

University of Groningen

Adaptive basis functions for prototype-based classification of functional data

Melchert, Friedrich; Bani, Gabriele; Seiffert, Udo; Biehl, Michael

Published in:
Neural Computing and Applications

DOI:
[10.1007/s00521-019-04299-2](https://doi.org/10.1007/s00521-019-04299-2)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Version created as part of publication process; publisher's layout; not normally made publicly available

Publication date:
2019

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):
Melchert, F., Bani, G., Seiffert, U., & Biehl, M. (2019). Adaptive basis functions for prototype-based classification of functional data. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-019-04299-2>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).


Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.



Adaptive basis functions for prototype-based classification of functional data

Friedrich Melchert^{1,3}  · Gabriele Bani² · Udo Seiffert¹ · Michael Biehl³

Received: 15 March 2018 / Accepted: 17 June 2019
© The Author(s) 2019

Abstract

We present a framework for distance-based classification of functional data. We consider the analysis of labeled spectral data and time series by means of generalized matrix relevance learning vector quantization (GMLVQ) as an example. To take advantage of the functional nature, a functional expansion of the input data is considered. Instead of using a predefined set of basis functions for the expansion, a more flexible scheme of an adaptive functional basis is employed. GMLVQ is applied on the resulting functional parameters to solve the classification task. For comparison of the classification, a GMLVQ system is also applied to the raw input data, as well as on data expanded by a different predefined functional basis. Computer experiments show that the methods offer potential to improve classification performance significantly. Furthermore, the analysis of the adapted set of basis functions give further insights into the data structure and yields an option for a drastic reduction of dimensionality.

Keywords Functional data · Machine learning · Adaptive basis · GMLVQ · Relevance learning

1 Introduction

In a variety of modern systems, high-dimensional sensor data are produced that is usually difficult to handle with traditional methods. To cope with the high number of input dimensions, the use of prior knowledge of the data generating systems can be used for simplification and dimensionality reduction. In many cases, high-dimensional data vectors can be assumed to represent discretized versions of continuous functions. Although time series might come to mind first as an example, similar properties are also present in other types of sequential data, such as spectral image data.

Such data are frequently recorded to serve as input for a classification task. Various machine learning algorithms can be applied, having specific advantages and disadvantages. The popularity of prototype- and distance-based classification systems results from their intuitive interpretation and straightforward implementation [2, 7]. In this paper, an extension of the popular learning vector quantization (LVQ) [21] is used. LVQ systems comprise different prototypes which represent characteristic properties of their corresponding classes. Together with an appropriate distance measure, they constitute an efficient classification method.

The selection of a suitable distance measure is a key issue in the design of a prototype-based classifier [6]. While using predefined distances such as the Euclidean or other Minkowski-like measures often yields reasonable results, the application of more flexible relevance schemes has shown to be beneficial [8]. In generalized matrix relevance LVQ (GMLVQ) [34], a parameterized distance measure is employed whose parameters are determined in a data-driven training process [35]. This offers greater flexibility, since only the general structure of the distance measure has to be specified in advance. The analysis of the

✉ Friedrich Melchert
friedrich.melchert@iff.fraunhofer.de

¹ Fraunhofer Institute for Factory Operation and Automation IFF, Magdeburg, Germany

² Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands

³ Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, The Netherlands

emerging distance measure can give further insights into the nature of the input data [34–36].

In this paper, we introduce a further extension of GMLVQ with respect to functional data. An expansion of functional data by means of suitable basis functions in combination with GMLVQ has proven to make superior classification performance and drastic reduction of input dimensions possible. In [26, 27], the representation of spectral data and time series as weighted sums of basis functions facilitated a reduction of input dimensions. As a first example, Chebyshev polynomials of first kind were examined as possible basis functions. In [35], a wavelet transformation was employed for the dimensionality reduction, but was mainly motivated as a definition of discriminative features for sharply peaked mass spectra.

The use of functional data analysis for classification has been investigated in recent years in combination with different approaches including support vector machines [31], random forests [16] and self-organizing maps [30]. The increasing availability of deep learning frameworks and suitable hardware has further motivated the analysis of functional data by using convolutional neural networks [19] and auto-encoders [18]. While the latter studies follow a data-driven approach to incorporate the functional nature of the data, the interpretation of the obtained results remains tough, due to the high number of free parameters in deep learning models. Most of the other previous studies take only a single predefined generic functional expansion into account. The problem of an appropriate functional basis selection (with respect to the classification task) remains and is scarcely discussed in the literature.

Since the choice of the functional basis has direct influence on the resulting distance measure, it becomes a key problem in the design of functional data classifiers. Along the lines of the adaptive distance matrix in GMLVQ, we propose an adaptive functional basis that is optimized during the same data-driven training process. Please note that GMLVQ serves here as an example classification system and is favored based on its intuitive interpretation. The principle idea of an adaptive functional expansion can be transferred to other classification systems. This paper extends the approach of adaptive functional basis in combination with GMLVQ that was originally presented in the context of the WSOM+ Conference 2017 [4].

In the following section, we outline the mathematical framework including the classification using GMLVQ and its extension toward an adaptive functional basis. The presented approach is then applied to several publicly available benchmark datasets of time series and spectral data. Results of computer experiments are discussed in Sect. 4. To validate the plausibility of the adapted functional basis, results are transferred to other machine

learning algorithm in Sect. 5 before we conclude with a summary and outlook.

2 The mathematical framework

In a general supervised classification task, a set of P labeled d -dimensional feature vectors

$$\mathbf{v}_\mu \in \mathbb{R}^d \text{ and labels } S_\mu = S(\mathbf{v}_\mu) \in \{1, 2, \dots, C\} \quad (1)$$

for $\mu = 1, 2, \dots, P$ is considered as a training set which is exploited for the adaptation of a classifier. As an example of prototype- and distance-based classification, we consider generalized matrix relevance LVQ in the following.

2.1 Generalized matrix relevance LVQ

In learning vector quantization (LVQ), the classifier is composed of a set of M labeled prototypes

$$\{\mathbf{w}_V^j \in \mathbb{R}^d\}_{j=1}^M \text{ with labels } S_j \in \{1, 2, \dots, C\}, \quad (2)$$

in combination with an appropriate distance measure $d(\mathbf{w}_V, \mathbf{v})$ [6]. An unlabeled feature vector \mathbf{v} is assigned to the class of the closest prototype \mathbf{w}_V^* with respect to the employed distance measure with $d(\mathbf{w}_V^*, \mathbf{v}) = \min_m \{d(\mathbf{w}_V^m, \mathbf{v})\}_{m=1}^M$.

The prototypes are adapted during a training process that is either of heuristic nature like the popular LVQ1 algorithm [21], or guided by a suitable cost function, e.g., [32]. Please note that the subscript V of the prototypes indicates that they are defined in the same space as the feature vectors \mathbf{v}_μ . The assumption that they serve as representatives for their corresponding class facilitates an intuitive interpretation of the classifier with respect to the data.

The selection of an appropriate distance measure is a key issue in the design of an LVQ system [6]. Instead of using a fixed measure, like the Euclidean or other Minkowski-like distances, the use of flexible relevance schemes has shown to improve classification performance [17]. In generalized matrix relevance LVQ (GMLVQ) [34], a parameterized quadratic distance measure of the form

$$\begin{aligned} d_V(\mathbf{w}_V, \mathbf{v}) &= (\mathbf{w}_V - \mathbf{v})^\top \Lambda_V (\mathbf{w}_V - \mathbf{v}) \\ &= [\Omega_V (\mathbf{w}_V - \mathbf{v})]^2 \end{aligned} \quad (3)$$

is employed. The parametrization $\Lambda_V = \Omega_V^\top \Omega_V$ ensures positive semi-definiteness of the measure [34]. To avoid numerical problems, we impose a normalization $\text{Tr}(\Lambda_V) = \sum_{i,j} [\Omega_V]_{ij}^2 = 1$, as suggested in [34].

For the sake of simplicity, we restrict the definition here to a single global square distance measure $\Omega_V \in \mathbb{R}^{d \times d}$.

Extensions of the system to multiple local distance measures or rectangular matrices have been studied in [10].

Prototypes \mathbf{w}_V and the distance matrix Ω_V are adapted during a single data-driven training process. This optimization can be implemented as a stochastic gradient descent, as in [34]. Here, we employ a batch gradient descent algorithm together with an automatic step size adaption scheme described in [28]. The optimization is guided by the cost function introduced in [32]

$$E = \sum_{\mu=1}^P \Phi \left(\frac{d(\mathbf{w}_V^J, \mathbf{v}_\mu) - d(\mathbf{w}_V^K, \mathbf{v}_\mu)}{d(\mathbf{w}_V^J, \mathbf{v}_\mu) + d(\mathbf{w}_V^K, \mathbf{v}_\mu)} \right). \tag{4}$$

For a given feature vector \mathbf{v}_μ , \mathbf{w}_V^J denotes the closest prototype of the correct class $S(\mathbf{w}_V^J) = S(\mathbf{v}_\mu)$, while \mathbf{w}_V^K denotes the closest prototype of any other class $S(\mathbf{w}_V^K) \neq S(\mathbf{v}_\mu)$. Often the function $\Phi(\dots)$ in (4) is chosen to be sigmoidal. To avoid the introduction of further parameters, we resort to the simplest setting with $\Phi(x) = x$. All results presented here are achieved using a Python implementation of GMLVQ, based on the code available at [5].

2.2 Adaptive basis for functional data

In contrast to a generic classification task, where feature vectors simply represent the concatenation of more or less independent quantities, functional input data are assumed to be a discretized representation of an unknown continuous function

$$v_{\mu,j} = f_\mu(x_j). \tag{5}$$

At this point, we restrict the discussion to regularly sampled functions with equidistant x_j . An example for prototype-based classification of functional data that incorporate irregular sampling can be found in [26].

Given a suitable set of basis functions $g_k(x)$ it is possible to find a parameterized representation of f_μ as

$$f_\mu(x) = \sum_{k=0}^{\infty} c_{\mu,k} g_k(x), \tag{6}$$

resulting in an alternative feature vector \mathbf{c}_μ . Limiting the number of coefficients to n makes $\mathbf{c}_\mu \in \mathbb{R}^n$ an approximative representation of $f_\mu(x)$. The parameters $c_{\mu,k}$ can be determined by a suitable optimization criterion such as the quadratic or maximum deviation. In [26], a matrix representation for the functional expansion was introduced as

$$\mathbf{c}_\mu = \Psi \mathbf{v}_\mu, \tag{7}$$

where the rows of $\Psi \in \mathbb{R}^{n \times d}$ represent the employed basis functions evaluated at the sampling points x_j .

Under consideration of the functional expansion of the input data, the distance measure introduced in (3) becomes

$$d_c(\mathbf{w}_V, \mathbf{v}) = (\mathbf{w}_V - \mathbf{v})^\top \Psi^\top \Lambda_c \Psi (\mathbf{w}_V - \mathbf{v}). \tag{8}$$

Note that the relevance matrix $\Lambda_c \in \mathbb{R}^{n \times n}$ is defined with respect to the space of functional parameters. Given this relation, the functional approximation matrix Ψ can be optimized in a similar way as the relevance matrix Λ in the training. In contrast to a predefined fixed functional basis, this facilitates a more flexible framework. Not only it does eliminate the need of a prior basis selection, also the interpretation of the emerging functional basis can give insights into the input data structure with respect to the classification task.

To ensure a functional characteristic for the rows of Ψ the optimization has to be constrained. We consider a matrix Ψ to have a functional characteristic if its rows represent a set of basis functions. A desirable functional basis is a set of continuous, linear independent or even orthogonal functions; In our case, a discretized representation, respectively.

To achieve these properties, we propose two methods. As a first approach, we alter the update rule of Ψ to incorporate the desired properties. We make use of a Gram–Schmidt orthonormalization [33] coupled with a row-wise moving average filter, resulting in a set of smooth orthogonal basis functions. During training the modified update is repeatedly applied as

$$\Psi \leftarrow (1 - \alpha - \beta) \Psi + \alpha \cdot \tilde{\Psi} + \beta \cdot \Psi^*, \tag{9}$$

with $\alpha, \beta \in [0, 1]$ and $\alpha + \beta \leq 1$

where $\tilde{\Psi}$ represents the row-wise smoothed version of Ψ and Ψ^* its orthonormalization. Using the parameters α and β the behavior of functional constraint is controlled, where $\alpha = 0, \beta = 0$ voids the functional constraint completely, while $\alpha = 1$ only incorporates a smoothing of the basis functions and $\beta = 1$ forces a full orthonormalization of the matrix. Previous studies [4] have shown that the results of proposed algorithm are robust over a wide range of values for α and β .

A second approach extends the GMLVQ cost function (4) to favor models with the desired functional properties. Two additional terms representing the discontinuity and non-orthogonality of Ψ are added to the cost function as

$$E \leftarrow E + \gamma \cdot d(\Psi) + \eta \cdot o(\Psi). \tag{10}$$

As a measure for the discontinuity $d(\Psi)$ represents the mean local (discrete) gradient of each basis function ψ_i

$$d(\Psi) = \frac{1}{n \cdot d} \sum_{i=1}^n \sum_{k=1}^{d-1} (\psi_i(k) - \psi_i(k+1))^2 \tag{11}$$

The orthogonality score $o(\Psi)$ is computed as

$$o(\Psi) = \sum_{i \neq j} [\Psi^\top \Psi]_{i,j}^2 \tag{12}$$

Similar to the first approach, the parameters γ and η control the behavior of the functional constraint. Both approaches are evaluated in the following.

3 Computer experiments

To validate the presented approaches, they are applied to four publicly available benchmark datasets comprising different kinds of functional data.

The *Tecator* dataset [37] is a prominent benchmark dataset for spectral data classification. It contains 215 reflectance spectra of meat probes in the range from 850 to 1050 nm wavelength. Each of the spectra has 100 sampling points and is labeled according to the fat content (high/low).

As a second dataset, we use the a subset of the *Sugar* dataset [23, 24]. The *Sugar* dataset is a collection of spectral information from various sugar and sugar-related chemical compounds, acquired with different sensors. In particular, we use the data acquired with the *FieldSpec* sensor which offers the widest spectral bandwidth ranging from 350 to 2500 nm wavelength. It contains 450 spectra labeled to represent one of the nine compounds. The spectra are sampled at a resolution of 1 nm resulting in 2151 sampling points per spectrum.

In addition to these two spectral datasets, two time series datasets are selected from the *UCR Time Series Repository* [11], namely the *Plane* and *Symbols* data sets. Apart from the key properties of the datasets given in *Table 1*, no further information on the interpretation of the data or associated classification tasks is available at the repository.

For each of the datasets, we consider four alternative scenarios for the classification.

Scenario A To obtain a natural baseline for the classification performance in a first experiment, the functional nature of the data is ignored and it is used directly for the training of a GMLVQ system.

Scenario B In a second set of experiments, the input data are expanded by fixed basis functions. To investigate and illustrate the influence of a proper choice Chebyshev polynomials of first kind (which yielded reasonable results in previous studies [25, 27])

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_k(x) &= 2xT_{k-1} - T_{k-2}(x), \end{aligned} \tag{13}$$

or Gaussian functions

$$g_k(x) = a_k \exp\left(-\frac{(x - b_k)^2}{2c_k^2}\right), \quad k = 0 \dots n - 1 \tag{14}$$

are employed. In both setups, the number of basis functions is varied as $n = 4, 8, 16, 32$. For the Gaussian functions, the parameters are chosen as $a_k = 1.0$, $b_k = -1 + 2k/n$ and $c_k = 0.25$, what yields an even distribution of the functions with respect to the functional input data space.

Scenario C As a third set of experiments, the proposed extension of GMLVQ introducing an adaptive functional basis Ψ is considered using the altered update rule introduced in (9). It is applied every 64 optimization steps. In analogy to *Scenario B* the size of Ψ and therefore the number of employed basis functions is varied $n = 4, 8, 16, 32$. To investigate the influence of different initializations of Ψ it is initialized as in *Scenario B* either by using Chebyshev polynomials of first kind or by distributing n Gaussian functions $g_k(x)$ evenly over the input space. The control parameters for the update rule are kept constant $\alpha = \beta = 0.25$, since previous studies [4] have shown minor influence on the results over a wide range of values.

Scenario D The final set of experiment resembles all experiments performed in *Scenario C* including all parameter variations but employs the altered cost function (10) instead of the modified update rule. The control parameters for the cost function are chosen as $\gamma = \eta = 0.25$. Note that the parameters γ and η serve as a weighting within the cost function. Assigning the same value to both parameters implies equal importance to orthogonality and smoothness of the emerging functional basis, which is a reasonable choice regarding the experiment.

Besides the different model setups, the same settings and parameters were used in the different scenarios. All GMLVQ systems comprised a single prototype per class. The distance matrix Ω was initialized proportional to the identity matrix and prototypes were initialized as the corresponding class conditional mean prior to the training.

Model parameters were optimized using a Python implementation of the Adam [20] optimizer included in the

Table 1 Key properties of the evaluated benchmark datasets

Name	Dimensions	Classes	Samples	
			Training	Validation
Tecator	100	2	215	
Sugar	2151	9	360	90
Plane	100	7	105	105
Symbols	398	6	25	995

PyTorch library [1, 29]. The step size for the optimization was initialized to 10^{-3} . In each experiment 2048 optimization steps were performed, to ensure final convergence of the model parameters

To avoid numerical problems, a row-wise normalization $\sum_{\mu=1}^d [\Psi]_{k,\mu}^2 = 1$ was applied. Normalization of Ω was kept unchanged as described in Sect. 2.1.

In all experiments, the classification performance of the resulting GMLVQ systems is evaluated by means of the overall correct classification rate on a validation set. For the *Sugar*, *Plane* and *Symbol* datasets dedicated validation datasets are provided along with the data. Results for the experiments on the *Tecator* dataset are obtained by averaging over 16 random splits of the dataset into 60% training and 40% validation data. The achieved classification accuracies are shown in Table 2.

To highlight different benefits of the proposed method, Figs. 1, 2 and 3 depict additional details of the resulting GMLVQ systems for the *Tecator* dataset.

4 Discussion

The observed classification accuracy as given in Table 2 for *scenarios B, C* and *D* shows that a functional representation of the data has the potential to increase the classification performance significantly with respect to the naive approach which ignores the functional nature of the data (*Scenario A*). For all examined datasets, the classification accuracy achieved using a functional expansion of the data exceeds the one obtained by training in the original input feature space. For the datasets *Tecator*, *Sugar* and *Symbols*, the best observed accuracy was achieved using the adaptive functional expansion (*Scenario C* and *D*). The best accuracy for the *Plane* dataset was observed in *Scenario B*, where a fixed functional basis was employed. Nevertheless, the use of an adaptive functional basis achieved classification performances exceeding the one achieved in *Scenario A*. As already discussed in [27], the positive effect of the functional representation of the input data is not only due to its noise reducing character. Therefore, the results confirm that a functional encoding of the input data yields a representation more suitable for prototype-based classification.

Furthermore, Table 2 shows that the resulting classification performance achieved in *Scenario C* is sensitive to the initialization of Ψ . The moving average filter introduced in the context of the functional constraint in (9) should void the influence of the initial basis choice. However, in the examined experimental setup, the functional constraint is applied every 64 optimization steps, so over the complete training of 2048 steps only 32 times.

Table 2 Achieved classification accuracy on the validation datasets for different scenarios

	Tecator	Sugar	Plane	Symbols
Scenario A	0.928	0.956	0.962	0.796
<i>Scenario B Chebyshev</i>				
$n = 4$	0.892	0.767	0.552	0.745
$n = 8$	0.918	0.944	0.943	0.779
$n = 16$	0.949	1.000	0.990	0.743
$n = 32$	0.946	1.000	0.962	0.817
<i>Scenario B Gaussian</i>				
$n = 4$	0.904	0.733	0.895	0.674
$n = 8$	0.908	0.933	0.981	0.738
$n = 16$	0.908	0.911	0.981	0.757
$n = 32$	0.906	0.822	0.981	0.757
<i>Scenario C Chebyshev</i>				
$n = 4$	0.902	1.000	0.952	0.816
$n = 8$	0.921	1.000	0.971	0.810
$n = 16$	0.937	0.967	0.962	0.809
$n = 32$	0.935	1.000	0.971	0.781
<i>Scenario C Gaussian</i>				
$n = 4$	0.931	1.000	0.952	0.772
$n = 8$	0.874	1.000	0.962	0.803
$n = 16$	0.901	1.000	0.962	0.813
$n = 32$	0.921	1.000	0.990	0.839
<i>Scenario D Chebyshev</i>				
$n = 4$	0.938	1.000	0.952	0.800
$n = 8$	0.936	1.000	0.981	0.796
$n = 16$	0.942	1.000	0.952	0.810
$n = 32$	0.943	1.000	0.952	0.820
<i>Scenario D Gaussian</i>				
$n = 4$	0.941	1.000	0.952	0.796
$n = 8$	0.938	1.000	0.952	0.808
$n = 16$	0.932	1.000	0.952	0.804
$n = 32$	0.931	1.000	0.962	0.812

All accuracies that exceed the accuracy achieved in *Scenario A* are highlighted in bold face. The highest observed accuracy for each dataset is additionally emphasized in bold italic

Therefore, the initialization of the functional expansion matrix Ψ is not eroded completely over the training period. *Scenario D* overcomes this problem by directly incorporating the desired functional characteristics of Ψ into an altered cost function, cf. (10). Since the smoothness constraint is taken into account in every optimization step, the influence of the initial values are voided comparatively fast. Therefore, the achieved classification performance is more stable with respect to the initialization of Ψ .

Besides the classification accuracy, the interpretability of machine learning models is a key issue in the design of reliable classification systems. Prototype-based models are

Fig. 1 Prototypes and relevance profiles obtained for the *Tecator* dataset from the different experimental setups. Only the prototype for the high fat class is shown. The different curves are labeled according to the experimental scenario. Besides, *Scenario A* all results were obtained with $n = 4$

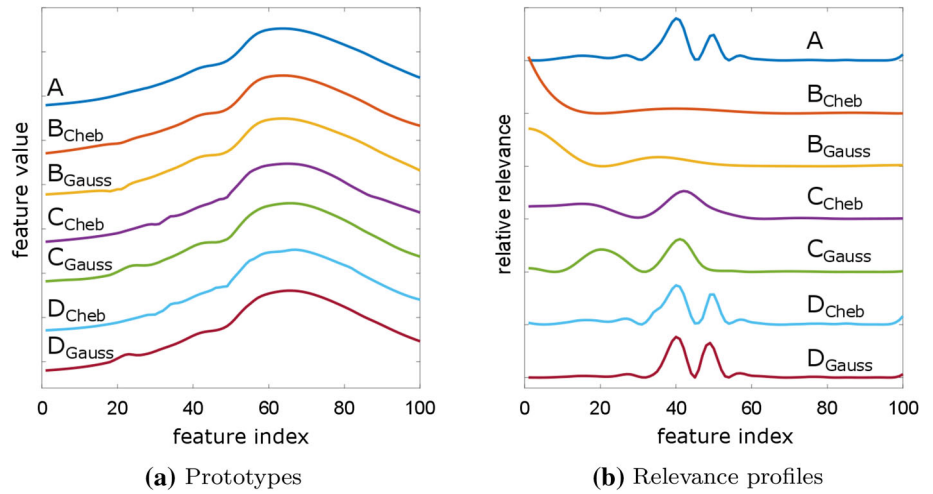


Fig. 2 Adapted basis functions obtained from *Scenario C* on the *Tecator* dataset and the corresponding basis relevance. Only the first (solid line), second (dashed line) and third (dotted line) basis functions are depicted. All depicted basis functions result from training with $n = 4$ after initialization of Ψ with Chebyshev polynomials

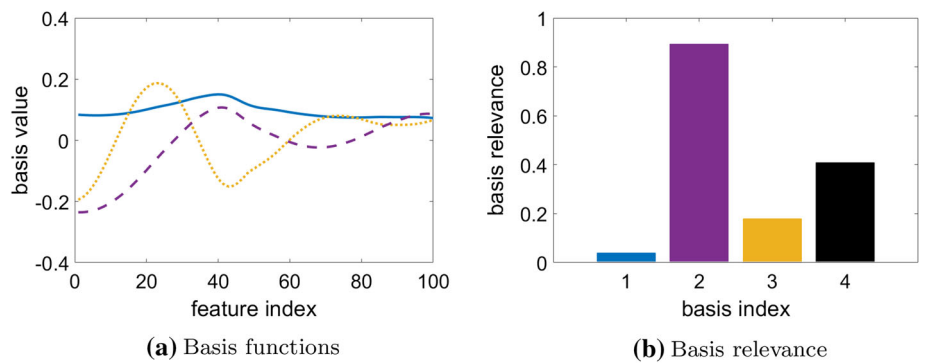
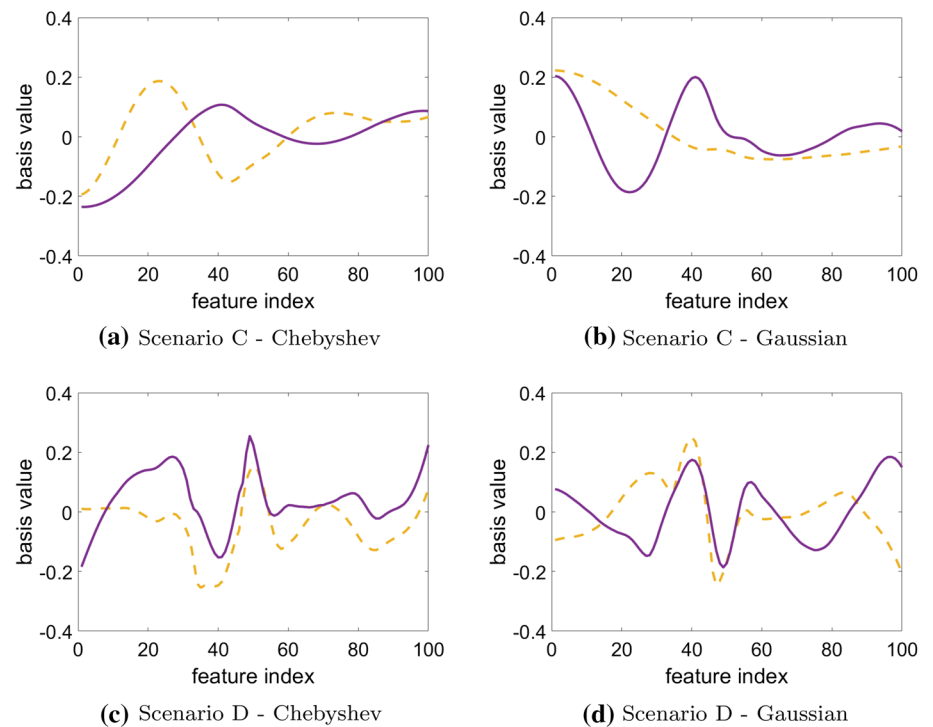


Fig. 3 Adapted basis function obtained from experiments on the *Tecator* dataset. Only the basis functions with the highest (solid line) and second highest (dashed line) assigned relevance are depicted for each scenario. All depicted basis functions result from training with $n = 4$



often straightforward to interpret, since prototypes, which serve as characteristic exemplars of their class, are adapted in the original feature space. To validate the plausibility of the obtained models incorporating a functional transformation, we analyze the prototypes and relevance profiles of the resulting GMLVQ systems.

In Fig. 1, the prototypes and the corresponding relevance profiles resulting from training on the *Tecator* dataset in the different scenarios are depicted. Please note that the relevance profile is obtained as $R = \text{diag}(A_V)$ for *Scenario A* and $R = \text{diag}(\Psi^\top A_c \Psi)$ for *scenarios B, C* and *D*, respectively. All obtained prototypes have a very similar shape.

The obtained relevance profiles, as illustrated in Fig. 1b, show more diversity with respect to the different scenarios. Note that the presented results were obtained using a set of only 4 basis functions. The limited capabilities of a small fixed functional basis to represent complex curves become clearly visible in the obtained relevance profiles of *Scenario B*. In the following scenarios the basis functions are adaptive, thus capable to represent more complex but data specific patterns even though only 4 basis functions are employed. All scenarios incorporating an adaptive functional basis create a relevance profile that is similarly interpretable as the one obtained in the original feature space. In particular, a concentration of relevance around feature index 40 is clearly visible in all obtained profiles. This observed maximum has been associated with a specific peak of spectral absorbance associated to fat alone in previous studies [30]. The relevance profiles resulting from *Scenario D* resemble the original relevance profile best. Not only does this emphasize the conversation of model interpretability, it also backs the reduced sensitivity with respect to the initialization of Ψ discussed before.

Please note that the analysis of model interpretability here assumes prototypes and relevance profile of *Scenario A* to be correct. A detailed discussion that the prototypes and relevance profiles reflect the underlying biochemistry most faithfully is beyond the scope of this contribution.

Since the functional basis (in *Scenarios C* and *D*) is adapted during training, it incorporates specific features of the data. In other words, we expect the functional basis not only to yield a good representation of the data, but also to give further insights into it. To follow this idea, it is essential to understand that the presented approach can be separated into three different aspects. During the training phase, representative examples for each class are determined (prototypes) while a distance measure is optimized. The distance measure itself is separated into the adaptation of a functional basis and a weighting of the different basis functions realized by the relevance matrix A .

As an illustrative example, the relation between the adapted basis functions and their particular relevance is given in Fig. 2. As a first interesting point, we observe the small relevance assigned to the first basis function which is represented by a solid line in Fig. 2a. This particular function is approximately constant and thus can be understood as an offset in the input data. A small relevance implies that this offset present in the data plays a minor role for the classification task, which confirms the results of previous studies [25]. The basis function with the highest relevance according to Fig. 2b is the second basis functions, which is depicted as a dashed line in Fig. 2a. This basis function clearly shows a linear trend superimposed by a local maximum centered at feature index 40. In accordance with the previous discussion, this emphasizes the importance of the fat-related spectral peak once more.

As stated before, the resulting classification performance is influenced by the initialization of Ψ . This motivates the question if the interpretation of the most relevant basis function for the *Tecator* dataset by means of the spectral characteristics of fat is based on a lucky coincidence. Figure 3 depicts the two most relevant basis functions obtained from the different combinations of Ψ initialization and adaptive learning rule. Although the specific shape of each set of basis functions is rather unique, the most relevant basis function displays the characteristic fat peak at feature index 40 in all experiments. The presence of this most prominent feature (with respect to the classification task) together with the high assigned relevance suggests that all of the obtained basis function sets are plausible.

Besides the superior classification accuracy and enhanced possibilities of model interpretation, the presented approach also reduces the computational effort for the model generation. The introduction of a rectangular matrix Ψ reduces the number of free parameters for the classification model drastically. While there are d^2 free parameters for a simple quadratic Ω in GMLVQ, with the presented approach the number reduces to $n \cdot d + n^2$. Note that this is only beneficial for $n < d(\sqrt{5} - 1)/2 \approx 0.62d$. Hence, for a reasonable number of basis functions the advantage of fewer free parameters persists. This does not only reduce the computational effort of the optimization process but also reduces the risk of overfitting and convergence problems.

5 Generalization of the learned basis

The proposed method to find a data- and problem-specific functional basis is tightly coupled to the optimization of a GMLVQ system. The observed classification performance benefits from the compact representation of the functional

data by the means of abstract approximation coefficients. Nevertheless, the question, if this beneficial effect is caused by the problem-specific functional basis or results from the combination with GMLVQ, remains.

To investigate this issue, we perform additional experiments in which the adapted functional basis is used in combination with other machine learning approaches. The original input data from the dataset are transformed to a new set of feature vectors according to (7) that is then used as input to the training of a classifier. We restrict the discussion to the *Sugar* dataset as an illustrative example for the generalization of the adapted functional basis, since it has shown the most impressive reduction of input dimensionality while preserving or even enhancing the achieved classification performance in the previous experiments.

A wide variety of classifiers is available each having different advantages and disadvantages. To cover a broad range, we selected four basic algorithms, namely k-nearest-neighbor (kNN) [14], multilayer perceptron (MLP) [22], linear support vector machine (SVM) [12] and decision tree (Tree) [9] as illustrative examples. Please note that an in-depth description of the mathematical background for all of the employed classifiers goes beyond the scope of this document. For details on the training and classification algorithms consult the according references.

For each of the classification algorithms, an additional set of computer experiments is performed. The first experiment serves, similar to the previous setup, as baseline for the classification performance and trains the classifier using the original input data without further preprocessing. In a second set of experiments, the input data are expanded by a predefined set of basis functions, namely Chebyshev polynomials of first kind and Gaussian functions, equivalent to *Scenario B* in the previous experiments. The number of basis functions is also varied as $n = 4, 8, 16, 32$. In a final set of experiments, the adapted functional basis from the experiments in *Scenario D* in Sect. 3 is employed to transform the input data into problem-specific approximation coefficients that are then used to optimize the classification system. The problem-specific expansion coefficients are obtained by transformation of the original feature vectors according (7). The employed transformation matrices Ψ result from the corresponding experiments in *Scenario D* of Sect. 3 with Chebyshev polynomial initialization. As in the previous experiments the size of the functional basis is varied as $n = 4, 8, 16, 32$. Please note that the transformation matrix Ψ serves in these experiments as a fixed functional basis and is not altered by any means during the training of the classifier.

Apart from the different representations of the input data, all experiments use the same set of parameters. For the kNN classifier $k = 1$ together with the squared Euclidean distance serves as the simplest classifier possible. The

MLP model employs a single hidden layer containing 16 neurons with sigmoidal activation. Since SVMs are only suited for binary classification problems, an ensemble of linear kernel SVMs is trained using error-correcting output codes [13] to map the nine different classes to multiple binary classification tasks. A standard classification and regression tree (CART) algorithm introduced in [9] employing Gini's diversity index [15] is used to construct the decision tree classifier.

All other parameters for the classification are kept at the default values specified in the corresponding MATLABTM toolboxes. The used implementation for kNN, SVM and decision tree classification is provided by the Statistics and Machine Learning Toolbox (v10.2), while the MLP is provided by the Neural Network Toolbox (v9.0).

Equivalent to the previous experiments the classification performance of the different algorithms is measured by the accuracy with respect to the validation set provided with the input data. All experiments are repeated 64 times to account for random model initializations and the mean accuracy over all experiments is evaluated.

Table 3 shows the observed accuracies. The table results indicate that a superior classification performance is achieved for all examined classifiers when a functional representation of the data is used. The use of a generic functional basis can be beneficial, but the performance is highly dependent on the selected basis function as well as the classification algorithm. While the use of Chebyshev polynomials of first kind can be beneficial for the kNN, Tree and ANN classifier, the achieved accuracy is significantly harmed for the SVM classification system. Even worse, all of the evaluated classification algorithms showed poor performance using simple Gaussian basis functions. This observation confirms the hypothesis that Gaussian basis functions are not suitable for a classification focused approximation of the *sugar* dataset.

Using the learned problem-specific functional basis outperforms the classification in the original input space in all experiments. The constant superior performance with respect to the classification in the original input space proves that the functional basis provides a meaningful, low-dimensional and problem-specific feature space which is not limited to use of GMLVQ classifiers. The drastic reduction in input dimensions (here $d = 2151 \rightarrow n = 8$) reduces the complexity of the resulting classifiers significantly.

Table 3 Validation accuracy for different classification algorithms achieved on the *Sugar* dataset

	GMLVQ	kNN	SVM	Tree	ANN
Original input	0.956	0.878	0.911	0.756	0.872
<i>Fixed basis Chebyshev</i>					
$n = 4$	0.767	0.589	0.778	0.556	0.705
$n = 8$	0.944	0.744	0.889	0.578	0.781
$n = 16$	1.000	0.844	0.911	0.889	0.839
$n = 32$	1.000	0.867	0.911	0.889	0.873
<i>Fixed basis Gaussian</i>					
$n = 4$	0.733	0.611	0.767	0.611	0.683
$n = 8$	0.933	0.589	0.789	0.633	0.681
$n = 16$	0.911	0.556	0.822	0.667	0.699
$n = 32$	0.822	0.556	0.856	0.644	0.695
<i>Adaptive basis</i>					
$n = 4$	1.000	1.000	1.000	0.956	1.000
$n = 8$	1.000	1.000	1.000	1.000	1.000
$n = 16$	1.000	1.000	1.000	0.867	0.995
$n = 32$	1.000	0.967	0.933	1.000	0.991

All accuracies that exceed the accuracy achieved on the original input data are highlighted in bold face. The highest observed accuracy for each classification algorithm is additionally emphasized in bold italic

6 Summary and outlook

This contribution proposes an extension of GMLVQ for functional data that takes the special characteristics of the input data into account by using an adaptive functional expansion. We have shown that this approach has the potential to improve overall classification accuracy. For all of the examined benchmark datasets, models using an adaptive functional basis outperformed the classification accuracy achieved on the raw input data.

Furthermore, the interpretation of the adapted functional basis can give additional insights into the structure of the input. For the *Tecator* dataset, one of the learned basis functions can be interpreted as a data offset, while other basis functions can be associated with the spectrum of fatty tissue. The experiments have shown that the presented approach has the potential to decompose signal into different compounds with respect to the classification task. These findings may motivate a new approach for the recovery of spectral end-members that overcomes the assumption that end-members have to be present in the data, like in [3].

In combination with the relevance scheme used in GMLVQ, the importance of different data characteristics with respect to the classification task is weighted. For the *Tecator* example, the data offset is almost ignored with respect to the classification task, while the basis function associated with the fat spectrum is assigned the highest relevance.

In addition to the separate interpretation of functional basis and assigned relevances, it is possible to map the model parameters to the original feature space, which facilitates a direct analysis of relevances in the space of input data. Due to the limited number of basis functions employed, a perfect reconstruction is not possible and the back projected relevances (and relevance matrices) can only serve as an estimation for the input space relevances. Nevertheless, our experiments have shown that the general shape of the relevance profiles is preserved.

Therefore, the presented approach does not only allow for the interpretation of model parameters in the space of functional coefficients, but preserves the interpretability of prototypes and the relevance matrix in the original input space. Additionally, the introduction of the rectangular matrix Ψ drastically decreases the number of free model parameters. This leads to a reduced computational effort for the model generation and a lower risk to encounter overfitting effects and convergence problems.

In additional experiments, the use of the adapted functional basis in combination with different classification algorithms, namely k-nearest-neighbor, support vector machines, decision trees and artificial neural networks, was investigated. All classifiers have shown superior performance to the brute-force approach when trained on the data projected by the adaptive basis. In contrast, the classification performance in combination with generic basis functions could not significantly exceed the one on the original input data. This proves that the adapted set of basis

functions provides a meaningful, data- and problem-specific domain with respect to the classification task.

Still there are various questions remaining to be addressed in future research. For the introduction of the adaptive functional basis, the number of basis functions has to be specified in advance. The performed computer experiments have shown that the approach is sensitive to the size of the functional basis and an increase in the number of basis functions can lead to a decrease in classification performance. A consistent procedure to estimate the number of necessary basis functions is desirable.

Furthermore, the experiments have shown that different initial values for Ψ as well as different optimization criteria lead to significant differences in the adapted functional basis. The question if there is a unique optimal set of basis functions, given a set of training data and a classification task, remains. For the experiments performed in the scope of this work, the optimization was canceled after a defined number of optimization steps. It is possible that the convergence of Ψ toward a unique optimal solution was therefore aborted too early.

Finally, the presented results are obtained on a comparatively low number of example datasets and classification algorithms. With the increasing availability of annotated functional datasets, the proposed approach should be further validated.

Acknowledgements F. Melchert thanks for support through an Ubbo-Emmius Sandwich Scholarship by the Faculty of Science and Engineering.

Compliance with ethical standards

Conflict of interest F. Melchert has received an Ubbo-Emmius Sandwich Scholarship by the Faculty of Science and Engineering of the University of Groningen, The Netherlands.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. pytorch—tensors and dynamic neural networks in python with strong gpu acceleration. <http://pytorch.org/>. Accessed 22 Jan 2015
2. Backhaus A, Seiffert U (2014) Classification in high-dimensional spectral data: accuracy vs. interpretability vs. model size. *Neurocomputing* 131:15–22
3. Bandfield JL, Christensen PR, Smith MD (2000) Spectral data set factor analysis and end-member recovery—application to analysis of Martian atmospheric particulates. *J Geophys Res* 105(E4):9573–9587
4. Bani G, Seiffert U, Biehl M, Melchert F (2017) Adaptive basis functions for prototype-based classification of functional data. In: 2017 12th International workshop on self-organizing maps and learning vector quantization, clustering and data visualization (WSOM), pp 145–152
5. Biehl M (2016) A no-nonsense beginner's tool for GMLVQ. University of Groningen. <http://www.cs.rug.nl/~biehl/gmlvq>. Accessed 17 Oct 2016
6. Biehl M, Hammer B, Villmann T (2014) Distance measures for prototype based classification. In: Grandinetti L, Petkov N, LipPERT I (eds) *BrainComp 2013, Proceedings international workshop on brain-inspired computing*, Cetraro/Italy, 2013, Lecture Notes in Computer Science, vol 8603, pp 100–116. Springer
7. Biehl M, Hammer B, Villmann T (2016) Prototype-based models in machine learning. *Wiley interdisciplinary reviews. Cognit Sci* 7(2):92–111
8. Bojer T, Hammer B, Schunk D, von Toschanowitz K.T (2001) Relevance determination in learning vector quantization. In: Verleysen M (ed) *European symposium on artificial neural networks ESANN 2001*, pp 271–276. d-facto
9. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) *Classification and regression trees*. CRC Press, Boca Raton
10. Bunte K, Schneider P, Hammer B, Schleif FM, Villmann T, Biehl M (2012) Limited rank matrix learning, discriminative dimension reduction and visualization. *Neural Netw* 26:159–173
11. Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/. Accessed 6 Jan 2016
12. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297. <https://doi.org/10.1007/bf00994018>
13. Dietterich TG, Bakiri G (1995) Solving multiclass learning problems via error-correcting output codes. *J Artif Intell Res* 2:263–286
14. Fix E, Hodges JL Jr (1951) Discriminatory analysis-nonparametric discrimination: consistency properties. California Univ Berkeley, Technical report
15. Gini C (1912) Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche. Studi economico-giuridici pubblicati per cura della facoltà di Giurisprudenza della R. Università di Cagliari. Tipogr. di P. Cuppini
16. Gregorutti B, Michel B, Saint-Pierre P (2015) Grouped variable importance with random forests and application to multiple functional data analysis. *Comput Stat Data Anal* 90:15–35
17. Hammer B, Villmann T (2002) Generalized relevance learning vector quantization. *Neural Netw* 15:1059–1068
18. Jia F, Lei Y, Guo L, Lin J, Xing S (2018) A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines. *Neurocomputing* 272:619–628
19. Jia F, Lei Y, Lu N, Xing S (2018) Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mech Syst Signal Process* 110:349–367
20. Kingma D, Ba J (2015) Adam: a method for stochastic optimization. In: *Proceedings of international conference learning represent*
21. Kohonen T (1995) *Self-organizing maps*. Springer, Berlin
22. Lippmann RP (1987) An introduction to computing with neural nets". *IEEE Assp Mag* 4(2):4–22
23. Melchert F, Matros A, Biehl M, Seiffert U (2016) The sugar dataset—a multimodal hyperspectral dataset for classification and research. In: Schleif EM, Villmann I (eds) *MIWOCI workshop. Machine Learning Reports*, vol 03
24. Melchert F, Matros A, Biehl M, Seiffert U (2018) The sugar dataset. Online Release. <http://github.com/melchert/sugarDataset>

25. Melchert F, Seiffert U, Biehl M (2015) Polynomial approximation of spectral data in LVQ and relevance learning. *Machine Learning Reports*, vol 03, 2015, pp 25–32. Bielefeld University
26. Melchert F, Seiffert U, Biehl M (2016) Functional approximation for the classification of smooth time series, pp 24–31. *Machine Learning Reports*. University of Bielefeld
27. Melchert F, Seiffert U, Biehl M (2016) Functional representation of prototypes in LVQ and relevance learning. In: *Advances in self-organizing maps and learning vector quantization*, pp 317–327. Springer
28. Papari G, Bunte K, Biehl M (2011) Waypoint averaging and step size control in learning by gradient descent. *Machine Learning Reports*. MLR-06/2011, 16
29. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in PyTorch. In: *NIPS 2017 autodiff workshop: the future of gradient-based machine learning software and techniques*, Long Beach, CA, US
30. Rossi F, Conan-guez B, Golli AE (2004) Clustering functional data with the SOM algorithm. *Proc ESANN 2004*:305–312
31. Rossi F, Villa N (2006) Support vector machine for functional data classification. *Neurocomputing* 69(7–9):730–742
32. Sato A, Yamada K (1995) Generalized learning vector quantization. In: Tesauro G, Touretzky D, Leen T (eds) *Advances in neural information processing systems*, vol 7. MIT Press, Cambridge, pp 423–429
33. Schmidt E (1908) Über die Auflösung linearer Gleichungen mit unendlich vielen Unbekannten. *Rendiconti del Circolo Matematico di Palermo (1884–1940)* 25(1):53–77
34. Schneider P, Biehl M, Hammer B (2009) Adaptive relevance matrices in learning vector quantization. *Neural Comput* 21:3532–3561
35. Schneider P, Biehl M, Schleif F.M, Hammer B (2007) Advanced metric adaptation in Generalized LVQ for classification of mass spectrometry data. In: *Proceedings of 6th international workshop on self-organizing-maps (WSOM)*. Bielefeld University. 5 pages
36. Strickert M, Hammer B, Villmann T, Biehl M (2013) Regularization and improved interpretation of linear data mappings and adaptive distance measures. In: *2013 IEEE symposium on computational intelligence and data mining (CIDM)*, pp 10–17
37. Thodberg HH (2015) Tecator meat sample dataset. *StatLib Datasets Archive*. <http://lib.stat.cmu.edu/datasets/tecator>. Accessed 29 Oct 2015

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.