symmetry

MDPI

Article

Generalized Eigenvalue Proximal Support Vector Machine for Functional Data Classification

Yuanyuan Chen and Zhixia Yang *

College of Mathematics and Systems Science, Xinjiang University, Urumuqi 830046, China; 107551900444@stu.xju.edu.cn

* Correspondence: yangzhx@xju.edu.cn; Tel.: +86-189-9913-4166

Abstract: Functional data analysis has become a research hotspot in the field of data mining. Traditional data mining methods regard functional data as a discrete and limited observation sequence, ignoring the continuity. In this paper, the functional data classification is addressed, proposing a functional generalized eigenvalue proximal support vector machine (FGEPSVM). Specifically, we find two nonparallel hyperplanes in function space, a positive functional hyperplane, and a functional negative hyperplane. The former is closest to the positive functional data and furthest from the negative functional data, while the latter has the opposite properties. By introducing the orthonormal basis, the problem in function space is transformed into the ones in vector space. It should be pointed out that the higher-order derivative information is applied from two aspects. We apply the derivatives alone or the weighted linear combination of the original function and the derivatives. It can be expected that to improve the classification accuracy by using more data information. Experiments on artificial datasets and benchmark datasets show the effectiveness of our FGEPSVM for functional data classification.

Keywords: functional data analysis; generalized eigenvalue proximal support vector machine; orthonormal basis; functional data classification; weighted linear combination

check for updates

Citation: Chen, Y.; Yang, Z. Generalized Eigenvalue Proximal Support Vector Machine for Functional Data Classification. *Symmetry* **2021**, *13*, 833. https://doi.org/10.3390/sym13050833

Academic Editor: Ioan Raşa

Received: 17 April 2021
Accepted: 6 May 2021
Published: 9 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The concept of functional data was first put forward by Ramsay and showed that functional data can be obtained through advanced equipment [1]. Then, these data can be regarded as dynamic and continuous data, rather than static and discrete data in the traditional data analysis. Ramsay and Dalzel further proposed a systematic tool for functional data analysis (FDA) and promoted some traditional analysis methods [2]. Ramsay and Silverman popularized functional data analysis and have brought widespread attention for the last two decades [3]. In recent years, with the improvement of data acquisition technology and data storage capacity, more and more complex data are constantly appearing, presented in the form of functions. These data appear in every aspect of human production and life, such as speech recognition, spectral analysis, meteorology, etc. Functional data contains more complete and sufficient information, which can avoid information loss caused by discrete data. Therefore, it has become a research hotspot [4–8].

We deal with the problem of classifying functional data. There are numerous functional data classification problems [9–13]. For example, a doctor can determine whether the arteries of a patient are narrowly based on a Doppler ultrasonography, and marble can be automatically graded using the spectral curve. Many scholars have researched this area. Rossi and Villa showed how to use support vector machines (SVMs) for functional data classification [14]. Li and Yu suggested a classification and feature extraction method on functional data where the predictor variables are curves; it combines the classical linear discriminant analysis and SVM [15]. Muñoz proposed a functional analysis technique to obtain finite-dimensional representations of functional data. The key idea is to consider each functional datum as a point in public function space and then project these points onto

a Reproducing Kernel Hilbert Space with SVM [16]. Chang proposed a new semi-metric based on wavelet thresholding for functional data classification [17]. Martin-Barragan faced the problem of obtaining an SVM-based classifier for functional data with good classification ability and provides a classifier that is easy to interpret [18]. Blanquero proposed a new approach to optimally select the most informative time instants in multivariate functional data to get reasonable classification rates [19]. Moreover, other researchers applied different approaches to solve the functional data classification problems [20–22].

SVMs are one of the most comment methods to address classification. According to supervised learning, it is a generalized linear classifier. Its decision boundary maximizes the margin between the hyperplane of the training set [23]. The pioneering work can be traced to a study by Vapnik and Lerner in 1963 [24]. Vapnik and Chervonenkis established the linear SVM with a hard margin in 1964 [25]. Boser, Guyon, and Vapnik obtained the nonlinear SVM through the kernel method in 1992 [26]. Cortes and Vapnik proposed the nonlinear SVM of soft margins in 1995 [27], and this study received attention and citation after publication. SVM seeks a hyperplane and finally solves a quadratic programming problem in the dual space. As an extension of the SVM, the nonparallel support vector machine (NPSVM) constructs corresponding support hyperplanes for each class to describe the difference in data distribution between different classes [28–31]. The idea was first proposed by Magasarian in 2006, who came up with a generalized eigenvalue proximal support vector machine (GEPSVM) [32]. The model obtained two nonparallel hyperplanes by solving two generalized eigenvalue problems [33]. GEPSVM has two advantages over SVM. The first is the computation time, the former is solving a pair of generalized eigenvalue problems, while the latter is solving a quadratic programming problem, obviously because the former has low computational complexity. Next, GEPSVM performs better than SVM on crossed datasets.

This paper considers the binary functional data classification. Inspired by the success of GEPSVM, our proposal finds two nonparallel hyperplanes in function space. Each of the functional hyperplanes is generated such that it is close to the functional data in one class while is far away from the other class. This yields the optimization problem in function space. Then it is transformed into the vector space through the orthonormal basis. It is worthwhile to mention that our methodology is not only restricted to the original function. Indeed, the approach proposed here can be directly applied to higher-order derivatives. Only the functional data can employ the higher-order derivative information, which is impossible for the discrete data in the traditional data analysis because the discrete data does not have the property of functions. We mainly consider higher-order derivative information from two aspects. First, using derivative alone can improve the classification accuracy when the difference between original functions may not be distinct in the training set. Moreover, we employ the weighted linear combination of derivatives, which contains more information to improve the classification accuracy. The former is a particular case of the latter. Our model is based on the weighted linear combination of derivatives, and this paper considers the original function, first derivative, and second derivative in the experiment. The numerical experiment shows that the higher-order derivative information will be crucial in the classifier performance. Moreover, when we apply the weighted linear combination of the original function and the derivatives, the performance of our method is excellent on some datasets as well, which indicates that it is feasible to employ the weighted linear combination. Two main advantages are obtained from our concept. One is the classification accuracy of FGEPSVM on some datasets is better than that of SVM, such as crossed datasets. Moreover, given that the computational complexity of GEPSVM is much lower than SVM, especially when dealing with large-scale datasets.

The overall structure of this paper is as follows. Section 1 introduces the research background and significance. Section 2 reviews the related work. Section 3 introduces the functional data and proposes our method, FGEPSM, to solve the functional data classification. Section 4 is the numerical experiment, which is divided into three parts. The first part is an experiment on artificial datasets, which demonstrates the feasibility

of the proposed method. In the second part, experiments on classical functional datasets are carried out to further illustrate the effectiveness of our proposal. The last part is the parameter analysis, which discusses the influence of the important parameter on classification accuracy. Finally, the summary part briefly describes the content of this paper and the proposed methods.

2. Related Work

Here, we introduce the basic notation. Throughout this paper, uppercase and lowercase characters are used to denote matrices and vectors. $\mathbb{R}^{m\times n}$ and $\mathbb{R}^n$ denote the set of $m \times n$ matrices and n-dimensional vectors, respectively. The inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in the n-dimensional real space $\mathbb{R}^n$ is defined by $< \boldsymbol{x}, \boldsymbol{y} >= \boldsymbol{x}^T\boldsymbol{y}$. Furthermore, the inner product of two functions $x(t)$ and $y(t)$ in the square-integrable function space $L^2(T)$ is denoted by $< x(t), y(t) >= \int_T x(t)y(t)dt$. The norm in this paper refers to the 2-norm of a vector, which is denoted by $\|x\| = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}$, where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$.

Our work is closely related with the generalized eigenvalue proximal support vector machine (GEPSVM) [32]. Given a training set

$$T = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{m_+}, y_{m_+}), (\boldsymbol{x}_{m_++1}, y_{m_++1}), \ldots, (\boldsymbol{x}_{m_++m_-}, y_{m_++m_-})\}, \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, m_+, m_+ + 1, \ldots, m_+ + m_-$. $y_1 = \cdots = y_{m_+} = +1$, $y_{m_++1} = \cdots = y_{m_++m_-} = -1$. The goal of GEPSVM is to find two nonparallel hyperplanes in R^n

$$< \boldsymbol{w}_+, \boldsymbol{x} > +b_+ = 0, \tag{2}$$

and

$$< \boldsymbol{w}_-, \boldsymbol{x} > +b_- = 0, \tag{3}$$

where $\boldsymbol{w}_+, \boldsymbol{w}_- \in \mathbb{R}^n$, $b_+, b_- \in \mathbb{R}$. The positive plane (2) is closest to the data points of class $+1$ and furthest from the data points in class -1, while the negative plane (3) is closest to the points in class -1 and furthest from the points in class $+1$. Thus, the optimization problems are formulated as

$$\min_{(\boldsymbol{w}_+^T, b_+)\neq 0} \frac{\sum_{i=1}^{m_+} (< \boldsymbol{w}_+, \boldsymbol{x_i} > +b_+)^2 + \Omega(f_+)}{\sum_{i=m_++1}^{m_++m_-} (< \boldsymbol{w}_+, \boldsymbol{x_i} > +b_+)^2}, \tag{4}$$

and

$$\min_{(\boldsymbol{w}_-^T, b_-)\neq 0} \frac{\sum_{i=m_++1}^{m_++m_-} (< \boldsymbol{w}_-, \boldsymbol{x_i} > +b_-)^2 + \Omega(f_-)}{\sum_{i=1}^{m_+} (< \boldsymbol{w}_-, \boldsymbol{x_i} > +b_-)^2}, \tag{5}$$

where $f_+ =< \boldsymbol{w}_+, \boldsymbol{x} > +b_+$, $f_- =< \boldsymbol{w}_-, \boldsymbol{x} > +b_-$. $\Omega(f_+)$ and $\Omega(f_-)$ are regularization terms.

We organize the m_+ data points of Class $+1$ by $\boldsymbol{X}_+ = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m_+})^T \in \mathbb{R}^{m_+\times n}$ and the m_- data points of Class -1 by $\boldsymbol{X}_- = (\boldsymbol{x}_{m_++1}, \ldots, \boldsymbol{x}_{m_++m_-})^T \in \mathbb{R}^{m_-\times n}$. Selecting the Tikhonov regularization terms [34] to be $\Omega(f_+) = \delta \left\| \begin{pmatrix} \boldsymbol{w}_+ \\ b_+ \end{pmatrix} \right\|^2$ and $\Omega(f_-) = \delta \left\| \begin{pmatrix} \boldsymbol{w}_- \\ b_- \end{pmatrix} \right\|^2$ with a nonnegative parameter δ, (4) and (5) can be written as

$$\min_{(\boldsymbol{w}_+^T, b_+)\neq 0} \frac{\|\boldsymbol{X}_+\boldsymbol{w}_+ + b_+\boldsymbol{e}_+\|^2 + \delta \left\| \begin{pmatrix} \boldsymbol{w}_+ \\ b_+ \end{pmatrix} \right\|^2}{\|\boldsymbol{X}_-\boldsymbol{w}_+ + b_+\boldsymbol{e}_-\|^2}, \tag{6}$$

and

$$\min_{(\boldsymbol{w}_-^T, b_-)\neq 0} \frac{\|\boldsymbol{X}_-\boldsymbol{w}_- + b_-\boldsymbol{e}_-\|^2 + \delta\left\|\begin{pmatrix}\boldsymbol{w}_- \\ b_-\end{pmatrix}\right\|^2}{\|\boldsymbol{X}_+\boldsymbol{w}_- + b_-\boldsymbol{e}_+\|^2}, \tag{7}$$

where $\boldsymbol{e}_+$ and $\boldsymbol{e}_-$ are all ones vectors with length m_+ and m_-, respectively.

Starting from the identity matrix $\boldsymbol{I}$ of appropriate dimensions, define the symmetric matrices $\boldsymbol{G}, \boldsymbol{H}, \boldsymbol{R}, \boldsymbol{S} \in \mathbb{R}^{(n+1)\times(n+1)}$

$$\begin{aligned} \boldsymbol{G} &= (\boldsymbol{X}_+, \boldsymbol{e}_+)^T(\boldsymbol{X}_+, \boldsymbol{e}_+) + \delta\boldsymbol{I}, \quad \boldsymbol{H} = (\boldsymbol{X}_-, \boldsymbol{e}_-)^T(\boldsymbol{X}_-, \boldsymbol{e}_-), \\ \boldsymbol{R} &= (\boldsymbol{X}_-, \boldsymbol{e}_-)^T(\boldsymbol{X}_-, \boldsymbol{e}_-) + \delta\boldsymbol{I}, \quad \boldsymbol{S} = (\boldsymbol{X}_+, \boldsymbol{e}_+)^T(\boldsymbol{X}_+, \boldsymbol{e}_+), \end{aligned} \tag{8}$$

and the vectors $\boldsymbol{z}_+, \boldsymbol{z}_- \in \mathbb{R}^{(n+1)}$

$$\boldsymbol{z}_+ = \begin{pmatrix}\boldsymbol{w}_+ \\ b_+\end{pmatrix}, \ \boldsymbol{z}_- = \begin{pmatrix}\boldsymbol{w}_- \\ b_-\end{pmatrix}, \tag{9}$$

then the optimization problems become

$$\min_{\boldsymbol{z}_+\neq 0} \frac{\boldsymbol{z}_+^T\boldsymbol{G}\boldsymbol{z}_+}{\boldsymbol{z}_+^T\boldsymbol{H}\boldsymbol{z}_+}, \tag{10}$$

and

$$\min_{\boldsymbol{z}_-\neq 0} \frac{\boldsymbol{z}_-^T\boldsymbol{R}\boldsymbol{z}_-}{\boldsymbol{z}_-^T\boldsymbol{S}\boldsymbol{z}_-}. \tag{11}$$

Note that the objective functions in (10) and (11) are known as the Rayleigh quotients, which can be transformed into a pair of generalized eigenvalue problems

$$\boldsymbol{G}\boldsymbol{z}_+ = \lambda_+\boldsymbol{H}\boldsymbol{z}_+, \tag{12}$$

and

$$\boldsymbol{R}\boldsymbol{z}_- = \lambda_-\boldsymbol{S}\boldsymbol{z}_-. \tag{13}$$

Obviously, the global minimum of (10) is achieved at the eigenvector $\boldsymbol{z}_+^*$ of the generalized eigenvalue problem (12) corresponding to the smallest eigenvalue. Similarly, the global minimum $\boldsymbol{z}_-^*$ of (11) can be obtained. At last, the solutions $\begin{pmatrix}\boldsymbol{w}_+^* \\ b_+^*\end{pmatrix} = \boldsymbol{z}_+^*$, $\begin{pmatrix}\boldsymbol{w}_-^* \\ b_-^*\end{pmatrix} = \boldsymbol{z}_-^*$ are obtained.

A new sample $\boldsymbol{x} \in \mathbb{R}^n$ is assigned to either class $+1$ or class -1, according to which of the nonparallel hyperplanes it is closer to, i.e.,

$$Class\ \boldsymbol{x} = \begin{cases} +1, & if\ \frac{|<\boldsymbol{w}_+^*, \boldsymbol{x}>+b_+^*|}{\|\boldsymbol{w}_+^*\|} \leq \frac{|<\boldsymbol{w}_-^*, \boldsymbol{x}>+b_-^*|}{\|\boldsymbol{w}_-^*\|}, \\ -1, & otherwise. \end{cases} \tag{14}$$

3. Functional GEPSVM

Now we are in a position to address the functional data classification problem. Given a training set

$$D = \{(x_1(t), y_1), \ldots, (x_{m_++1}(t), y_{m_++1}), \ldots, (x_{m_++m_-}(t), y_{m_++m_-})\}. \tag{15}$$

where $x_i(t)$ is the functional data from the square-integrabel space $L^2(T)$, $t \in T$. T is the interval in $\mathbb{R}$, which is the domain of definition of $x_i(t)$, $i = 1, \ldots, m_+, m_+ + 1, \ldots, m_+ + m_-$.

$y_1 = \cdots = y_{m_+} = +1$, $y_{m_++1} = \cdots = y_{m_++m_-} = -1$. Corresponding to (2) and (3), we seek two nonparallel hyperplanes in function space $L^2(T)$

$$< w_+(t), x(t) > +b_+ = 0, \tag{16}$$

and

$$< w_-(t), x(t) > +b_- = 0, \tag{17}$$

where $w_+(t)$, $w_-(t) \in L^2(T)$, $b_+, b_- \in \mathbb{R}$. It is required that each of the functional hyperplanes is generated such that it is close to the functional data in one class while is far away from the functional data in the other class. Corresponding to (4) and (5), the optimization problems are formulated as

$$\min_{w_+(t),b_+} \frac{\sum_{i=1}^{m_+} [< w_+(t), x_i(t) > +b_+]^2 + \Omega(f_+)}{\sum_{i=m_++1}^{m_++m_-} [< w_+(t), x_i(t) > +b_+]^2}, \tag{18}$$

and

$$\min_{w_-(t),b_-} \frac{\sum_{i=m_++1}^{m_++m_-} [< w_-(t), x_i(t) > +b_-]^2 + \Omega(f_-)}{\sum_{i=1}^{m_+} [< w_-(t), x_i(t) > +b_-]^2}, \tag{19}$$

where $f_+ =< w_+(t), x(t) > +b_+$, $f_- =< w_-(t), x(t) > +b_-$. $\Omega(f_+)$ and $\Omega(f_-)$ are regularization terms.

Our methodology is not restricted to use the original functional data only, but also its j-th derivative $x_i^{(j)}(t)$, $j = 0, 1, \ldots, d$, where $x_i^{(0)}(t)$ is defined as the original functional data $x_i(t)$. So, more generally, it is more flexible to replace the original functional data $x_i(t)$ in (15) by the weighted linear combination of derivatives $\overline{x}_i(t) = \sum_{j=0}^{d} c_j x_i^{(j)}(t)$, where c_j is the coefficient and $\sum_{j=0}^{d} c_j = 1, c_j \geq 0$. This process yields data of the form

$$\overline{D} = \{(\overline{x}_1(t), y_1), \ldots, (\overline{x}_2(t), y_{m_+}), (\overline{x}_{m_++1}(t), y_{m_++1}) \ldots, (\overline{x}_{m_++m_-}(t), y_{m_++m_-})\} \tag{20}$$

Correspondingly, the original function data $x_i(t)$ in (18) and (19) should be taken place by the weighted linear combination $\overline{x}_i(t) = \sum_{j=0}^{d} c_j x_i^{(j)}(t)$ as well. This leads to the following optimization problems

$$\min_{w_+(t),b_+} \frac{\sum_{i=1}^{m_+} [< w_+(t), \overline{x}_i(t) > +b_+]^2 + \Omega(f_+)}{\sum_{i=m_++1}^{m_++m_-} [< w_+(t), \overline{x}_i(t) > +b_+]^2}, \tag{21}$$

and

$$\min_{w_-(t),b_-} \frac{\sum_{i=m_++1}^{m_++m_-} [< w_-(t), \overline{x}_i(t) > +b_-]^2 + \Omega(f_-)}{\sum_{i=1}^{m_+} [< w_-(t), \overline{x}_i(t) > +b_-]^2}, \tag{22}$$

where $f_+ =< w_+(t), \overline{x}_i(t) > +b_+$, $f_- =< w_-(t), \overline{x}_i(t) > +b_-$. $\Omega(f_+)$ and $\Omega(f_-)$ are regularization terms.

Next, we shall transfer the optimization problem in function space into the ones in vector space. In the square-integrable space, $L^2(T)$, so we need the following proposition:

Proposition 1. *Let $\{\phi_k(t)\}_{k=1}^{\infty}$ be the complete orthonormal basis in $L^2(T)$. Suppose that the above basis is used to expand two functions $u(t)$, $v(t) \in L^2(T)$*

$$\begin{aligned} u(t) &= \sum_{k=1}^{\infty} u_k \phi_k(t) \approx \boldsymbol{u}^T \boldsymbol{\phi}(t), \\ v(t) &= \sum_{k=1}^{\infty} v_k \phi_k(t) \approx \boldsymbol{v}^T \boldsymbol{\phi}(t), \end{aligned} \tag{23}$$

where $\boldsymbol{\phi}(t) = (\phi_1(t), \phi_2(t), \ldots, \phi_l(t))^T$, $\boldsymbol{u} = (u_1, u_2, \ldots, u_l)^T$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_l)^T$. Then, according to the property of the orthonormal basis, we get

$$< u(t), v(t) > \approx < \boldsymbol{u}, \boldsymbol{v} > . \tag{24}$$

From the above proposition, the inner product between $\overline{x}_i(t)$ and $w_+(t)$, $w_-(t)$ can be approximated by the inner product between their coefficient vectors. More precisely, for the orthonormal basis $\boldsymbol{\phi}(t) = (\phi_1(t), \phi_2(t), \ldots, \phi_l(t))^T$ in $L^2(T)$, we have

$$\overline{x}_i(t) = \sum_{j=0}^{d} c_j x_i^{(j)}(t) \approx \sum_{j=0}^{d} c_j \sum_{k=1}^{l} x_{ik}^{(j)} \phi_k(t) = \boldsymbol{c}^T \boldsymbol{X_i} \boldsymbol{\phi}(t), \; i = 1, \ldots, m_+ + m_-, \tag{25}$$

where $\boldsymbol{c} = (c_0, c_1, \ldots, c_d)^T$ is the coefficient of weighted linear combination, $x_{ik}^{(j)}$ is the coefficient of the i-th functional data with the j-th derivative corresponding to the k-th basis function, $\boldsymbol{X}_i = \begin{pmatrix} x_{i1}^{(0)} & \cdots & x_{il}^{(0)} \\ \vdots & \ddots & \vdots \\ x_{i1}^{(d)} & \cdots & x_{il}^{(d)} \end{pmatrix} \in \mathbb{R}^{(d+1)\times l}$, $i = 1, \ldots, m_+, m_+ + 1, \ldots, m_+ + m_-$. In the same way, we get

$$\begin{aligned} w_+(t) &\approx \sum_{k=1}^{l} w_k^+ \phi_k(t) = \boldsymbol{w}_+^T \boldsymbol{\phi}(t), \\ w_-(t) &\approx \sum_{k=1}^{l} w_k^- \phi_k(t) = \boldsymbol{w}_-^T \boldsymbol{\phi}(t), \end{aligned} \tag{26}$$

where $\boldsymbol{w}_+ = (w_1^+, w_2^+, \ldots, w_l^+)^T$, $\boldsymbol{w}_- = (w_1^-, w_2^-, \ldots, w_l^-)^T$. Therefore, (21) and (22) in function space can be transformed into the ones in vector space

$$\min_{(\boldsymbol{w}_+^T, b_+) \neq 0} \frac{\sum_{i=1}^{m_+} [< \boldsymbol{w}_+, (\boldsymbol{c}\boldsymbol{X}_i)^T > + b_+]^2 + \Omega(f_+)}{\sum_{i=m_+ + 1}^{m_+ + m_-} [< \boldsymbol{w}_+, (\boldsymbol{c}\boldsymbol{X}_i)^T > + b_+]^2}, \tag{27}$$

and

$$\min_{(\boldsymbol{w}_-^T, b_-) \neq 0} \frac{\sum_{i=m_+ + 1}^{m_+ + m_-} [< \boldsymbol{w}_-, (\boldsymbol{c}\boldsymbol{X}_i)^T > + b_-]^2 + \Omega(f_-)}{\sum_{i=1}^{m_+} [< \boldsymbol{w}_-, (\boldsymbol{c}\boldsymbol{X}_i)^T > + b_-]^2}. \tag{28}$$

By defining

$$\boldsymbol{X}_+ = \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_{m_+} \end{pmatrix} \in \mathbb{R}^{((d+1)\times m_+)\times l}, \quad \boldsymbol{X}_- = \begin{pmatrix} \boldsymbol{X}_{m_+ + 1} \\ \boldsymbol{X}_{m_+ + 2} \\ \vdots \\ \boldsymbol{X}_{m_+ + m_-} \end{pmatrix} \in \mathbb{R}^{((d+1)\times m_-)\times l}, \tag{29}$$

$$\begin{aligned}
\boldsymbol{C}_+ \equiv \boldsymbol{I}_+ \otimes \boldsymbol{c} &= \begin{pmatrix} I^+_{1,1}\boldsymbol{c}^T & \cdots & I^+_{m_+,1}\boldsymbol{c}^T \\ \vdots & \ddots & \vdots \\ I^+_{m_+,1}\boldsymbol{c}^T & \cdots & I^+_{m_+,m_+}\boldsymbol{c}^T \end{pmatrix} \in \mathbb{R}^{m_+ \times ((d+1)\times m_+)}, \\
\boldsymbol{C}_- \equiv \boldsymbol{I}_- \otimes \boldsymbol{c} &= \begin{pmatrix} I^-_{1,1}\boldsymbol{c}^T & \cdots & I^-_{m_-,1}\boldsymbol{c}^T \\ \vdots & \ddots & \vdots \\ I^-_{m_-,1}\boldsymbol{c}^T & \cdots & I^-_{m_-,m_-}\boldsymbol{c}^T \end{pmatrix} \in \mathbb{R}^{m_- \times ((d+1)\times m_-)},
\end{aligned} \tag{30}$$

where $\otimes$ is Kronecker product, $\boldsymbol{I}_+$ and $\boldsymbol{I}_-$ are identity matrix of appropriate dimensions. Equations (27) and (28) are transformed into the following form:

$$\min_{(\boldsymbol{w}_+^T, b_+)\neq 0} \frac{\|\boldsymbol{C}_+\boldsymbol{X}_+\boldsymbol{w}_+ + b_+\boldsymbol{e}_+\|^2 + \delta\left\|\begin{pmatrix} \boldsymbol{w}_+ \\ b_+ \end{pmatrix}\right\|^2}{\|\boldsymbol{C}_-\boldsymbol{X}_-\boldsymbol{w}_+ + b_+\boldsymbol{e}_-\|^2}, \tag{31}$$

and

$$\min_{(\boldsymbol{w}_-^T, b_-)\neq 0} \frac{\|\boldsymbol{C}_-\boldsymbol{X}_-\boldsymbol{w}_- + b_-\boldsymbol{e}_-\|^2 + \delta\left\|\begin{pmatrix} \boldsymbol{w}_- \\ b_- \end{pmatrix}\right\|^2}{\|\boldsymbol{C}_+\boldsymbol{X}_+\boldsymbol{w}_- + b_-\boldsymbol{e}_+\|^2}, \tag{32}$$

where $\boldsymbol{e}_+$ and $\boldsymbol{e}_-$ are all ones vectors with length m_+ and m_-, respectively. $\Omega(f_+) = \delta\left\|\begin{pmatrix} \boldsymbol{w}_+ \\ b_+ \end{pmatrix}\right\|^2$ and $\Omega(f_-) = \delta\left\|\begin{pmatrix} \boldsymbol{w}_- \\ b_- \end{pmatrix}\right\|^2$ are the regularization terms with a nonnegative parameter δ. For example, supposing $m_+ = 2$, $m_- = 3$, $d = 1$, $l = 2$, $\boldsymbol{C}_+$ and $\boldsymbol{C}_-$ should be defined as

$$\boldsymbol{C}_+ = \begin{pmatrix} c_0 & c_1 & 0 & 0 \\ 0 & 0 & c_0 & c_1 \end{pmatrix}, \ \boldsymbol{C}_- = \begin{pmatrix} c_0 & c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_0 & c_1 \end{pmatrix}.$$

To date, we transfer the optimization problem in function space into the vector space, and then we can solve (31) and (32) according to the method in vector space.

Starting from the identity matrix $\boldsymbol{I}$ of appropriate dimensions, define the symmetric matrices $\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{(l+1)\times(l+1)}$

$$\begin{aligned}
\boldsymbol{P} &= (\boldsymbol{C}_+\boldsymbol{X}_+, \boldsymbol{e}_+)^T(\boldsymbol{C}_+\boldsymbol{X}_+, \boldsymbol{e}_+) + \delta\boldsymbol{I}, \quad \boldsymbol{Q} = (\boldsymbol{C}_-\boldsymbol{X}_-, \boldsymbol{e}_-)^T(\boldsymbol{C}_-\boldsymbol{X}_-, \boldsymbol{e}_-), \\
\boldsymbol{U} &= (\boldsymbol{C}_-\boldsymbol{X}_-, \boldsymbol{e}_-)^T(\boldsymbol{C}_-\boldsymbol{X}_-, \boldsymbol{e}_-) + \delta\boldsymbol{I}, \quad \boldsymbol{V} = (\boldsymbol{C}_+\boldsymbol{X}_+, \boldsymbol{e}_+)^T(\boldsymbol{C}_+\boldsymbol{X}_+, \boldsymbol{e}_+),
\end{aligned} \tag{33}$$

and the vectors $\boldsymbol{z}_+, \boldsymbol{z}_- \in \mathbb{R}^{(l+1)}$

$$\boldsymbol{z}_+ = \begin{pmatrix} \boldsymbol{w}_+ \\ b_+ \end{pmatrix}, \ \boldsymbol{z}_- = \begin{pmatrix} \boldsymbol{w}_- \\ b_- \end{pmatrix}, \tag{34}$$

we can reformulate (31) and (32) as

$$\min_{\boldsymbol{z}_+\neq 0} \frac{\boldsymbol{z}_+^T\boldsymbol{P}\boldsymbol{z}_+}{\boldsymbol{z}_+^T\boldsymbol{Q}\boldsymbol{z}_+}, \tag{35}$$

and

$$\min_{\boldsymbol{z}_-\neq 0} \frac{\boldsymbol{z}_-^T\boldsymbol{U}\boldsymbol{z}_-}{\boldsymbol{z}_-^T\boldsymbol{V}\boldsymbol{z}_-}. \tag{36}$$

The above are exactly Rayleigh quotients and the global optimal solutions to the minimization problems can be readily computed by solving the following two generalized eigenvalue problems

$$\boldsymbol{P}\boldsymbol{z}_+ = \lambda_+ \boldsymbol{Q}\boldsymbol{z}_+, \tag{37}$$

and

$$\boldsymbol{U}\boldsymbol{z}_- = \lambda_- \boldsymbol{V}\boldsymbol{z}_-. \tag{38}$$

More precisely, the minimum of (35) is achieved at the eigenvector $\boldsymbol{z}_+^*$ corresponding to the smallest generalized eigenvalue. The same is true for the optimal solution $\boldsymbol{z}_-^*$ of (36). At last, the solutions $\begin{pmatrix} \boldsymbol{w}_+^* \\ b_+^* \end{pmatrix} = \boldsymbol{z}_+^*$, $\begin{pmatrix} \boldsymbol{w}_-^* \\ b_-^* \end{pmatrix} = \boldsymbol{z}_-^*$ are obtained.

A new functional data $x(t) \in L^2(T)$, it can be represented by the orthonormal basis as $x(t) \approx \sum_{k=1}^{l} x_k \phi_k(t) = \boldsymbol{x}^T \boldsymbol{\phi}(t)$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_l)^T$ is the coefficient vector. After obtaining the solutions $\boldsymbol{w}_+^*$, b_+^*, $\boldsymbol{w}_-^*$, b_-^* by Algorithm 1, $x(t)$ is assigned to class $+1$ or class -1, according to which of the hyperplanes it is closer to, i.e.,

$$Class\ x(t) = \begin{cases} +1, & if\ \frac{|<\boldsymbol{w}_+^*, \boldsymbol{x}> + b_+|}{\|\boldsymbol{w}_+^*\|} \leq \frac{|<\boldsymbol{w}_-^*, \boldsymbol{x}> + b_-|}{\|\boldsymbol{w}_-^*\|}, \\ -1, & otherwise. \end{cases} \tag{39}$$

Algorithm 1 Functional GEPSVM

Input: Training set $\overline{D}$, orthonormal basis $\boldsymbol{\phi}(t)$, $\boldsymbol{c}$, parameter δ, l, d.

Output: $\boldsymbol{w}_+^*$, b_+^*, $\boldsymbol{w}_-^*$, b_-^*

1: Calculate the coefficients $\boldsymbol{X}_i$ by (25), $i = 1, 2, \ldots, m_+ + m_-$.
2: Obtain the matrix $\boldsymbol{X}_+$, $\boldsymbol{X}_-$, $\boldsymbol{C}_+$, $\boldsymbol{C}_-$ by (29) and (30).
3: Calculate the matrix $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{U}$, $\boldsymbol{V}$ by (33).
4: Solve the generalized eigenvalue problems (37) and (38), obtain $\boldsymbol{z}_+^*$ and $\boldsymbol{z}_-^*$.
5: The solutions $\begin{pmatrix} \boldsymbol{w}_+^* \\ b_+^* \end{pmatrix} = \boldsymbol{z}_+^*$ and $\begin{pmatrix} \boldsymbol{w}_-^* \\ b_-^* \end{pmatrix} = \boldsymbol{z}_-^*$ are obtained.

4. Numerical Experiments

This section illustrates the performance of our proposal in three artificial datasets and six benchmark datasets. Comparing the performance of FGEPSVM against FSVM [14], we analyze the improvements in performance obtained when, instead of the functional data alone, up to d derivatives of the functional data are also included in the input. For this reason, we experiment with three different values of d, namely $d = 0, 1, 2$. This is a special case of our proposed weighted linear combination. Many scholars have used the higher-order derivative information of functional data to improve classification accuracy. We further consider the weighted linear combination between the original function, the first derivative, and the second derivative. There are three corresponding parameters, c_0, c_1, and c_2, and we will adjust these three parameters to select the optimal parameter combination.

To obtain stable results, k-fold cross-validation is performed. The number of folds, k, depends on the dataset considered. Particularly, if a database is small, k coincides with the number of individuals. That is to say, leave-one-out is applied. For a large-scale dataset, $k = 10$ has been chosen. This paper considers that a dataset is small if its cardinal is smaller than 100 individuals. We also apply k-fold cross-validation to select the optimal parameters in the training process. There is an essential parameter in our method, the regularization parameter δ. At the end of the numerical experiment, we analyze the effects of the parameter on the classification accuracy. We experiment on Matlab, and the processor of our computer is 11th Gen Intel(R) Core(TM) i5-1135G7.

4.1. Artificial Datasets

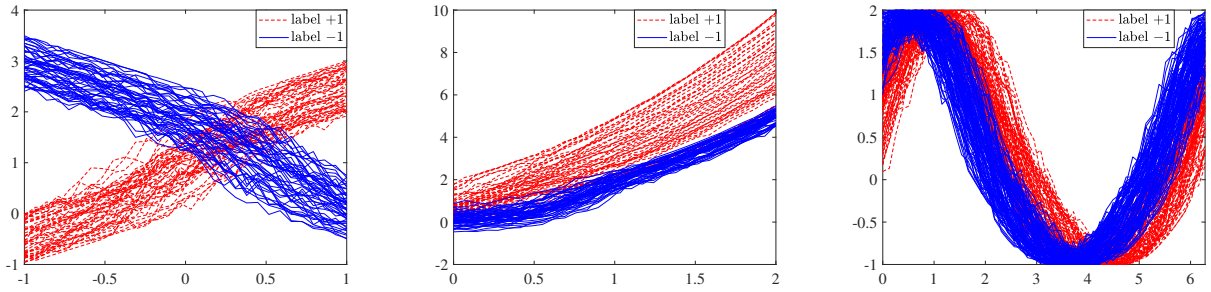
In order to illustrate the performance of our method, we use two classification methods on artificial datasets. We describe the following three artificial datasets.

Example 1. $x_+(t) = t + U_+ + \epsilon; x_-(t) = -t + U_- + \epsilon,$

Example 2. $x_+(t) = (t + U_+)^2 + \epsilon; x_-(t) = t^2 + U_- + \epsilon,$

Example 3. $x_+(t) = sin(t + U_+) + \epsilon; x_-(t) = sin(t + U_-) + \epsilon.$

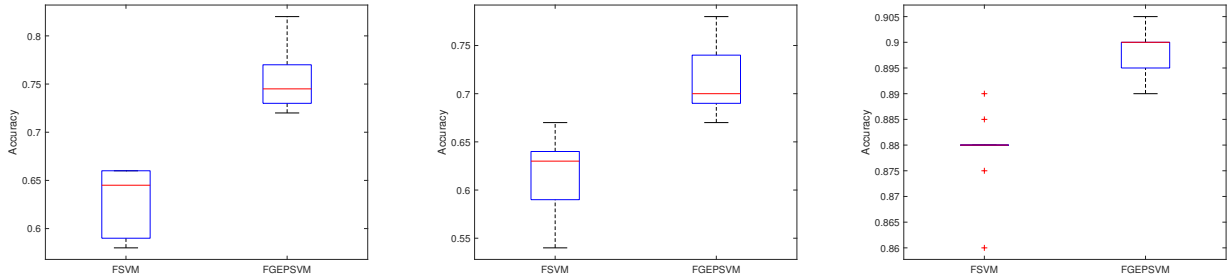
Where U_+ is a uniform random variable on the interval $(0, 1)$, and U_- is a uniform random variable on the interval $(0, 1)$, ϵ is Gaussian noise. The positive functions and negative functions in Example 1 and Example 2 are evaluated in 50 points between -1 and 1, respectively. In the same way, the positive functions and negative functions in Example 3 are evaluated in 100 points between 0 and 2π, respectively. Figure 1 displays the samples of three artificial numerical examples.



(**a**) Example 1 (**b**) Example 2 (**c**) Example 3

Figure 1. Samples of artificial numerical examples.

We perform a 10-fold cross-validation on these three artificial numerical examples to obtain the final experimental results and use the boxplot to denote the results, as shown in Figure 2. At the same time, we record the CPU time of two methods on the three datasets, as shown in Figure 3. It should be noted that the CPU time of the two methods is very different in Example 3, which leads to the time of FGEPSVM can hardly be seen when drawing the bar chart, so we have to deal with the time to some extent. We use e^t instead of t, and t is the CPU time.



(**a**) Example 1 (**b**) Example 2 (**c**) Example 3

Figure 2. Boxplots of the results on artificial numerical examples.

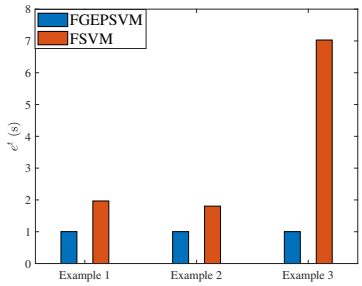


Figure 3. CPU time for two methods on artificial numerical examples.

We can see from Figure 2 that the results of the method in this paper are superior to FSVM on all three artificial datasets. The accuracy of FGEPSVM on the three artificial datasets are 0.7520, 0.7130 and 0.8985, respectively, while the corresponding results of FSVM are 0.6280, 0.6160 and 0.8780. Furthermore, our method is more stable on these three datasets. Figure 3 shows that our method has great advantages in terms of computation time. It is worth noting that nonparallel hyperplanes have certain advantages in dealing with crossed datasets. Therefore, we would like to explore whether the method proposed in this paper has the same advantages for crossed functional data in the functional space. The result of Example 1 illustrates this problem. The above artificial examples prove the feasibility and effectiveness of the proposed method.

4.2. Benchmark Datasets

We start with a detailed look at the six benchmark datasets. Coffee dataset is a two-class problem to distinguish between Robusta and Arabica coffee beans. ECG dataset contains the electrical activity recorded during one heartbeat and the two classes are a normal heartbeat and a Myocardial Infarction. Ham dataset includes measurements from Spanish and French dry-cured hams. The above three datasets are from the UCR time series classification and clustering website at http://www.cs.ucr.edu/~eamonn/time_series_data/(accessed on 7 July 2020). Weather dataset consists of one year of daily temperature measurements from 35 Canadian weather stations, and the two classes are considered depending if the total yearly amount of precipitations are above or below 600 [3]. Growth dataset consists of 93 growth curves for a sample of 54 boys and 39 girls, and the observations were measured at a set of thirty-one ages from one to eighteen years old [3]. Tecator dataset consists of 215 near-infrared absorbance spectra of meat samples. Each observation consists of a 100-channel absorbance spectrum in the wavelength range 8,501,050 nm. The goal here is to predict whether the fat percentage is greater than 20% from the spectra. The dataset is available at http://lib.stat.cmu.edu/datasets/tecator (accessed on 1 January 2020). Table 1 shows the number of each dataset, dimension, and the functions of each class.

Table 1. Data description summary.

Datasets	Size	Dimension	Lable $+1$	Lable -1
weather	35	365	15	20
coffee	56	286	27	29
growth	93	31	54	39
ECG	200	96	133	67
ham	214	431	103	111
tecator	215	100	138	77

Samples of ten individuals of each dataset are plotted in Figure 4 ($d = 0$), Figure 5 ($d = 1$) and Figure 6 ($d = 2$) are the first derivative and second derivative of the curves for functional datasets, respectively. The functional data in class -1 are plotted in solid blue line, whereas functional data in class $+1$ are depicted with a red dashed line.

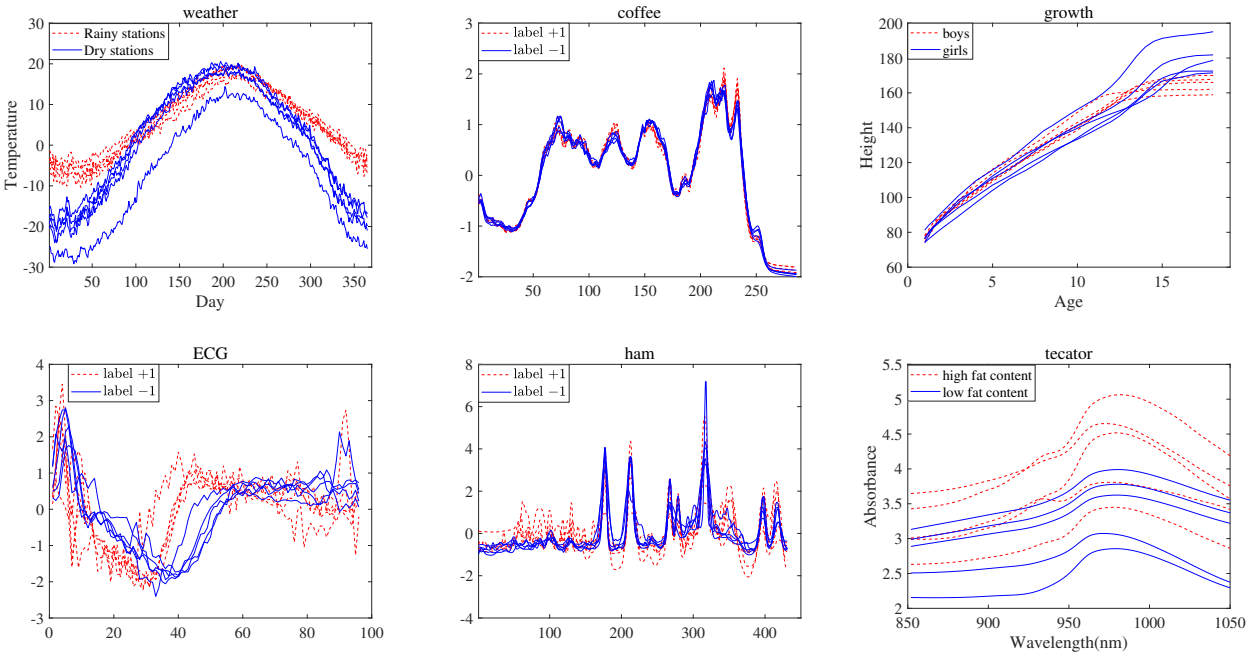


Figure 4. Original data for datasets ($d = 0$).

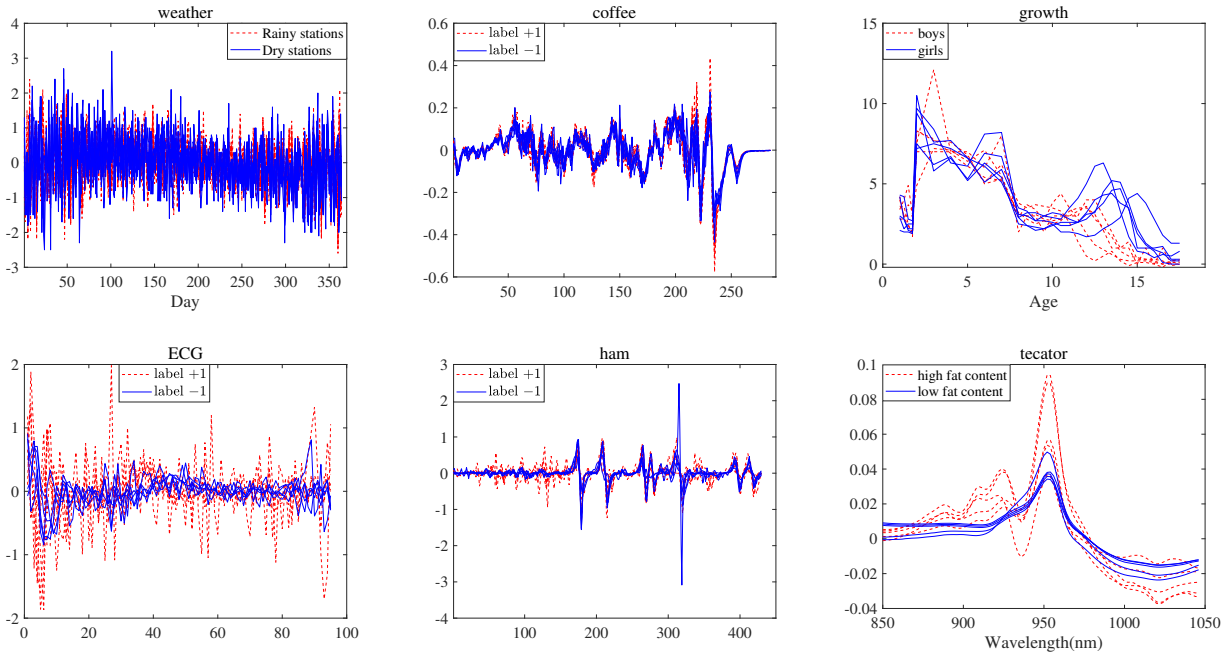


Figure 5. First derivative of the curves for datasets ($d = 1$).

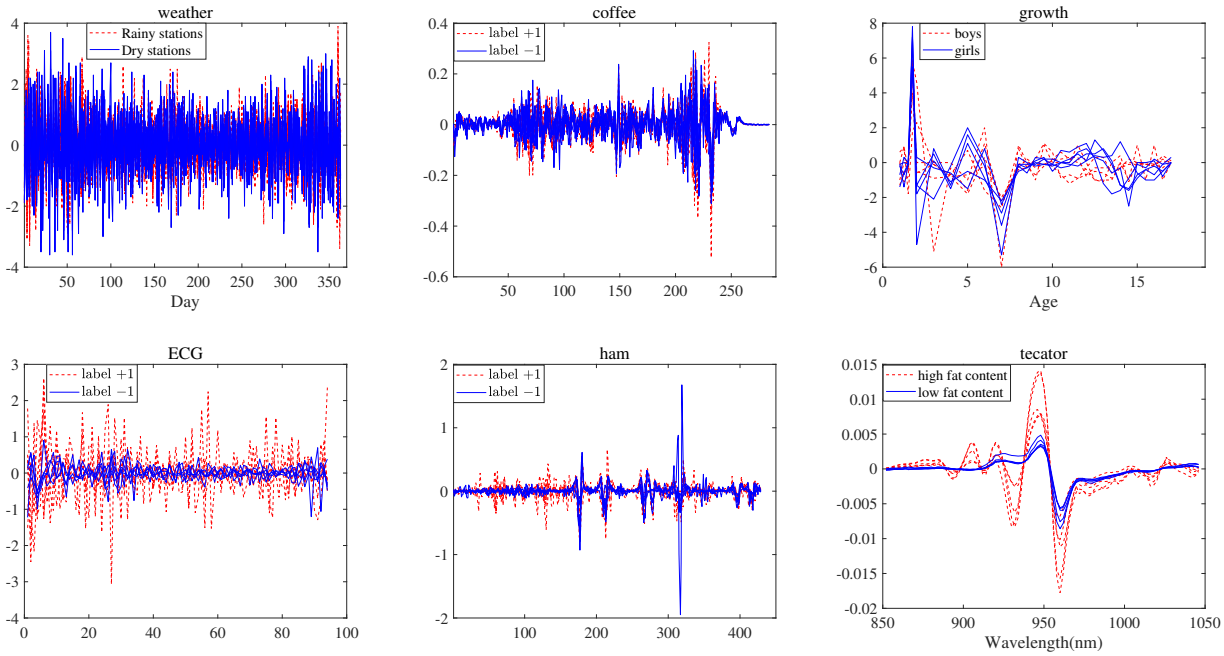


Figure 6. Second derivative of the curves for datasets ($d = 2$).

We conduct experiments to evaluate the performance of FGEPSVM on six functional datasets. We record the averaged accuracy and CPU time on the testing set provided with the information given by the data ($d = 0$), the first derivative ($d = 1$), and the second derivative ($d = 2$). Leave-one-out is performed on the weather, coffee, and growth datasets, whereas 10-fold cross-validation is implemented in ECG, ham, and tecator datasets. The classification results of the two methods are given in Table 2. The best classification accuracy of the two methods are highlighted in bold type.

Table 2. Accuracy results on functional datasets.

Dataset	*d*	FSVM	CPU Time (s)	FGEPSVM	CPU Time (s)
weather	0	0.8857	0.5911	0.8286	0.0018
	1	0.8286	0.8103	0.8000	0.0036
	2	0.6857	0.5692	**0.9143**	0.0156
coffee	0	0.9821	3.5711	**0.9821**	0.0031
	1	0.9464	1.3956	**0.9464**	0.0001
	2	0.7500	1.5059	**0.7500**	0.0056
growth	0	0.6882	4.8594	**0.7634**	0.0030
	1	0.9247	5.5469	**0.9570**	0.0025
	2	0.6667	5.0469	**0.7634**	0.0042
ECG	0	0.7280 ± 0.0127	2.1422	0.6840 ± 0.0084	0.0031
	1	0.6650 ± 0.0001	1.9187	**0.6650** ± 0.0001	0.0031
	2	0.6680 ± 0.0035	2.0594	0.6660 ± 0.0120	0.0001
ham	0	0.6015 ± 0.0164	2.7484	**0.6165** ± 0.0114	0.0031
	1	0.6033 ± 0.0203	3.7813	**0.6139** ± 0.0179	0.0047
	2	0.5932 ± 0.0114	3.8891	**0.6147** ± 0.0174	0.0016
tecator	0	0.8796 ± 0.0068	2.3344	0.8384 ± 0.0135	0.0001
	1	0.9902 ± 0.0033	2.3281	0.9196 ± 0.0055	0.0016
	2	0.9676 ± 0.0045	2.3031	**0.9786** ± 0.0059	0.0016

In Table 2, FGEPSVM performed better than FSVM on most datasets. It is important to note that when the two methods produce the same results on the dataset, we consider our method more advantageous. This is based on computation time because we solve a pair of generalized eigenvalue problems, while the latter solves a quadratic programming problem. When dealing with the same dataset, the former calculation speed is significantly faster than the latter. Therefore, when the classification results are the same, we believe that FGEPSVM is more advantageous than FSVM. Furthermore, FSVM takes much more time than FGEPSVM on large-scale datasets, so that our approach has advantages in dealing with large-scale datasets.

Moreover, using higher-order derivative information also has a particular impact on classification performance. Higher-order derivative information can improve classification accuracy reflected in weather, growth, and tecator datasets. The classification accuracy of the second derivative for the weather dataset is much higher than that of the original function and the first derivative. Similarly, the classification results of the first derivative for the growth dataset are much higher than that of the original function and second derivative. Compared with the classification accuracy of the original function results, using the first derivative and the second derivative are improve the result for the tecator dataset. However, not all higher-order derivative information of datasets can do this. Therefore, when we solve functional data classification, we can appropriately consider using higher-order derivative information to improve the classification accuracy. In this paper, the first derivative and second derivative are considered in the numerical experiments. Any derivative can be used, which depends on the characteristics of the datasets.

We further apply the weighted linear combination of derivatives. The dimension of the weighted linear combination coefficient vector $\boldsymbol{c}$ depends on how we use the d derivatives. This article uses the original function, the first derivative, and the second derivative, so $\boldsymbol{c}$ is a three-dimensional vector. There are three weight coefficients, c_0, c_1, c_2. Since the sum of them is one, only two parameters are needed to select. Parameters c_0 and c_1 are selected from the values $\{0, \frac{1}{12}, \ldots, \frac{11}{12}, 1\}$. One of the ways to select the optimal weighted linear combination coefficient vector is through k-fold cross-validation to obtain higher classification accuracy. Table 3 shows the classification results corresponding to the optimal parameters.

Table 3. Accuracy results on the weighted linear combination.

Dataset	$\boldsymbol{c}$	FGEPSVM	CPU Time
weather	$\boldsymbol{c} = (0, 0, 1)^T$	0.9143	0.0156
coffee	$\boldsymbol{c} = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})^T$	**1.0000**	0.0001
growth	$\boldsymbol{c} = (\frac{1}{3}, 0, \frac{2}{3})^T$	**0.9892**	0.0001
ECG	$\boldsymbol{c} = (\frac{11}{12}, \frac{1}{12}, 0)^T$	**0.7005** $\pm$ 0.0169	0.0016
ham	$\boldsymbol{c} = (\frac{1}{3}, 0, \frac{2}{3})^T$	**0.6326** $\pm$ 0.0232	0.0016
tecator	$\boldsymbol{c} = (0, 0, 1)^T$	0.9786 $\pm$ 0.0059	0.0016

In Table 3, the results in bold indicate that the classification results obtained by using weighted linear combination are better than those obtained by employing specific derivatives. The classification result of the original function on the coffee dataset is 0.9821, while the corresponding classification accuracy of the weighted linear combination is 1.0000. The classification accuracy corresponding to the first derivative on the growth dataset is 0.9570, while the classification accuracy is improved by 0.0322 after applying the weighted linear combination. Similarly, using the weighted linear combination of ham and ECG datasets also improves the accuracy compared to the previous results. Therefore, it also indicates that it is feasible and practical for us to consider employing the weight combination of derivatives.

4.3. *Parameter Sensitivity Analysis*

There is a key parameter δ in FGEPSVM. To analyze the sensitivity of the parameter, we study the effect of the parameter on classification accuracy. The parameter δ is selected from the values $\{10^i | i = -12, -11, \ldots, 12\}$. For this study, we conduct experiments on datasets, as seen Figure 7.

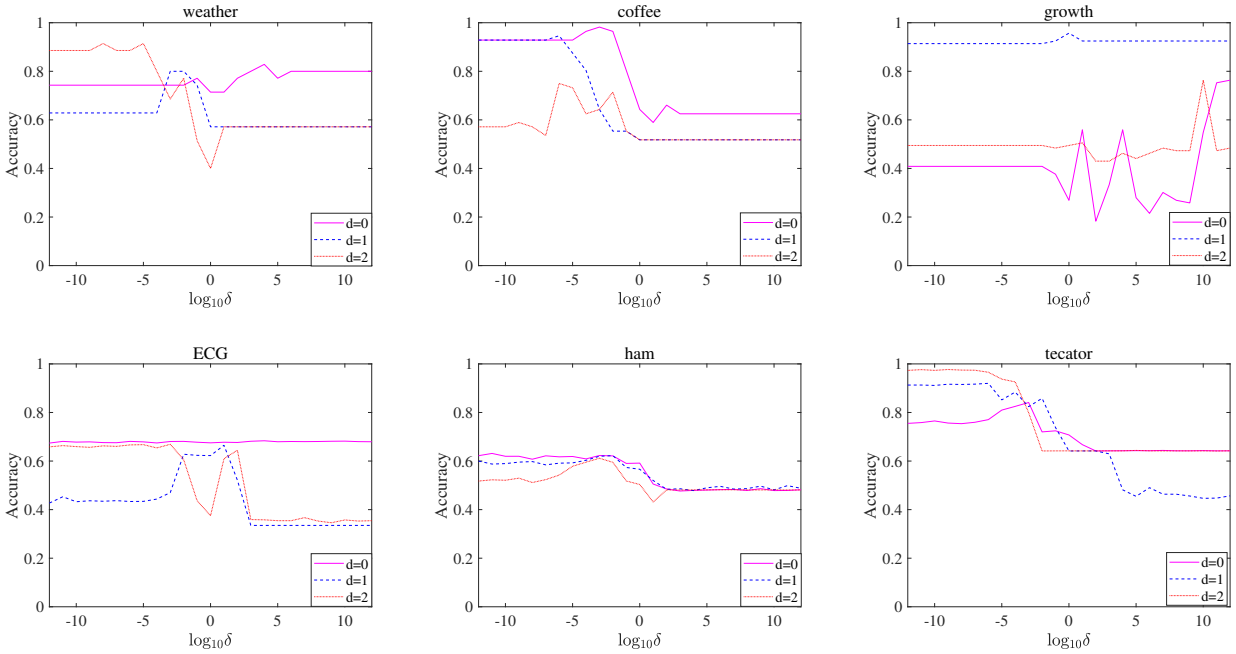


Figure 7. The classification performance of FGEPSVM with δ on datasets.

We find that FGEPSVM is sensitive to δ. This indicates that δ has a greater impact on the performance of our proposal, so it needs to be adjusted carefully. It can also be seen that when the parameter δ is relatively small, the classification accuracy of most datasets is the best.

5. Conclusions

This paper proposes a functional generalized eigenvalue proximal support vector machine, which looks for two nonparallel hyperplanes in function space, making each functional hyperplane close to the functional data in one class and far away from the functional data in the other class. The corresponding optimization problem involves the inner product between two functional data, which can be replaced by the inner product between two coefficient vectors by introducing the orthonormal basis. Thus, the optimization problem in function space is transformed into the vector space through this orthonormal basis. Furthermore, our method is not restricted to the original functional data. The method presented in this paper can be applied to higher-order derivatives. Only when dealing with functional data can we consider using higher-order derivative information. We use the derivatives alone. Besides, we employ the weighted linear combination of derivatives that contain more information. It can be seen that the former is a particular case of the latter. We consider the weighted linear combination of the original function, first derivative, and second derivative. The results of numerical experiments show that the higher-order derivative information is critical to the performance of the classifier.

Author Contributions: Y.C.: Methodology, software, writing—original draft. Z.Y.: project administration, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Xinjiang Provincial Natural Science Foundation of China (No.2020D01C028), the Xinjiang Provincial University Research Foundation of China (No. XJEDU2018I002) and the National Natural Science Foundation of China (No. 12061071).

Data Availability Statement: All data used during the study were provided by a third party. In the numerical experiment, we have given the sources of all datasets for easy download.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ramsay, J.O. When the data are functions. *Psychometrika* **1982**, *47*, 379–396. [CrossRef]
2. Ramsay, J.O.; Dalzell, C.J. Some Tools for Functional Data Analysis. *J. R. Stat. Soc. B* **1991**, *53*, 539–572. [CrossRef]
3. Ramsay, J.O.; Silverman, B.W. *Functional Data Analysis*; Springer: New York, NY, USA, 2005.
4. Altland, H.W. Applied Functional Data Analysis. *Technometrics* **2003**, *45*, 101–102. [CrossRef]
5. Auton, T. Applied Functional Data Analysis: Methods and Case Studies. *J. R. Stat. Soc. A Stat.* **2004**, *167*, 378–379. [CrossRef]
6. Yao, F.; Muller, H.G.; Wang, J.L. Functional Data Analysis for Sparse Longitudinal Data. *J. Am. Stat. Assoc.* **2005**, *100*, 577–590. [CrossRef]
7. Greenwood, M. Nonparametric Functional Data Analysis: Theory and Practice. *J. R. Stat. Soc. A Stat.* **2007**, *102*, 1077. [CrossRef]
8. Aneiros, G.; Cao, R.; Fraiman, R.; Genest, C.; Vieu, P. Recent advances in functional data analysis and high-dimensional statistics. *J. Multivar. Anal.* **2018**, *170*, 3–9. [CrossRef]
9. Ferraty, F.; Vieu, P. Curves discrimination: A nonparametric functional approach. *Comput. Stat. Data Anal.* **2003**, *44*, 161–173. [CrossRef]
10. Ferraty, F. Most-predictive design points for functional data predictors. *Biometrika* **2010**, *97*, 807–824. [CrossRef]
11. Leng, X.Y.; Müller, H.J. Classification using functional data analysis for temporal gene expression data. *Bioinformatics* **2006**, *22*, 68–76. [CrossRef]
12. Lopez, M.; Martinez, J.; Matias, J.M.; Taboada, J.; Vilan, J.A. Functional classification of ornamental stone using machine learning techniques. *J. Comput. Appl. Math.* **2010**, *234*, 1338–1345. [CrossRef]
13. Fortuna, F.; Battista, T.D. Functional unsupervised classification of spatial biodiversity. *Ecol. Indic.* **2020**, *111*, 106027. [CrossRef]
14. Rossi, F.; Villa, N. Support vector machine for functional data classification. *Neurocomputing* **2005**, *69*, 730–742. [CrossRef]
15. Li, B.; Yu, Q.Z. Classification of functional data: A segmentation approach. *Comput. Stat. Data Anal.* **2008**, *52*, 4790–4800. [CrossRef]
16. Muñoz, A.; Gonzalez, J. Representing functional data using support vector machines. *Pattern Recognit. Lett.* **2009**, *31*, 511–516.
17. Chang, C.; Chen, Y.K.; Todd Ogden, R. Functional data classification: A wavelet approach. *Comput. Stat.* **2014**, *29*, 1497–1513. [CrossRef]
18. Martin-Barragan, B.; Lillo, R.; Romo, J. Interpretable support vector machines for functional data. *Eur. J. Oper. Res.* **2014**, *232*, 146–155. [CrossRef]
19. Blanquero, R.; Carrizosa, E.; Jimenez-Cordero, A.; Martin-Barragan, B. Variable Selection in Classification for Multivariate Functional Data. *Inf. Sci.* **2018**, *481*, 445–462. [CrossRef]
20. Berrendero, J.R.; Cuevas, A. Variable selection in functional data classification: A maxima-hunting proposal. *Stat. Sin.* **2016**, *26*, 619–638.
21. Berrendero, J.R.; Cuevas, A.; Torrecilla, J.L. On the Use of Reproducing Kernel Hilbert Spaces in Functional Classification. *J. Am. Stat. Assoc.* **2018**, *113*, 1210–1218. [CrossRef]
22. Blanquero, R.; Carrizosa, E.; Jimenez-Cordero, A.; Martin-Barragan, B. Functional-bandwidth Kernel for Support Vector Machine with Functional Data: An Alternating Optimization Algorithm. *Eur. J. Oper. Res.* **2018**, *275*, 195–207. [CrossRef]
23. Vapnik, V.N. Statistical Learning Theory. *Encycl. Sci. Learn.* **1998**, *41*, 3185.
24. Vapnik, V.N.; Lerner, A.Y. Recognition of Patterns with help of Generalized Portraits. *Avtomat. U Telemekh* **1963**, *24*, 774–780.
25. Vapnik, V.N.; Chervonenkis, A.Y. A note on class of perceptron. *Autom. Remote Control* **1964**, *25*. Available online: http://www.kernel-machines.org/publications/VapChe64 (accessed on 10 April 2021).
26. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. *Comput. Learn. Theory* **1992**, 144–152. [CrossRef]
27. Cortes, C.; Vapnik, V.N. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
28. Jayadeva, R.K.; Khemchandani, R.; Chandra, S. Twin support vector machines for pattern classification. *IEEE Trans. Pattern Anal.* **2007**, *29*, 905–910. [CrossRef]
29. Shao, Y.H.; Zhang, C.H.; Wang, X.B.; Deng, N.Y. Improvements on twin support vector machines. *IEEE Trans. Neural. Netw.* **2011**, *22*, 962–968. [CrossRef]
30. Shao, Y.H.; Chen, W.J.; Deng, N.Y. Nonparallel hyperplane support vector machine for binary classification problems. *Inf. Sci.* **2014**, *263*, 22–35. [CrossRef]
31. Tian, Y.J.; Qi, Z.Q.; Ju, X.C.; Shi, Y.; Liu, X.H. Nonparallel support vector machines for pattern classification. *IEEE Trans. Cybern.* **2014**, *44*, 1067–1079. [CrossRef] [PubMed]

32. Mangasarian, O.L.; Wild, E.W. Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Trans. Pattern Anal.* **2006**, *28*, 69–74. [CrossRef] [PubMed]
33. Guarracino, M.R.; Cifarelli, C.; Seref, O.; Pardalos, P.M. A classification method based on generalized eigenvalue problems. *Optim. Methods Softw.* **2007**, *22*, 73–81. [CrossRef]
34. Tikhonov, A.N.; Arsen, V.Y. *Solutions of Ill-Posed Problems*; Wiley: New York, NY, USA, 1977.