Statistica Sinica **34** (2024), 1545-1564
doi:https://doi.org/10.5705/ss.202022.0057

OPTIMAL CLASSIFICATION FOR FUNCTIONAL DATA

Shuoyang Wang, Zuofeng Shang, Guanqun Cao* and Jun S. Liu

University of Louisville, New Jersey Institute of Technology, Michigan State University and Harvard University

Abstract: A central topic in functional data analysis is how to design an optimal decision rule, based on training samples, to classify a data function. We exploit the optimal classification problem in which the data functions are Gaussian processes. We derive sharp convergence rates for the minimax excess misclassification risk both when the data functions are fully observed and when they are discretely observed. We explore two easily implementable classifiers, based on a discriminant analysis and on a deep neural network, respectively, which both achieve optimality in Gaussian settings. Our deep neural network classifier is new in the literature, and demonstrates outstanding performance, even when the data functions are nonGaussian. For discretely observed data, we discover a novel critical sampling frequency that governs the sharp convergence rates. The proposed classifiers perform favorably in finite-sample applications, shown in comparisons with other functional classifiers in simulations and one real-data application.

Key words and phrases: Functional classification, functional deep neural network, functional quadratic discriminant analysis, Gaussian process, minimax excess misclassification risk.

1. Introduction

In many applications, data are collected in the form of functions, such as curves or images. Such data are referred to as functional data. A fundamental problem in functional data analysis is to classify a data function based on training samples. For instance, in the speech recognition data extracted from the TIMIT database (Ferraty and Vieu (2003)), the training samples are digitized speech curves of American English speakers from different phoneme groups, and the task is to predict the phoneme of a new speech curve. Classic multivariate analysis techniques, such as logistic regression or discriminant analysis, are not directly applicable, because functional data are intrinsically infinite-dimensional (Wang, Chiou and Müller (2016)). A common strategy is to adapt a multivariate analysis to functional settings, such as functional logistic regression (Araki et al. (2009)) and functional discriminant analysis (Shin (2008); Delaigle, Hall and Bathia (2012); Delaigle and Hall (2012, 2013); Galeano, Joseph and Lillo (2015); Dai, Müller and Yao (2017); Berrendero, Cuevas and Torrecilla (2018);

*Corresponding author.

Park, Ahn and Jeon (2020)), among others. However, despite their impressive performance, we may wish to know whether and which of these approaches is statistically optimal, and, how to construct an optimal functional classifier that performs even better.

Optimal classification has been investigated in multivariate settings (Mammen and Tsybakov (1999); Tsybakov (2004); Lecué (2008); Farnia and Tse (2016); Cai and Zhang (2019a,b); Mazuelas, Zanoni and Perez (2020)). Here, the term "optimality" refers to minimizing the excess misclassification risk relative to the oracle Bayes rule, which provides a theoretical understanding of the nature of the problem and a benchmark against which to measure the performance of a classifier. Optimal classification in a functional setting is more challenging, because the data are infinite-dimensional. Existing works, such as that of Delaigle and Hall (2012), focus on the special case that the Bayes risk vanishes, referred to perfect classification. As revealed in Berrendero, Cuevas and Torrecilla (2018), the Bayes risk vanishes when the probability measures of the populations are mutually singular. If the two populations have equivalent probability measures, that is, the singularity fails, then the density functions of the two populations are finite, and the Bayes risk does not vanish. The latter scenario is more challenging, because the two populations are much "closer" to each other, in the sense that the differences between the population means and the covariances are sufficiently smooth. There is a lack of literature on how to design an optimal functional classifier in this situation.

In this study, we investigate the optimal classification problem under the Gaussian setting, that is, the observed data are Gaussian processes. In the nonvanishing Bayes risk setting, we derive sharp rates for the minimax Excess misclassification risk (MEMR), which provides a theoretical understanding of how to approximate the Bayes risk based on training samples. Our results cover both fully observed data and discretely observed data. We also show that a functional quadratic discriminant analysis (FQDA) and a functional deep neural network (FDNN) both achieve sharp rates of MEMR, and hence are minimax optimal.

Although functional discriminant analysis is a popular technique for classifying Gaussian data (Galeano, Joseph and Lillo (2015); Dai, Müller and Yao (2017)), its optimality remains an open problem. Hence, we provides the first rigorous analysis to fill this gap. Specifically, we derive an upper bound for the excess misclassification risk of an FQDA in a Gaussian setting that matches the sharp rate of MEMR. In conventional settings, such as low- or high-dimensional data classification, the optimality of the discriminant analysis has been established by Anderson (2003) and Cai and Zhang (2019a,b). Our work can be viewed as a nontrivial extension of their results to functional data. In practice, an FQDA is known to perform poorly when the data are nonGaussian, so it is desirable to design a classifier that is robust to a violation of the Gaussian assumption. We propose a novel FDNN classifier based on a deep neural network

(DNN) to address this issue. FDNNs have been proven to achieve the same optimality as that of an FQDA in the Gaussian setting, and exhibit better classification accuracy when the data are nonGaussian. DNNs have been applied in various nonparametric problems; see Schmidt-Hieber (2020), Bauer and Kohler (2019), Kim, Ohn and Kim (2021), Liu, Boukai and Shang (2022),Liu, Shang and Cheng (2021), and Hu, Shang and Cheng (2020). The present work provides the first application of a DNN to functional data classification with provable guarantees.

In the setting of discretely observed data, the rate of convergence for MEMR demonstrates an interesting phase transition phenomenon; jointly characterized by the number of data curves and the sampling frequency. The discretely observed data scenario is practically meaningful, because in real-world problems, functional data can only be observed at discrete sampling points. Our analysis reveals that when the sampling frequency is relatively small, the number of data curves has little effect on the rate of MEMR. When the sampling frequency is relatively large, the rate of MEMR depends more on the number of data curves. In other words, there exists a critical sampling frequency that governs the performance of the minimax optimal classifier. Cai and Yuan (2011) show the existence of a critical sampling frequency that governs the optimal estimation in a functional regression. The present work has made a relevant and new discovery in functional classification.

The rest of the paper is organized as follows. Section 2 provides background on the functional Bayes classifier and optimal functional classification. Section 3 establishes sharp rates for MEMR for both fully observed data and discretely observed data. Sections 4 and 5 propose FQDA and FDNN classifiers, respectively, both of which are proven optimal. Section 6 compares FQDA and FDNN with existing functional classification methods using simulations. Section 7 applies our method to analyze a speech recognition data set. Section 8 concludes the paper. Major technical details for the proofs of the main results are deferred to the Supplementary Material.

Notation and Terminology. We introduce some basic notation and definitions that we use throughout the rest of the paper. Vectors and matrices are denoted by boldface letters. For a matrix $\mathbf{A} \in \mathbb{R}_{p\times p}$, $|\mathbf{A}|$ is the determinant of $\mathbf{A}$, and $\mathbf{I}_p$ is the $p \times p$ identity matrix. For two sequences of positive numbers a_n and b_n, $a_n \lesssim b_n$ means that for some constant $c > 0$, $a_n \leq cb_n$, for all n, $a_n \asymp b_n$ means $a_n \lesssim b_n$ and $b_n \lesssim a_n$, and $a_n \ll b_n$ means $\lim_{n\to\infty} a_n/b_n = 0$. We also use $c, c_0, c_1, \ldots, C, C_0, C_1, \ldots$ to denote absolute constants, the values of which may change, depending on the context.

2. Preliminaries

In this section, we provide some background onthe functional Bayes classifier and an optimal classification in a Gaussian setting.

Let $Z(t), t \in \mathcal{T} := [0,1]$ be a random process. We say that Z belongs to class k if $Z \sim \mathcal{GP}(\eta_k, \Omega_k)$, for $k = 1, 2$, where $\mathcal{GP}(\eta_k, \Omega_k)$ is a Gaussian process with unknown mean function η_k and unknown covariance function Ω_k. For $k = 1, 2$, let $\pi_k \in (0,1)$ be the unknown probability of Z belonging to class k, satistying $\pi_1 + \pi_2 = 1$. Suppose that Ω_k satisfies the eigen-decomposition

$$\Omega_k(s,t) = \sum_{j=1}^{\infty} \lambda_j^{(k)} \psi_j(s)\psi_j(t), s, t \in \mathcal{T}, \tag{2.1}$$

where $\psi_j, j \geq 1$ is an orthonormal basis of $L^2(\mathcal{T})$ w.r.t. the usual L^2 inner product $\langle \cdot, \cdot \rangle$, and $\lambda_j^{(k)}$ are positive eigenvalues. Note that (2.1) requires that the covariance functions possess the same eigenfunctions, which is a common assumption for technical convenience; see Delaigle and Hall (2012) and Dai, Müller and Yao (2017). Write $\eta_k(t) = \sum_{j=1}^{\infty} \mu_{kj}\psi_j(t) \in L^2(\mathcal{T})$ and $Z(t) = \sum_{j=1}^{\infty} z_j \psi_j(t)$, where μ_{kj} represent the projection scores of η_k, and z_j represent the projection scores of Z. When Z belongs to class k, z_j are pairwise uncorrelated with the mean μ_{kj} and variance $\lambda_j^{(k)}$.

Define $\boldsymbol{\theta} = \left(\pi_1, \pi_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}\right)$, in which $\boldsymbol{\mu}_k = (\mu_{k1}, \mu_{k2}, \ldots)$ and $\boldsymbol{\lambda}^{(k)} = (\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots)$ are infinite sequences of mean and variance projection scores, respectively. Given $\boldsymbol{\theta}$, it follows from Berrendero, Cuevas and Torrecilla (2018) and Torrecilla et al. (2020) that the functional Bayes rule for classifying a new data function $Z \in L^2(\mathcal{T})$ has the expression

$$G^*_{\boldsymbol{\theta}}(Z) = \begin{cases} 1, & Q^*(Z, \boldsymbol{\theta}) \geq 0, \\ 2, & Q^*(Z, \boldsymbol{\theta}) < 0, \end{cases} \tag{2.2}$$

where

$$Q^*(Z, \boldsymbol{\theta}) = -\langle Z - \eta_1, Z - \eta_1 \rangle_{\Omega_1} + \langle Z - \eta_2, Z - \eta_2 \rangle_{\Omega_2} - \log\left(\frac{|\Omega_2|}{|\Omega_1|}\right) + 2\log\left(\frac{\pi_1}{\pi_2}\right),$$

where $\langle Z - \eta_k, Z - \eta_k \rangle_{\Omega_k} = \sum_{j=1}^{\infty}((z_j - \mu_{kj})^2/\lambda_j^{(k)})$, $|\Omega_2|/|\Omega_1| = \prod_{j=1}^{\infty} \lambda_j^{(2)}/\lambda_j^{(1)}$. Berrendero, Cuevas and Torrecilla (2018) and Torrecilla et al. (2020) show that $Q^*(Z, \boldsymbol{\theta})$ is well defined and almost surely finite when the probability measures of the two classes are equivalent.

In practice, $G^*_{\boldsymbol{\theta}}$ is unobservable, because $\boldsymbol{\theta}$ is unknown. Suppose we observe a training sample $\{X_i^{(k)}(t) : 1 \leq i \leq n_k, k = 1, 2, t \in \mathcal{T}\}$, where n_k is the sample size for class k, $X_i^{(k)} \sim \mathcal{GP}(\eta_k, \Omega_k)$, all $X_i^{(k)}$ are independent, and are independent of Z to be classified. For a generic classifier $\widehat{G}$ constructed using the training samples, its performance is measured by the misclassification risk

$R_{\boldsymbol{\theta}}(\widehat{G}) = E_{\boldsymbol{\theta}}[\mathbb{I}\{\widehat{G}(Z) \neq Y(Z)\}]$ under the true parameter $\boldsymbol{\theta}$, where $Y(Z)$ denotes the unknown label of Z.

Following Delaigle and Hall (2012) and Dai, Müller and Yao (2017), if

$$\text{both } \sum_{j=1}^{\infty} \frac{(\mu_{1j} - \mu_{2j})^2}{\lambda_j^{(2)}} \text{ and } \sum_{j=1}^{\infty} \left(\frac{\lambda_j^{(1)}}{\lambda_j^{(2)}} - 1 \right)^2 \text{ are convergent,} \tag{2.3}$$

then $R_{\boldsymbol{\theta}}(G^*_{\boldsymbol{\theta}}) > 0$. Classification under (2.3) is challenging, because the two Gaussian measures are asymptotically equivalent; see Berrendero, Cuevas and Torrecilla (2018) for a special case when $\lambda_j^{(1)} = \lambda_j^{(2)}$. Because $G^*_{\boldsymbol{\theta}}$ achieves the smallest risk, it is impossible to design a classifier with zero risk. Instead, we aim to construct a classifier $\widehat{G}$, based on training samples, that performs similarly to $G^*_{\boldsymbol{\theta}}$, which motivates the study of MEMR,

$$\inf_{\widehat{G}} \sup_{\theta \in \Theta} E[R_{\boldsymbol{\theta}}(\widehat{G}) - R_{\boldsymbol{\theta}}(G^*_{\boldsymbol{\theta}})],$$

where the infimum is taken over all functional classifiers constructed using the training samples, and Θ is a parameter space, described in the following section.

3. Sharp Rates for MEMR

We derive sharp rates for MEMR for fully observed data and for discretely observed data. To the best of our knowledge, these are the first results exploring MEMR in a functional setting.

3.1. Parameter space

Our MEMR results rely on an explicit parameter space for $\boldsymbol{\theta}$. We first introduce the concepts of hyperrectangles and Sobolev balls.

Definition 1. A hyperrectangle of order $\omega > 0$ and length $A > 0$ is defined as

$$H^{\omega}(A) = \left\{ \boldsymbol{a} = (a_1, a_2, \ldots) : \sup_{j \geq 1} |a_j| j^{1+\omega} \leq A \right\}. \tag{3.1}$$

An implication of $\boldsymbol{a} \in H^{\omega}(A)$ is that $|a_k| \leq A k^{-(1+\omega)}$, for any $k \geq 1$, in which ω governs the decay rate of the coordinates.

Definition 2. An ℓ_1-Sobolev ball of order $\omega > 0$ and radius $A > 0$ is defined as

$$S^{\omega}(A) = \left\{ \boldsymbol{a} = (a_1, a_2, \ldots) : \sum_{j=1}^{\infty} |a_j| j^{\omega} \leq A \right\}. \tag{3.2}$$

An implication of $\boldsymbol{a} \in S^{\omega}(A)$ is that $\sum_{k=L}^{\infty} |a_k| \leq A L^{-\omega}$, for any $L \geq 1$, in which ω governs the decay rate of the tail sum.

Hyperrectangles and Sobolev balls depict different perspectives of a real sequence: the former controls a sequence in an element-wise manner, and the latter controls its tail sum. Although overlapping, hyperrectangles and Sobolev balls do not include each other.

In the rest of this article, consider the following two parameter spaces for $\boldsymbol{\theta}$. For $\nu_1, \nu_2 > 0$,

$$\begin{aligned}\Theta_H(\nu_1,\nu_2) := \{\, \boldsymbol{\theta} : \left\{\mu_{1j}^2 \vee \mu_{2j}^2\right\}_{j\geq 1} \in H^{\nu_1}, \left\{\lambda_j^{(1)} \vee \lambda_j^{(2)}\right\}_{j\geq 1} \in H^{\nu_1}, \\ \left\{\frac{(\mu_{1j}-\mu_{2j})^2}{\lambda_j^{(2)}}\right\}_{j\geq 1} \in H^{\nu_2}, \left\{\left(\frac{\lambda_j^{(1)}}{\lambda_j^{(2)}}-1\right)^2\right\}_{j\geq 1} \in H^{\nu_2}, \\ C_0 \leq \pi_1, \pi_2 \leq 1 - C_0 \,\}, \end{aligned} \tag{3.3}$$

and

$$\begin{aligned}\Theta_S(\nu_1,\nu_2) := \{\, \boldsymbol{\theta} : \left\{\mu_{1j}^2 \vee \mu_{2j}^2\right\}_{j\geq 1} \in S^{\nu_1}, \left\{\lambda_j^{(1)} \vee \lambda_j^{(2)}\right\}_{j\geq 1} \in S^{\nu_1}, \\ \left\{\frac{(\mu_{1j}-\mu_{2j})^2}{\lambda_j^{(2)}}\right\}_{j\geq 1} \in S^{\nu_2}, \left\{\left(\frac{\lambda_j^{(1)}}{\lambda_j^{(2)}}-1\right)^2\right\}_{j\geq 1} \in S^{\nu_2}, \\ C_0 \leq \pi_1, \pi_2 \leq 1 - C_0 \,\}, \end{aligned} \tag{3.4}$$

where $C_0 \in (0, 1/2)$ is a constant, $H^\omega = H^\omega(A)$, and $S^\omega = S^\omega(A)$. For notational simplicity, A is omitted. Specifically, $\boldsymbol{\theta} \in \Theta_H(\nu_1, \nu_2)$ implies that μ_{kj}^2 and $\lambda_j^{(k)}$ belong to H^{ν_1}, and that $(\mu_{1j} - \mu_{2j})^2/\lambda_j^{(2)}$ and $(\lambda_j^{(1)}/\lambda_j^{(2)} - 1)^2$ belong to H^{ν_2}. ν_1 governs the smoothness of the mean functions and the covariance functions, and ν_2 governs the smoothness of the separation of the two populations. Moreover, the series $\sum_{j=1}^\infty (\mu_{1j} - \mu_{2j})^2/\lambda_j^{(2)}$ and $\sum_{j=1}^\infty (\lambda_j^{(1)}/\lambda_j^{(2)} - 1)^2$ are both convergent, which implies that the Bayes risk is nonvanishing; see (2.3). One can interpret $\boldsymbol{\theta} \in \Theta_S(\nu_1, \nu_2)$ similarly. In the subsequent subsections, we derive the rate of MEMR under parameter spaces (3.3) and (3.4) for fully observed data and for discretely observed data.

3.2. Sharp rate of MEMR under fully observed data

Suppose that the data functions $X_i^{(k)}(t)$, for $i = 1, \ldots, n_k$, $k = 1, 2$, are fully observed, for arbitrary $t \in \mathcal{T}$. Throughout, let $n = n_1 \wedge n_2$.

Theorem 1. *For both $\Theta = \Theta_H(\nu_1, \nu_2)$ and $\Theta = \Theta_S(\nu_1, \nu_2)$, the following holds:*

$$\inf_{\widehat{G}} \sup_{\boldsymbol{\theta}\in\Theta} E\left[R_{\boldsymbol{\theta}}(\widehat{G}) - R_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}^*)\right] \asymp \left(\frac{\log n}{n}\right)^{\nu_2/(1+\nu_2)},$$

where the infimum is taken over all functional classifiers.

Theorem 1 provides a sharp rate for MEMR under parameter spaces (3.3) and (3.4). Interestingly, the rate relies on ν_2 rather than ν_1, implying that the smoothness of the population mean and covariance differences plays a more crucial role than the smoothness of the mean and covariance functions in terms of the performance of the optimal functional classifier. Specifically, the sharp rate for MEMR becomes faster when ν_2 increases, which may be because of the fully observed data. In fact, as discussed in Section 3.3, when the data are observed discretely , this phenomenon may not hold. Moreover, the optimal rate appears to depend only on the smoothness, rather than the size, of the difference between the two populations. This means that the optimal rate does not change if the size of the population difference changes and its smoothness remains the same.

3.3. Sharp rate of MEMR under discretely observed data

Suppose we observe $X_i^{(k)}(t_1), \ldots, X_i^{(k)}(t_M)$,for $i = 1, \ldots, n_k$, $k = 1, 2$, on evenly spaced $t_1, \ldots, t_M \in \mathcal{T}$; that is, the data functions are observed over M evenly spaced sampling points. For technical convenience, we make an additional assumption that ψ_j in (2.1) are Fourier bases of $L^2(\mathcal{T})$, that is, $\psi_1(t) = 1, \psi_{2j}(t) = \sqrt{2}\cos(2j\pi t)$, and $\psi_{2j+1}(t) = \sqrt{2}\sin(2j\pi t)$, for $j \geq 1$, $t \in \mathcal{T}$.

Theorem 2. *Let $\nu_1, \nu_2 > 0$ with $\nu_1 \leq 1 + \nu_2$. For both $\Theta = \Theta_H(\nu_1, \nu_2)$ and $\Theta = \Theta_S(\nu_1, \nu_2)$, the following holds:*

$$\inf_{\widehat{G}} \sup_{\boldsymbol{\theta} \in \Theta} E\left[R_{\boldsymbol{\theta}}(\widehat{G}) - R_{\boldsymbol{\theta}}(G^*_{\boldsymbol{\theta}})\right] \asymp \left(\frac{\log n}{n} + \frac{1}{M^{\nu_1}}\right)^{\nu_2/(1+\nu_2)},$$

where the infimum is taken over all functional classifiers.

Theorem 2 reveals that $M^* = (n/\log n)^{1/\nu_1}$ is a critical sampling frequency for the rate of MEMR over the parameter space $\Theta_H(\nu_1, \nu_2)$ and $\Theta_S(\nu_1, \nu_2)$. When $M \geq M^*$, the MEMR is of rate $(\log n/n)^{\nu_2/(1+\nu_2)}$, which is free of M and is consistent with the rate derived in Theorem 1. In other words, when $M \geq M^*$, the optimal classifier performs as well as the one based on fully observed data. When $M < M^*$, the MEMR is of rate $M^{-\nu_1\nu_2/(1+\nu_2)}$, which relies solely on M. Another interesting finding is that, when $M < M^*$, the rate of MEMR relies on both ν_1 and ν_2, that is, the smoothness of the mean and covariance functions, as well as the separation between the populations. This differs from estimation or testing problems in which the minimax optimal rate relies only on the smoothness of the mean function (see Cai and Yuan (2011, 2012); Hilgert, Mas and Verzelen (2013); Shang and Cheng (2015)).

4. Functional Quadratic Discriminant Analysis

In this section, we establish an optimal functional classifier based on FQDA that requires accurately estimating the functional Bayes classifier by estimating

the principle mean projection scores and principle eigenvalues. FQDA is a popular technique in the functional classification literature (See Galeano, Joseph and Lillo (2015); Dai, Müller and Yao (2017)). The basic idea is to first project the data functions onto an orthonormal basis and extract the principle projection scores, and then to perform a conventional QDA over the extracted scores. FQDA performs well when the data are Gaussian processes, but there is a lack of rigorous proof on the optimality of FQDA. Here, we construct a FQDA classifier and prove its optimality in both fully observed data and discretely observed data.

4.1. FQDA for fully observed data

Consider the ideal case that the data functions are fully observed, as in Section 3.2. Write $X_i^{(k)}(t) = \sum_{j=1}^{\infty} \xi_{ij}^{(k)} \psi_j(t)$, for $i = 1, \ldots, n_k$, $k = 1, 2$, where $\xi_{ij}^{(k)}$ are the observed projection scores. For $J \geq 1$, let

$$\widehat{\boldsymbol{\mu}}_k = (\bar{\xi}_{\cdot 1}^{(k)}, \ldots, \bar{\xi}_{\cdot J}^{(k)})^\top, \quad \widehat{\mathbf{D}} = \widehat{\boldsymbol{\Sigma}}_2^{-1} - \widehat{\boldsymbol{\Sigma}}_1^{-1}, \quad \widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\Sigma}}_2^{-1}(\widehat{\boldsymbol{\mu}}_2 - \widehat{\boldsymbol{\mu}}_1), \tag{4.1}$$

where $\bar{\xi}_{\cdot j}^{(k)} = {n_k}^{-1} \sum_{i=1}^{n_k} \xi_{ij}^{(k)}$ is the estimation of the mean projection score, $\widehat{\lambda}_j^{(k)} = n_k^{-1} \sum_{i=1}^{n_k} (\xi_{ij}^{(k)} - \bar{\xi}_{\cdot j}^{(k)})^2$ is the estimation of the eigenvalue, and $\widehat{\boldsymbol{\Sigma}}_k = \mathrm{diag}(\widehat{\lambda}_1^{(k)}, \ldots, \widehat{\lambda}_J^{(k)})$ is the estimation of the covariance operator. The FQDA classifier is designed as follows:

$$\widehat{G}_J^{FQDA}(Z) = \begin{cases} 1, & \widehat{Q}(\boldsymbol{z}) \geq 0, \\ 2, & \widehat{Q}(\boldsymbol{z}) < 0, \end{cases} \tag{4.2}$$

where

$$\widehat{Q}(\boldsymbol{z}) := (\boldsymbol{z} - \widehat{\boldsymbol{\mu}}_1)^\top \widehat{\mathbf{D}} (\boldsymbol{z} - \widehat{\boldsymbol{\mu}}_1) - 2\widehat{\boldsymbol{\beta}}^\top (\boldsymbol{z} - \widehat{\bar{\boldsymbol{\mu}}}) - \log\left(|\widehat{\mathbf{D}}\widehat{\boldsymbol{\Sigma}}_1 + \mathbf{I}_J|\right) + 2\log\left(\frac{\widehat{\pi}_1}{\widehat{\pi}_2}\right),$$

$\boldsymbol{z} = (z_1, \ldots, z_J)^\top$ includes the first J projection scores of Z (see Section 2), $\widehat{\bar{\boldsymbol{\mu}}} = (\widehat{\boldsymbol{\mu}}_1 + \widehat{\boldsymbol{\mu}}_2)/2$, and $\widehat{\pi}_k = n_k/(n_1 + n_2)$ is the sample proportion of class k. Heuristically, when J is suitably large, (4.2) performs similarly to the functional Bayes classifier (2.2).

Theorem 3. *For both $\Theta = \Theta_H(\nu_1, \nu_2)$ and $\Theta = \Theta_S(\nu_1, \nu_2)$, the proposed FQDA classifier* (4.2) *satisfies*

$$\sup_{\boldsymbol{\theta} \in \Theta} E\left[R_{\boldsymbol{\theta}}(\widehat{G}_{J^*}^{FQDA}) - R_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}^*)\right] \lesssim \left(\frac{\log n}{n}\right)^{\nu_2/(1+\nu_2)},$$

where $J^* \asymp (n/\log n)^{1/(1+\nu_2)}$.

Theorem 3 provides an upper bound for the excess misclassification risk of (4.2) with $J = J^*$. Because the upper bound matches Theorem 1, we claim that FQDA attains minimax optimality if the leading J^* basis functions are used to construct the classifier.

4.2. FQDA for discretely observed data

Consider the more realistic case in which the data functions are observed discretely, as in Section 3.3. For $1 \le J \le M$, define

$$\mathbf{B} = \begin{pmatrix} \psi_1(t_1) & \psi_2(t_1) & \cdots & \psi_J(t_1) \\ \psi_1(t_2) & \psi_2(t_2) & \cdots & \psi_J(t_2) \\ \vdots & \vdots & & \vdots \\ \psi_1(t_M) & \psi_2(t_M) & \cdots & \psi_J(t_M) \end{pmatrix}.$$

Heuristically, when J is suitably large, the data vector $\mathbf{X}_i^{(k)} = (X_i^{(k)}(t_1), \ldots, X_i^{(k)}(t_M))^\top$ has an approximate expression $n_k^{-1}\sum_{i=1}^{n_k}\mathbf{X}_i^{(k)} \approx \mathbf{B}\boldsymbol{\mu}_k$, for $i = 1, \ldots, n_k$, where $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kJ})^\top$ is the vector of J principle mean projection scores. When ψ_j are a Fourier basis, it holds that $\mathbf{B}^\top\mathbf{B} = M\mathbf{I}_J$, which leads to

$$\boldsymbol{\mu}_k \approx \frac{1}{n_k}\sum_{i=1}^{n_k}\boldsymbol{\zeta}_i^{(k)}, \tag{4.3}$$

where $\boldsymbol{\zeta}_i^{(k)} = M^{-1}\mathbf{B}^\top\mathbf{X}_i^{(k)}$. For $k = 1, 2$, let

$$\widehat{\boldsymbol{\mu}}_{sk} = \frac{1}{n_k}\sum_{i=1}^{n_k}\boldsymbol{\zeta}_i^{(k)}, \quad \widehat{\mathbf{D}}_s = \widehat{\boldsymbol{\Sigma}}_{s2}^{-1} - \widehat{\boldsymbol{\Sigma}}_{s1}^{-1}, \quad \widehat{\boldsymbol{\beta}}_s = \widehat{\boldsymbol{\Sigma}}_{s2}^{-1}\left(\widehat{\boldsymbol{\mu}}_{s2} - \widehat{\boldsymbol{\mu}}_{s1}\right), \tag{4.4}$$

where $\widehat{\boldsymbol{\Sigma}}_{sk} = \mathrm{diag}(\widehat{\lambda}_{s1}^{(k)}, \ldots, \widehat{\lambda}_{sJ}^{(k)})$ with $\widehat{\lambda}_{sj}^{(k)} = n_k^{-1}\sum_{i=1}^{n_k}(\zeta_{ij}^{(k)} - \bar{\zeta}_{\cdot j}^{(k)})^2$, $\bar{\zeta}_{\cdot j}^{(k)} = n_k^{-1}\sum_{i=1}^{n_k}\zeta_{ij}^{(k)}$, and $\zeta_{ij}^{(k)}$ are components of $\boldsymbol{\zeta}_i^{(k)}$. We then propose the following classification rule, called sampling FQDA (sFQDA):

$$\widehat{G}_J^{sFQDA}(Z) = \begin{cases} 1, & \widehat{Q}_s(\boldsymbol{z}) \ge 0, \\ 2, & \widehat{Q}_s(\boldsymbol{z}) < 0, \end{cases} \tag{4.5}$$

where

$$\widehat{Q}_s(\boldsymbol{z}) := (\boldsymbol{z} - \widehat{\boldsymbol{\mu}}_{s1})^\top\widehat{\mathbf{D}}_s(\boldsymbol{z} - \widehat{\boldsymbol{\mu}}_{s1}) - 2\widehat{\boldsymbol{\beta}}_s^\top(\boldsymbol{z} - \widehat{\bar{\boldsymbol{\mu}}}_s) - \log\left(|\widehat{\mathbf{D}}_s\widehat{\boldsymbol{\Sigma}}_{s1} + \mathbf{I}_J|\right) + 2\log\left(\frac{\widehat{\pi}_1}{\widehat{\pi}_2}\right),$$

with $\widehat{\bar{\boldsymbol{\mu}}}_s = (\widehat{\boldsymbol{\mu}}_{s1} + \widehat{\boldsymbol{\mu}}_{s2})/2$.

Theorem 4. *Let $\nu_1, \nu_2 > 0$ with $\nu_1 \le 1 + \nu_2$. For both $\Theta = \Theta_H(\nu_1, \nu_2)$ and $\Theta = \Theta_S(\nu_1, \nu_2)$, the sFQDA in (4.5) satisfies*

$$\sup_{\boldsymbol{\theta}\in\Theta} E\left[R_{\boldsymbol{\theta}}(\widehat{G}_{J^*}^{sFQDA}) - R_{\boldsymbol{\theta}}(G_{\boldsymbol{\theta}}^*)\right] \lesssim \left(\frac{\log n}{n} + \frac{1}{M^{\nu_1}}\right)^{\nu_2/(1+\nu_2)},$$

where $J^* \asymp M^{\nu_1/(1+\nu_2)}\mathbb{I}(M < M^*) + (n/\log n)^{1/(1+\nu_2)}\mathbb{I}(M \ge M^*)$, $M^* = (n/\log n)^{1/\nu_1}$ and $\mathbb{I}(\cdot)$ is the indicator function.

Theorem 4 provides an upper bound for the excess misclassification risk of (4.5) with $J = J^*$, which matches Theorem 2. Therefore, we claim that sFQDA attains minimax optimality if the leading J^* basis functions are used to construct the classifier.

Although FQDA is optimal in a Gaussian setting, in general, it performs poorly when the data are nonGaussian. Hence, it is desirable to design a more accurate classifier for nonGaussian data that preserves the same optimality in the Gaussian case. In the next section, we propose a novel approach to do so, based on a DNN.

5. FDNN

DNNs are used in nonparametric regression and classification problems; see Schmidt-Hieber (2020), Bauer and Kohler (2019), Kim, Ohn and Kim (2021), Liu, Boukai and Shang (2022), Liu, Shang and Cheng (2021), Wang, Cao and Shang (2021), andHu, Shang and Cheng (2020). To the best of our knowledge, this is the first application of a DNN to functional data classification. The basic idea is to train a DNN classifier using the observed principle projection scores. Intuitively, when the network architectures are well selected, the DNN should have high expressive power, so that the functional Bayes classifier can be well approximated, even when its explicit form is not known. Hence, FDNN is expected to be more resistant than FQDA for nonGaussian data. We first define a sparse DNN, and then construct FDNN classifiers for fully observed data and for discretely observed data, and prove their optimality.

5.1. Sparse DNN

A DNN tends to overfit the training data, owing to too much capacity of the network class. A common practice is to sparsify the network parameters, using methods such as dropout (Ian, Yoshua and Aaron (2016)). Our approach is to train a functional classifier using a sparse DNN that addresses the overfitting issue problem effectively.

Let σ denote the rectifier linear unit (ReLU) activation function, that is, $\sigma(x) = (x)_+$ for $x \in \mathbb{R}$. For any real vectors $\boldsymbol{V} = (v_1, \ldots, v_r)^\top$ and $\boldsymbol{y} = (y_1, \ldots, y_r)^\top$, define the shift activation function $\sigma_{\boldsymbol{V}}(\boldsymbol{y}) = (\sigma(y_1 - v_1), \ldots, \sigma(y_r - v_r))^\top$. For $L, J \geq 1$ and $\boldsymbol{p} = (p_0, p_1, \ldots, p_L, p_{L+1}) \in \mathbb{N}^{L+2}$, let $\mathcal{F}(L, J, \boldsymbol{p})$ denote the class of DNNs over J inputs, with L hidden layers and p_l nodes in the hidden layer l, for $l = 1, \ldots, L$. Let $p_0 = J$ and $p_{L+1} = 1$. Any $f \in \mathcal{F}(L, J, \boldsymbol{p})$ has an expression

$$f(\mathbf{x}) = \mathbf{W}_L \sigma_{\boldsymbol{V}_L} \mathbf{W}_{L-1} \sigma_{\boldsymbol{V}_{L-1}} \cdots \mathbf{W}_1 \sigma_{\boldsymbol{V}_1} \mathbf{W}_0 \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^J, \tag{5.1}$$

where $\mathbf{W}_l \in \mathbb{R}^{p_{l+1} \times p_l}$, for $l = 0, \ldots, L$, are weight matrices, and $\boldsymbol{V}_l \in \mathbb{R}^{p_l}$, for

$l = 1, \ldots, L$, are shift vectors. The sparse DNN class is defined as

$$\begin{aligned}&\mathcal{F}(L, J, \boldsymbol{p}, s, B) \\ &= \Bigg\{ f \in \mathcal{F}(L, J, \boldsymbol{p}) : \max_{l=0,\ldots,L} \|\mathbf{W}_l\|_\infty + \|\mathbf{v}_l\|_\infty \le B, \sum_{l=0}^{L} \|\mathbf{W}_l\|_0 + \|\mathbf{v}_l\|_0 \le s, \\ &\quad \|f\|_\infty \le 1 \Bigg\},\end{aligned} \tag{5.2}$$

where $\|\cdot\|_\infty$ denotes the maximum-entry norm of a matrix/vector or supnorm of a function, $\|\cdot\|_0$ denotes the number of nonzero entries of a matrix or vector, $s > 0$ controls the number of nonzero weights and shifts, and $B > 0$ controls the largest weights and shifts. For notational convenience, we assume that the supnorm of f has a unit upper bound, which can be replaced by an arbitrary positive constant.

5.2. FDNN classifier for fully observed data

Let $\phi : \mathbb{R} \to [0, \infty)$ denote a surrogate loss such as the hinge loss $\phi(x) = (1 - x)_+$. For $k = 1, 2$ and $i = 1, \ldots, n_k$, recall $X_i^{(k)}(t) = \sum_{j=1}^{\infty} \xi_{ij}^{(k)} \psi_j(t)$ (see Section 4.1), and for $J \ge 1$, let $\boldsymbol{\xi}_i^{(k)} = (\xi_{i1}^{(k)}, \xi_{i2}^{(k)}, \ldots, \xi_{iJ}^{(k)})$ be the vector of J principle projection scores corresponding to $X_i^{(k)}$. Define the decision function

$$\widehat{f}_\phi(\cdot) = \operatorname*{argmin}_{f \in \mathcal{F}(L, J, \boldsymbol{p}, s, B)} \sum_{k=1}^{2} \sum_{i=1}^{n_k} \phi((2k - 3) f(\boldsymbol{\xi}_i^{(k)})).$$

Specifically, $\widehat{f}_\phi$ is the best network in $\mathcal{F}(L, J, \boldsymbol{p}, s, B)$ minimizing the empirical surrogate loss. In practice, we suggest using the R package "`Keras`" to find $\widehat{f}_\phi$.

We then propose the following FDNN classifier:

$$\widehat{G}^{FDNN}(Z) = \begin{cases} 1, & \widehat{f}_\phi(\boldsymbol{z}) \ge 0, \\ 2, & \widehat{f}_\phi(\boldsymbol{z}) < 0. \end{cases} \tag{5.3}$$

Theorem 5. *Suppose the network class $\mathcal{F}(L, J, \boldsymbol{p}, s, B)$ satisfies*

(i) $L \asymp \log n$;

(ii) $J \asymp n^{\frac{1}{1+\nu_2}} (\log n)^{-4/(1+\nu_2)}$;

(iii) $\max_{0 \le \ell \le L} p_\ell \asymp n^{1/(1+\nu_2)} (\log n)^{(\nu_2 - 3)/(1+\nu_2)}$;

(iv) $s \asymp n^{\frac{1}{1+\nu_2}} (\log n)^{(2\nu_2 - 2)/(1+\nu_2)}$;

(v) $B \asymp n^{\nu_2/(2+2\nu_2)} (\log n)^{(2-2\nu_2)/(1+\nu_2)}$.

For both $\Theta = \Theta_H(\nu_1, \nu_2)$ *and* $\Theta = \Theta_S(\nu_1, \nu_2)$*, the FDNN classifier* (5.3) *satisfies*

$$\sup_{\boldsymbol{\theta}\in\Theta} E\left[R_{\boldsymbol{\theta}}(\widehat{G}^{FDNN}) - R_{\boldsymbol{\theta}}(G^*_{\boldsymbol{\theta}})\right] \lesssim \left(\frac{\log^4 n}{n}\right)^{\nu_2/(1+\nu_2)}.$$

Theorem 5 provides an upper bound for the excess misclassification risk of (5.3). When the neural network architectures $(L, J, \boldsymbol{p}, s, B)$ are properly selected, the upper bound matches Theorem 1 up to a log factor. Therefore, the FDNN is proven to be minimax optimal.

5.3. FDNN classifier for discretely observed data

For $i = 1, \ldots, n_k$, $k = 1, 2$, let $\boldsymbol{\zeta}_i^{(k)}$ be given in (4.3). Define the decision function

$$\widehat{f}_\phi^{(s)}(\cdot) = \operatorname*{argmin}_{f\in\mathcal{F}(L,J,\boldsymbol{p},s,B)} \sum_{k=1}^{2}\sum_{i=1}^{n_k} \phi((2k-3)f(\boldsymbol{\zeta}_i^{(k)})).$$

We then propose the following sampling FDNN (sFDNN) classifier:

$$\widehat{G}^{sFDNN}(Z) = \begin{cases} 1, & \widehat{f}_\phi^{(s)}(\boldsymbol{z}) \geq 0, \\ 2, & \widehat{f}_\phi^{(s)}(\boldsymbol{z}) < 0. \end{cases} \tag{5.4}$$

Theorem 6. *Suppose the network class* $\mathcal{F}(L, J, \boldsymbol{p}, s, B)$ *satisfies*

(i) $L \asymp (\log M)\mathbb{I}(M \leq M^*) + (\log n)\mathbb{I}(M \geq M^*)$*;*

(ii) $J \asymp M^{\nu_1/(1+\nu_2)}\mathbb{I}(M \leq M^*) + n^{1/(1+\nu_2)}(\log n)^{-4/(1+\nu_2)}\mathbb{I}(M \geq M^*)$*;*

(iii) $\max_{0\leq\ell\leq L} p_\ell \asymp M^{\nu_1/(1+\nu_2)}(\log M)\mathbb{I}(M \leq M^*) + n^{1/(1+\nu_2)}(\log n)^{(\nu_2-3)/(1+\nu_2)}\mathbb{I}(M \geq M^*)$*;*

(iv) $s \asymp M^{\nu_1/(1+\nu_2)}(\log^2 M)\mathbb{I}(M \leq M^*) + n^{1/(1+\nu_2)}(\log n)^{(2\nu_2-2)/(1+\nu_2)}\mathbb{I}(M \geq M^*)$*;*

(v) $B \asymp M^{\nu_1\nu_2/(2+2\nu_2)}\mathbb{I}(M \leq M^*) + n^{\nu_2/(2+2\nu_2)}(\log n)^{(2-2\nu_2)/(1+\nu_2)}\mathbb{I}(M \geq M^*)$*,*

where $M^* = \left(n/\log^4 n\right)^{1/\nu_1}$*. Let* $\nu_1, \nu_2 > 0$*, with* $\nu_1 \leq 1 + \nu_2$*. For both* $\Theta = \Theta_H(\nu_1, \nu_2)$ *and* $\Theta = \Theta_S(\nu_1, \nu_2)$*, the sFDNN classifier in* (5.4) *satisfies*

$$\sup_{\boldsymbol{\theta}\in\Theta} E\left[R_{\boldsymbol{\theta}}(\widehat{G}^{sFDNN}) - R_{\boldsymbol{\theta}}(G^*_{\boldsymbol{\theta}})\right] \lesssim \left(\frac{\log^4 n}{n} + \frac{1}{M^{\nu_1}}\right)^{\nu_2/(1+\nu_2)}.$$

Theorem 6 provides an upper bound for the excess misclassification risk of (5.3). When the architectures $(L, J, \boldsymbol{p}, s, B)$ are properly selected, the upper bound matches the result in Theorem 2 up to a log factor. Therefore, the sFDNN is able to attain minimax optimality. The critical sampling frequency $M^* = \left(n/\log^4 n\right)^{1/\nu_1}$ differs from the one in Theorem 2 by a log factor as well.

6. Simulation

Here, we examine the performances of FQDA and FDNN using simulations.

6.1. Gaussian setting

In this section, we provide numerical evidence to demonstrate the superior performance of FQDA and FDNN compared with two popular functional classifiers: the quadratic discriminant method (QD) proposed by Delaigle and Hall (2013), and the nonparametric Bayes classifier (NB) proposed by Dai, Müller and Yao (2017). We do not include the functional logistic regression because it performs worse than NB when covariance differences in the populations are present (Dai, Müller and Yao (2017)). The difference between FQDA and QD lies in how they estimate principle projection scores. Specifically, FQDA estimates the projection scores by projecting the functional data onto a Fourier basis, and QD applies a functional principal component analysis to estimate the principle projection scores in which the eigenfunctions are data-driven. We evaluated all methods using four synthetic data sets. In all simulations, we generated $n = n_1 = n_2 = 50, 100$ training samples for each class, and thus $\pi_1 = \pi_2 = 0.5$. We generated functional data $X_i^{(k)}(t) = \sum_{j=1}^{J} \xi_{ij}^{(k)} \psi_j(t)$, where $\xi_{ij}^{(k)} \sim N(\mu_{kj}, \lambda_j^{(k)})$, for $i = 1, \ldots, n_k$, $k = 1, 2$. In the following, $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kJ})^\top$, $\boldsymbol{\Sigma}_k = \operatorname{diag}(\lambda_1^{(k)}, \ldots, \lambda_J^{(k)})$, and $\psi_j(t)$ are specified in different models for $t \in [0, 1]$.

Model 1: Let $J = 3$, $\boldsymbol{\mu}_1 = (-1, 2, -3)^\top$, $\boldsymbol{\Sigma}_1^{1/2} = \operatorname{diag}(3/5, 2/5, 1/5)$, $\boldsymbol{\mu}_2 = (-1/2, 5/2, -5/2)^\top$, $\boldsymbol{\Sigma}_2^{1/2} = \operatorname{diag}(9/10, 1/2, 3/10)$, $\psi_1(t) = \log(t + 2)$, $\psi_2(t) = t$, and $\psi_3(t) = t^3$.

Model 2: Let $J = 3$, $\boldsymbol{\mu}_1 = (-6, 12, -18)^\top$, $\boldsymbol{\Sigma}_1^{1/2} = \operatorname{diag}(3, 2, 1)$, $\boldsymbol{\mu}_2 = (-3, 9, -15)^\top$, $\boldsymbol{\Sigma}_2^{1/2} = \operatorname{diag}(9/2, 5/2, 3/2)$, $\psi_1(t) = \log(t+2)$, $\psi_2(t) = t$, and $\psi_3(t) = t^3$.

Model 3: Let $J = 4$, $\boldsymbol{\mu}_1 = (1, -1, 2, -3)^\top$, $\boldsymbol{\Sigma}_1^{1/2} = \operatorname{diag}(4/5, 3/5, 2/5, 1/5)$, $\boldsymbol{\mu}_2 = (1/2, -1/2, 5/2, -5/2)^\top$, $\boldsymbol{\Sigma}_2^{1/2} = \operatorname{diag}(1, 1, 1/2, 3/10)$, $\psi_1(t) = \sin 2\pi t$, $\psi_2(t) = \log(t + 2)$, $\psi_3(t) = t$, and $\psi_4(t) = t^3$.

Model 4: Let $J = 4$, $\boldsymbol{\mu}_1 = (6, -6, 12, -18)^\top$, $\boldsymbol{\Sigma}_1^{1/2} = \operatorname{diag}(4, 3, 2, 1)$, $\boldsymbol{\mu}_2 = (3, -3, 9, -15)^\top$, $\boldsymbol{\Sigma}_2^{1/2} = \operatorname{diag}(5, 5, 5/2, 3/2)$, $\psi_1(t) = \sin 2\pi t$, $\psi_2(t) = \log(t + 2)$, $\psi_3(t) = t$, and $\psi_4(t) = t^3$.

Note that the setting $J \leq 4$ indicates that the two classes have fewer than four different terms in the density functions. Therefore, the two populations are much "closer" to each other, which leads to relatively larger misclassification errors. This setting is more challenging than those with a relatively larger value J and many different terms. In each model above, the parameter $\boldsymbol{\theta}$ belongs to $\Theta_H(1, 1)$ or $\Theta_S(1, 1)$, that is, $\nu_1 = \nu_2 = 1$ in (3.3) or (3.4). The random

functions are sampled at M equally spaced sampling points from zero to one. We chose M from $\{10, 20, 30, 40, 50\}$ to detect how the sampling frequency affects the classification error, where we regarded $M = 50$ as the full observation. In each scenario, the number of repetitions is set to 100, and the classification errors are evaluated using 500 samples.

To select the tuning parameter for FQDA, we selected J using cross-validation, as proposed by Delaigle and Hall (2012) and Delaigle and Hall (2013), and for FDNN, we chose $L = \lceil \log M \rceil \vee \lceil \log n \rceil$, $J = c\lceil M^{1/2} \rceil \vee \lceil n^{1/2} \rceil$ for $1 \leq c \leq 4$, depending on different settings, $p_\ell = 20\lceil M^{1/2} \rceil \vee \lceil n^{1/2} \rceil$, $B = 5\lceil M^{1/4} \rceil \vee \lceil n^{1/4} \rceil$, and $s = 20\lceil M^{1/2} \rceil \vee \lceil n^{1/2} \rceil$. Note that the above selection of the architecture parameters is based on Theorem 6.

Tables 1 to 4 summarize the misclassification rates for four classifiers, given combinations of different mean and covariance models. Given the explicit definition of $X_i^{(k)}$, it is not surprising that of FQDA and FDNN significantly outperformQD and NB, which require that the two series in (2.3) are divergent. The discrepancy increases with the number of observations per subject. In particular, under the fully observed cases, the classification risks of our FQDA and FDNN classifiers are less than half of the risks generated by QD and NB. When the data are sparsely sampled ($M = 10$), all classifiers have larger misclassification risks, because there is less available information. However, the proposed FQDA and FDNN still outperform their two counterparts.

6.2. NonGaussian setting

To evaluate the performance of the proposed classifiers under nonGaussian process situations, we consider the following two models:

Model 5: Let $X_i^{(k)}(t) = \sum_{j=1}^{3} \xi_{ij}^{(k)} \psi_j(t)$, where $\xi_{ij}^{(1)} \sim N(\mu_{1j}, \lambda_j^{(1)})$, for $i = 1, \ldots, n_1$, $\xi_{ij}^{(2)} \sim t_{7-2j}$, $i = 1, \ldots, n_2$, $\boldsymbol{\mu}_1 = (-1, 2, -3)^\top$, $\boldsymbol{\Sigma}_1^{1/2} = \mathrm{diag}\,(3, 2, 1)$, $\psi_1(t) = \log(t + 2)$, $\psi_2(t) = t$, and $\psi_3(t) = t^3$.

Model 6: Let $X_i^{(k)}(t) = \sum_{j=1}^{3} \xi_{ij}^{(k)} \psi_j(t)$, where $\xi_{ij}^{(1)} \sim Exp(r_j)$, for $i = 1, \ldots, n_1$, $\boldsymbol{r} = (r_1, r_2, r_3)^\intercal = (0.3, 0.8, 1.5)^\intercal$, $\xi_{ij}^{(2)} \sim t_{7-2j}$, for $i = 1, \ldots, n_2$, $\psi_1(t) = \log(t + 2)$, $\psi_2(t) = t$, and $\psi_3(t) = t^3$.

It is easy to see that $\boldsymbol{\theta}$ in Models 5 and 6 also belong to $\Theta_H(1, 1)$ or $\Theta_S(1, 1)$. We select the tuning parameters for FQDA and FDNN in the same way as in Section 6.1. Tables 5 and 6 report the misclassification rates for the four classifiers when the functional data of one of the classes are nonGaussian. Because the three competitors are designed only for the Gaussian process, FDNN dominates in terms of performance for both sparsely and densely sampled functional data cases. In most scenarios, the misclassification rates of FDNN are approximately one-third of those of QD and NB. FQDA incurred larger risks than FDNN in both cases, but is still superior to QD and NB.

Table 1. Misclassification rates (%), with standard errors in parentheses, for Model 1.

M	n	FQDA	FDNN	QD	NB
50	50	18.75(0.02)	19.46(0.08)	39.15(0.02)	42.09(0.02)
	100	18.54(0.01)	16.86(0.09)	38.53(0.02)	40.96(0.02)
40	50	19.97(0.02)	19.91(0.08)	39.12(0.02)	42.10(0.02)
	100	19.85(0.02)	18.58(0.10)	38.49(0.02)	40.91(0.02)
30	50	22.17(0.02)	24.82(0.12)	39.14(0.02)	42.04(0.02)
	100	22.00(0.02)	18.70(0.10)	38.48(0.02)	40.87(0.02)
20	50	25.99(0.02)	26.04(0.12)	39.00(0.02)	41.97(0.02)
	100	26.04(0.02)	24.27(0.01)	38.47(0.02)	40.75(0.05)
10	50	32.10(0.02)	28.59(0.10)	38.98(0.02)	41.79(0.02)
	100	31.91(0.02)	25.24(0.09)	38.28(0.02)	40.70(0.02)

Table 2. Misclassification rates (%), with standard errors in parentheses, for Model 2.

M	n	FQDA	FDNN	QD	NB
50	50	14.77(0.02)	18.82(0.10)	37.91(0.02)	41.03(0.02)
	100	14.58(0.01)	13.19(0.10)	37.35(0.02)	39.92(0.02)
40	50	15.99(0.02)	18.52(0.10)	37.85(0.02)	40.99(0.02)
	100	15.92(0.01)	12.92(0.02)	37.32(0.02)	40.07(0.02)
30	50	18.29(0.02)	21.71(0.12)	37.86(0.02)	40.89(0.02)
	100	18.37(0.02)	12.95(0.09)	37.33(0.02)	39.91(0.02)
20	50	22.27(0.02)	24.01(0.14)	37.83(0.02)	40.90(0.02)
	100	22.39(0.02)	21.70(0.11)	37.28(0.02)	39.81(0.02)
10	50	29.12(0.02)	27.74(0.13)	37.66(0.02)	40.72 (0.02)
	100	29.16(0.02)	27.33(0.12)	37.18(0.02)	39.57(0.02)

Table 3. Misclassification rates (%), with standard errors in parentheses, for Model 3.

M	n	FQDA	FDNN	QD	NB
50	50	18.63(0.02)	20.02(0.04)	34.95(0.03)	40.26(0.03)
	100	18.06(0.02)	19.96(0.06)	34.69(0.02)	38.89(0.02)
40	50	19.85(0.02)	22.46(0.07)	34.96(0.03)	40.41(0.03)
	100	19.31(0.02)	19.34(0.09)	34.67(0.02)	38.95(0.02)
30	50	21.79(0.02)	24.35(0.07)	34.96(0.03)	40.42(0.03)
	100	21.33(0.02)	20.05(0.08)	34.70(0.02)	39.05(0.02)
20	50	25.36(0.02)	26.07(0.09)	34.92(0.03)	40.42(0.03)
	100	24.16(0.02)	21.22(0.08)	34.60(0.02)	38.98(0.03)
10	50	30.25(0.02)	26.03(0.08)	34.72(0.03)	40.35(0.03)
	100	30.00(0.02)	24.13(0.08)	34.15(0.03)	38.83(0.09)

7. Real-Data Illustrations

This benchmark data example was extracted from the TIMIT database (TIMIT Acoustic-Phonetic Continuous Speech Corpus, NTIS, US Dept of

Table 4. Misclassification rates (%), with standard errors in parentheses, for Model 4.

M	n	FQDA	FDNN	QD	NB
50	50	14.56(0.02)	21.16(0.10)	32.76(0.02)	38.76(0.03)
	100	14.26(0.02)	16.85(0.10)	32.64(0.02)	36.77(0.03)
40	50	15.89(0.02)	20.42(0.10)	32.78(0.02)	38.65(0.03)
	100	19.31(0.02)	20.18(0.09)	34.67(0.02)	38.95(0.02)
30	50	18.26(0.02)	22.75(0.10)	32.72(0.02)	38.58(0.03)
	100	17.81(0.02)	16.29(0.10)	32.60(0.02)	36.74(0.03)
20	50	21.93(0.02)	22.76(0.11)	32.72(0.03)	38.36(0.03)
	100	21.54(0.02)	21.29(0.09)	32.59(0.02)	36.88(0.03)
10	50	27.46(0.02)	27.73(0.10)	32.52(0.02)	38.78(0.03)
	100	27.08(0.02)	24.85(0.10)	32.34(0.02)	37.00(0.02)

Table 5. Misclassification rates (%), with standard errors in parentheses, for Model 5.

M	n	FQDA	FDNN	QD	NB
50	50	18.11(0.04)	13.20(0.01)	42.63(0.02)	40.27(0.03)
	100	17.11(0.04)	12.29(0.01)	38.42(0.09)	39.84(0.04)
40	50	19.47(0.04)	13.40(0.02)	42.61(0.10)	40.38(0.04)
	100	18.62(0.04)	12.35(0.01)	38.38(0.09)	39.79(0.04)
30	50	22.14(0.05)	12.89(0.01)	42.73(0.01)	40.50(0.03)
	100	24.19(0.05)	12.21(0.01)	38.30(0.09)	40.11(0.04)
20	50	27.00(0.08)	13.00(0.01)	42.77(0.10)	40.69(0.04)
	100	22.75(0.07)	12.21(0.01)	38.17(0.09)	40.26(0.04)
10	50	36.75(0.08)	23.01(0.16)	43.16(0.04)	41.38(0.04)
	100	32.14(0.09)	19.52(0.15)	37.87(0.09)	40.90(0.04)

Commerce), which is a widely used resource for research in speech recognition and functional data classification (Ferraty and Vieu (2003)). Our data set is constructed by selecting five phonemes for classification based on digitized speech from this database. From each speech frame, a log-periodogram transformation is applied to cast the speech data in a form suitable for speech recognition. The five phonemes in this data set are as follows: "sh," as in "she," "dcl," as in "dark," "iy," as the vowel in "she," "aa," as the vowel in "dark," and "ao," as the first vowel in "water." For illustration purposes, we focus on the "aa," "ao," "iy," and "dcl" phoneme classes. Each speech frame is represented by $n = 400$ samples at a 16 kHz sampling rate; the first $M = 150$ frequencies from each subject are retained. Figure 1 displays 10 log-periodograms for each class phoneme.

We randomly select training sample size $n_1 = n_2 = 100$ to train the classifiers of the three methods, and the rest of the 300 samples remain as test samples. The tuning parameter selections for FQDA and FDNN are the same as those in Section 6.1. Table 7 reports the mean percentage (averaged over 100 repetitions) of misclassified test curves. Both FQDA and FDNN outperform QD and NB in

Table 6. Misclassification rates (%), with standard errors in parentheses, for Model 6.

M	n	FQDA	FDNN	QD	NB
50	50	13.38(0.08)	8.98(0.01)	20.54(0.09)	19.81(0.08)
	100	10.11(0.02)	8.31(0.01)	15.86(0.03)	17.07(0.06)
40	50	13.72(0.08)	9.45(0.01)	19.36(0.08)	19.25(0.08)
	100	12.12(0.06)	8.54(0.01)	16.98(0.05)	16.13(0.06)
30	50	13.94(0.08)	10.57(0.07)	19.35(0.08)	19.78(0.09)
	100	12.82(0.06)	8.92(0.04)	17.00(0.05)	16.69(0.04)
20	50	15.33(0.09)	10.52(0.04)	19.93(0.09)	20.32(0.10)
	100	13.91(0.06)	8.97(0.04)	17.00(0.06)	17.72(0.08)
10	50	15.58(0.07)	12.07(0.08)	19.33(0.08)	23.04(0.12)
	100	15.04(0.05)	8.90(0.01)	17.16(0.06)	20.71(0.10)

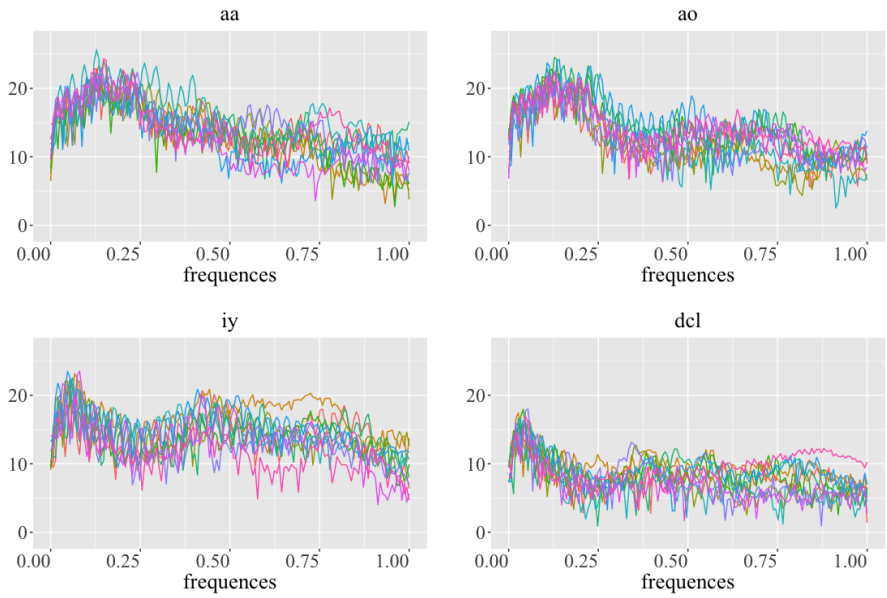

Figure 1. A sample of 10 log-periodograms per class.

all three classification tasks. For "ao" versus "iy," the misclassification rates of FQDA and FDNN are less than one-third of that of QD; for "ao" versus "dcl," the misclassification rates of FQDA and FDNN are around half that of NB. The most difficult task is to distinguish between "aa" and "ao" and all three classifiers have much larger risks. However, the proposed FQDA and FDNN classifiers still provide smaller risks and smaller standard errors compared with those of QD and NB classifiers.

Table 7. Misclassification rates (%), with standard errors in parentheses, for the speech recognition data.

Classes	FQDA	FDNN	QD	NB
"aa" vs "ao"	20.278(0.014)	20.744(0.016)	25.402(0.026)	25.378(0.021)
"aa" vs "iy"	0.196(0.001)	0.193(0.002)	0.288(0.005)	0.273(0.006)
"ao" vs "iy"	0.153(0.004)	0.183(0.004)	0.578(0.005)	0.232(0.005)
"ao" vs "dcl"	0.270(0.003)	0.229(0.002)	0.391(0.005)	0.472(0.006)

8. Conclusion

We present a new minimax optimality viewpoint for solving functional classification problems. In comparison with methods in the existing literature, our results deal with the more practical scenarios where the two populations are relatively "close," so that the optimal Bayes risk is asymptotically nonvanishing. Our contributions are threefold. First, we provide sharp convergence rates for MEMR when the data are either fully or discretely observed, as well as a critical sampling frequency that governs the rate in the latter case. Second, we propose novel classifiers based on FQDA and FDNN that we prove to achieve minimax optimality. Third, we use simulations and real-data examples to show that the proposed FDNN classifier exhibits outstanding performance, even when the Gaussian assumption is invalid.

Supplementary Material

Technical lemmas and proofs of Theorems 1 to 6 are provided in the online Supplementary Material.

Acknowkedgments

We thank the associate editor and two referees for their helpful and constructive comments. Wang's and Cao's research was partially supported by NSF award DMS 1736470. Cao's research was also partially supported by the Simons Foundation under Grant #849413. Shang's research was supported, in part, by NSF DMS 1764280 and 1821157. Jun S. Liu acknowledges NSF DMS grant 2015411.

References

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. 3rd Edition. Wiley-Intersceince, New York.

Araki, Y., Konishi, S., Kawano, S. and Matsui, H. (2009). Functional logistic discrimination via regularized basis expansions. *Communications in Statistics–Theory and Methods* **38**, 2944–2957.

Bauer, B. and Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality

in nonparametric regression. *The Annals of Statistics* **47**, 2261–2285.

Berrendero, J. R., Cuevas, A. and Torrecilla, J. L. (2018). On the use of reproducing kernel Hilbert spaces in functional classification. *Journal of the American Statistical Association* **113**, 1210–1218.

Cai, T. T. and Yuan, M. (2011). Optimal estimation of the mean function based on discretely sampled functional data: Phase transition. *The Annals of Statistics* **39**, 2330–2355.

Cai, T. T. and Yuan, M. (2012). Minimax and adaptive prediction for functional linear regression. *Journal of the American Statistical Association* **107**, 1201–1216.

Cai, T. T. and Zhang, L. (2019a). A convex optimization approach to high-dimensional sparse quadratic discriminant analysis. *arXiv:1912.02872*.

Cai, T. T. and Zhang, L. (2019b). High dimensional linear discriminant analysis: Optimality, adaptive algorithm and missing data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **81**, 675–705.

Dai, X., Müller, H.-G. and Yao, F. (2017). Optimal Bayes classifiers for functional data and density ratios. *Biometrika* **104**, 545–560.

Delaigle, A. and Hall, P. (2012). Achieving near-perfect classification for functional data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **74**, 267–286.

Delaigle, A. and Hall, P. (2013). Classification using censored functional data. *Journal of the American Statistical Association* **108**, 1269–1283.

Delaigle, A., Hall, P. and Bathia, N. (2012). Componentwise classification and clustering of functional data. *Biometrika* **99**, 299–313.

Farnia, F. and Tse, D. (2016). A minimax approach to supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* NIPS'16, 4240–4248.

Ferraty, F. and Vieu, P. (2003). Curves discrimination: A nonparametric functional approach. *Computational Statistics & Data Analysis* **44**, 161–173.

Galeano, P., Joseph, E. and Lillo, R. E. (2015). The Mahalanobis distance for functional data with applications to classification. *Technometrics* **57**, 281–291.

Hilgert, N., Mas, A. and Verzelen, N. (2013). Minimax adaptive tests for the functional linear model. *The Annals of Statistics* **41**, 838–869.

Hu, T., Shang, Z. and Cheng, G. (2020). Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. *arXiv:2001.06892*.

Ian, G., Yoshua, B. and Aaron, C. (2016). *Deep Learning*. MIT Press.

Kim, Y., Ohn, I. and Kim, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Networks* **138**, 179–197.

Lecué, G. (2008). Classification with minimax fast rates for classes of Bayes rules with sparse representation. *Electronic Journal of Statistics* **2**, 741–773.

Liu, R., Boukai, B. and Shang, Z. (2022). Optimal nonparametric inference via deep neural network. *Journal of Mathematical Analysis and Applications* **505**, 125561.

Liu, R., Shang, Z. and Cheng, G. (2021). On deep instrumental variables estimate. *arXiv:2004.14954*.

Mammen, E. and Tsybakov, A. B. (1999). Smooth discrimination analysis. *The Annals of Statistics* **27**, 1808–1829.

Mazuelas, S., Zanoni, A. and Perez, A. (2020). Minimax classification with 0-1 loss and performance guarantees. *arXiv:2010.07964*.

Park, J., Ahn, J. and Jeon, Y. (2020). Sparse functional linear discriminant analysis. *arXiv:2012.06488*.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics* **48**, 1875–1897.

Shang, Z. and Cheng, G. (2015). Nonparametric inference in generalized functional linear models. *The Annals of Statistics* **43**, 1742–1773.

Shin, H. (2008). An extension of fisher's discriminant analysis for stochastic processes. *Journal of Multivariate Analysis* **99**, 1191–1216.

Torrecilla, J. L., Ramos-Carreno, C., Sanchez-Montanes, M. and Alberto, S. (2020). Optimal classification of Gaussian processes in homo- and heteroscedastic settings. *Statistics and Computing* **30**, 1091–1111.

Tsybakov, A. B. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics* **32**, 135–166.

Wang, J., Chiou, J. M. and Müller, H. G. (2016). Functional data analysis. *Annual Review of Statistics and Its Application* **3**, 257–295.

Wang, S., Cao, G. and Shang, Z. (2021). Estimation of the mean function of functional data via deep neural networks. *Stat* **10**, e393.

Shuoyang Wang

Department of Bioinformatics and Biostatistics, University of Louisville, Louisville, KY 40202, USA.

E-mail: shuoyang.wang@louisville.edu

Zuofeng Shang

Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA.

E-mail: zshang@njit.edu

Guanqun Cao

Department of Statistics and Probability, Michigan State University, East Lansing, MI 48824, USA.

E-mail: caoguanq@msu.edu

Jun S. Liu

Department of Statistics, Harvard University, Cambridge, MA 02138-2901, USA.

E-mail: jliu@stat.harvard.edu

(Received February 2022; accepted November 2022)