generate the counterfactual explanation. Let $\boldsymbol{x}^b$, $b = 1, \ldots, B$, be all the instances that have been classified by the model in class k^* and are close enough to $\boldsymbol{x}^0$ so that they can be seen as references for $\boldsymbol{x}^0$. For an instance $\boldsymbol{x}^0 = (x_1^0, \ldots, x_J^0)$, feature j of the counterfactual explanation x_j is defined as the convex combination of the original feature x_j^0 and the feature j of all B prototypes x_j^b. Thus, the counterfactual explanation $\boldsymbol{x}$ is defined for each feature j as $x_j = \alpha_{0j} x^0 + \sum_{b=1}^{B} \alpha_j^b x_j^b$, where $\sum_{b=0}^{B} \alpha_j^b = 1, \quad \forall j = 1, \ldots, J$.

In order to gain interpretability of the so obtained counterfactual explanation $\boldsymbol{x}$, we want to use as few prototypes $\boldsymbol{x}^b$ as possible in the construction of $\boldsymbol{x}$. For this reason we will impose a maximum of $B^{\max}$ prototypes to be used, where $B^{\max}$ is a parameter defined by the user.

In $\mathcal{X}^0$ we may also impose the unmovable constraints or other constraints like upper or lower limits on the static variables.

2.2 Cost function

Recall that $C(\boldsymbol{x}, \boldsymbol{x}^0)$ is the cost of changing $\boldsymbol{x}^0$ to $\boldsymbol{x}$, which can be measured by the proximity between the curves defining $\boldsymbol{x}$ and $\boldsymbol{x}^0$.

The proximity between curves can be measured in several ways. One can use for instance the squared Euclidean distance:

$$\|\boldsymbol{x} - \boldsymbol{x}^0\|_2^2 = \int_0^T \sum_{j=1}^{J} (x_j(t) - x_j^0(t))^2 dt \tag{2}$$

Needless to say, different weights can be assigned to each feature in the expression above.

Another popular distance used in the literature (Esling and Agon 2012; Xing et al. 2010) is the Dynamic Time Warping (DTW) distance, which measures the dissimilarity between two functions that may be inspected at different speed, see Fig. 1. More explicitly, suppose we have $\boldsymbol{x}$ and $\boldsymbol{x}'$, discretised in two sequences of length n, so that the J-variate functions $\boldsymbol{x}$ and $\boldsymbol{x}'$ are replaced by $(\boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_n))$ and $(\boldsymbol{x}'(t_1), \ldots, \boldsymbol{x}'(t_n))$. Observe that each $\boldsymbol{x}(t), \boldsymbol{x}'(t)$ are vectors in $\mathbb{R}^J$. A warping path π is a chain of pairs of the form $\pi = (q_{11}, q_{21}) \rightarrow (q_{12}, q_{22}) \rightarrow \ldots \rightarrow (q_{1Q}, q_{2Q})$ of length Q, $n \leq Q \leq 2n-1$, satisfying the following two conditions:

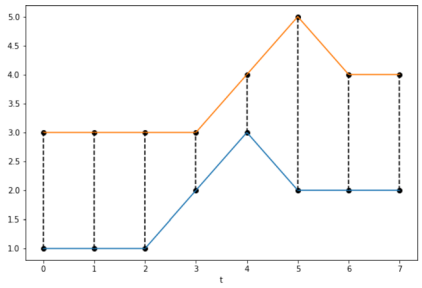
1. $(q_{11}, q_{21}) = (t_1, t_1)$, and $(q_{1Q}, q_{2Q}) = (t_n, t_n)$
2. $q_{1r} \leq q_{1(r+1)} \leq q_{1r} + 1$, and $q_{2r} \leq q_{2(r+1)} \leq q_{2r} + 1$, $r = 1, 2, \ldots, Q-1$

Let $\mathcal{W}$ denote the set of all warping paths. Then, the DTW distance $\mathrm{DTW}(\boldsymbol{x}, \boldsymbol{x}')$ between $\boldsymbol{x}$ and $\boldsymbol{x}'$ is the minimal squared Euclidean distance between pairs of the form $(\boldsymbol{x}(q_{11}), \ldots, \boldsymbol{x}(q_{1Q}))$ and $(\boldsymbol{x}'(q_{21}), \ldots, \boldsymbol{x}'(q_{2Q}))$ when $(q_{11}, q_{21}) \rightarrow (q_{12}, q_{22}) \rightarrow \ldots \rightarrow (q_{1Q}, q_{2Q})$ is a warping path, i.e.,

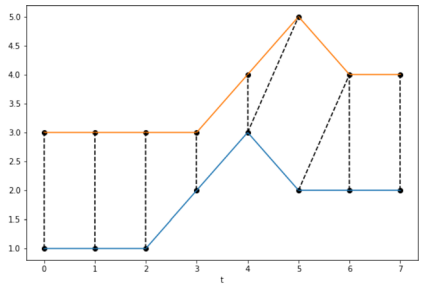
$$\begin{aligned} \mathrm{DTW}(\boldsymbol{x}, \boldsymbol{x}') = \min \quad & \sum_{r=1}^{Q} \sum_{j=1}^{J} \Big(x_j(q_{1r}) - x'_j(q_{2r}) \Big)^2 \\ \text{s.t.} \quad & \pi = (q_{11}, q_{21}) \rightarrow (q_{12}, q_{22}) \rightarrow \ldots \rightarrow (q_{1Q}, q_{2Q}) \in \mathcal{W} \end{aligned} \tag{3}$$

Observe that *DTW* can be efficiently evaluated using dynamic programming (Berndt and Clifford 1994).

Additionally, C may contain, on top of the distance-based term described above, other terms measuring, e.g., the number of features altered when moving from $\boldsymbol{x}^0$ to $\boldsymbol{x}$. In particular, in Sect. 3 we will discuss in detail the cases



(a) Warping path $\pi = (0,0) \rightarrow (1,1) \rightarrow (2,2) \rightarrow (3,3) \rightarrow (4,4) \rightarrow (5,5) \rightarrow (6,6) \rightarrow (7,7)$



(b) Optimal warping path $\pi^* = (0,0) \rightarrow (1,1) \rightarrow (2,2) \rightarrow (3,3) \rightarrow (4,4) \rightarrow (4,5) \rightarrow (5,6) \rightarrow (6,6) \rightarrow (7,7)$

Fig. 1 Comparison between different warping paths between the functions $\boldsymbol{x}$ (in blue) and $\boldsymbol{x}'$ (in orange)

$$C(\boldsymbol{x}, \boldsymbol{x}^0) = \lambda_0 \|\boldsymbol{x}^0 - \boldsymbol{x}\|_0 + \lambda_2 \int_0^T \sum_{j=1}^{J} (x_j(t) - x_j^0(t))^2 dt, \tag{4}$$

and

$$C(\boldsymbol{x}, \boldsymbol{x}^0) = \lambda_0 \|\boldsymbol{x}^0 - \boldsymbol{x}\|_0 + \lambda_2 \mathrm{DTW}(\boldsymbol{x}, \boldsymbol{x}^0), \tag{5}$$

where $\|\boldsymbol{x}^0 - \boldsymbol{x}\|_0$ indicates how many components of $\boldsymbol{x}^0 = (x_1^0, \ldots, x_J^0)$ and $\boldsymbol{x} = (x_1, \ldots, x_J)$ are not equal,

$$\|\boldsymbol{x}^0 - \boldsymbol{x}\|_0 = |\{j : x_j^0 \neq x_j\}|, \tag{6}$$

and $\lambda_0, \lambda_2 \geq 0$, not simultaneously 0.

3 Additive tree models

Problem (1) under the modelling assumptions in Sect. 2 can be addressed for several score-based classifiers. These include, among others, additive tree models (ATM) such as Random Forest (Breiman 2001) or XGBoost models (Chen and Guestrin 2016), as well as linear models such as logistic regression and linear support vector machines. Below, we focus on ATM, and extend to functional data the analysis for tabular data described in (Carrizosa et al. 2021).

The ATM is composed of T classification trees. Each tree t has a series of branching nodes s, each having associated a feature $v(s)$, a time instant t_s, and a threshold value c_s, so that records $\boldsymbol{x}$ go through the left or the right of the branching node depending on whether $x_{v(s)}(t_s) \leq c_s$ or not. Moreover, the tree t has associated a weight $w^t \geq 0$, so that the class predicted for an instance $\boldsymbol{x}$ is the most voted class according to the weights w^t. The ATM can be viewed as a score-based classifier by associating to class k the score f_k defined as:

$$f_k(\boldsymbol{x}) = \sum_{t \in \{1,\ldots,T\}/t \in \mathcal{T}_k(\boldsymbol{x})} w^t, \tag{7}$$

where $\mathcal{T}_k(\boldsymbol{x})$ denotes the subset of trees that classify $\boldsymbol{x}$ in class k.

To model Problem (1) for functional data and additive tree models, the parameters and decision variables in Fig. 2 will be used.

Recall that the ATM is already known, i.e., the whole structure, including the topology of the trees and the feature and threshold used in each split, is given. Thus, in order to compute the score of the counterfactual instance, the only requirement is to know in which leaf node it has ended up. When we end up in a specific leaf, the corresponding branching conditions are activated. For each split $s \in \mathrm{Left}(l, t)$ if the condition is true, then $x_{v(s)}(t_s) \leq c_s$, otherwise $x_{v(s)}(t_s) > c_s$. To introduce these

Parameters	
$\boldsymbol{x}^0$	the instance for which a minimum cost counterfactual $\boldsymbol{x}$ is sought
$\mathcal{L}_k^t$	subset of leaves in tree t whose output is class $k = 1, \ldots, K$, $t = 1, \ldots, T$
$\mathcal{L}^t$	set of leaves in tree t, with $\mathcal{L}^t = \cup_k \mathcal{L}_k^t$, $t = 1, \ldots, T$
$\mathcal{T}_k(\boldsymbol{x})$	subset of trees that classify $\boldsymbol{x}$ in class k
$\text{Left}(l, t)$	the set of ancestor nodes of leaf l in tree t whose left branch takes part in the path that ends in l, $l \in \mathcal{L}_t$, $t = 1, \ldots, T$
$\text{Right}(l, t)$	the set of ancestor nodes of leaf l in tree t whose right branch takes part in the path that ends in l, $l \in \mathcal{L}_t$, $t = 1, \ldots, T$
$v(s)$	feature used in split s, $s \in \text{Left}(l, t) \cup \text{Right}(l, t)$
t_s	time point used in split s, $s \in \text{Left}(l, t) \cup \text{Right}(l, t)$
c_s	threshold value used for split s, $s \in \text{Left}(l, t) \cup \text{Right}(l, t)$
w^t	weight of tree t, $t = 1, \ldots, T$
$\boldsymbol{x}^b$	instances of the dataset that have been classified in class k^*, $b = 1, \ldots, B$
$B^{\max}$	maximum number of prototypes allowed to construct the counterfactual explanation $\boldsymbol{x}$
M_1, M_2	big-M constants
Decision variables	
$\boldsymbol{x}$	counterfactual, $\boldsymbol{x} \in \mathcal{X}^0$
z_l^t	binary decision variable that indicates whether the counterfactual instance ends in leaf $l \in \mathcal{L}_t$ ($z_l^t = 1$) or not ($z_l^t = 0$), $t = 1, \ldots, T$
α_j^b	coefficient associated to the prototype $\boldsymbol{x}^b$ of the convex combination to construct feature j x_j of the counterfactual explanation, $b = 0, \ldots, B$, $j = 1, \ldots, J$
u_b	binary decision variable that indicates whether prototype $\boldsymbol{x}^b$ is used in the construction of the counterfactual explanation $\boldsymbol{x}$, $b = 0, \ldots, B$

Fig. 2 Parameters and decision variables used to model Problem (1) for additive tree models, when data are functions

logical conditions, we use the following big M constraints:

$$x_{v(s)}(t_s) - M_1(1 - z_l^t) + \epsilon \leq c_s \quad s \in \text{Left}(l, t) \tag{8}$$

$$x_{v(s)}(t_s) + M_2(1 - z_l^t) - \epsilon \geq c_s \quad s \in \text{Right}(l, t). \tag{9}$$

Due to the impossibility of the Mixed-Integer Optimization solvers to model a strict inequality, a small positive quantity ϵ is introduced in Eqs. (8) and (9), as is done in Bertsimas and Dunn (2017). With this, our counterfactual variable $x_{v(s)}$ at point t_s is not allowed to take values around the threshold value in c_s at the split s. Please note that the value of M_1 and M_2 can be tightened for each split.

The score function in (7) can be rewritten as a linear expression as follows:

$$\sum_{t=1}^{T} \sum_{l \in \mathcal{L}_k^t} w^t z_l^t,$$

for $k = 1, \ldots, K$.

Recall that one type of sparsity that we wanted was to use few prototypes to build our counterfactual explanation. To model this, we introduce binary decision variables u_b, which control the number of prototypes that can be used in the convex combination yielding $\boldsymbol{x}$ through parameter $B^{\max}$.

Given instance $\boldsymbol{x}^0$ and a cost function C, the formulation associated with Problem (1), the problem of finding the minimal cost perturbation that causes the classifier to

classify it in class k^* is as follows:

$$\min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{\alpha},\boldsymbol{u}} \quad C(\boldsymbol{x}, \boldsymbol{x}^0) \tag{10}$$

$$\text{s.t.} \quad x_{v(s)}(t_s) - M_1(1 - z_l^t) + \epsilon \leq c_s \quad \forall s \in \text{Left}(l,t) \quad \forall l \in \mathcal{L}^t \quad \forall t = 1, \ldots, T \tag{11}$$

$$x_{v(s)}(t_s) + M_2(1 - z_l^t) - \epsilon \geq c_s \quad \forall s \in \text{Right}(l,t) \quad \forall l \in \mathcal{L}^t \quad \forall t = 1, \ldots, T \tag{12}$$

$$\sum_{l \in \mathcal{L}^t} z_l^t = 1 \quad \forall t = 1, \ldots, T \tag{13}$$

$$\sum_{t=1}^{T} \sum_{l \in \mathcal{L}_{k^*}^t} w^t z_l^t \geq \sum_{t=1}^{T} \sum_{l \in \mathcal{L}_k^t} w^t z_l^t \quad \forall k = 1, \ldots, K \quad k \neq k^* \tag{14}$$

$$x_j = \alpha_{0j} x_j^0 + \sum_{b=1}^{B} \alpha_j^b x_j^b \quad \forall j = 1, \ldots, J \tag{15}$$

$$\sum_{b=0}^{B} \alpha_j^b = 1 \quad \forall j = 1, \ldots, J \tag{16}$$

$$\alpha_j^b \leq u_b \quad \forall b = 1, \ldots, B \quad \forall j = 1, \ldots, J \tag{17}$$

$$\sum_{b=1}^{B} u_b \leq B^{\max} \tag{18}$$

$$u_b \in \{0, 1\} \quad \forall b = 1, \ldots, B \tag{19}$$

$$z_l^t \in \{0, 1\} \quad \forall l \in \mathcal{L}^t \quad \forall t = 1, \ldots, T \tag{20}$$

$$\alpha_j^b \geq 0 \quad \forall b = 1, \ldots, B \quad \forall j = 1, \ldots, J \tag{21}$$

$$\boldsymbol{x} \in \mathcal{X}^0. \tag{22}$$

The cost function in (10) is discussed in more detail below, where we measure the movement from the original instance $\boldsymbol{x}^0$ to its counterfactual explanation $\boldsymbol{x}$ for functional data. Constraints (11) and (12) control to which leaf the counterfactual instance is assigned and constraint (13) enforces that only one leaf is active for each tree. Constraint (14) ensures that the counterfactual instance is assigned to class k^*, i.e., the score of class k^* is the highest one among all classes. Constraints (15) and (16) define for each feature j the counterfactual instance as the convex combination of x_j^0 and the prototypes x_j^b. To ensure sparsity in the prototypes, constraints (17)–(18) restrict the number of prototypes used in the convex combination to $B^{\max}$. Constraints

Constraints	Meaning
$x_{v(s)}(t_s) - M_1(1 - z_l^t) + \epsilon \leq c_s$ $\forall s \in \text{Left}(l,t) \quad \forall l \in \mathcal{L}^t \quad \forall t = 1, \ldots, T$	in which leaf ends the counterfactual
$x_{v(s)}(t_s) + M_2(1 - z_l^t) - \epsilon \geq c_s$ $\forall s \in \text{Right}(l,t) \quad \forall l \in \mathcal{L}^t \quad \forall t = 1, \ldots, T$	in which leaf ends the counterfactual
$\sum_{l \in \mathcal{L}^t} z_l^t = 1 \ \forall t = 1, \ldots, T$	only one leaf is active
$\sum_{t=1}^{T} \sum_{l \in \mathcal{L}_{k^*}^t} w^t z_l^t \geq \sum_{t=1}^{T} \sum_{l \in \mathcal{L}_k^t} w^t z_l^t$ $\forall k = 1, \ldots, K \quad k \neq k^*$	the counterfactual is assigned to class k^*
$x_j = \alpha_{0j} x_j^0 + \sum_{b=1}^{B} \alpha_j^b x_j^b \quad \forall j = 1, \ldots, J$	counterfactual structure
$\sum_{b=0}^{B} \alpha_j^b = 1 \quad \forall j = 1, \ldots, J$	coefficients of the convex combination
$\alpha_j^b \leq u_b \quad \forall b = 1, \ldots, B \quad \forall j = 1, \ldots, J$	sparsity in the prototypes
$\sum_{b=1}^{B} u_b \leq B^{\max}$	sparsity in the prototypes
$u_b \in \{0,1\} \quad \forall b = 1, \ldots, B$	binary variables
$z_l^t \in \{0,1\} \quad \forall l \in \mathcal{L}^t \quad \forall t = 1, \ldots, T$	binary variables
$\alpha_j^b \geq 0 \quad \forall b = 1, \ldots, B \quad \forall j = 1, \ldots, J$	nonnegative coefficients
$\boldsymbol{x} \in \mathcal{X}^0$	feasibility constraints

Fig. 3 Description of constraints (11)–(22), used to model Problem (1) for additive tree models, when data are functions

(19) and (20) ensure that all u_b and z_l^t are binary, constraint (21) that the coefficients α_j^b are nonnegative and constraint (22) that the counterfactual $\boldsymbol{x}$ is in $\mathcal{X}^0$, the set containing the rest of the actionability and plausibility constraints. An overview of all the constraints is detailed in Fig. 3.

Let us now discuss the objective function in (10) for the particular choices of C introduced in Sect. 2, namely, (4) and (5). In order to model the ℓ_0 term defined in (6), binary decision variables ξ_j are introduced. For every feature $j = 1, \ldots, J$, $\xi_j = 0$ if and only if $\alpha_{0j} = 1$, i.e., if $x_j = x_j^0$. This is expressed as

$$-\xi_j \leq 1 - \alpha_{0j} \leq \xi_j \quad j = 1, \ldots, J \tag{23}$$

$$\xi_j \in \{0,1\}, \quad j = 1, \ldots, J. \tag{24}$$

Moreover, we have that

$$\|\boldsymbol{x}^0 - \boldsymbol{x}\|_0 = \sum_{j=1}^{J} \xi_j.$$

Thus, for the cost function C in (4), we have the following reformulation of (10)–(22):

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{\alpha},\boldsymbol{u},\boldsymbol{\xi}} \quad & \lambda_0 \sum_{j=1}^{J} \xi_j + \lambda_2 \int_0^T \sum_{j=1}^{J} (x_j(t) - x_j^0(t))^2 dt \\ \text{s.t.} \quad & (11) - (22), (23) - (24). \end{aligned} \tag{CEF}$$

For the particular case of (CEF) where only the ℓ_0 distance is considered, i.e., $\lambda_2 = 0$, the objective function as well as the constraints are linear, (assuming $\mathcal{X}^0$ is also defined through linear constraints), while we have both binary and continuous decision variables. Therefore, Problem (CEF) can be solved using a Mixed Integer Linear Programming (MILP) solver. For arbitrary $\lambda_2 \geq 0$, taking into account that, by (15),

$$(x_j(t) - x_j^0(t))^2 = \left(\sum_{b=0}^{B} \alpha_j^b x_j^b(t) - x_j^0(t)\right)^2,$$

the second term in the objective can be expressed as a convex quadratic function in the decision variables α_j^b, and thus (again, assuming $\mathcal{X}^0$ is also defined through linear constraints) Problem (CEF) is a Mixed Integer Convex Quadratic Model with linear constraints.

Let us address Problem (1) when the cost function C has the form (5), and thus the DTW distance is involved. As in Sect. 2, the time interval $[0, T]$ is discretised in time instants $t_1, \ldots, t_n$, and thus the DTW distance is the minimal squared Euclidean distance among the warping paths W, yielding

$$\begin{aligned} \min_{\boldsymbol{x},z,\boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{u}} \quad & \lambda_0 \sum_{j=1}^{J} \xi_j + \lambda_2 \sum_{r=1}^{Q} \sum_{j=1}^{J} \Big(x_j(q_{1r}) - x_j^0(q_{2r})\Big)^2 \\ \text{s.t.} \quad & (11) - (22), (23) - (24) \\ & (q_{11}, q_{21}) \rightarrow (q_{12}, q_{22}) \rightarrow \ldots \rightarrow (q_{1Q}, q_{2Q}) \in \mathcal{W} \end{aligned} \qquad \text{(CEFDTW)}$$

Notice how for a fixed warping path in $\mathcal{W}$, constraints (11)–(24) are all linear, while we have both binary and continuous variables. Hence, if $\mathcal{X}^0$ is again defined by linear constraints, since the objective function is quadratic, Problem (CEFDTW) is a Mixed Integer Convex Quadratic Model with linear constraints, that can be solved using standard optimization packages. For this reason we propose an alternating heuristic to solve Problem (CEFDTW):

1: **Initialisation:** *Let* $\pi(0)$ *be the warping path* $(t_1, t_1) \rightarrow (t_2, t_2) \ldots \rightarrow (t_n, t_n)$
2: $r = 0$
3: Find the optimal warping path π^* and its corresponding δ^* by solving

$$\begin{aligned} \delta^* = \min\ & \sum_{r=1}^{Q} \sum_{j=1}^{J} \left(x_j(q_{1r}) - x_j^0(q_{2r})\right)^2 \\ \text{s.t.}\ & (q_{11}, q_{21}) \rightarrow (q_{12}, q_{22}) \rightarrow \ldots \rightarrow (q_{1Q}, q_{2Q}) \in \mathcal{W}, \end{aligned}$$

4: **if** $\delta^* = \delta(r)$ **then**
5: **stop**
6: **else**
7: update $\pi(r+1) = \pi^*$, $r = r+1$ and go to *Step 3*
8: **end if**
9: **Output**: counterfactual instance $\boldsymbol{x}$

Algorithm to calculate counterfactual explanations with the DTW-based cost function (5)

4 Numerical illustration

We will illustrate our methodology in two real-world datasets with functional data, one univariate and another multivariate, from the UCR archive (Dau et al. 2019). For a given instance, we are able to identify the individuals of the dataset from which the corresponding counterfactual is made up and what their contribution is. Furthermore, we show the two different sparsities that we can obtain with our model, namely, the number of prototypes used for the counterfactual and the number of functional features that change. The use of different distances, i.e., the Euclidean and the DTW distances, is also displayed.

All the mathematical optimization problems have been implemented using Pyomo optimization modeling language (Hart et al. 2017, 2011) in Python 3.8. As solver, we have used Gurobi 9.0 (Gurobi Optimization 2021). A value of $\epsilon = 1\text{e}{-6}$ has been imposed in (11) and (12). The values of the big-M in (11) and (12) are node dependent, and they have been tightened following the process described in Carrizosa et al. (2021). For all the computational experiments, the classification model considered has been a Random Forest with $T = 200$ trees and a maximum depth of 4. Our experiments have been conducted on a PC, with an Intel R CoreTM i7-1065G7 CPU @ 1.30GHz 1.50 GHz processor and 16 gigabytes RAM. The operating system is 64 bits.

The first dataset, *ItalyPowerDemand* (Keogh et al. 2006), has one functional feature. There are 1096 instances and each instance is a time series of length 24, representing the power demand in Italy in six months. The binary classification task is to distinguish days from October to March (response value -1) from April to September (response value $+1$).

The second dataset, *NATOPS* (Ghouaiel et al. 2017), has 24 functional time series of length 51 representing the X, Y, and Z coordinates of the left and right hand, wrist, thumb and elbows as captured by a Kinect 2 sensor. There are 260 instances and we

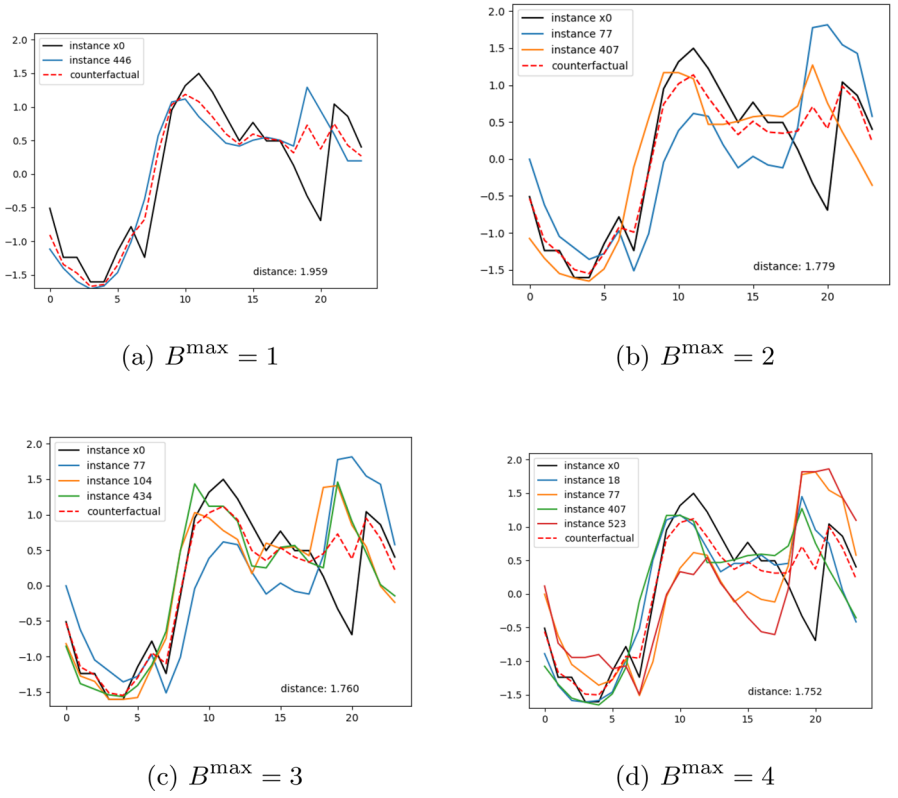
chose two classes of the 6 that there are in the dataset. The binary classification task is thus to distinguish the gesture “All Clear" (response value -1) from “Not Clear" (response value $+1$).

4.1 Experimental results

4.1.1 ItalyPowerDemand

We present the counterfactual for an instance $\boldsymbol{x}^0$ of the dataset *ItalyPowerDemand* in Fig. 4. In each case, we represent the original curve, the prototypes, and the final counterfactual.

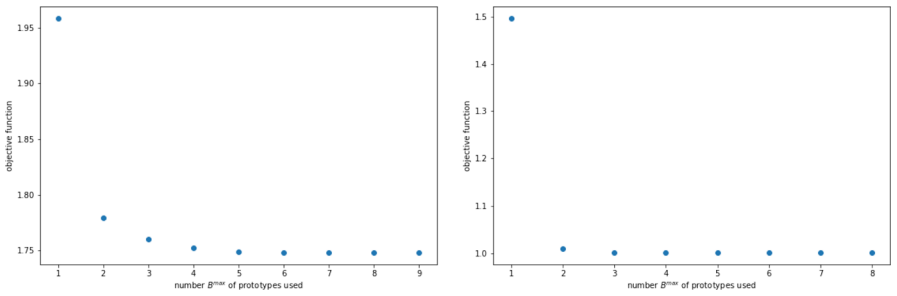
The first cost model analysed is the squared Euclidean model (4) with $\lambda_0 = 0$ (since we have only one feature, $\lambda_0 > 0$ is meaningless). Different values of $B^{\max}$ have been used. The smaller the value of $B^{\max}$ is, the more sparse the counterfactual is in terms of prototypes, while the larger the value of $B^{\max}$ is, the higher the freedom



(a) $B^{\max} = 1$ (b) $B^{\max} = 2$

(c) $B^{\max} = 3$ (d) $B^{\max} = 4$

Fig. 4 Counterfactual explanations for $\boldsymbol{x}^0$ of the ItalyPowerDemand data set which has been predicted by the Random Forest in $k^0 = -1$ and whose counterfactual $\boldsymbol{x}$ has to be predicted in class $k^* = +1$. Different values of $B^{\max}$, i.e., the number of prototypes used for the convex combination, have been imposed. The cost function is model (4) with $\lambda_0 = 0, \lambda_2 = 1$



(a) Euclidean distance (b) DTW distance

Fig. 5 Distance obtained versus number of prototypes used in a counterfactual explanation $\boldsymbol{x}$ for $\boldsymbol{x}^0$ of the ItalyPowerDemand data set which has been predicted by the Random Forest in $k^0 = -1$ and it is imposed $k^* = +1$

to use prototypes and therefore the smaller the distances obtained. In Fig. 5a we plot the relation between the number of prototypes and the distance. It is illustrated how using more than one prototype may be beneficial, but using more than 4 prototypes gives us less sparsity without smaller distances.

To show the flexibility of our model, the same experiments have been carried out but changing the cost based on DTW distances (5), again with $\lambda_0 = 0$. The counterfactual solutions have been calculated with the heuristic procedure described in Algorithm 1. The results are depicted in Fig. 6. As before, one can see how the objective function decreases as the number of prototypes $B^{\max}$ increases. However, in this case, it is sufficient to use 2 prototypes, as 3 or more will not improve much the objective function, see Fig. 5b.

4.1.2 NATOPS

We present now the counterfactual for an instance $\boldsymbol{x}^0$ of the multivariate dataset *NATOPS*. The cost function used has been of the form is the squared Euclidean model (4) with $\lambda_0 = 1$, $\lambda_2 = 0.005$. As we are interested in detecting the critical functional features to flip the classifier's decision, we give more weight in the cost function to the ℓ_0 as an illustration.

In Fig. 7 the counterfactual instance $\boldsymbol{x}$ for $\boldsymbol{x}^0$ for $B^{\max} = 1$ is shown. As the cost function C contains as its first term the ℓ_0 norm, we obtain a sparse solution in the sense of the features we need to change to move from $\boldsymbol{x}^0$ to $\boldsymbol{x}$. Indeed, to change its class, only three functional features have to be modified. In Fig. 8 the changed features are presented.

As in the univariate case, we can impose different values of $B^{\max}$. In Fig. 9 we show the counterfactual explanation for $B^{\max} = 2$ and for the same cost function. Note how giving the flexibility to use more than one prototype, results in only having to change two features, see Fig. 10.

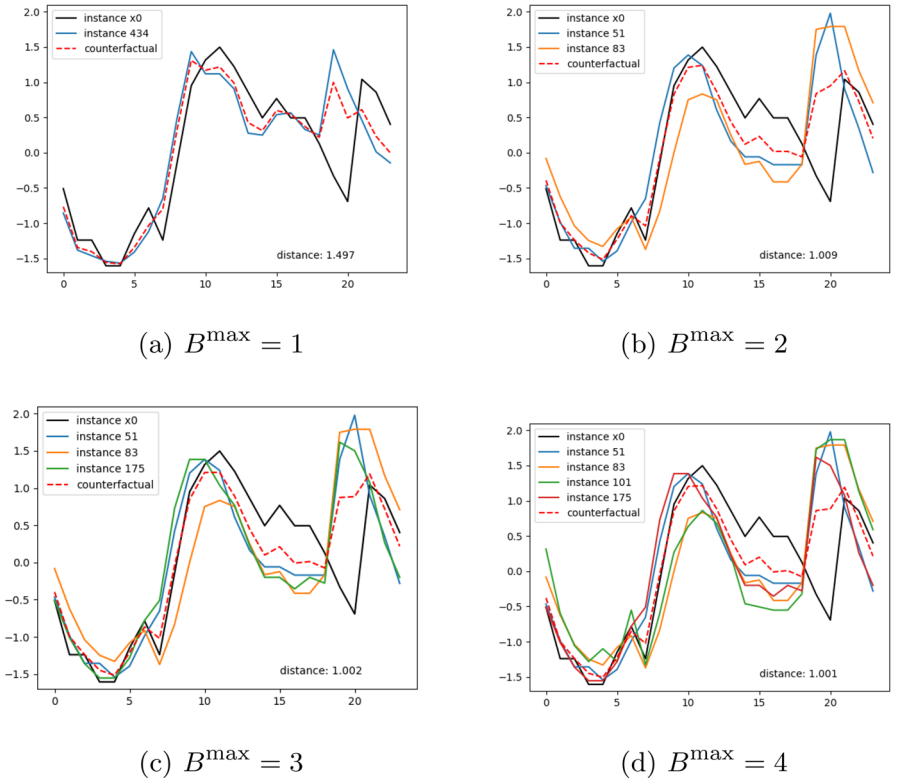


Fig. 6 Counterfactual explanations for $\boldsymbol{x}^0$ of the ItalyPowerDemand data set which has been predicted with a Random Forest in $k^0 = -1$ and whose counterfactual $\boldsymbol{x}$ has to be predicted in class $k^* = +1$. Different values of $B^{\max}$, i.e., the number of prototypes used for the convex combination, have been imposed. The cost function is model (5) with $\lambda_0 = 0, \lambda_2 = 1$

5 Conclusions

In this paper, we have proposed a novel approach to build counterfactual explanations when dealing with multivariate functional data in classification problems by means of mathematical optimization. With our method, we ensure plausible and sparse explanations, controlling not only the number of prototypes of the dataset used to create the counterfactuals, but also the number of features that need to be changed. Our model is also flexible enough to be used with different distance measures, e.g., the Euclidean distance or the DTW distance. Moreover, our methodology is applicable to score-

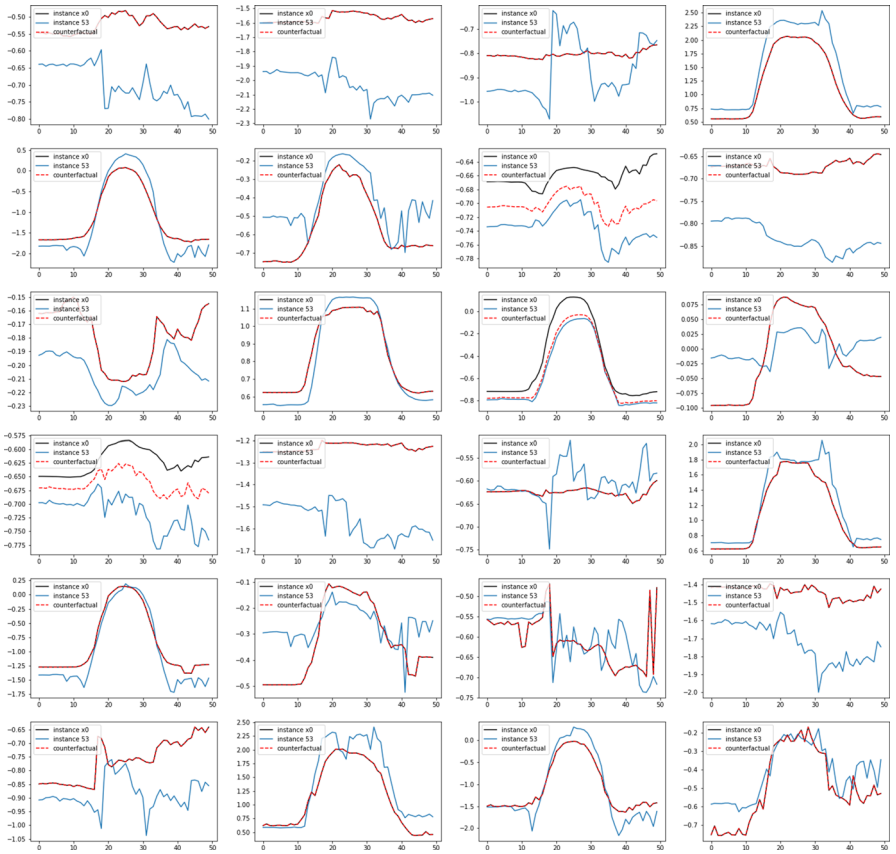
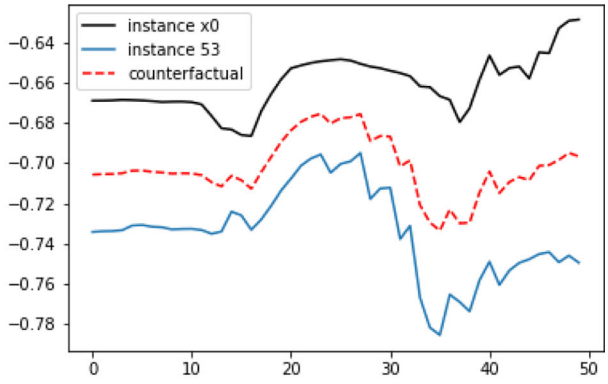


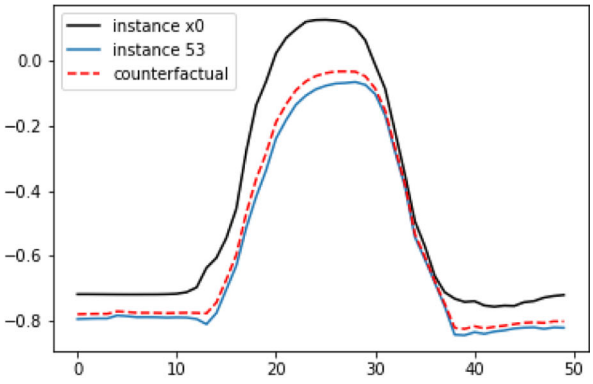
Fig. 7 Counterfactual explanations for $\boldsymbol{x}^0$ of the NATOPS data set which has been predicted by the Random Forest in $k^0 = +1$ and whose counterfactual $\boldsymbol{x}$ has to be predicted in class $k^* = -1$. $B^{\max} = 1$ prototype has been imposed. The cost function is model (4) with $\lambda_0 = 1$, $\lambda_2 = 0.005$

based classifiers, including additive tree models, such as random forest or XGBoost models, as well as linear models, such as logistic regression and linear support vector machines. We have illustrated our methodology on various real-world datasets.

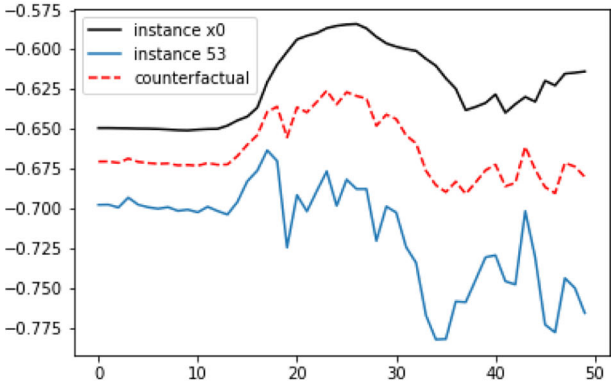
There are several interesting lines of future research. First, an extension to other non score-based classifiers, like k-NN classifiers, deserve some study. Secondly, to define counterfactual explanations for functional data one could be interested in keeping fixed a part of the curves defining the features. With our method we build the counterfactuals from scratch using the combinations of prototypes in the interval $[0, T]$, but suppose



(a) Feature 7



(b) Feature 11



(c) Feature 13

Fig. 8 Changed features in the counterfactual explanation for $\boldsymbol{x}^0$ of the NATOPS data set which has been predicted by the Random Forest in $k^0 = -1$ and whose counterfactual $\boldsymbol{x}$ has to be predicted in class $k^* = +1$ with $B^{\max} = 1$. The cost function is model (4) with $\lambda_0 = 1, \lambda_2 = 0.005$

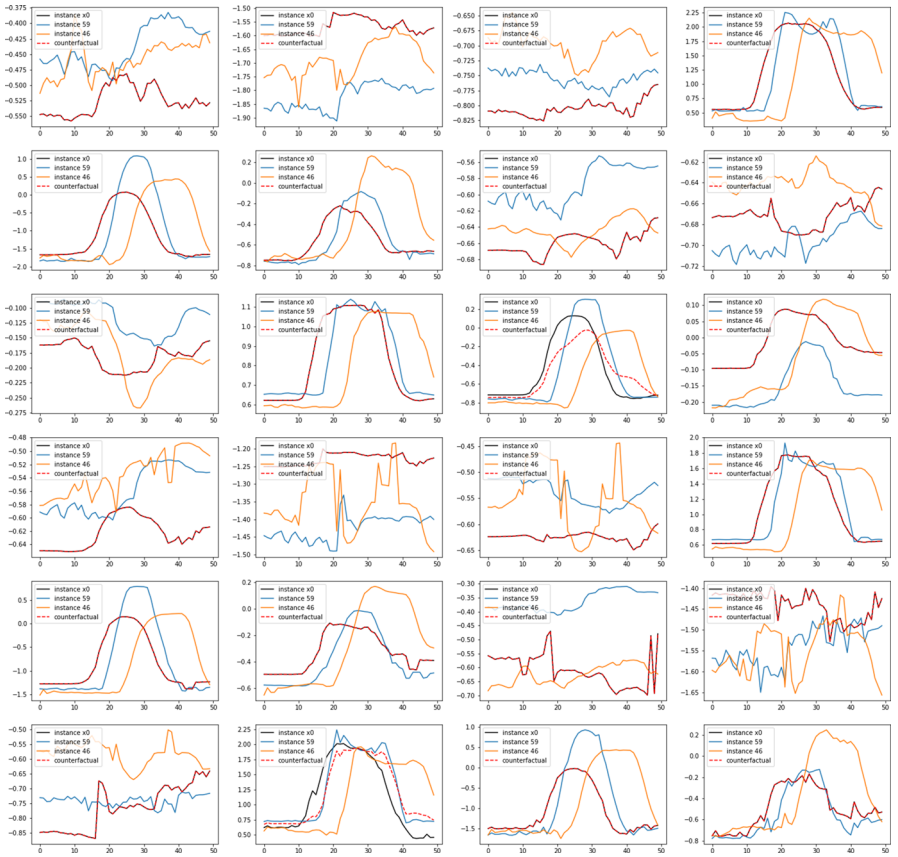
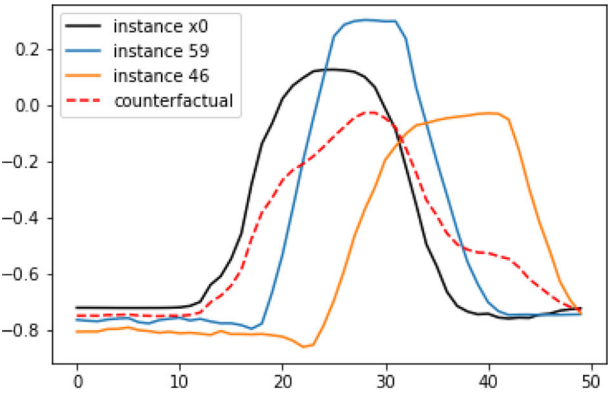
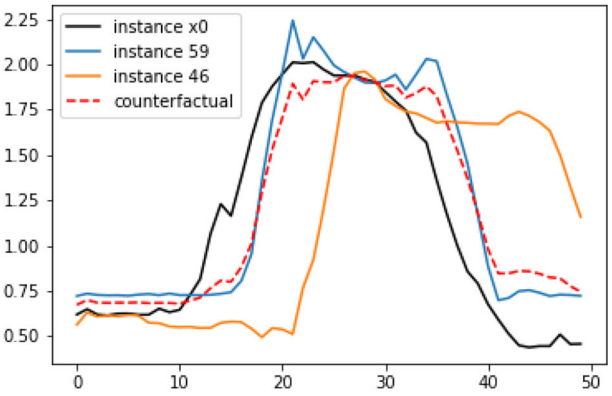


Fig. 9 Counterfactual explanations for $\boldsymbol{x}^0$ of the NATOPS data set which has been predicted by the Random Forest in $k^0 = +1$ and whose counterfactual $\boldsymbol{x}$ has to be predicted in class $k^* = -1$. $B^{\max} = 2$ prototypes has been imposed. Cost function: model (4) with $\lambda_0 = 1, \lambda_2 = 0.005$

we have an instance defined in the interval $[0, t_0)$, and one might want to find out how the rest of the curve in the interval $[t_0, T]$ would have to be like to make the overall curve being classified in class k^*. When constructing the rest of the curve, one would need to maintain the smoothness and other properties of the curve. Finally, the case in which other distance, such as the optimal transportation distance are used to measure closeness, is a topic of interest.



(a) Feature 11



(b) Feature 22

Fig. 10 Changed features in the counterfactual explanation for $\boldsymbol{x}^0$ of the NATOPS data set which has been predicted by the Random Forest in $k^0 = +1$ and whose counterfactual $\boldsymbol{x}$ has to be predicted in class $k^* = -1$ with $B^{\max} = 2$. The cost function is model (4) with $\lambda_0 = 1, \lambda_2 = 0.005$

Acknowledgements This research has been financed in part by research projects EC H2020 MSCA RISE NeEDS (Grant agreement ID: 822214), FQM-329, P18-FR2369 and US-1381178 (Junta de Andalucía), and PID2019-110886RB-I00 and PID2022-137818OB-I00 (Ministerio de Ciencia, Innovación y Universidades, Spain). This support is gratefully acknowledged.

Funding Funding for open access publishing: Universidad de Sevilla/CBUA.

Declarations

Conflict of interest The authors state that there are not financial conflicts of interest related to the paper and certify that they are complying with the journal's ethical policies.

References

Aneiros G, Horová I, Hušková M, Vieu P (2022) On functional data analysis and related topics. J Multivar Anal 189:104861

Ates E, Aksar B, Leung VJ, Coskun AK (2021) Counterfactual explanations for multivariate time series. In: 2021 international conference on applied artificial intelligence (ICAPAI), pp 1–8. https://doi.org/10.1109/ICAPAI49758.2021.9462056

Benítez-Peña S, Carrizosa E, Guerrero V, Jiménez-Gamero M, MartínBarragán B, Molero-Río C, Ramírez-Cobo P, Romero Morales D, Sillero-Denamiel M (2021) On sparse ensemble methods: an application to short-term predictions of the evolution of COVID-19. Eur J Oper Res 295(2):648–663

Berndt DJ, Clifford J (1994) Using dynamic time warping to find patterns in time series. KDD Workshop 10:359–370

Bertsimas D, Dunn J (2017) Optimal classification trees. Mach Learn 106(7):1039–1082

Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. Ann Stat 44(2):813–852

Blanquero R, Carrizosa E, Jiménez-Cordero A, Martín-Barragán B (2019) Functional-bandwidth kernel for support vector machine with functional data: an alternating optimization algorithm. Eur J Oper Res 275(1):195–207

Blanquero R, Carrizosa E, Molero-Río C, Romero Morales D (2023) On optimal regression trees to detect critical intervals for multivariate functional data. Comput Oper Res 152:106152

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Carrizosa E, Kurishchenko K, Marín A, Romero Morales D (2022) Interpreting clusters by prototype optimization. Omega 107:102543

Carrizosa E, Ramírez Ayerbe J, Romero Morales D (2023) Mathematical optimization modelling for group counterfactual explanations (Tech. Rep.): IMUS, Sevilla, Spain. https://www.researchgate.net/publication/368958766_Mathematical_Optimization_Modelling_for_Group_Counterfactual_Explanations

Carrizosa E, Ramírez-Ayerbe J, Romero Morales D (2024) Generating collective counterfactual explanations in score-based classification via mathematical optimization. Expert Syst Appl 238:121954

Chaovalitwongse W, Fan Y, Sachdeo R (2008) Novel optimization models for abnormal brain activity classification. Oper Res 56:1450–1460

Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 785–794. Retrieved from https://doi.org/10.1145/2939672.2939785

Dau HA, Bagnall A, Kamgar K, Yeh C-CM, Zhu Y, Gharghabi S, Ratanamahatana CA, Keogh E (2019) The UCR time series archive. IEEE/CAA J Autom Sin 6(6):1293–1305

Delaney E, Greene D, Keane MT (2021) Instance-based counterfactual explanations for time series classification. In: International conference on case-based reasoning, pp 32–47

Du M, Liu N, Hu X (2019) Techniques for interpretable machine learning. Commun ACM 63(1):68–77

Eiras-Franco C, Guijarro-Berdinas B, Alonso-Betanzos A, Bahamonde A (2019) A scalable decision-tree-based method to explain interactions in dyadic data. Decis Support Syst 127:113141

Esling P, Agon C (2012) Time-series data mining. ACM Comput Surv (CSUR) 45(1):1–34

Fu R, Aseri M, Singh P, Srinivasan K (2022) Un fair machine learning algorithms. Manage Sci 68(6):4173–4195

Ghouaiel N, Marteau P-F, Dupont M (2017) Continuous pattern detection and recognition in stream—a benchmark for online gesture recognition. Int J Appl Pattern Recogn 4(2):146–160

Goodman B, Flaxman S (2017) European Union regulations on algorithmic decision-making and a right to explanation. AI Mag 38(3):50–57
Guidotti R (2022) Counterfactual explanations and how to find them: literature review and benchmarking. Data Min Knowl Discov (**forthcoming**)
Gurobi Optimization L (2021) Gurobi optimizer reference manual. Retrieved from http://www.gurobi.com
Hart WE, Laird CD, Watson J-P, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) Pyomo-optimization modeling in Python, vol 67, 2nd edn. Springer, New York
Hart WE, Watson J-P, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in Python. Math Program Comput 3(3):219–260
Jank W, Shmueli G (2006) Functional data analysis in electronic commerce research. Stat Sci 21(2):155–166
Karimi A-H, Barthe G, Schölkopf B, Valera I (2022) A survey of algorithmic recourse: contrastive explanations and consequential recommendations. ACM Comput Surv 55(5):1–29
Keogh E, Wei L, Xi X, Lonardi S, Shieh J, Sirowy S (2006) Intelligent icons: integrating lite-weight data mining and visualization into GUI operating systems. In: Sixth international conference on data mining (ICDM'06), pp 912–916
Lundberg S, Erion G, Chen H, DeGrave A, Prutkin J, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I (2020) From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2(1):2522–5839
Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems, pp 4768–4777
Martens D, Provost F (2014) Explaining data-driven document classifications. MIS Q 38(1):73–99
Martín-Barragán B, Lillo R, Romo J (2014) Interpretable support vector machines for functional data. Eur J Oper Res 232(1):146–155
Miller T (2019) Explanation in artificial intelligence: insights from the social sciences. Artif Intell 267:1–38
Mohammadi K, Karimi A-H, Barthe G, Valera I (2021) Scaling guarantees for nearest counterfactual explanations. In: Proceedings of the 2021 AAAI/ACM conference on AI, ethics, and society, pp 177–187
Ramon Y, Martens D, Provost F, Evgeniou T (2020) A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. Adv Data Anal Classif 14:801–819
Ramsay JO (2006) Functional data analysis. In: Encyclopedia of statistical sciences, vol 4. https://doi.org/10.1002/0471667196.ess3138
Ribeiro MT, Singh S, Guestrin C (2016) "Why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
Sood A, James GM, Tellis GJ (2009) Functional regression: a new model for predicting market penetration of new products. Mark Sci 28(1):36–51
Sunar N, Swaminathan JM (2021) Net-metered distributed renewable energy: a peril for utilities? Manag Sci 67(11):6716–6733
Tolkachev G, Mell S, Zdancewic S, Bastani O (2022) Counterfactual explanations for natural language interfaces. In: Proceedings of the 60th annual meeting of the association for computational linguistics, pp 113–118
Verma S, Dickerson J, Hines K (2020) Counterfactual explanations for machine learning: a review. arXiv preprint arXiv:2010.10596
Wachter S, Mittelstadt B, Russell C (2017) Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv JL Tech 31:841
Xing Z, Pei J, Keogh E (2010) A brief survey on sequence classification. ACM SIGKDD Explor Newsl 12(1):40–48
Zhdanov D, Bhattacharjee S, Bragin MA (2022) Incorporating FAT and privacy aware AI modeling approaches into business decision making frameworks. Decis Support Syst 155:113715
Zheng Z, Lv J, Lin W (2021) Nonsparse learning with latent variables. Oper Res 69(1):346–359