Prototypes and Matrix Relevance Learning in Complex Fourier Space

M. Straat*, M. Kaden†, M. Gay†§, T. Villmann†, A. Lampe†, U. Seiffert‡, M. Biehl*, and F. Melchert*‡
*University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands
†University of Applied Sciences Mittweida, Computational Intelligence Group, Technikumplatz 17, 09648 Mittweida, Germany
‡Fraunhofer Institute for Factory Operation and Automation IFF,
Sandtorstrasse 22, 39106 Magdeburg, Germany
§Fraunhofer Institute for Transportation and Infrastructure Systems IVI,
Zeunerstrasse 38, 01069 Dresden, Germany

***Abstract*—In this contribution, we consider the classification of time-series and similar functional data which can be represented in complex Fourier coefficient space. We apply versions of Learning Vector Quantization (LVQ) which are suitable for complex-valued data, based on the so-called Wirtinger calculus. It makes possible the formulation of gradient based update rules in the framework of cost-function based Generalized Matrix Relevance LVQ (GMLVQ). Alternatively, we consider the concatenation of real and imaginary parts of Fourier coefficients in a real-valued feature vector and the classification of time domain representations by means of conventional GMLVQ.**

***Keywords*—Classification, supervised learning, functional data, Learning Vector Quantization, relevance learning, dimensionality reduction**

I. Introduction

Time series constitute an important example of *functional data* [1]: Their time-domain discretized vector representations comprise components which reflect the temporal order and are often highly-correlated over characteristic times. This is in contrast to more general datasets, where the feature vectors are concatenations of more or less independent quantities and their order does not play a role.

The machine learning analysis of time series data, e.g. for the purpose of classification, should take into account their functional nature. Recently, prototype-based systems have been put forward, which employ the representation of data and prototypes in terms of suitable basis functions [2], [3]. In addition, corresponding adaptive distance measures can be defined and trained in the space of expansion coefficients [4], [5], [6]. Hence, the functional nature of data is taken advantage of, explicitly. At the same time, it is possible to compress high-dimensional data by functional approximations, thus reducing computational effort and - potentially - the risk of over-fitting.

Examples of the basic approach include the application of wavelet representations of mass spectra [7] and polynomial expansions of smooth functional data [2], [3].

In the context of signal processing, the Discrete Fourier transform (DFT) to the frequency domain is a popular tool which can be applied to time series or more general, sequential data. In the following the discussion is presented mostly in terms of actual time series, but it is understood that methods and results would carry over to suitable sequential data from other contexts.

The standard formulation of the DFT resorts to the determination of complex coefficients, conveniently. Hence, we suggest and study the combination of DFT functional representations with the extension of GMLVQ [8], [9] to complex feature space [10].

We present furthermore the formalism to back-transform the resulting prototypes and relevance matrix to the time domain, thus retaining the intuitive interpretability of the LVQ approach.

We apply the suggested framework to a number of benchmark datasets [11] and study, among other aspects, the dependence of the performance on the approximation quality, i.e. the number of coefficients considered.

In addition, we compare performance with an approach that resorts to the concatenation of the imaginary and real parts of coefficients in a real-valued feature vector. The application of conventional GMLVQ classification in the time domain serves as an important baseline for comparison of performances and for the interpretation of the obtained relevance matrices.

II. The mathematical framework

In this section we present the mathematical framework that underlies the method. This consists of the Discrete Fourier-Transform (DFT), the adaptation of the machine learning algorithm GMLVQ for complex-valued data using Wirtinger calculus [12] and the backtransformation of the classifier that retains the intuitiveness in the original time-domain of the data.

A. Discrete Fourier transform

Sampling a continuous process $f(t)$ with sampling interval ΔT results in a high-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^N$ containing the values of $f(t)$ at the sampling times, $f(i\Delta T), i = 0, 1, ..., N-1$. The time domain vector $\boldsymbol{x} \in \mathbb{R}^N$ can be written as a linear combination of sampled complex sinusoids:

$$\boldsymbol{x}[t] = \sum_{k=0}^{N-1} \boldsymbol{x}_f[\omega_k] e^{-j2\pi tk/N}, t = 0, 1, 2, ..., N-1, \quad (1)$$

where the coefficients $\boldsymbol{x}_f[\omega_k] \in \mathbb{C}$ can be calculated efficiently by the DFT [13]:

$$\boldsymbol{x}_f[\omega_k] = \sum_{t=0}^{N-1} \boldsymbol{x}[t] e^{-j2\pi kt/N}\,, k = 0, 1, 2, ..., N-1\,. \quad (2)$$

As in Eq. (1) and Eq. (2) and the rest of the discussion, the subscript f is used to denote a vector or matrix in the Fourier domain. As can be observed in Eq. (2), the transformed feature vectors consist of N coefficients. It should be noted that the coefficients of $\boldsymbol{x}_f[\omega_k]$ are conjugate symmetric and therefore all the information is contained in the first $\lfloor N/2 \rfloor + 1$ coefficients: $\boldsymbol{x}_f[\omega_k], k = 0, 1, ..., \lfloor N/2 \rfloor$. By restricting the number of coefficients to a number $n < \lfloor N/2 \rfloor + 1$ in Eq. (1), an approximation $\hat{\boldsymbol{x}}[t]$ of the original time domain vector $\boldsymbol{x}[t]$ is obtained. Note that for the purpose of classification, in some datasets the discriminative information may lie in the higher band of frequencies as well. However, in this contribution we consider smooth versions of the time series which are obtained from cutting off high frequencies.

Note that according to Eq. (2), the computation of a single coefficient $\boldsymbol{x}_f[\omega_k] \in \mathbb{C}$ for the kth frequency is defined as the dot product between the time domain vector $\boldsymbol{x}[t]$ and the sampled complex sinusoid of the kth frequency, $g_k[t] = e^{-j2\pi kt/N}$. We could therefore equivalently write the transformation in Eq. (2) as a matrix equation:

$$\boldsymbol{x}_f[\omega] = \boldsymbol{F}\boldsymbol{x}\,, \quad (3)$$

where $\boldsymbol{x}_f[\omega] \in \mathbb{C}^n$ is the complex Fourier approximation of $\boldsymbol{x} \in \mathbb{R}^N$ truncated at n frequency coefficients and $\boldsymbol{F} \in \mathbb{C}^{n \times N}$ is the transformation matrix where the sampled complex sinusoids appear on the rows. The multiplication with $\boldsymbol{F}$ in Eq. (3) could be done using the FFT, which reduces computational cost to $O(N \log N)$, as compared to computing the DFT directly as it is defined in Eq. (3) which has a cost of $O(N^2)$ in case of a DFT that considers all the N frequencies.

B. GMLVQ with Wirtinger calculus

Having transformed the data to Fourier space as described in the previous section, we consider a classification setup in which GMLVQ works directly on complex-valued data, following the prescription outlined in [6]. In our case the complex-valued data vectors are frequency domain representations of the time series obtained by means of the DFT and therefore we use the f subscript for the vectors. Let the dataset consist of labeled feature vectors $(\boldsymbol{x}_f, y) \in \mathbb{C}^n \times \{1, ..., C\}$, i.e. each feature vector $\boldsymbol{x}_f \in \mathbb{C}^n$ being a member of one of the C distinct classes in the dataset. During the GMLVQ training process, complex-valued prototypes $\boldsymbol{w}_f \in \mathbb{C}^n$ representing the classes in the dataset are learned and a quadratic distance measure $d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f, \boldsymbol{w}_f]$ parameterized by a matrix $\boldsymbol{\Lambda} = \boldsymbol{\Omega}^T\boldsymbol{\Omega}$ is adapted according to the relevance of the features. In the end we obtain a classifier defined in terms of distance measure $d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f, \boldsymbol{w}_f]$ and the set of prototypes $\boldsymbol{W} = \{\boldsymbol{w}_f^1, \boldsymbol{w}_f^2, ..., \boldsymbol{w}_f^K\}$, where in the case of multiple prototypes per class $K > C$. A novel data point $\boldsymbol{x}_f^\mu$ is then assigned the class label of the nearest prototype according to the learned distance measure $d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f^\mu, \boldsymbol{w}_f]$.

Given an example $(\boldsymbol{x}_f^\mu, c = j)$ of class j, the closest prototype of the same class $(\boldsymbol{w}_f^+, c = j)$ and the closest prototype of a different class $(\boldsymbol{w}_f^-, c \neq j)$, the cost for example $\boldsymbol{x}_f^\mu$ in cost-function based GMLVQ is defined as:

$$e^\mu = \frac{d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f^\mu, \boldsymbol{w}_f^+] - d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f^\mu, \boldsymbol{w}_f^-]}{d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f^\mu, \boldsymbol{w}_f^+] + d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f^\mu, \boldsymbol{w}_f^-]} \in [-1, 1]\,. \quad (4)$$

The total cost is then the sum of the individual cost contributions of all data points:

$$E = \sum_\mu e^\mu\,. \quad (5)$$

Note that, for simplicity, we refrain from introducing a non-linear function $\Phi(e^\mu)$ in the sum, as originally suggested in [14].

Upon presentation of vector $\boldsymbol{x}_f^\mu$, the prototypes $\boldsymbol{w}_f^+$, $\boldsymbol{w}_f^-$ and the matrix $\boldsymbol{\Omega}$ are adapted according to steepest descent of the cost function:

$$\boldsymbol{w}_f^+ := \boldsymbol{w}_f^+ - \eta \nabla_{\boldsymbol{w}_f^+} e^\mu\,, \quad (6)$$

$$\boldsymbol{w}_f^- := \boldsymbol{w}_f^- - \eta \nabla_{\boldsymbol{w}_f^-} e^\mu\,, \quad (7)$$

$$\boldsymbol{\Omega} := \boldsymbol{\Omega} - \eta \nabla_{\boldsymbol{\Omega}} e^\mu\,. \quad (8)$$

Derivations of the gradients with respect to complex-valued $\boldsymbol{w}_f^+$, $\boldsymbol{w}_f^-$ and $\boldsymbol{\Omega}$ as appear in the above equations can be found in [10], [15]. In [10] the learning rules for updating the prototypes $\boldsymbol{w}_f^+$ and $\boldsymbol{w}_f^-$ and adaptive distance matrix $\boldsymbol{\Lambda}$ used in cost-function based GMLVQ are formulated for complex-valued data, relying on the mathematical formalism of Wirtinger calculus [12] for the computation of the gradients which yields intuitive adaptation rules. Note that $\nabla_{\boldsymbol{w}_f^+} e^\mu = \frac{\partial e^\mu}{\partial d_{\boldsymbol{\Lambda}}} \frac{\partial d_{\boldsymbol{\Lambda}}}{\partial \boldsymbol{w}_f^+}$, and therefore only the inner gradient is taken with respect to complex variables, for which Wirtinger calculus is used. The distance between a data vector $\boldsymbol{x}_f \in \mathbb{C}^n$ and a prototype $\boldsymbol{w}_f \in \mathbb{C}^n$ is defined as:

$$d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f, \boldsymbol{w}_f] = (\boldsymbol{x}_f - \boldsymbol{w}_f)^H \boldsymbol{\Omega}^H \boldsymbol{\Omega} (\boldsymbol{x}_f - \boldsymbol{w}_f)\,, \quad (9)$$

where $\boldsymbol{A}^H$ denotes the Hermitian transpose of a matrix, which is obtained by the transpose operation on $\boldsymbol{A}$ and the complex conjugation of each element A_{ij}.

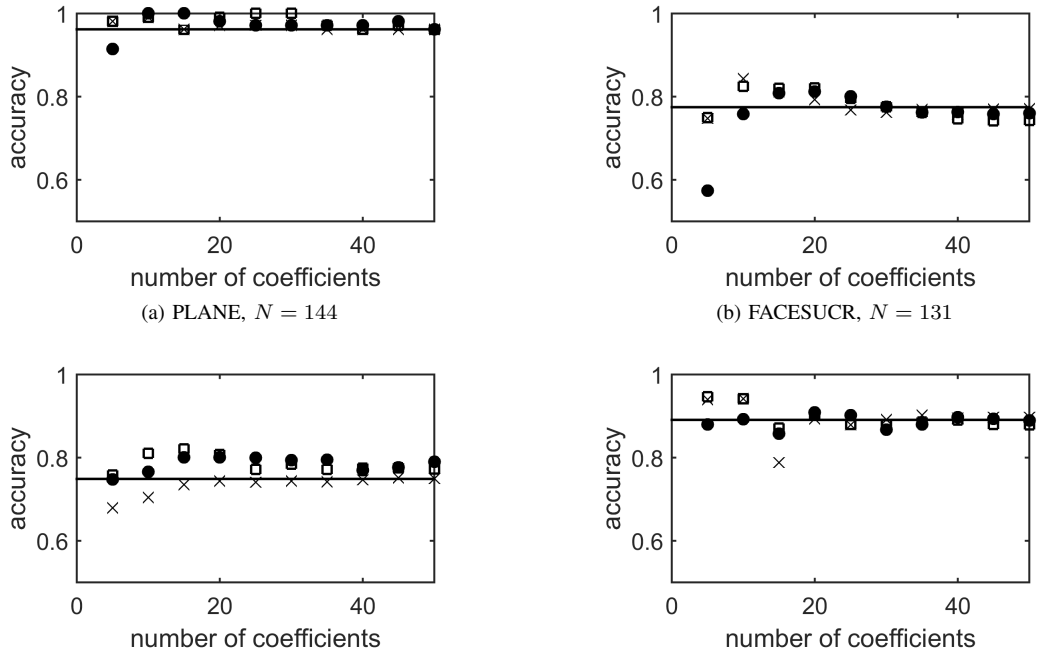
The gradient of $d_{\boldsymbol{\Lambda}}$ with respect to complex prototype $\boldsymbol{w}_f \in \mathbb{C}^n$ is then, using the Wirtinger gradient, intuitively formulated as:

$$\nabla_{\boldsymbol{w}_f^*} d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f, \boldsymbol{w}_f] = -\boldsymbol{\Omega}^H \boldsymbol{\Omega} (\boldsymbol{x}_f - \boldsymbol{w}_f)\,. \quad (10)$$

The gradient of $d_{\boldsymbol{\Lambda}}$ w.r.t. matrix $\boldsymbol{\Omega}$ is defined as:

$$\nabla_{\boldsymbol{\Omega}^*} d_{\boldsymbol{\Lambda}}[\boldsymbol{x}_f, \boldsymbol{w}_f] = \boldsymbol{\Omega} (\boldsymbol{x}_f - \boldsymbol{w}_f)(\boldsymbol{x}_f - \boldsymbol{w}_f)^H\,. \quad (11)$$

A comparison of the above gradients for complex-valued data with the gradients for real-valued GMLVQ [8], [9] reveals that the two are formally very similar, and therefore naturally, by substitution of the gradients of the complex variables into equations (6), (7), (8), the learning rules for prototypes $\boldsymbol{w}_f^+$ and $\boldsymbol{w}_f^-$ and relevance matrix $\boldsymbol{\Lambda}$ in the complex case are formally similar to the learning rules in the real case.



(a) PLANE, $N = 144$ (b) FACESUCR, $N = 131$

(c) SYMBOLS, $N = 398$ (d) MALLAT, $N = 1024$

Fig. 1. Percentage of correctly classified vectors in the test sets for each dataset. The solid line represents the classification result in the original time domain of the data. *Filled circles* show the classification accuracy in the n-coefficient complex Fourier space of the data. *Empty squares* show the classification accuracy in the n-coefficient Fourier space where the real and imaginary parts of the complex features are concatenated yielding real feature vectors. *Crosses* show the classification accuracy on the smooth data in the original space that was obtained by an inverse transform of the Fourier representation. For each dataset the number of dimensions N of the original feature vectors is indicated.

C. Backtransformation

Training on the data in complex Fourier space as described in the previous section yields complex-valued prototypes $\boldsymbol{w}_f \in \mathbb{C}^n$ and relevance matrix $\boldsymbol{\Lambda}_f \in \mathbb{C}^{n\times n}$. A transformation of the prototypes to the time domain using the inverse Discrete Fourier Transform (iDFT) retains the time domain intuitiveness of the prototypes [13]:

$$\boldsymbol{w}[t] = \frac{1}{N}\sum_{k=0}^{N-1} \boldsymbol{w}_f[k] e^{j2\pi tk/N}, t = 0, 1, 2, ..., N-1\,. \quad (12)$$

We further note that the distance measure in Fourier space can be written in terms of the Fourier transformation matrix $\boldsymbol{F}$:

$$d[\boldsymbol{x}_f, \boldsymbol{w}_f] = (\boldsymbol{x}-\boldsymbol{w})^H \boldsymbol{F}^H \boldsymbol{\Lambda}_f \boldsymbol{F}(\boldsymbol{x}-\boldsymbol{w})\,, \quad (13)$$

where $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{w} \in \mathbb{R}^N$ are vectors in the time domain. By Eq. (13), the matrix $\boldsymbol{\Lambda} = \boldsymbol{F}^H \boldsymbol{\Lambda}_f \boldsymbol{F}$ yields a time domain interpretation of the feature relevances.

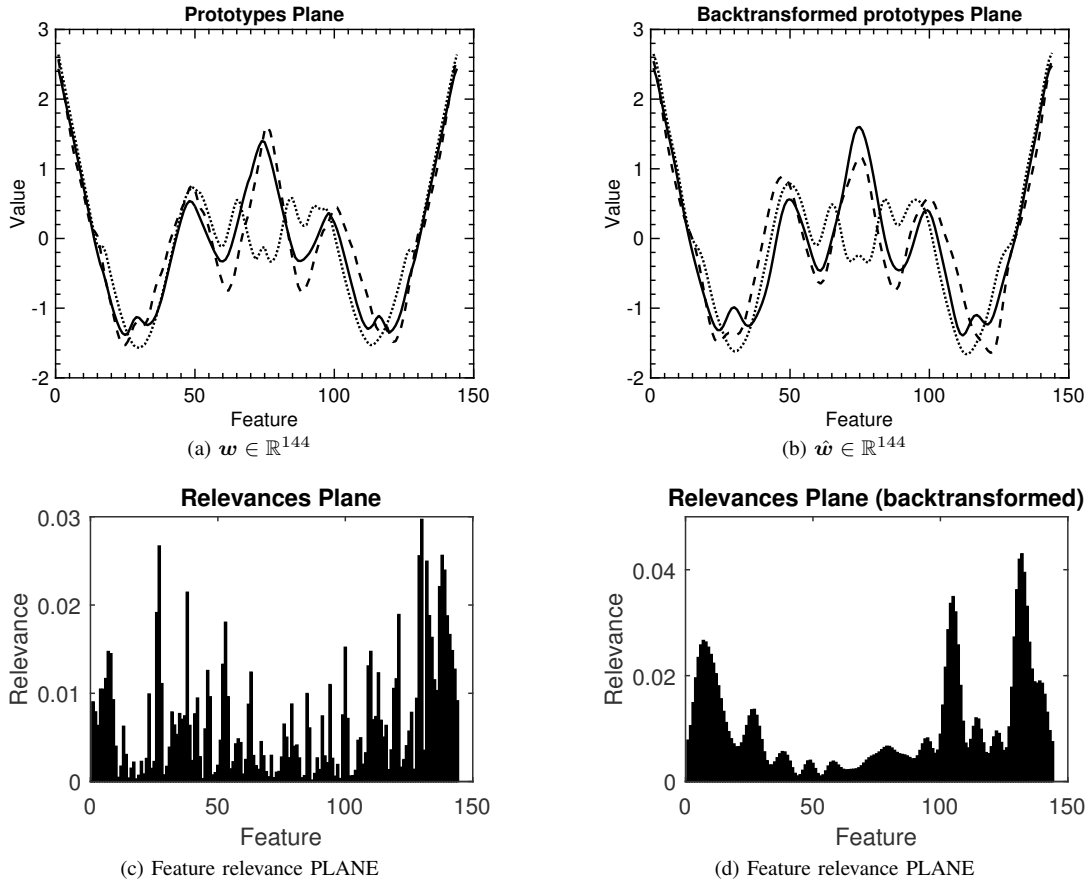
As was stated before, GMLVQ could also be used directly on the data in the time-domain. Besides the potential of improving classification accuracy, we must note that our approach also has the ability to reduce the number of parameters in GMLVQ ($n < N$) and at the same time keep the time-domain intuitiveness. Since the number of parameters in GMLVQ scales quadratically with the number of dimensions, the computational effort in the training process is considerably reduced.

III. Experiments

A. Workflows

For our investigation into the usefulness and performance of the proposed method, we compare and study the results for the following scenarios:

1) Train a GMLVQ system using the feature vectors $\boldsymbol{x} \in \mathbb{R}^N$ in the original time domain and evaluate the system on the test data. This serves as the baseline performance. Note that it is required that $\lfloor N/2 \rfloor + 1 \geq n_{max}$ (see Section III-C).
2) Transform the feature vectors to complex Fourier space truncated at different numbers of Fourier coefficients $n = [6, 11, ..., 51]$ yielding feature vectors $\boldsymbol{x}_f \in \mathbb{C}^n$. On each of these representations a GMLVQ system is trained. The training results in a classifier defined by prototypes $\boldsymbol{w}_f \in \mathbb{C}^n$ and complex relevance matrix $\boldsymbol{\Lambda}_f \in \mathbb{C}^{n\times n}$, which is evaluated on the corresponding test set.
3) As in scenario 2, transform the data to complex Fourier space truncated at $n = [6, 11, ..., 51]$ coefficients obtaining vectors $\boldsymbol{x}_f \in \mathbb{C}^n$, but here we consider the representation that concatenates the real and imaginary parts forming real-valued feature vectors $\boldsymbol{x}_f = \begin{bmatrix} \Re(\boldsymbol{x}_f) \\ \Im(\boldsymbol{x}_f) \end{bmatrix} \in \mathbb{R}^{2n}$. We train a GMLVQ system on each of these representations resulting in a classifier defined by prototypes $\boldsymbol{w}_f \in \mathbb{R}^{2n}$ and a real-valued relevance matrix $\Lambda_C \in \mathbb{R}^{2n\times 2n}$, which is evaluated on the corresponding test set.



(a) $\boldsymbol{w} \in \mathbb{R}^{144}$ (b) $\hat{\boldsymbol{w}} \in \mathbb{R}^{144}$

(c) Feature relevance PLANE (d) Feature relevance PLANE

Fig. 2. In Figure 2a, the resulting class prototypes of the PLANE dataset are shown for training in the original 144-dimensional space. For clarity, only three of the seven prototypes are shown. The corresponding feature relevances, which are the diagonal elements of the resulting relevance matrix for the PLANE dataset, are shown in Figure 2c. In Figure 2b the backtracktransformed prototypes obtained from training in 20-coefficient Fourier space are shown. Figure 2d shows the corresponding feature relevances, obtained from backtransforming the complex relevance matrix via the method discussed in Section II-C.

4) Transform the feature vectors $\boldsymbol{x} \in \mathbb{R}^N$ to Fourier space for the same numbers $n = [6, 11, ..., 51]$ of coefficients as in scenarios 2 and 3, after which the data are transformed back to the original space yielding feature vectors $\hat{\boldsymbol{x}} \in \mathbb{R}^N$, which are smoothed versions of the original feature vectors. The GMLVQ systems are now trained and evaluated on these smoothed feature vectors in the time domain. The comparison of the obtained performance with the performance of scenarios B and C allows an estimate of the performance gain that results from the noise reduction caused by the truncation of high frequencies.

B. Training settings and parameter values

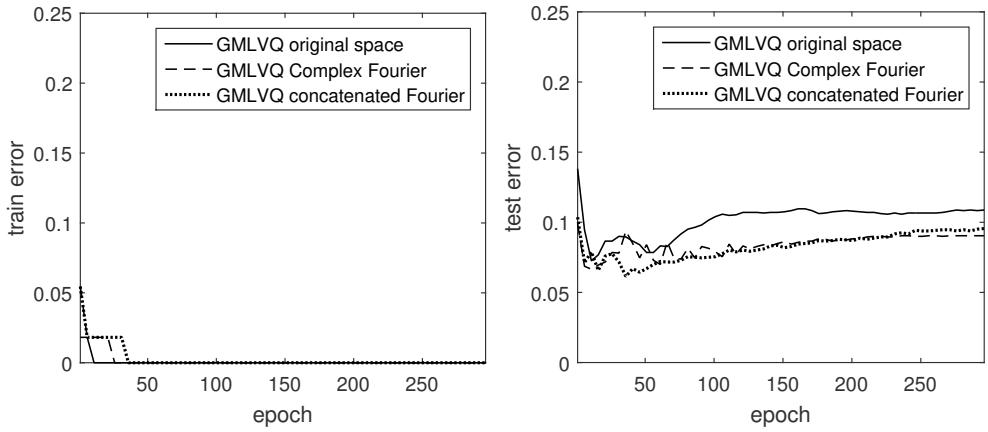
Prior to training, the training data were transformed such that all dimensions have zero mean and unit variance. The test data were transformed correspondingly using the mean and standard deviation of the features in the training data. This normalization is useful for the intuitive interpretation of the relevance matrix, since the relevance matrix does not have to account for the different scales of the features. The relevance values will therefore be directly comparable. All systems used one prototype per class, which was initialized to a small random deviation from the corresponding class conditional mean. The relevance matrix was initialized proportional to the identity matrix. Furthermore, a batch gradient descent along the lines of [16] was applied as the optimization procedure using the default parameters from [17].

C. Example Datasets

The suggested approach was applied to four time series datasets from the UCR repository [11]. The names of the datasets and their properties are in Table I. The four datasets all contain time series with more or less periodic behavior. The repository does not provide further details and annotations about the origin of the datasets. Note that it is required that $\lfloor N/2 \rfloor + 1 \geq n_{max}$, where $n_{max} = 51$, the maximum number of coefficients we consider in the experiments (see scenario 2). As mentioned in Section II-A, all information is contained in $\lfloor N/2 \rfloor + 1$ coefficients which is therefore the upper-bound for the number of approximation coefficients n. As can be seen in Table I all the considered datasets satisfy $\lfloor N/2 \rfloor + 1 \geq 51$.

D. Performance evaluation

The performance for the different scenarios is evaluated by the classification accuracy, i.e. the percentage of correctly classified feature vectors on the validation set as indicated in Table I. For scenario 1 this is one baseline classification accuracy. For the functional approximation scenarios, 2, 3



(a) Train set error (b) Validation set error

Fig. 3. Training and validation error for the MALLAT dataset in the course of training. The *dashed line* is the error development in 20-coefficient complex Fourier space. The *solid line* shows the error development in the original space of the data. The *dotted line* shows the error development in 20-coefficient concatenated Fourier space.

TABLE I. TIME SERIES DATASETS

Dataset name	classes	sampling points	samples	
			training	validation
PLANE	7	144	105	105
MALLAT	8	1024	55	2345
SYMBOLS	6	398	25	995
FACESUCR	14	131	200	2050

and 4, each level of approximation n yields a classification accuracy, which will then be compared and discussed.

IV. RESULTS AND DISCUSSION

The results displayed in Figure 1 suggest that, in general, the classification results of functional data using a Fourier representation are comparable to or better than the baseline performance in the original time domain of the data.

The results on the PLANE dataset in Figure 1a show that for all numbers of complex Fourier coefficients $n > 5$ the classification accuracy is at least as good as the accuracy in the original 144-dimensional feature space. The obtained accuracies are robust with respect to n, as there are no large fluctuations in performance. For this particular dataset, a functional approximation with 15 or 20 complex Fourier coefficients already seems sufficient to accurately distinguish between the classes. The representation with concatenated Fourier coefficients of Scenario 3 achieves a similar accuracy as the complex representation.

In the results of the method on the FACESUCR dataset as shown in Figure 1b, the best performance is achieved for 20 Fourier coefficients. For $n \geq 15$, the performance of the two Fourier representations is similar. The performance in Fourier space is better than the performance in original space in the $n = 20$ region, but the classification becomes less accurate the more higher frequency components are added. This indicates the presence of higher frequency noise in the original signals that negatively affects the classification accuracy.

On the SYMBOLS dataset the functional Fourier representations structurally achieve a better performance than the baseline performance in the original 398-dimensional space, even with a number of coefficients as low as $n = 15$. The accuracies of the complex representation and the concatenated real representation are similar. On the other hand, the accuracies achieved on the smoothed time series of Scenario 4 are systematically lower than the accuracies in Fourier space. Therefore the observed improvement achieved from the transformation of the feature vectors to Fourier space cannot only be explained by the smoothing that the functional approximation brings about.

For the further investigation of the performance of the method for even higher dimensional functional data, the dataset MALLAT is considered consisting of feature vectors with dimensions $N = 1024$. Figure 1d shows that the results in complex and concatenated Fourier space do not deviate significantly from the achieved accuracy in the original space. A functional Fourier approximation with 20 coefficients provides the same classification accuracy as in the original space, i.e., the system was able to achieve a similar accuracy on the 20-coefficient Fourier space representation compared to using all 1024 available original features. Despite the result on this dataset showing no improvement in accuracy, the dimensionality in the classification problem was reduced by 99.6% without loss of classification accuracy, yielding a large computational advantage in the training- and classification stage.

The prototypes that arise in the training process in complex Fourier coefficient space can be interpreted as class-specific contributions of the complex sinusoidal components of different frequencies in the corresponding classes. In Figure 2b, the backtransformation of the prototypes as formulated in section II-C has been applied to the resulting complex prototypes of the PLANE dataset in 21-coefficient Fourier space, $\boldsymbol{w}_f \in \mathbb{C}^{21}$, yielding a representation of the prototypes in the original time domain. A comparison with the prototypes resulting from training in the original time domain (Figure 2a) reveals that the backtransformed prototypes are smoother, but resemble the

prototypes from training in the full original space closely. Correspondingly, Figure 2d shows the backtransformed relevance values. A comparison with the relevance values obtained in the original time domain shown in Figure 2c reveals that the general relevance profiles are similar.

Figure 3 shows the error development on the training- and validation set of the MALLAT dataset. The three methods all achieve zero training error before 50 training epochs. After 50 epochs the increased error in the original space on the validation set indicates an overfitting effect. Both Fourier representations, complex and concatenated real- and imaginary parts, are less affected by overfitting, as the error on the validation set for these representations does not increase significantly. This confirms the conjecture that training in reduced Fourier-coefficient space can help to alleviate overfitting effects that arise in the original space. On this dataset, the complex Fourier representation eventually achieves the lowest error, followed by the concatenated Fourier representation.

Besides the potential to improve performance with a transformation to Fourier space, we must note that the difference in accuracy between the complex Fourier representation of scenario 2 and the concatenated representation of scenario 3 is small. However, training on the complex-valued data directly with GMLVQ using learning rules derived with Wirtinger calculus has the advantage of treating the complex dimensions as such and is therefore mathematically well-formulated.

V. Summary and Outlook

In this contribution we have shown and discussed the benefits of transforming smooth (periodic) time series, which are essentially functional data, to the complex Fourier domain for the classification. In our experiments, the classification accuracy for even a reasonably small number of coefficients ($n = 20$) was similar and usually better than the classification accuracy on the corresponding dataset in the original time domain. Besides the potential of improving the classification accuracy, this suggests that the method can be used to reduce the number of dimensions of the feature vectors to a large extent. As the number of parameters in GMLVQ scales quadratically with the number of dimensions, this reduces the computational effort in the training phase considerably. Moreover, the training curves of the MALLAT dataset have indicated that training in reduced coefficient space has the potential to attenuate over-fitting. The optimal number of Fourier coefficients to use is of course dependent on the properties of the dataset. For future study, an automatic method could be devised that suggests a number of coefficients based on the available training and validation data according to a criterion of optimality, which seeks the best balance between accuracy and the number of coefficients.

References

[1] J. Ramsay and B. Silverman, *Functional Data Analysis*. Springer, 2006.

[2] F. Melchert, U. Seiffert, and M. Biehl, "Functional representation of prototypes in LVQ and Relevance Learning," in *Advances in Self-Organizing Maps and Learning Vector Quantization: Proc. of the 11th Intl. Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016*, E. Merényi, J. M. Mendenhall, and P. O'Driscoll, Eds. Cham: Springer, 2016, pp. 317–327.

[3] ——, "Functional approximation for the classification of smooth time series," in *GCPR Workshop on New Challenges in Neural Computation 2016*, ser. Machine Learning Reports, B. Hammer, T. Martinetz, and T. Villmann, Eds., vol. MLR-2016-04, 2016, pp. 24–31.

[4] M. Kästner, B. Hammer, M. Biehl, and T. Villmann, "Generalized functional relevance learning vector quantization," in *Proc. Europ. Symp. on Artificial Neural Networks (ESANN)*, M. Verleysen, Ed. d-side, 2011, pp. 93–98.

[5] M. Biehl, B. Hammer, and T. Villmann, "Distance measures for prototype based classfication," in *BrainComp 2013, Proc. International Workshop on Brain-Inspired Computing, Cetraro/Italy, 2013*, ser. Lecture Notes in Computer Science, L. Grandinetti, N. Petkov, and T. Lippert, Eds., vol. 8603. Springer, 2014, pp. 100–116.

[6] ——, "Prototype-based models in machine learning," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, pp. 92–111, 2016.

[7] P. Schneider, M. Biehl, F.-M. Schleif, and B. Hammer, "Advanced metric adaptation in Generalized LVQ for classification of mass spectrometry data," in *Proc. 6th Intl. Workshop on Self-Organizing-Maps (WSOM)*. Bielefeld University, 2007, 5 pages.

[8] P. Schneider, M. Biehl, and B. Hammer, "Relevance matrices in LVQ," in *Proc. European Symposium on Artificial Neural Networks*, M. Verleysen, Ed. d-side publishing, 2007, pp. 37–42.

[9] ——, "Adaptive relevance matrices in learning vector quantization," *Neural computation*, vol. 21, no. 12, pp. 3532–3561, 12 2009.

[10] M. Gay, M. Kaden, M. Biehl, A. Lampe, and T. Villmann, "Complex variants of GLVQ based on Wirtinger's calculus," in *Advances in Self-Organizing Maps and Learning Vector Quantization: Proc. of the 11th Intl. Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016*, E. Merényi, J. M. Mendenhall, and P. O'Driscoll, Eds. Cham: Springer, 2016, pp. 293–303.

[11] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR time series classification archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/, accessed: 02-2017.

[12] W. Wirtinger, "Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen," *Mathematische Annalen*, vol. 97, pp. 357–376, 1927.

[13] E. O. Brigham, "The discrete fourier transform," *The Fast Fourier Transform*, pp. 91–109, 1974.

[14] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. MIT Press, 1995, pp. 423–429.

[15] K. Bunte, F.-M. Schleif, and M. Biehl, "Adaptive learning for complex valued data," in *20th European Symposium on Artificial Neural Networks, ESANN 2012*, M. Verleysen, Ed. d-side publishing, 2012, pp. 387–392.

[16] G. Papari, K. Bunte, and M. Biehl, "Waypoint averaging and step size control in learning by gradient descent," *Machine Learning Reports*, vol. MLR-06/2011, p. 16, 2011.

[17] M. Biehl, "A no-nonsense beginner's tool for GMLVQ," accessed: 02-2017. [Online]. Available: http://www.cs.rug.nl/~biehl/gmlvq