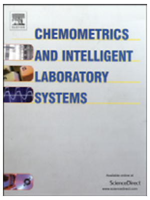
Chemometrics and Intelligent Laboratory Systems 146 (2015) 186–197

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab



Nearest neighbor ensembles for functional data with interpretable feature selection☆

Karen Fuchs [a,c,*], Jan Gertheiss [b], Gerhard Tutz [c]

[a] Siemens AG, CT RTC SET CPS-DE, Otto-Hahn-Ring 6, D-81739 Munich, Germany
[b] Department of Animal Sciences, Biometrics and Bioinformatics Group, Georg-August-University Göttingen, Carl-Sprengel-Weg 1, D-37075 Göttingen, Germany
[c] Department of Statistics, Seminar of Applied Stochastics, Ludwig-Maximilians-University Munich, Akademiestr. 1, D-80799 Munich, Germany

ARTICLE INFO

Article history:
Received 13 January 2015
Received in revised form 14 April 2015
Accepted 20 April 2015
Available online 2 May 2015

Keywords:
Functional classification
Ensemble method
Variable selection
Functional data analysis
Nearest neighbor method
Cell chip data

ABSTRACT

Functional data becomes increasingly common in many fields of application. Although much research has been done on functional regression and clustering approaches for chemometric data, so far few classification methods exist. This paper introduces an ensemble method for classification that inherently provides automatic and interpretable feature selection. It is designed for single as well as multiple functional (and non-functional) covariates. The ensemble members are posterior probability estimates that are based on a *k*-nearest-neighbor approach. The ensemble allows for feature selection by including members that are calculated from various semi-metrics used in the *k*-nearest-neighbor approach, where a particular semi-metric represents a specific curve feature. Each ensemble member, and thus each curve feature, is weighted by an unknown coefficient. These coefficients are estimated using a proper scoring rule with implicit Lasso-type penalty, such that some coefficients can be estimated to be exactly zero. Thus, the ensemble automatically provides feature selection, and also, in the case of multiple functional (and non-functional) covariates, variable selection. The selection performance and the interpretability of the coefficients are investigated in simulation studies. Data of a cell chip used for water quality monitoring experiments is examined. Here, the relevance of especially the feature selection aspect of the ensemble is illustrated.

1. Introduction

Nearest neighbors are discrimination techniques belonging to the group of supervised learning methods. Originally proposed by Fix and Hodges [1], they have become popular in chemometrics [2–7]. Nearest neighbor methods have also been used in other fields of application (see for example Refs. [8–11]). Nearest neighbor approaches are non-parametric and memory based (see also Hastie et al. [12]). For multivariate data, the basic principle of *k*-nearest-neighbors is as follows:

Let a learning sample be given by $(y_i, \boldsymbol{x}_i)$, $i = 1,\ldots N$, where $\boldsymbol{x}_i = (x_{i1},\ldots x_{iq})^T$ is a vector of predictors and $y_i \in \{1,\ldots G\}$ denotes the class membership of observation *i*. Moreover, let $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denote a distance measure in the feature space. For a new observation $\boldsymbol{x}^* = (x_1^*, \ldots, x_q^*)^T$, one determines the *k* observations which are closest to $\boldsymbol{x}^*$. This means one seeks the *k*-nearest-neighbors of $\boldsymbol{x}^*$, denoted by $\boldsymbol{x}_{(1)},\ldots\boldsymbol{x}_{(k)}$, with nearness defined by the distance measure $d(\cdot,\cdot)$. The *k* nearest neighbors fulfill

$$d\left(\boldsymbol{x}^*, \boldsymbol{x}_{(1)}\right) \leq \ldots \leq d\left(\boldsymbol{x}^*, \boldsymbol{x}_{(k)}\right).$$

Let $y_{(i)}$ denote the observed class linked to the neighbor $\boldsymbol{x}_{(i)}$. For the assignment of $\boldsymbol{x}^*$ to a class y^*, one uses the majority rule,

$$y^* = g \Leftrightarrow g \text{ is most frequent in observations } \left\{y_{(1)}, \ldots y_{(k)}\right\}.$$

The resulting classifier is called the *k*-nearest-neighbor classifier.

The basic *k*-nearest-neighbor approach can be modified in various ways. For example, Gertheiss and Tutz [13] extended it to an ensemble of weighted nearest neighbor posterior probability estimates. Further extensions can be found in Ji and Zhao [14] and Hayat et al. [15]; see also Bischl et al. [16] for a comparison of the *k*-nearest-neighbor approach with other local discrimination techniques.

In the present paper we introduce a *k*-nearest-neighbor classification ensemble for functional data, which are infinite dimensional in theory, and high dimensional in practice. For an introduction to functional data see the monograph by Ramsay and Silverman [17]. In functional data analysis, the predictors that are used for classification are curves $x_i(t)$, with *t* from a domain $\mathbb{D}$, typically an interval from $\mathbb{R}$. The general ensemble methodology used here is related to "model stacking" and "super learning" [18–20]. One of the first approaches setting up such an ensemble for functional data is the method by Goldsmith and Scheipl [21]. Here, scalar-on-function regression is done by building an ensembler, for example a linear model, with the fits of many candidate

☆ Please see the online appendix for supplementary material.

* Corresponding author. Tel.: +49 89 636 40709.
E-mail addresses: karenfuchs@gmx.de (K. Fuchs), jgerthe@uni-goettingen.de (J. Gertheiss), gerhard.tutz@stat.uni-muenchen.de (G. Tutz).

http://dx.doi.org/10.1016/j.chemolab.2015.04.019
0169-7439/© 2015 Elsevier B.V. All rights reserved.

estimators constituting the ensemble members. In this paper, we will not deal with regression but classification problems.

The data motivating our approach are measurements of cell based sensor chips, which are promising tools for environmental monitoring, such as water quality monitoring. The chip used in this study is covered with a monolayer of a living cell population, and the signals of three kinds of sensors, ion-sensitive field-effect transistors (ISFET), interdigital electrode structures (IDES) and oxygen sensitive (CLARK) electrodes, are measured concurrently over time. The sensors record different cell reactions which are related to the cells' metabolism. Fig. 1 depicts the measurements. In the experiment two classes are considered, one class with 2.5 mM paracetamol (short: AAP) applied on the cells, and one class without the use of paracetamol.

One approach to two-class discrimination problems with functional covariates is to use a logistic functional model [22]. Such generalized functional linear models use the whole observation, i.e., the curve $x_i(t)$ across the entire domain $\mathbb{D}$. In many applications, however, it is reasonable to assume that only parts of the signal contain discriminative information. For example, biological considerations concerning the cell chip data suggest that especially the range around 220 min is of importance (see also Section 4).

The k-nearest-neighbor ensemble approach presented here is especially designed to perform automated and interpretable feature selection on functional covariates. Ferraty and Vieu [23] showed that the concept of nearness in functional data analysis is adequately met by so-called semi-metrics in the space of the functional predictors. Alonso et al. [24], for example, use a pre-defined semi-metric on derivatives of functional observations to generate multivariate data points that are then used as input in a classification algorithm. The idea of our nearest neighbor ensemble is not to use a single semi-metric, but a set of semi-metrics, where each semi-metric focuses on a certain feature of the curve. For example, we use a semi-metric that focuses on the absolute distance of two curves on a limited range $\mathbb{D}_{small} \subset \mathbb{D}$ of domain $\mathbb{D}$, or one that focuses on jump heights at specific points from $\mathbb{D}$. The basic concept is to select from the set of potential semi-metrics the best ones and combine them in a smart and data-driven way: By assigning weights to the members of the ensemble, information on the discriminative power of different semi-metrics is obtained. The estimated weights reflect which signal parts, or which forms of data preprocessing are most relevant for discrimination. Thus, the resulting k-nearest-neighbor ensemble allows for an automated and interpretable selection of curve features.

The rest of the paper is organized as follows. In Section 2, the semi-metrics and the functional k-nearest-neighbor ensemble are introduced in detail. In Section 3, our approach is evaluated by means of simulation studies and compared to alternative classification methods. All classification approaches are applied to cell chip data in Section 4. The paper ends with a discussion of further developments. In an online supplement, we provide the cell chip data as well as code reproducing our results.

2. Construction of functional nearest neighbor ensembles

2.1. Distance measures

Ferraty and Vieu [23] postulate that a semi-metric d on space $\mathcal{F}$ fulfills $d(a,a) = 0 \wedge d(a,b) \leq d(a,e) + \mathrm{d}(e,b)\ \forall a,b,e \in \mathcal{F}$. They point out that semi-metrics, if chosen appropriately, may override the curse of dimensionality by taking functional features of the functional observations into account. We put an additional constraint on our semi-metrics: they should also fulfill $d(a,b) = d(b,a)\ \forall a,b \in \mathcal{F}$. This ensures that the similarity of two curves $x_i(t)$ and $x_j(t)$ is based on curve characteristics and ignores, for example, the orientation of curve shifts, i.e., whether curve $x_i(t)$ lies above or beneath a curve $x_j(t)$ with identical shape. An important difference between metrics and semi-metrics lies in the implications of a distance $d = 0$. While $d(a,a) = 0$ holds for semi-metrics as well as for metrics, the property $d(a,b) = 0 \Leftrightarrow a \equiv b$ of a metric space does not necessarily hold for semi-metrics, such that $d(a,b) = 0$ can occur for $a \neq b$. In principle, every distance measure operating on curves $x_i(t)$ and fulfilling the above equations is allowed in our approach. Nonetheless, the semi-metrics we will consider are supposed to account for specific characteristics of the functional covariates.

In what follows, let $x_i^{(a)}(t)$ denote the ath order differentiation of $x_i(t)$. We restrict the set of semi-metrics we use to semi-metrics that focus on specific curve characteristics. For example, a measure that focuses on the curve distances is the Euclidian distance

$$d_a^{Eucl}(x_i(t), x_j(t)) = \sqrt{\int_{\mathbb{D}} \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)^2 dt}.$$

It represents the absolute distance of two curves, or their derivatives, which might contain information concerning the class, for example, if the curves have similar shapes within classes. Instead of such a "static" semi-metric, especially appealing distance measures are adaptive ones that locate points or regions of discriminative power. An example for such a more sophisticated semi-metric is

$$d_{a\tau}^{Scan}(x_i(t), x_j(t)) = \sqrt{\int_{\mathbb{D}} \left(\phi_\tau(t)\left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)\right)^2 dt},$$

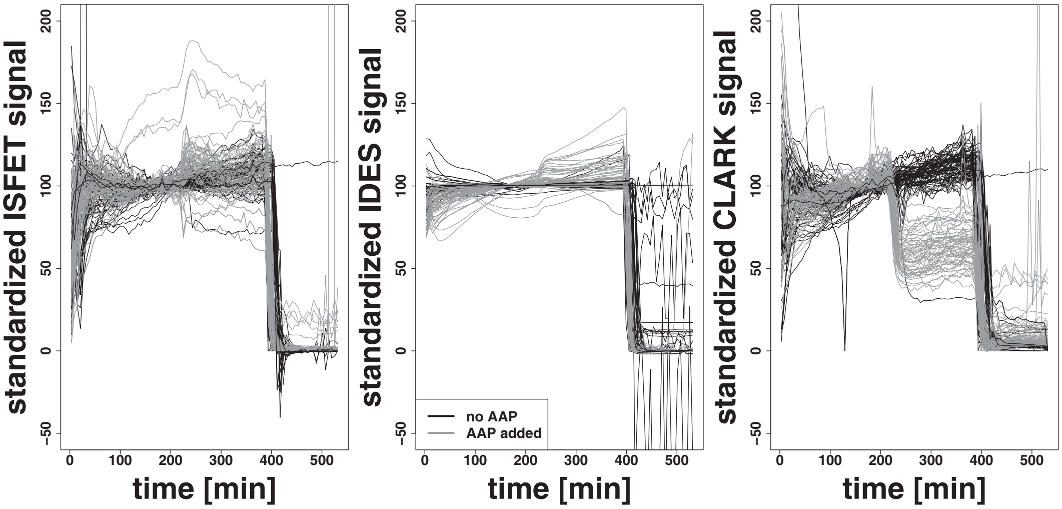


Fig. 1. The $N = 120$ standardized signals for each of the three sensor types, measured at 89 time points on an equidistant grid. The gray shades represent the presence (gray lines) or absence (black lines) of AAP.

with $\tau \in \mathbb{D}$, where the function $\phi_\tau(t)$ is an appropriate scan function, for example a Gaussian kernel

$$\phi_\tau(t) = \frac{1}{\sqrt{2\pi}} exp^{-\frac{(t-\tau)^2}{2}},$$

giving a weight profile to the variable t that is centered around τ. Further semi-metrics used in our k-nearest-neighbor ensemble are given in Table 1. They were chosen because they seemed to be the most appropriate distance measures for the analysis of the cell chip data described above. Naturally, our approach can be extended to other semi-metrics. In applications, however, it might be difficult to judge which semi-metric is most appropriate for the discrimination task. Therefore our strategy is to combine several semi-metrics in an ensemble. Which semi-metric yields most information for discrimination is reflected in ensemble weights that are estimated using the data at hand.

2.2. The functional nearest neighbor ensemble

Let $(y_i, x_i(t))$, $i = 1,\ldots N$, be a learning sample and $(y^*, x^*(t))$ a new observation with unknown class membership y^*, and let $d(\cdot,\cdot)$ denote a semi-metric. Then the observations are ordered such that

$$d\left(x^*(t), x_{(1)}(t)\right) \le \ldots d\left(x^*(t), x_{(k)}(t)\right) \le \ldots d\left(x^*(t), x_{(N)}(t)\right), \tag{1}$$

with the $x_{(1)}(t),\ldots x_{(N)}(t)$ being observations from the learning sample. Using (1), we define the neighborhood $\mathcal{N}(x^*(t))$ of the k nearest neighbors of $x^*(t)$,

$$\mathcal{N}(x^*(t)) = \left\{x_j(t) : d\left(x^*(t), x_j(t)\right) \le d\left(x^*(t), x_{(k)}(t)\right)\right\}. \tag{2}$$

With $I(\cdot)$ denoting the indicator function, the estimated probability $\hat{\pi}_g$ that covariate $x^*(t)$ belongs to class g is given by

$$\hat{\pi}_g = \frac{1}{k} \sum_{x_j(t) \in \mathcal{N}(x^*(t))} I\left(y_j = g\right).$$

Similar to the k-nearest-neighbor classifier for multivariate data described in the introduction, the unknown y^* is assigned to the class that is most frequent within the neighborhood $\mathcal{N}(x^*(t))$, i.e., to the class of highest probability, $y^* = \underset{g}{\text{argmax}}\,(\hat{\pi}_g)$.

This simple functional k-nearest-neighbor approach can be extended to a functional ensemble which, in its structure, is similar to the model presented in Gertheiss and Tutz [13]. If we use several semi-metrics $d_l(\cdot,\cdot)$, $l = 1,\ldots p$, instead of one specific semi-metric only, the order (1) of the observations relative to the new observation $x^*(t)$ depends on the distance measure $d_l(\cdot,\cdot)$. For distance $d_l\ (\cdot,\cdot)$, we define the neighborhood $\mathcal{N}_l(x^*(t))$ of the k nearest neighbors of $x^*(t)$ analogously to neighborhood (2). The corresponding posterior probability estimates are denoted by $\hat{\pi}_{gl}$ and given by

Table 1
Semi-metrics used in the k-nearest-neighbor ensemble.

Semi-metric	Takes into account…
$d_{a,\mathbb{D}_{small}}^{shortEucl}(x_i(t), x_j(t)) = \sqrt{\int_{\mathbb{D}_{small}} \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)^2 dt}$	… the absolute distance on a limited part of the domain of definition $\mathbb{D}_{small} \subset \mathbb{D}$ of two curves (or their derivatives).
$d_a^{Mean}(x_i(t), x_j(t)) = \left\|\int_{\mathbb{D}} x_i^{(a)}(t)dt - \int_{\mathbb{D}} x_j^{(a)}(t)dt\right\|$	… the similarity of mean values of the whole curves (or their derivatives).
$d_a^{relAreas}(x_i(t), x_j(t)) = \left\| \left\|\frac{\int_{\mathbb{D}_1} x_i^{(a)}(t)dt}{\int_{\mathbb{D}_2} x_i^{(a)}(t)dt}\right\| - \left\|\frac{\int_{\mathbb{D}_1} x_j^{(a)}(t)dt}{\int_{\mathbb{D}_2} x_j^{(a)}(t)dt}\right\| \right\|$	… the similarity of the relation of areas on parts of the domain of definition $\mathbb{D}_1$, $\mathbb{D}_2 \subset \mathbb{D}$.
$d_{no}^{Jump}(x_i(t), x_j(t)) = \left\|(x_i(t_n) - x_i(t_o)) - (x_j(t_n) - x_j(t_o))\right\|$	… the similarity of jump heights at points t_n, $t_o \in \mathbb{D}$.
$d_a^{\text{Max}}(x_i(t), x_j(t)) = \left\| \max\left(x_i^{(a)}(t)\right) - \max\left(x_j^{(a)}(t)\right) \right\|$	… the difference of the curves' (or their derivatives') global maxima.
$d_a^{Min}(x_i(t), x_j(t)) = \left\| \min\left(x_i^{(a)}(t)\right) - \min\left(x_j^{(a)} t\right) \right\|$	… the difference of the curves' (or their derivatives') global minima.
$d_a^{Points}(x_i(t), x_j(t)) = \frac{1}{S}\sum_{q=1}^{S} \left\|x_i^{(a)}(t) - x_j^{(a)}(t)\right\|_{t=t_q}$	… the differences at certain observation points (also called "points of impact").

$$\hat{\pi}_{gl} = \frac{1}{k} \sum_{x_j(t) \in \mathcal{N}_l(x^*(t))} I\left(y_j = g\right).$$

The overall posterior probability estimate $\hat{\pi}_g$ that function $x^*(t)$ is from class g is set up as an ensemble

$$\hat{\pi}_g = \sum_{l=1}^{p} c_l \hat{\pi}_{gl} \tag{3}$$

$$\text{with } \ c_l \ge 0\ \forall\, l,\ \sum_{l=1}^{p} c_l = 1, \tag{4}$$

where the coefficients c_l are unknown and have to be estimated. The constraint (4) not only yields identifiability of the coefficients (in a least-square sense [25]), but also ensures that the probability estimates $\hat{\pi}_g$ are proper probabilities in the sense that $0 \le \hat{\pi}_g \le 1\ \forall\ g$ and for all potential, or future $x^*(t)$; see Proposition (1) in Ref. [13]. The coefficients give a weight to each estimate $\hat{\pi}_{gl}$, and with that to every semi-metric $d_l(\cdot,\cdot)$. This enables the ensemble to determine which semi-metric $d_l(\cdot,\cdot)$, i.e., which curve characteristic, yields the highest contribution to $\hat{\pi}_g$ and is the most informative concerning the discrimination of the classes. The strength of the method is that feature selection is a built-in feature of the method; in contrast to, for example, functional principal component analysis (FPCA) [26,27]. As in the multivariate case, FPCA is a method to project the feature space on the eigenfunction space of the covariates' covariance matrix (see also Section 3). Another popular approach, without automatic feature selection, however, is the method by Ferraty and Vieu [23], where a fixed kernel function and a semi-metric have to be chosen by the user (see also Section 3.1).

The ensemble (3) can be extended to include further parameters. For instance, the order of derivation a and the number of nearest neighbors k have to be chosen to calculate the semi-metrics and with that the posterior probability estimates $\hat{\pi}_{gl}$. We include the order a of the derivative of the covariates in the ensemble (3) by using it inherently in the semi-metrics, such that $d(\cdot,\cdot) = d(x^{*(a)}(t), x_j^{(a)}(t))$. Moreover, the number of nearest neighbors k is no longer assumed to be fixed, but to be from a given set of M numbers of nearest neighbors, $k \in \mathcal{K} = \{k_1, \ldots k_M\}$. This means that the index l now represents a tuple $\{d(\cdot,\cdot), a, k\}$, with $d(\cdot,\cdot)$ denoting the distance measure, a denoting the order of the derivative and k denoting the number of nearest neighbors used. The corresponding neighborhood is denoted by $\mathcal{N}_{(l)}\left(x^{*(a)}(t)\right)$, which is used to calculate the single posterior probability estimate $\hat{\pi}_{g(l)}$. Ensemble (3) thus extends to an ensemble including $l = 1, \ldots, p$ ensemble members, each one characterized by an unique tuple $\{d(\cdot,\cdot), a, k\}$. By assigning weights c_l to every ensemble member, the relevance of a combination of $d(\cdot,\cdot)$, a and k is automatically determined. The weighting of the single k's from the set $\mathcal{K}$ is another important advantage of our ensemble, since only few techniques exist for determining an optimal choice of k, see for example Hall et al. [28] for the case of multivariate data.

Since one can choose the semi-metrics that are used in the k-nearest-neighbor ensemble, the ensemble can also be applied to functional covariates that are not square-integrable simply by adapting the semi-metrics. Also, the approach is quite robust against single outliers because it focuses on various curve characteristics.

2.3. Estimation of weights

The weights c_l can be estimated from the learning sample by minimizing the global Brier score [29]

$$Q = \sum_{i=1}^{N}\sum_{g=1}^{G}\left(z_{ig} - \hat{\pi}_{ig}\right)^2, \tag{5}$$

where $z_{ig} = 1$ if $y_i = g$ and $z_{ig} = 0$ otherwise codes the response. The Brier score is a strictly proper scoring rule [30], and the only one that (up to a positive linear transformation) fulfills the properties Selten [31] demands of scoring rules. Among others, one advantage of the Brier score over other measures as, for example, the logarithmic score, is that it is neither hypersensitive nor insensitive. Not being hypersensitive means that the score does not react strongly on small differences between small probabilities, especially probabilities of value (around) zero. Not being insensitive means that the expected score loss $\sum_{g=1}^{G}\left(\pi_{ig} - \hat{\pi}_{ig}\right)^2$ corresponding to the Brier score adequately reflects the difference between the underlying true and the predicted distribution of the probabilities [31].

2.4. Estimation in practice

The global Brier score (5) is interpreted as a function of the coefficient vector $\mathbf{c} = (c_1, \ldots c_p)^T$ of the coefficients c_l,

$$Q(\boldsymbol{c}) = \Bigl(\underset{NG\times 1}{\boldsymbol{z}} - \underset{NG\times \mathrm{p}}{\boldsymbol{P}}\underset{\mathrm{p}\times 1}{\boldsymbol{c}}\Bigr)^T\Bigl(\underset{NG\times 1}{\boldsymbol{z}} - \underset{NG\times \mathrm{p}}{\boldsymbol{P}}\underset{\mathrm{p}\times 1}{\boldsymbol{c}}\Bigr), \tag{6}$$

with vector $\boldsymbol{z} = (\boldsymbol{z}_1|\ldots|\boldsymbol{z}_N)^T, \boldsymbol{z}_i = (z_{i1}, \ldots z_{iG})^T, i = 1, \ldots N, g = 1, \ldots G$, and matrix $\boldsymbol{P} = (\boldsymbol{P}_1^T| \ldots |\boldsymbol{P}_N^T)^T$, where

$$\boldsymbol{P}_i = \begin{pmatrix} \hat{\pi}_{i1(1)} & \cdots & \hat{\pi}_{i1(p)} \\ \vdots & \cdots & \vdots \\ \hat{\pi}_{iG(1)} & \cdots & \hat{\pi}_{iG(p)} \end{pmatrix}$$

merges the estimates of the single posterior probabilities $\hat{\pi}_{ig(l)}$, with classes g per row, and all combinations of semi-metrics, orders of derivation of the covariates, and numbers of nearest neighbors per column. Here, the single posterior probabilities $\hat{\pi}_{ig(l)}$ are estimated via leave-one-out cross-validation for each $x_i(t)$ from the learning sample (as otherwise the nearest neighbor of observation i would always be observation i itself). Alternatively, other procedures such as K-fold cross-validation could be used.

Minimizing Eq. (6) with respect to the coefficients c_l by $\min_c (Q(\boldsymbol{c}))$ yields a way of estimating the coefficients in terms of a quadratic programming problem. By employing the constraints (4) on the coefficients, the estimation procedure implicitly uses a (positive) Lasso-type penalty (see e.g. Tibshirani [32]), which typically sets some coefficients c_l to be exactly zero and thus enables feature selection. For solving the quadratic programming problem, we use the lsei-function of the R package limSolve [25,33].

2.5. The functional nearest neighbor ensemble including multiple covariates

If V functional covariates $x_v(t)$, $v = 1,\ldots V$, instead of a single functional covariate $x(t)$ are available, two problems have to be addressed. First, the covariates might be defined on different scales, or represent totally different situations, as for example measurements on a time and a spatial scale. Concerning the k-nearest-neighbor ensemble, this means that the semi-metrics possibly differ in their adequacy concerning different covariates. Second, the content of information of the covariates might differ, and with that their individual importance for the discrimination task. All this, however, is easily accounted for when using our nearest neighbor ensemble.

Let $x_{iv}(t)$ denote the ith observation of covariate $x_v(t)$, and $x_v^*(t)$ a new observation of that covariate. Further, let $d_v(\cdot,\cdot)$ denote semi-metrics that are used on covariate $x_v(t)$. Again, each tuple $\{d_v(\cdot,\cdot), a, k\}$ is represented by the index l. There are now V neighborhoods denoted by $\mathcal{N}_{v(l)}\left(x_v^{*(a)}(t)\right)$, and defined in the same way as before. The posterior probability estimate $\hat{\pi}_{gv(l)}$ that covariate $x_v^*(t)$ belongs to class g when considering $k \in \mathcal{K}$ nearest neighbors, and semi-metric $d_\mathrm{v}(\cdot,\cdot)$ defined on the derivative of order a of the covariates, is given by

$$\hat{\pi}_{gv(l)} = \frac{1}{k}\sum_{x_{jv}^{(a)}(t)\in\mathcal{N}_{v(l)}\left(x_v^{*(a)}(t)\right)} I\left(y_{vj} = g\right).$$

Analogously to the univariate ensemble, the overall posterior probability estimate $\hat{\pi}_g$ that $y^* = g$ is set up as an ensemble

$$\begin{aligned} &\hat{\pi}_g = \sum_{v=1}^{V}\sum_{l=1}^{p} c_{vl}\hat{\pi}_{gv(l)} \\ &\text{with } c_{vl} \geq 0 \forall v, l, \sum_{v=1}^{\mathrm{V}}\sum_{l=1}^{\mathrm{p}} c_{vl} = 1. \end{aligned} \tag{7}$$

With the assignment of a coefficient c_{vl} per covariate type $x_v(t)$, our functional k-nearest-neighbor ensemble permits not only for feature selection, but additionally allows for variable selection from the V covariates. Moreover, we may include and exclude additional non-functional covariates: simply by defining appropriate distance measures on the corresponding predictors' space and including the resulting posterior probability estimates in the ensemble. This flexibility and general applicability is a huge advantage of our approach over existing methods for nonparametric functional discrimination. Sometimes, and depending on the data, however, the general model can be simplified, for example, by using the same set of semi-metrics for all V covariates.

The weight estimation can be performed analogously to the univariate case described in Section 2.3. For each covariate type $x_v(t)$, a matrix $\boldsymbol{P}_v = (\boldsymbol{P}_{1v}^T| \ldots | \boldsymbol{P}_{Nv}^T)^T$ is calculated, and these single matrices are merged to a final matrix $\boldsymbol{P} = (\boldsymbol{P}_1| \ldots |\boldsymbol{P}_v)$, which is used for the coefficient estimation.

3. Simulation studies

The performance of ensemble (3) and its value concerning the interpretability of the estimated coefficients is investigated in simulation studies. All results will be compared to alternative classification methods. Existing methods take either the whole curve or few of its characteristics into account, as in Refs. [34–36]. Some are interpretable in terms of a common (functional or non-functional) statistical model, e.g., a functional logistic model, some rather act like “black boxes”. The main advantage of our approach is that its interpretability is based on a wide range of, potentially very different, curve characteristics through the ensemble of semi-metrics. All classification methods used are listed in Table 2.

Since only a limited number of classification methods for functional data has been developed, and only a few come with an implementation, we also include multivariate models. For the multivariate models, the functional principal component (FPC) scores instead of the functional covariates will be used (as has also been done, for example, by Ramsay and Silverman [37]). Those scores have been computed with the fpca.sc-function of the R-package refund [26,27,33,38]. The covariance matrices $\Sigma_i(t; t_0) = \mathrm{cov}(x_i(t); x_i(t_0))$ per curve $x_i(t)$ are estimated in two steps. The eigenfunctions $\phi_e(t)$ and eigenvalues λ_e, $e = 1, \ldots E$, of the corresponding smoothed covariance matrices constitute the functional principal component basis functions and score variances. Here, the final number of scores E has to be chosen. The number of scores is chosen such that at least 95% of the learning samples' variability can be explained. The considerably large proportion of 95% ensures that all

Table 2
The classification methods used for comparison. The second column gives the abbreviations that are used when presenting the results; the third column gives details concerning the implementations.

Method	Abbreviation	R function used (package-name)
Functional *k*-nearest-neighbor ensemble	kNN Ensemble	see online supplement
Nonparametric functional classification (NPFC)	NPFC-deriv	funopadi.knn.lcv (http://www.math.univ-toulouse.fr/staph/npfda/)
NPFC	NPFC-fourier	see above
NPFC	NPFC-mplsr	see above
NPFC	NPFC-pca	see above
Functional linear model	FLM-log	gam (mgcv)
Support vector classifiers	SVM-cov.	svm (e1071)
Support vector classifiers	SVM-FPCs	see above
Random forests	RF-cov.	randomForest (randomForest)
Random forests	RF-FPCs	see above
Linear discriminant analysis	LDA	lda (MASS)
Penalized discriminant analysis	PDA-cov.	fda (mda)
Multinomial model	mM	maxent (maxent)

substantial features of the functions are covered, as the scores with largest variance (corresponding to the first one or two principal components only) are not necessarily those with largest discriminative power.

3.1. Competing methods

3.1.1. Nonparametric functional classification

A nonparametric functional classification (NPFC) approach was introduced in Ferraty and Vieu [39]. Analogously to our ensemble (3), posterior probabilities $\hat{\pi}_{g,h}(x^*(t))$ of the probability that a functional random covariate $x^*(t)$ is of class g are estimated. Estimation is based on one (pre-) chosen semi-metric, and done via a consistent kernel estimator

$$\hat{\pi}_{g,h}(x^*(t)) = \frac{\sum_{j=1}^{N} I\left(y_j = g\right) K\left(h^{-1} d\left(x^*(t), x_j(t)\right)\right)}{\sum_{j=1}^{N} K\left(h^{-1} d\left(x^*(t), x_j(t)\right)\right)},$$

with bandwidth h and $K(\cdot)$ being a fixed positive kernel function. $x^*(t)$ is assigned to the class with the highest estimated probability. There are especially two weak points in this estimation method. First, it uses a single, unweighted semi-metric, in contrast to our approach, which uses a variety of semi-metrics and estimates their weights with respect to their discriminative power. The user has to choose both, the kernel function $K(\cdot)$ and the semi-metric $d(\cdot,\cdot)$. The second drawback is that the NPFC approach allows only for a single covariate. The extension to multiple covariates is not straightforward, in contrast to the simple extension of our ensemble.

For our comparison, we use four semi-metrics implemented for this method. The first semi-metric will be called *NPFC-deriv*. After approximating covariates $x(t)$ by a B-spline basis of $\mathcal{B}$ B-spline functions $B(t)$ and coefficients a such that

$$x(t) \approx \tilde{x}(t) = \sum_{b=1}^{\mathcal{B}} \alpha_b B_b(t),$$

the semi-metric is defined on the approximated covariates $\tilde{x}(t)$ similar to our semi-metric $d_a^{Eucl}(\cdot,\cdot)$ by

$$d_a\left(\tilde{x}^*(t), \tilde{x}_j(t)\right) = \sqrt{\int \left(\tilde{x}^{*(a)}(t) - \tilde{x}_j^{(a)}(t)\right)^2 dt}.$$

Parameters that have to be chosen by the user are the order of derivation a and the number of the interior knots of the B-spline basis. The second semi-metric, called *NPFC-fourier*, builds the same semi-metric, but uses covariates approximated by a Fourier expansion. Parameter choices are the order of derivation a and the number of basis functions. The third semi-metric is denoted by *NPFC-mplsr*. It uses the decomposition of the covariates and response via multivariate partial least squares regression, where the user has to choose the number of retained factors. The last semi-metric is called *NPFC-pca* and is based on a FPCA decomposition. Again, the user has to choose the number of retained factors.

For more details on the semi-metrics and the NPFC method, see Ferraty and Vieu [23]. For all four semi-metrics, the parameters that have to be specified are chosen via *K*-fold cross-validation (minimizing the mean prediction error).

3.1.2. Functional linear model

In the case of a two-class problem, we use a parametric functional model. This means that the functional covariates $x_{iv}(t)$ are directly used as functional predictors. With a Bernoulli distributed response y_i based on the linear predictor η_i, with intercept β_0 and V smooth terms, the model takes the form

$$y_i \sim B\left(1, \frac{exp(\eta_i)}{1 + exp(\eta_i)}\right) \text{ with } \eta_i = \beta_0 + \sum_{v=1}^{V} \int_{\mathbb{D}} x_{iv}(t) \xi_v(t) dt.$$

This model was examined for instance in Reiss and Ogden, Wood, and Gertheiss et al. [40–42] and is implemented in the gam-function of the R package mgcv [43]. It will be referred to by the abbreviation *FLM-log*.

3.1.3. Support vector machines

Support vector machines (SVM) try to find a not necessarily linear decision boundary by transforming the given feature space in such a way that a linear boundary between classes exists. In the case of $G > 2$ classes, the SVM are trained as binary classifiers following the 'one-against-one' approach. We use the implementation of the R package e1071 [44], by using the function svm, with probability = TRUE and default settings else. The SVM is applied to both, the discretized data $x_i(t_q)$ (called *SVM-cov.*) and the FPC scores (called *SVM-FPCs*).

3.1.4. Random forests

This technique builds an ensemble of (classification) trees by growing a predefined (large) number of trees, with each tree being trained on a bootstrapped sample from the learning data. Class membership is then determined by majority vote of the ensemble. The method is implemented in the R package randomForest [45]. We used the randomForest-function with 500 trees. Random forests (RF) are applied to both the discretized data $x_i(t_q)$ (called *RF-cov.*) and the FPC scores (called *RF-FPCs*).

3.1.5. Linear discriminant analysis

We apply linear discriminant analysis (LDA) to the FPC scores, as done by Ramsay and Silverman [37]. LDA is implemented in the R package MASS [46], function lda. Results from the LDA are referred to by the abbreviation *LDA*.

3.1.6. Penalized discriminant analysis

Penalized discriminant analysis (PDA) was developed from LDA. PDA was especially designed for high-dimensional and highly correlated covariates [47], such that it can be applied on the discretized data. The approach is implemented in the R package mda [48], function fda. Results from the PDA are referred to by the abbreviation *PDA-cov.*

3.1.7. Multinomial model

A multinomial logistic regression model is used on the FPC scores. This method is implemented in the maxent-function of the R package maxent [49]. The abbreviation used in the results is *mM*.

3.2. Simulation study A

3.2.1. Set-up

Let $U(\tau_{1,}\tau_2)$ denote an uniform distribution with limits $[\tau_1,\tau_2]$, $N(\mu,\sigma^2)$ a normal distribution with mean μ and variance σ^2, and $f(t;\mu,\sigma^2)$ a normal density function with mean μ and variance σ^2.

Our generating process builds functional covariates

$$x_i(t) = \sum\nolimits_{m=1}^{L_i} f_m\left(t;\mu_m,\sigma_m^2\right)$$

as a sum of L_i normal densities $f_m(t;\mu_m,\sigma_m^2)$, with means $\mu_m \sim U(-1,3)$, variances $\sigma_m^2 = |\rho_m|$, $\rho_m \sim N(0,1)$, and L_i being chosen at random from $\{1,\ldots 11\}$. Since computation is only possible for a discretized covariate, let $x_i = (x_i(t_1),\ldots x_i(t_Q))$ denote the discretization of $x_i(t)$ at observation points $t_q \in \mathbb{D}$, $q = 1,\ldots Q$. The classes y_i are defined with respect to the position of the maximum of the curves. To this end, we divide the domain of definition in five equal sized parts and assign class $y_i = g$ if the maximum of curve $x_i(t)$, $\max(x_i(t)) = x_i(t)_{|t=t_{\max(x_i(t))}}$, lies in the gth part of the domain, with $g \in \{1,2,3,4,5\}$, namely

$$y_i = g \qquad \text{if} \qquad \left(t_{(gQ\text{-}Q)/5} < t_{\max(x_i(t))} \le t_{gQ/5}\right).$$

An example of covariates generated by this process can be found in Fig. 2. The number of observation points for the discretized covariates x_i, $i = 1,\ldots N$, is $Q = 100$, with $t_q \in \mathbb{D} = [0.1, 1]$, $q = 1,\ldots Q$ equidistant points. To be able to use the semi-metrics of Table 1 on the discretized curves, the integrals are approximated by quadrature sums (analogously to, for example, Ref. [41]). The number of observations N is one out of the set $\{100, 300, 1000\}$.

The data generation, and with that the estimation of the coefficients c_l of model (3), is repeated $W = 100$ times to draw conclusions concerning the stability of estimation. As numbers of nearest neighbors k, the set $k \in \mathcal{K} = \{1, 5, 11, 21\}$ is used. As orders of derivative a of the covariates, the set $a \in \{0, 1, 2\}$ is used. For semi-metric $d_a^{shortEucl}(\cdot,\cdot)$, one of the intervals $[t_1, t_{17}]$, $[t_{18}, t_{36}]$, $[t_{37}, t_{56}]$, $[t_{57}, t_{76}]$, $[t_{77}, t_{100}]$ or $[t_{30}, t_{65}]$ is used for $\mathbb{D}_{small}$. For semi-metric $d_a^{relAreas}(\cdot,\cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_1, t_{17}]$, $[t_{57}, t_{76}]$ or $[t_{30}, t_{65}]$ and $\mathbb{D}_2 = [t_{37}, t_{56}]$. For semi-metric $d_{no}^{Jump}(\cdot,\cdot)$, one of the sets $\{t_{15}, t_{19}\}$, $\{t_{34}, t_{40}\}$, $\{t_{54}, t_{58}\}$ or $\{t_{74}, t_{78}\}$ is used for $\{t_n, t_o\}$. For semi-metric $d_a^{Points}(\cdot,\cdot)$, an equidistant grid $t_q \in \{t_{mQ/10}\}$, $m = 1,\ldots 10$, is used. For $d_{a\tau}^{Scan}(\cdot,\cdot)$, function $\phi_\tau(t) = \left(\frac{max\left(x_i^{(a)}(t)\right)}{max\left(\phi_{1,\tau}(t)\right)}\right)\phi_{1,\tau}(t)$ with $\phi_{1,\tau}(t) = \frac{1}{\sqrt{2\pi}\sigma} exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}$ is used. The parameters are $\sigma = 0.05$, $\tau \in \{0.18, 0.36, 0.55, 0.73, 0.91\}$. It should be noted that all these choices are rather arbitrary with respect to the simulated data. This enables us to impartially test the performance and interpretability of the estimated coefficients. If special knowledge about the data at hand is available, one might optimize the above parameters, as has been done in the application Section 4. All of the semi-metrics introduced in Section 2.1, except $d_{no}^{Jump}(\cdot,\cdot)$, are employed on the generated covariates as well as on their centered counterparts $\tilde{x}_i(t) = x_i(t) - \bar{x}_i(t)$. Thus, $p = 504$ coefficients c_l have to be estimated.

The optimal parameters per semi-metric of the NPFC approach are chosen in such a way that they minimize the mean prediction error of a 10-fold cross-validation (CV).

3.2.2. Results

Fig. 3 illustrates the selection results of the proposed ensemble method. In the upper panel, the coefficients c_l that have been estimated to be of mean values above 0.001 are plotted as boxplots across $W = 100$ replications, with sample size $N = 100$. The lower panel shows the respective mean (gray +) and median (black ×) values of these coefficients. Only few of the 504 coefficients were selected. The estimation is similar across all three sample sizes, becoming more stable if more observations are used for estimation.

Recall that the classes y_i of the covariates $x_i(t)$ were assigned with respect to the position of the curves' maximum. But as seen from Table 3, the most important features for the discrimination of the curve classes are the curves' Euclidian distances. This can be seen from the curves' progression (see Fig. 2). Since the $x_i(t)$ follow normal densities, they are very smooth, and show only slight gradients to and from their maximum. Thus, the position of the maximum itself often does not offer more discriminative power than the whole curves' Euclidian distances.

To validate our results, a new data set of $N_{val} = 1000$ observations is generated, and the respective posterior probabilities $\hat{\pi}_{g(l)}$ are calculated. In addition to the Brier score, we give the misclassification rate, which is also a popular measure to judge classification performance. With y_i denoting the true class of observation $x_i(t)$ and $\hat{y}_i$ being the class assigned by the method considered, the misclassification rate is defined as $\text{MCR} = (1/N_{val})\sum_i I(y_i = \hat{y}_i)$. However, it should be kept in mind that, in contrast to the Brier score, the misclassification rate is not a proper scoring rule concerning the estimated posterior probabilities. The global Brier scores and misclassification rates with respect to the validation data across 100 independent replicates (of training data) can be found in Fig. 4. The results of the nearest neighbor ensemble are shown as the first boxplots. The other boxplots show the results for the competing

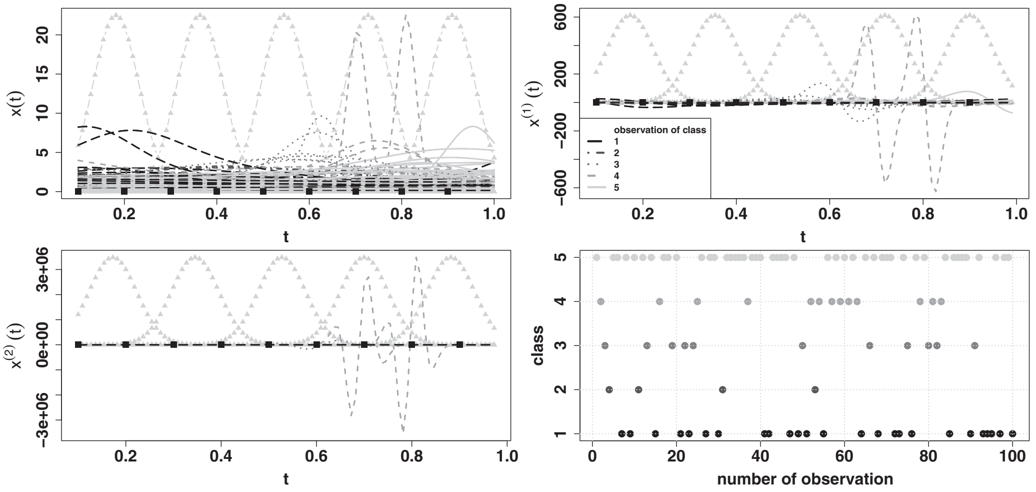


Fig. 2. The upper left panel shows $N = 100$ realizations of the covariates of the generating process. The upper right panel shows their first, the lower left panel their second derivatives. Curve color and line type coding is with respect to the curves' class. The function $\phi_\tau(t)$ used in $d_{a\tau}^{Scan}(\cdot,\cdot)$ is depicted as light gray, dotted lines, the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ as black boxes. The class of each covariate can be found in the lower right panel.

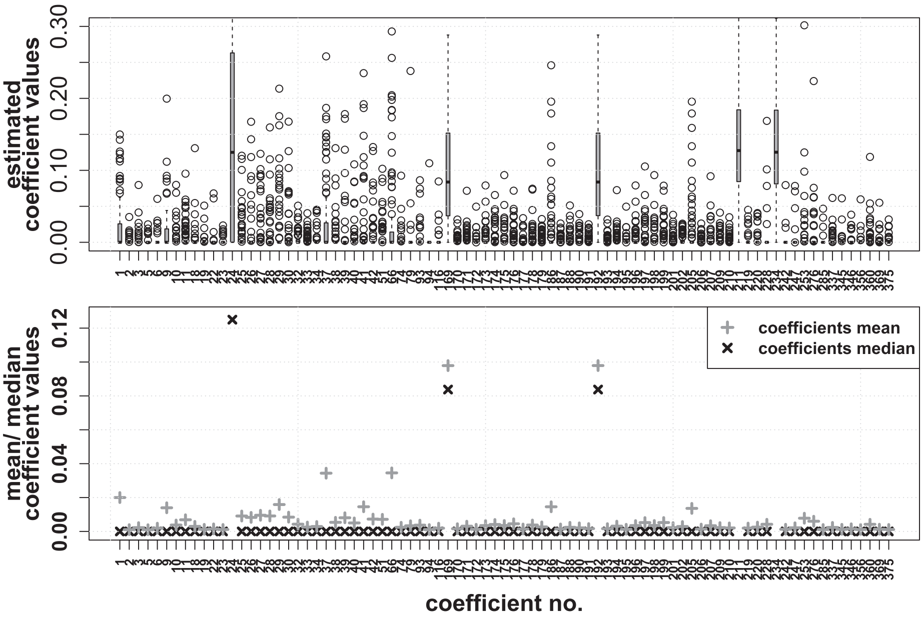


Fig. 3. Estimated coefficients yielding mean values above 0.001. The upper panel shows boxplots across 100 replications when $x_i(t) = \sum_{m=1}^{L_i} f_m(t; \mu_m, \sigma_m^2)$, $i = 1, \ldots N$, with $N = 100$ observations. The lower panel shows the mean and median values for these coefficients.

methods from Section 3.1. Results for different sample sizes are plotted in different colors: (a) white boxes for $N = 100$, (b) light gray for $N = 300$, (c) dark gray for $N = 1000$.

As expected, the Brier score as well as the misclassification rate decrease with an increasing number of observations, except for the Brier scores of the SVM method. It seems that SVM cannot adequately reflect the underlying true probability distribution. Our functional k-nearest-neighbor ensemble yields the lowest Brier scores and misclassification rates across all N. The most competitive method is the only other functional one, the nonparametric functional classification approach. The choice of the semi-metric used here has a non-negligible influence on the classification performance, with option *NPFC-deriv* using the first order of derivation $a = 1$ and 7 interior knots yielding best results. A combination of different semi-metrics as done by our ensemble, however, is apparently the optimal choice here. In general, functional classification approaches perform better than most multivariate approaches applied to the functional principal component scores.

Our estimated weights c_l yielded additional insight in the generated data, revealing the Euclidian distance to contain most information concerning the classification task.

Table 3
Left three columns: IDs (no. according to Fig. 3) of the five estimated coefficients that show the largest means (in decreasing order, for differing numbers of observations N). On the right, the chosen ensemble coefficients are decoded; the value of a indicates the order of derivation, k indicates the number of nearest neighbors used.

IDs (no.) of estimated coefficients			Coefficient number	Parameter tuple
			24	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0, k = 1\}$
N = 100	N = 300	N = 1000	66	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0, k = 5\}$
24	211	234		
211	234	211	169	$\{d^{Eucl}, a = 1, k = 1\}$
234	24	24	192	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1, k = 1\}$
192	169	66		
169	192	192	211	$\{d^{Eucl}, a = 1, k = 5\}$
			234	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1, k = 5\}$

3.3. Simulation study B: Waveform data

With regard to the results presented in Fig. 4, another advantage of the k-nearest-neighbor ensemble is its robust prediction performance with regard to high in-class variability of the functional covariates, compared to the competing methods.

Functional covariates which exhibit similar characteristics in each class can be simulated by the well-studied waveform data [34,39]. Let $u_i \sim U(0,1)$ and $\varepsilon_i(t) \sim N(0,1)$ denote curve specific variables, and define three waveform functions

$$\begin{aligned} &h_1(t) = max(6 - |t - 11|, 0),\\ &h_2(t) = h_1(t - 4), \quad \text{and}\\ &h_3(t) = h_1(t + 4). \end{aligned}$$

The functional covariates are generated by

$$\begin{aligned} &y_i = 1, \quad x_{1i}(t) = u_i h_1(t) + (1 - u_i) h_2(t) + \varepsilon_i(t),\\ &y_i = 2, \quad x_{2i}(t) = u_i h_1(t) + (1 - u_i) h_3(t) + \varepsilon_i(t), \quad \text{or}\\ &y_i = 3, \quad x_{3i}(t) = u_i h_2(t) + (1 - u_i) h_3(t) + \varepsilon_i(t). \end{aligned}$$

The waveform functions as well as the covariates are observed on an equidistant grid of $Q = 100$ points, with $t_q \in \mathbb{D} = [1, 21]$, $q = 1, \ldots Q$, see Fig. 5 for exemplarily curve realizations per class.

Analogously to previous studies [34,39], we simulate 50 training samples containing 150 curves per class, and 50 validation samples containing 250 curves per class. All competing methods were applied on the same sample sets. The estimated k-nearest-neighbor ensemble coefficients yielding the highest means across the 50 estimations correspond to tuples that include the semi-metrics $d_{a=0}^{Eucl}(\cdot,\cdot)$ and $d_{a=0,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, $\mathbb{D}_{small} = [t_{30}, t_{65}] = [6.86, 13.93]$, with $k = 11$ or $k = 21$. Thus, the covariates' Euclidian distances seem to contain more discriminative power than the positions of the covariates' maxima. Fig. 6 shows the classification results for the validation data. As can be seen, the PDA and SVM approaches perform worst. The other methods, including the k-nearest-neighbor ensemble, perform comparable.

In addition to the simulation study given above, we examined another popular low in-class variability benchmark data set, the phoneme data [34,39,50]. Again, classification results of the functional k-nearest-neighbor ensemble were highly competitive (not shown here).

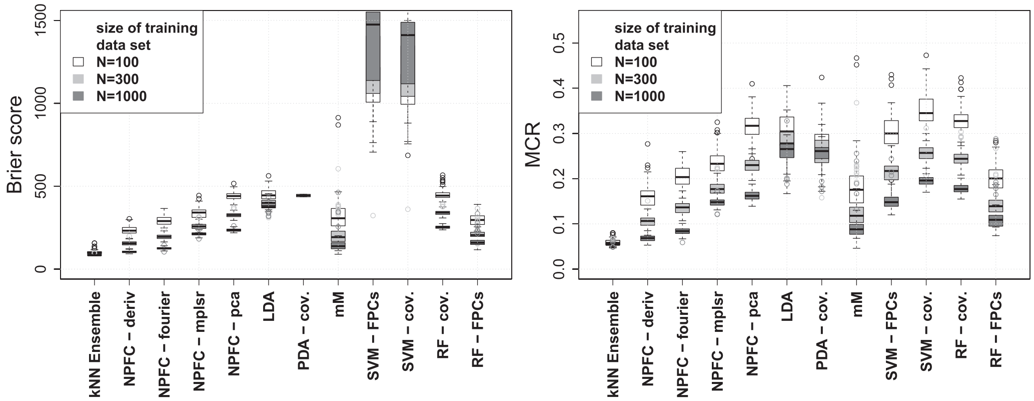


Fig. 4. Results for $N_{val} = 1000$ test observations. The models were estimated 100 times with sample sizes $N = 100$ (white boxes), $N = 300$ (light gray boxes) and $N = 1000$ (dark gray boxes). The left panel shows the Brier scores, the right panel the misclassification rates (MCR).

4. Application to real world data—cell based sensor chips

This section deals with data of cell based silicon sensor chips. Cell based sensor technologies are promising tools concerning environmental quality monitoring (see e.g. [51,52]).

The cell based chips in this study are covered with a monolayer of a living cell population. There are three different kinds of sensors distributed across the chip surface, which record three different cell reactions. Five ion-sensitive field-effect transistors (ISFET) measure the pH value of the extracellular medium. A high acidification of the medium correlates with a high metabolic rate of the cells. One interdigital electrode structure (IDES) is used to draw conclusions about the cell morphology and cell adhesion of the cells on the chip surface. Two oxygen sensitive (CLARK) electrodes measure the oxygen (O_2) contained in the medium, a proxy for the respiration activity of the cells [53,54]. The signals of the sensors are recorded concurrently over time, and we use the arithmetic mean of signals of the same type for our study.

We use Chinese hamster lung fibroblast cells as a cell detection layer because of their stable and reliable growth [55]. They are retained in nutrient medium. If a test substance such as paracetamol is added to the medium, the cells will react to the altered environment. Our goal is to discriminate between measurements with nutrient medium only, and measurements where paracetamol (2.5 mM) is added. Our data set includes $N = N_0 + N_1 = 120$ measurements per signal type of $Q = 89$ observation points, $N_0 = 63$ without and $N_1 = 57$ with AAP, depicted in Fig. 7. Since, just before the test substance reaches the cells, one expects the cells to exhibit 100% viability, all signals were standardized in such a way that, at the respective data point (about 215 min), the signals have a value of 100. The measurements can be divided into three phases: the first corresponds to an acclimatisation phase with medium (no AAP) flowing over the cells to let them adapt to the system and get stable physiological signals. At the second phase from about 220 min on, the AAP reaches the cells. At the last phase (from about 400 min on) 0.2% Triton X-100 is added, removing the cells from the chip surface. This last step is necessary to obtain a negative control.

It can be seen in Fig. 7 that the measurements show some variation even under equal conditions.

4.1. Results

Since the cell chip data consists of the three very different signal types ISFET, IDES, and CLARK, the adequate approach here is to deal with them as a number of $V = 3$ covariate types. The character of the single curves, however, is similar, exhibiting all three measurement phases, such that identical semi-metrics are used for each signal type. The members of ensemble (7) were calculated via leave-one-out. The parameters used for the semi-metrics are the numbers of nearest neighbors $k \in \mathcal{K} = \{1, 5, 11, 21\}$, and orders of derivation $a \in \{0, 1, 2\}$. The choices of $\mathbb{D}_{small}$, $\mathbb{D}_1$, $\mathbb{D}_2$, t_q and τ reflect the signal ranges and points where the AAP reaches the cells in phase two, and the changeover of phase two and three. For semi-metric $d^{shortEucl}_{a,\mathbb{D}_{small}}(\cdot, \cdot)$, one of the intervals $[t_1, t_{35}]$, $[t_{36}, t_{40}]$, $[t_{41}, t_{64}]$, $[t_{65}, t_{69}]$ and $[t_{70}, t_{89}]$ is used for $\mathbb{D}_{small}$; for semi-metric $d^{Jump}_{no}(\cdot,\cdot)$, one of the sets $\{t_{36}, t_{39}\}$ or $\{t_{65}, t_{68}\}$ is used for $\{t_n, t_o\}$; for semi-metric $d^{relAreas}_a(\cdot,\cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_1, t_{35}]$ or $[t_{41}, t_{64}]$ and $\mathbb{D}_2 = [t_{41}, t_{64}]$; for semi-metric $d^{Points}_a(\cdot,\cdot)$, an equidistant grid $t_q = t_{mQ/10}$, $m = 1, \ldots 10$, is used; and for semi-metric $d^{Scan}_{a\tau}(\cdot,\cdot)$, the function $\phi_\tau(t) = \left(\frac{max\left(x_i^{(a)}(t)\right)}{max\left(\phi_{1,\tau}(t)\right)}\right)\phi_{1,\tau}(t)$ with $\phi_{1,\tau}(t) =$

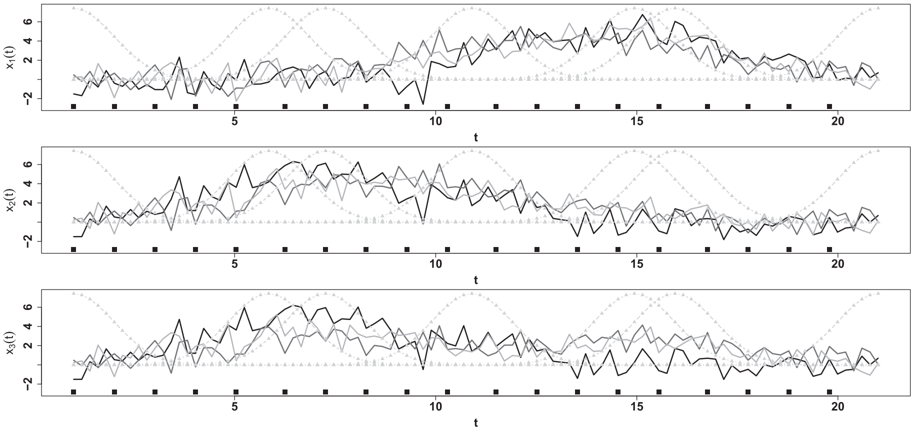


Fig. 5. The panels show $N = 3$ realizations of waveform covariates for each class. The function $\phi_\tau(t)$ used in $d^{Scan}_{a\tau}(\cdot,\cdot)$ is depicted as light gray, dotted lines, the impact points t_q used in $d^{Points}_a(\cdot,\cdot)$ as black boxes.

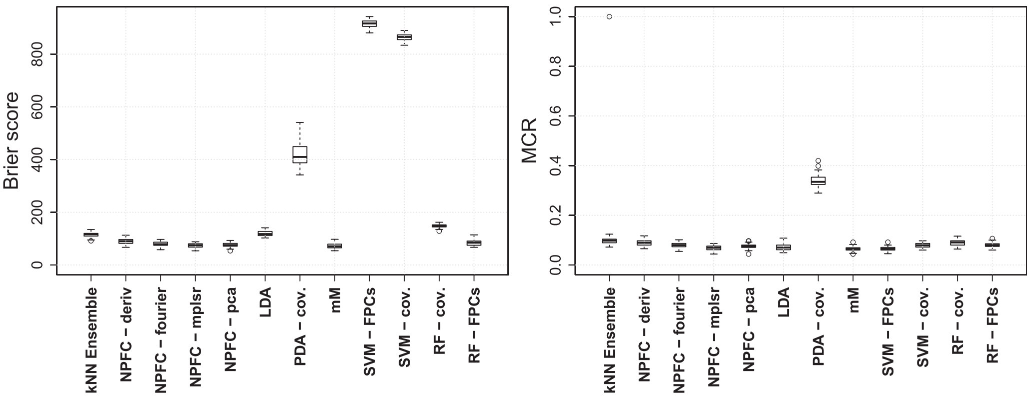


Fig. 6. Results for $N_{val} = 250$ test observations per class. The models were estimated 50 times with sample sizes of $N = 150$ covariates per class. The left panel shows the Brier scores, the right panel the misclassification rates (MCR).

$\frac{1}{\sqrt{2\pi}\sigma} exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}, \sigma = 10$ and $\tau \in \{3.120, 45.120, 87.120, 135.120, 177.120, 219.129, 267.129, 309.129, 351.129, 399.140, 441.140, 483.140, 531.140\}$ is used.

Only five of the $p = 1872$ coefficients (624 per signal type) are estimated to be of values unequal to zero. The respective coefficients are listed in Table 4. They correspond to the semi-metrics $d_{a\tau}^{Scan}(\cdot,\cdot)$ and $d_{a,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, which take the part of the signal where the AAP reaches the cells, around data point $t_{37} = 219.129$, into account. These results are sensible: In theory, the curve progression per signal type should be similar for two curves when the cells meet similar conditions. Furthermore, when AAP reaches the cells at about 220 min, this should stimulate a cell reaction, which is reflected by a jump in the curves of most measurements with AAP. In contrast, measurements without AAP should not notably alter their progression. This clustering of the curves representing non or 2.5 mM AAP is especially obvious in the CLARK-signals, which are selected by the ensemble to be the most informative signal type for this classification task.

To test the performance of our model and compare its prediction accuracy to other approaches, the data was split randomly $W = 25$ times into 15 subsets of eight observations each. With these, a 15-fold cross-validation is performed W times to estimate the coefficients of ensemble (7) and to validate the results. Fig. 8 summarizes the results for the validation data. The global Brier scores and misclassification rates are shown for all approaches, with the results of our functional k-nearest-neighbor ensemble being presented as the first boxes. For the NPFC approach, white boxes show results if only ISFET is used, light gray boxes if only IDES is used, and dark gray boxes if only CLARK is used. The optimal parameters per semi-metric and covariate type of the NPFC approach are chosen via minimization of the mean prediction error of a 10-fold CV.

For the validation data set, the multivariate approaches applied on the functional principal component scores are performing worse than the functional approaches in the Brier score. For the NPFC approach, the choices of the semi-metric as well as the covariate type are essential. In accordance to the results of our k-nearest-neighbor ensemble, the results of the NPFC method are best when using the CLARK-signals.

Our approach is highly competitive in terms of prediction performance. Only the NPFC approach using the CLARK-signals and random forests applied on the discretized covariates performed slightly better. When using the NPFC method, however, the input variable as well as the single semi-metric, i.e., one particular curve characteristic, have to be chosen by the user. Given those, NPFC rather acts like a "black box" but does not give interpretable results in terms of feature selection. The same is true for random forests. Thus, if automated, interpretable variable and feature selection is of main interest, accompanied by good prediction performance, our k-nearest-neighbor ensemble is a very attractive choice.

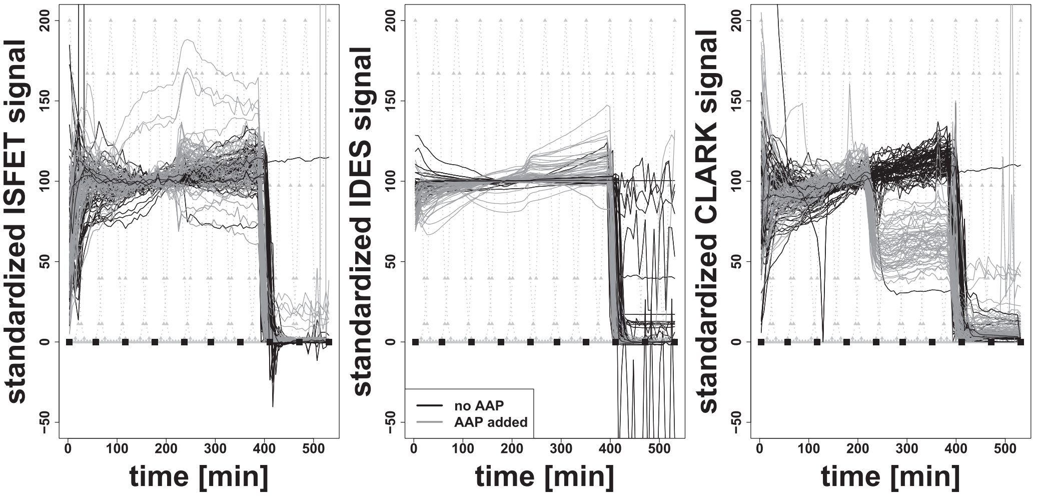


Fig. 7. The $N = 120$ standardized signals for each of the three sensor types, measured at 89 time points on an equidistant grid. The gray shades represent the presence (gray lines) or absence (black lines) of AAP. The light gray, dotted lines depict the function $\phi_\tau(t)$ in $d_{a\tau}^{Scan}(\cdot,\cdot)$ used at certain observation points, the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ are depicted as black boxes.

Table 4
The five coefficients that were selected for the cell chip data. Left column: the coefficient numbers of the selected ensemble members. On the right, the selected members are decoded. The value of *a* indicates the order of derivation, *k* indicates the number of nearest neighbors used.

Coefficient IDs	Parameter tuple	Covariate	Semi-metric parameters
1268	$\{d^{Scan}, a = 0, k = 1\}$	CLARK	$q = 37, \tau = 219.129$
1476	$\{d^{Scan}, a = 1, k = 1\}$	CLARK	$q = 37, \tau = 219.129$
1459	$\{d^{shortEucl}, a = 1, k = 1\}$	CLARK	$\mathbb{D}_{small} = [t_{36}, t_{40}]$
1486	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 1, k = 1\}$	CLARK	$\mathbb{D}_{small} = [t_{36}, t_{40}]$
1501	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 1, k = 1\}$	CLARK	$q = 37, \tau = 219.129$

5. Discussion

We introduced a functional *k*-nearest-neighbor ensemble that allows for automatic feature and, depending on the data at hand, variable selection. For that purpose, a set of semi-metrics was defined. Here, each semi-metric focused on a specific feature of the functional covariates. Additionally, sets of numbers of nearest neighbors and of orders of derivation of the covariates were defined. A particular combination of a semi-metric, a number of nearest neighbors and an order of derivation made up a parameter tuple. The ensemble members were then calculated by a *k*-nearest-neighbor approach and the leave-one-out technique, using a specific tuple. Each ensemble member was weighted by an unknown coefficient. These coefficients were estimated such that the global Brier score was minimized. A constraint put on the coefficients yielded an implicit Lasso-type penalty, such that some coefficients were estimated to be exactly zero. Zero-valued coefficients mean that the respective ensemble member, i.e., a certain tuple, has a weight of zero. Thus, an automatic feature selection was performed during the estimation process. In the case of multiple functional (and non-functional) covariates, the parameter tuple can also include the covariate type, such that the ensemble allows for additional variable selection.

Our ensemble presents a flexible and powerful tool for the classification of all sorts of functional, including, but not limited to, chemometric data. While competitive in terms of predictive classification performance, the automatic and interpretable feature selection is an important advantage compared to other discrimination methods. In simulation studies, it was shown that even a set of essentially arbitrarily

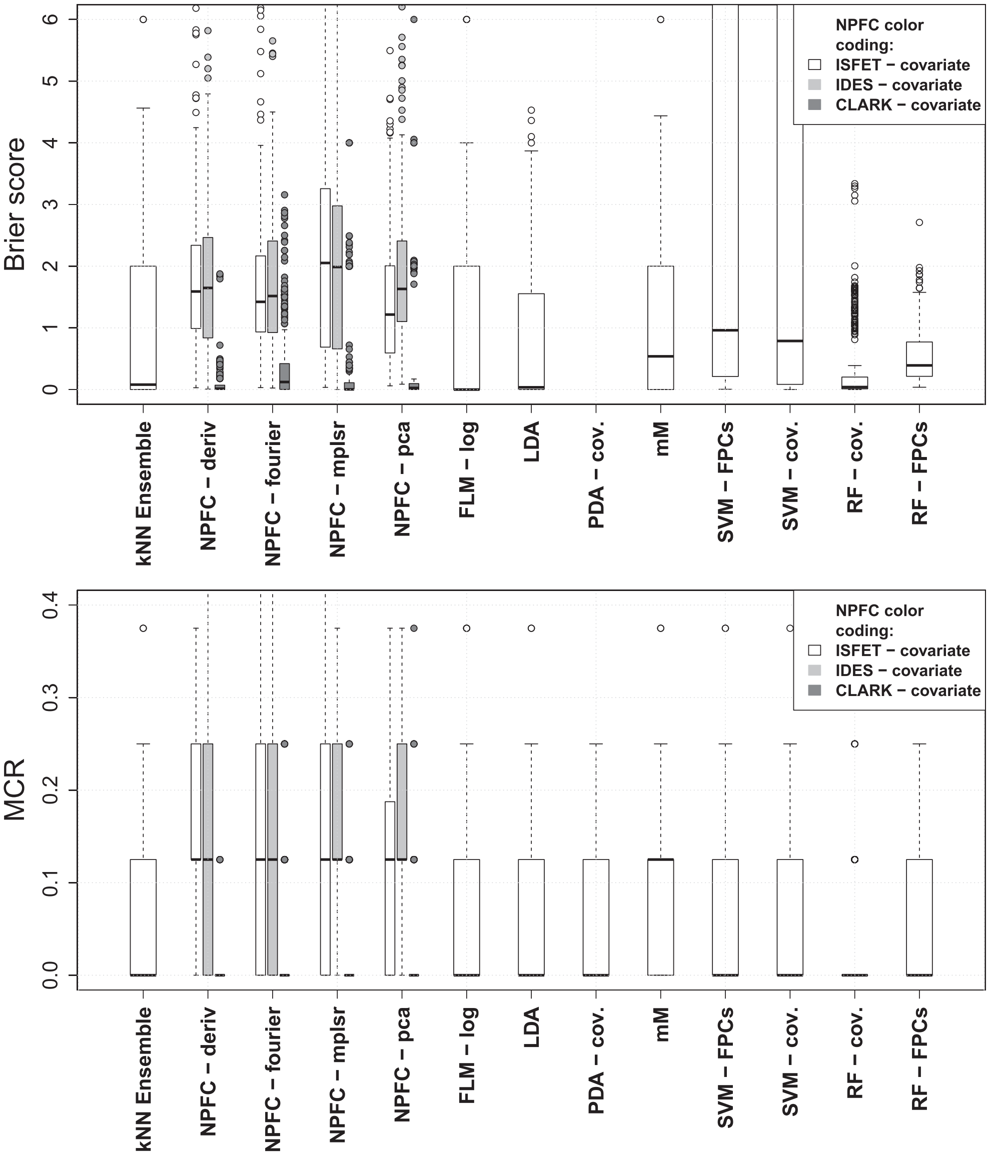


Fig. 8. Validation results of the cell chip data for all classification approaches on basis of 25 replications of a 15-fold CV. The upper panel shows the Brier scores, the lower panel the misclassification rates (MCR).

chosen semi-metrics yields excellent predictions and sensible results in terms of interpretability. In the cell chip data application, the prediction performance was competitive to or better than the alternative methods. The feature and variable selection here was outstanding since the estimated coefficients ideally agreed with the biological background knowledge.

All calculations were performed on a machine with 256 GB RAM and four AMD Opteron CPUs of 12 cores and 2.2 GHz, with software R version 3.1.0 [33] and the add-on packages mentioned above.

Thanks to the flexibility of the functional *k*-nearest-neighbor ensemble, there is even ample room for extension. Of course, the ensemble members are not limited to members that are based on distance measures. Other members could be included, for example the results of basic models such as a functional linear model. Also, the ensemble could be adapted to regression problems by a suitable modification of the underlying *k*-nearest-neighbor method and the optimization criterion. One could think about further developments in the direction of time series analysis. The implicit Lasso-type penalty imposed on the ensemble coefficients c_l by the constraints $c_l \geq 0 \; \forall \; l$ and $\sum_{l=1}^{p} c_l = 1$ could be relaxed by altering the second condition to $\sum_{l=1}^{p} c_l = \nu$. The estimation, thus, would be obtained by a Lasso-type estimator with tuning parameter ν which controls the sparseness of the ensemble.

In the analyses provided, a certain degree of smoothness of the functional data is implicitly assumed in the semi-metrics used. If this is not the case for the data at hand, some preprocessing is advisable. Options are for example to approximate the data by decomposition and to build a semi-metric based on the decomposition, or use smoothed signals in a semi-metric.

Although our approach provides built-in feature selection, the question which set of semi-metrics and respective parameters should be used must still be answered by the user. If background knowledge on the data is provided, the semi-metrics can be chosen adequately, as has been done with the cell chip data. If no such knowledge is available, one can, for example, use a very large set of semi-metrics and let the data decide using the proposed ensemble with built-in feature selection. If the set of potential semi-metrics becomes too large, similar to a random forest approach, the semi-metrics and parameters actually used in the ensemble could be subsets randomly drawn from the predefined sets. This procedure could be repeated until a model choice criterium is fulfilled.

Conflict of interest statement

The authors declare that they have no conflict of interest.

Acknowledgments

The first author gratefully acknowledges the support by the Siemens AG, Corporate Technology. We thank Evamaria Stütz, Ulrich Bohrn, and Christian Guijarro for providing parts of the cell chip data and motivating our research. Furthermore, we thank two anonymous referees for several helpful comments on the manuscript and the methodology.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.chemolab.2015.04.019.

References

[1] E. Fix, J.L. Hodges, Discriminatory analysis—nonparametric discrimination: consistency properties, Tech. rep., US Air Force School of Aviation Medicine, Randolph Field Texas, 1951.

[2] R.M. Alonso-Salces, S. Guyot, C. Herrero, L.A. Berrueta, J.-F. Drilleau, B. Gallo, F. Vicente, Chemometric classification of Basque and French ciders based on their total polyphenol contents and CIELab parameters, Food Chem. 91 (2005) 91–98.

[3] R. Japon-Lujan, J. Ruiz-Jiménez, M.D.L. de Castro, Discrimination and classification of olive tree varieties and cultivation zones by biophenol contents, J. Agric. Food Chem. 54 (2006) 9706–9712.

[4] B.M. Lukasiak, S. Zomer, R.G. Brereton, R. Faria, J.C. Duncan, Pattern recognition and feature selection for the discrimination between grades of commercial plastics, Chemom. Intell. Lab. Syst. 87 (2007) 18–25.

[5] D. Kruzlicova, J. Mocak, E. Katsoyannos, E. Lankmayr, Classification and characterization of olive oils by UV-Vis absorption spectrometry and sensorial analysis, J. Food Nutr. Res. 47 (4) (2008) 181–188.

[6] S. Fdez-Ortiz de Vallejuelo, G. Arana, A. de Diego, J.M. Madariaga, Pattern recognition and classification of sediments according to their metal content using chemometric tools. A case study: the estuary of Nerbioi-Ibaizabal River, Bilbao, Basque Country, Chemosphere 85 (2011) 1347–1352.

[7] L.A. Berrueta, R.M. Alonso-Salces, K. Héberger, Supervised pattern recognition in food analysis, J. Chromatogr. A 1158 (2007) 196–214.

[8] I. Melvin, J. Weston, C.S. Leslie, W.S. Noble, Combining classifiers for improved classification of proteins from sequence or structure, BMC Bioinforma. 9 (2008) 389–397.

[9] C. Wong, Y. Li, C. Lee, C.H. Huang, Ensemble learning algorithms for classification of mtDNA into haplogroups, Brief. Bioinform. 12 (1) (2010) 1–9.

[10] M. Przewozniczek, K. Walkowiak, M. Wozniak, Optimizing distributed computing systems for *k*-nearest neighbours classifiers – evolutionary approach, Log. J. IGPL 19 (2) (2011) 357–372.

[11] R. Nava, B. Escalante-Ramirez, G. Cristóbal, R.S.J. Estépar, Extended Gabor approach applied to classification of emphysematous patterns in computed tomography, Med. Biol. Eng. Comput. 52 (2014) 393–403.

[12] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Science and Business Media, New York, 2011.

[13] J. Gertheiss, G. Tutz, Feature selection and weighting by nearest neighbor ensembles, Chemom. Intell. Lab. Syst. 99 (2009) 30–38.

[14] J. Ji, Q. Zhao, A hybrid SVM based on nearest neighbor rule, Int. J. Wavelets Multiresolution Inf. Process. 11 (6) (2013).

[15] M. Hayat, M. Tahir, S.A. Khan, Prediction of protein structure classes using hybrid space of multi-profile Bayes and bi-gram probability feature spaces, J. Theor. Biol. 346 (2014) 8–15.

[16] B. Bischl, J. Schiffner, C. Weihs, Benchmarking local classification methods, Comput. Stat. 28 (2013) 2599–2619.

[17] J. Ramsay, B. Silverman, Functional Data Analysis, New York, Springer, 2005.

[18] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (1992) 241–259.

[19] M. LeBlanc, R. Tibshirani, Combining estimates in regression and classification, J. Am. Stat. Assoc. 91 (436) (1996) 1641–1650.

[20] M. van der Laan, S. Dudoit, Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: finite sample oracle inequalities and examples, Paper 130, University of California, Berkeley, Division of Biostatistics, 2003.

[21] J. Goldsmith, F. Scheipl, Estimator selection and combination in scalar-on-function regression, Comput. Stat. Data Anal. 70 (2014) 362–372.

[22] H.G. Müller, U. Stadtmüller, Generalized functional linear models, Ann. Stat. 33 (2) (2005) 774–805.

[23] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis, Springer Science and Business Media, New York, 2006.

[24] A.M. Alonso, D. Casado, J. Romo, Supervised classification for functional data: a weighted distance approach, Comput. Stat. Data Anal. 56 (2012) 2334–2346.

[25] K. Soetaert, K.V. den Meersche, D. van Oevelen, limSolve: solving: linear inverse models, R package version 1.5.5 (2013).

[26] C. Di, C.M. Crainiceanu, B.S. Caffo, N.M. Punjabi, Multilevel functional principal component analysis, Ann. Appl. Stat. 3 (1) (2009) 458–488.

[27] J. Goldsmith, S. Greven, C. Crainiceanu, Corrected confidence bands for functional data using principal components, Biometrics 69 (1) (2013) 41–51.

[28] P. Hall, B.U. Park, R.J. Samworth, Choice of neighbor order in nearest-neighbor classification, Ann. Stat. 36 (5) (2008) 2135–2152.

[29] G.W. Brier, Verification of forecasts expressed in terms of probability, Mon. Weather Rev. 78 (1) (1950) 1–3.

[30] T. Gneiting, A.E. Raftery, Strictly proper scoring rules prediction, and estimation, J. Am. Stat. Assoc. 102 (477) (2007) 359–378.

[31] R. Selten, Axiomatic characterization of the quadratic scoring rule, Exp. Econ. 1 (1998) 43–62.

[32] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B 58 (1) (1996) 267–288.

[33] R. Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2014. (URL http://www.R-project.org).

[34] I. Epifanio, Shape descriptors for classification of functional data, Technometrics 50 (3) (2008) 284–294.

[35] F. Rossi, N. Villa, Support vector machine for functional data classification, Neurocomputing 69 (2006) 730–742.

[36] G.M. James, Functional linear discriminant analysis for irregularly sampled curves, J. R. Stat. Soc. B 63 (3) (2001) 533–550.

[37] J.O. Ramsay, B.W. Silverman, Applied Functional Data Analysis, Springer-Verlag Inc., New York, 2002.

[38] C. Crainiceanu, P. Reiss, J. Goldsmith, L. Huang, L. Huo, F. Scheipl, B. Swihart, S. Greven, J. Harezlak, M. G. Kundu, Y. Zhao, M. McLean, L. Xiao, *refund*: *Regression with functional data*, R package version 0.1-9, 2013.

[39] F. Ferraty, P. Vieu, Curves discrimination: a nonparametric functional approach, Comput. Stat. Data Anal. 44 (2003) 161–173.

[40] P.T. Reiss, R.T. Ogden, Smoothing parameter selection for a class of semiparametric linear models, J. R. Stat. Soc. B 71 (2) (2009) 505–523.

[41] S. Wood, Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models, J. R. Stat. Soc. B 73 (1) (2011) 3–36.

[42] J. Gertheiss, A. Maity, A.-M. Staicu, Variable selection in generalized functional linear models, Stat. 2 (2013) 86–101.

[43] S. Wood, *mgcv*: *Mixed Gam Computation Vehicle with GCV/ AIC/ REML Smoothness Estimation*, R package version 1.8-4, 2014.

[44] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, C.-C. Chang, C.-C. Lin, *e*1071: *Misc Function of the Department of Statistic* (*e*1071), *TU Wien*, R package version 0.1-9, 2013.

[45] L. Breiman, A. Cutler, A. Liaw, M. Wiener, *random Forest*: *Breiman and Cutler's random forest for classification and regression*, R package version 4.6-7, 2012.

[46] B. Ripley, B. Venables, D. M. Bates, K. Hornik, A. Gebhardt, D. Firth, *MASS Support Functions and Datasets for Venables and Ripley's MASS*, R package version 7.3-30, 2014.

[47] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, Ann. Stat. 23 (1) (1995) 73–102.

[48] T. Hastie, R. Tibshirani, F. Leisch, K. Hornik, B. D. Ripley, *mda*: *mixture and flexible discriminant analysis*, R package version 0.4-4.

[49] T. P. Jurka, Y. Tsuruoka, *maxent Low – memory Multinomial Logistic Regression with Support for Text Classification*, R package version 1.3.3.1, 2013.

[50] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, New York, Chapman & Hall, 1984.

[51] U. Bohrn, E. Stütz, K. Fuchs, M. Fleischer, M.J. Schöning, P. Wagner, Monitoring of irritant gas using a whole-cell-based sensor system, Sensor Actuator B Chem. 175 (2012) 208–217.

[52] R. Kubisch, U. Bohrn, M. Fleischer, E. Stütz, Cell-based sensor system using L6 cells for broad band continuous pollutant monitoring in aquatic environments, Sensors 12 (3) (2012) 3370–3393.

[53] E. Thedinga, A. Kob, H. Holst, A. Keuer, S. Drechsler, R. Niendorf, W. Baumann, I. Freund, M. Lehmann, R. Ehret, Online monitoring of cell metabolism for studying pharmacodynamic effects, Toxicol. Appl. Pharmacol. 220 (2007) 33–44.

[54] L. Ceriotti, A. Kob, S. Drechsler, J. Ponti, E. Thedinga, P. Colpo, R. Ehret, F. Rossi, Online monitoring of BALB/3T3 metabolism and adhesion with multiparametric chip-based system, Anal. Biochem. 371 (2007) 92–104.

[55] U. Bohrn, A. Mucha, C. Werner, B. Trattner, M. Bäcker, C. Krumbe, M. Schienle, E. Stütz, D. Schmitt-Landsiedel, M. Fleischer, P. Wagner, M. Schöning, A critical comparison of cell-based sensor systems for the detection of Cr(VI) in aquatic environment, Sensors Actuators B 182 (2013) 58–65.