Functional approximation for the classification of smooth time series

Melchert, Friedrich; Seiffert, Udo; Biehl, Michael

Published in:
Workshop New Challenges in Neural Computation 2016

Functional approximation for the classification of smooth time series

Friedrich Melchert[1,2], Udo Seiffert[2], and Michael Biehl[1]

[1] University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, P.O. Box 407, 9700 AK Groningen, The Netherlands
[2] Fraunhofer Institute for Factory Operation and Automation IFF, Sandtorstrasse 22, 39106 Magdeburg, Germany

Abstract. Time series data are frequently analysed or classified by considering sequences of observations directly as high-dimensional feature vectors. The presence of several hundreds or thousands of input dimensions can lead to practical problems. Moreover, standard algorithms are not readily applicable when the time series data is non-equidistant or the sampling rate is non-uniform. We present an approach that allows for a massive reduction of input dimensions and explicitly takes advantage of the functional nature of the data. Furthermore, the application of standard classification algorithms becomes possible for inhomogeneously sampled time series. The presented approach is evaluated by applying it to four publicly available time series datasets.

Keywords: Classification; supervised learning; functional data; time series; Learning Vector Quantization; relevance learning; dimensionality reduction; missing values

1 Introduction

The classification of time series data is of interest in various domains including medicine, finance, entertainment and industry [19]. In many applications the time series data is sampled with high temporal resolution, resulting in high-dimensional feature vectors. Traditional classification schemes often display inferior performance when applied to nominally very high-dimensional data. However, due to temporal correlations, the large number of features does not necessarily correspond to high intrinsic dimension in time series data [18]. Although a variety of machine learning techniques are able to handle high-dimensional datasets, most of them were not designed to take advantage of the functional nature and temporal ordering of the features [8].

Here, we consider an explicit functional representation of time series data which exploits the correlation of subsequent measurements and reduces the number of input dimensions drastically. To implement the actual classification task, different machine learning algorithms can be applied, each having characteristic advantages and disadvantages. Here, we resort to prototype and distance based classifiers, such as *Learning Vector Quantization* (LVQ) [10], which are

straightforward to implement and allow for intuitive interpretation [1,3,4]. The prototypes in LVQ represent typical exemplars of their corresponding classes. Together with a suitable distance mesaure, they constitute an efficient classification system [3,4].

The choice of an appropriate distance is a key step in the design of any prototype based classification system. Although it is computational costly, *Dynamic Time Warping* (DTW) [14] is considered a standard choice for comparing time series [13]. Here, we employ a fast and adaptive quadratic distance measure in the framework of *Generalized Matrix Relevance LVQ* (GMLVQ), which is optimized in the training process [15,3]. This is not only more flexible than the use of fixed, predefined measures, it also facilitates the interpretation of the emerging distance measure which provides important insights into the structure of the input data with respect to the classification task [15,16].

Previously, similar variants of relevance LVQ were considered in the context of short term and long term predictions of time series in [17]. The use of a functional representation together with GMLVQ in coefficient space was discussed in [11] for spectral and other functional data. Here, we will transfer and extend this approach to smooth time series and their specific properties. In particular, we will show how the functional nature of the data can be exploited to cope with missing and non-equidistant sampled data.

In the next section we will outline the general framework of time series classification by combining GMLVQ with functional representations. In section 3 the performed experiments are described and their results are shown. We conclude with a discussion of the results and a brief outlook on open research questions.

2 Polynomial approximation of time series

We consider the general classification setup, where a training set of N labeled feature vectors $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{1 \dots A\}, i = 1 \dots N$ is used to train a classifier. Here d denotes the dimension of the data and A the number of different classes in the dataset. The trained classifier assigns a class label $y(\boldsymbol{x}) = 1 \dots A$ to any feature vector $\boldsymbol{x}$.

Furthermore, we assume that the feature vectors $\boldsymbol{x}_i$ represent discrete time series data, which result from sampling an unknown function $f_i(t)$ at some known time points t_j. In the following we will assume the time scale to be the interval $t \in [-1 \dots 1]$ and denote the discretized observations as

$$x_{i,j} = f_i\left(t_j\right). \tag{1}$$

Given a suitable set of basis function $g_k(t)$ it is possible to represent $f_i(t)$ as a weighted sum of the basis functions:

$$f_i\left(t\right) = \sum_{k=0}^{\infty} c_{i,k} g_k(t). \tag{2}$$

Restricting the number of coefficients to a finite number n, Eq. (2) becomes, in general, an approximation $\hat{f}_i(t)$ of the original function $f_i(t)$.

Although using a Fourier basis is first choice in many signal processing applications it is most suitable for periodic functions. Here we use Chebyshev polynomials of the first kind as basis functions. They provide an efficient way to represent non-periodic smooth functions and have favourable properties with respect to numerics [6]. The recursive definition reads

$$T_0(x) = 1; \qquad T_1(x) = x; \qquad T_n(x) = 2xT_{n-1} - T_{n-2}(x). \tag{3}$$

The approximation coefficients $c_{i,k}$ can be determined by minimizing a suitable optimization criterion, e.g. the quadratic error $e = \sum_{j=1}^{d}(f_i(t_j) - \hat{f}_i(t_j))^2$ or the maximum deviation $e = \max_{j=1\ldots d}(f_i(t_j) - \hat{f}_i(t_j))$. Here, we exploit the properties of truncated Chebyshev series to compute the coefficient values in an efficient way [9]:

$$c_{i,k} = \frac{2}{n+1}\sum_{l=0}^{n} f_i(t_l)T_k(t_l) \text{ , with } t_l = \cos\left(\left(l+\frac{1}{2}\right)\frac{\pi}{n+1}\right). \tag{4}$$

Given the maximum degree n, the sampling points t_l represent the roots of the Chebyshev polynomial of degree $(n+1)$. Since, in general, the original sampling points will not match these roots, we perform a simple, linear interpolation of the original data in order to obtain the values of $f_i(t_l)$. The linear interpolation is justified under the assumption that the distance of the t_l from the known sampling points is small compared to the overall length of the time series. It is, of course, possible to use more powerful interpolation schemes, e.g. Floater Hormann interpolants [7]. However, using a linear scheme has advantages in terms of computational effort and, moreover, its invertibility facilitates a suitable interpretation of the results as demonstrated and discussed below. Note that approximation quality is not the main goal in the following. The polynomial representation serves as a method fpr feature extraction in terms of the resulting coefficients.

We can summarize the transformation from the original data to the space of approximation coefficients by the equation

$$\boldsymbol{c}_i = \boldsymbol{S}\boldsymbol{P}\boldsymbol{x}_i = \boldsymbol{\Psi}\boldsymbol{x}_i, \tag{5}$$

where the matrix $\boldsymbol{S} \in \mathbb{R}^{n\times d}$ represents the linear interpolation of the original data at the sampling points t_l and the matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ represents the first n Chebyshev polynomials evaluated at the sampling points t_l.

The setup can be easily extended to non-equidistant and non-uniform sampled time series, since no assumption on the number and distribution of the original sampling points t_j is made. An extension to a particular sampling $t_{j,i}$, which could be even data point specific, is straightforward according to Eqs. (1-5) and only affects the interpolation matrix, introducing individual $\boldsymbol{S}_i$.

Under the assumption that the available data results from sampling a smooth time-dependent function, the presented approach allows for a transformation to the more abstract space of coefficients. This transformation is also feasible if the input data is not equidistant (different time intervals between sampled points) or not uniform (different number of time-points sampled).

Table 1: Selected datasets from the UCR Time Series Repository [5], together with the number of samples, sampling points and classes.

Dataset name	classes	sampling points	samples (training)	samples (validation)
ItalyPowerDemand	2	24	67	1029
Plane	7	144	105	105
StarLightCurves	3	1024	1000	8236
Strawberry	2	256	370	613

3 Application to example datasets

In order to evaluate the suggested approach, it is applied to four publicly available, relatively smooth time series datasets taken from the UCR repository [5]. The selected datasets and their key properties are listed in Table 1. Note that the repository does not provide detailed information with respect to, e.g., the interpretation of the values, the meaning of classes or the real world time scales.

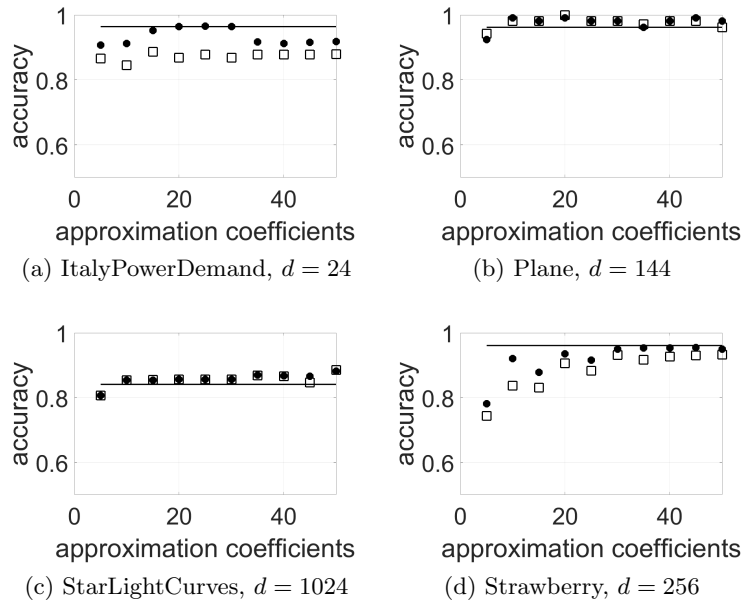
For each of the datasets three setups were considered for computer experiments. To obtain a natural baseline for the achievable classification performance in a first setup (A) the classifiers were trained from the original time series data.

For a second set of experiments (B) the data were transformed to vectors of approximation coefficients and GMLVQ training was performed in this space. The experiments were repeated for different numbers of coefficients: $n = 5, 10, \ldots 50$.

In the third expermimental setup (C) the original data was manipulated in order to simulate non-equidistant, non-uniform sampled data. To this end, a random number (between 20% and 60%) of values was discarded from each available feature vector. Which values were actually deleted was also chosen randomly and independently for each data point. This resulted in modified feature vectors with varying number of sampling points and randomized positions of the available points. The modified dataset $\{\tilde{\boldsymbol{x}}_i, \boldsymbol{t}_i\}$ was then used to transform the data to the space of approximation coefficients according to Eqs.(4,5). As in setup (B), the number of coefficients was varied as $n = 5, 10, \ldots 50$.

In all experiments a corresponding GMLVQ system was trained from the respective set of labeled feature vectors using the same set of parameters. All systems comprised one prototype per class. Before each training process the data was preprocessed in terms of a z-score transformation, yielding zero mean and unit variance in all dimensions, and therefore equalizing the magnitudes of the different features. The z-score transformation facilitates the intuitive interpretation of the emerging relevance matrices [15]. The relevance matrix was initialized as proportional to the identity, while the prototypes were initialized in the corresponding class-conditional means. As optimization scheme a batch gradient descent with adaptive step sizes along the lines of [12] was performed with default parameters as suggested in [2].

The performance of the emerging GMLVQ systems was evaluated as the overall classification accuracy with respect to the corresponding validation dataset



(a) ItalyPowerDemand, $d = 24$ (b) Plane, $d = 144$

(c) StarLightCurves, $d = 1024$ (d) Strawberry, $d = 256$

Fig. 1: Classification accuracies achieved in the respective validation sets as a function of the number of approximation coefficients. The solid lines represent the accuracy achieved in the full set of all available input features (experimental setup A). Filled circles correspond to accuracies resulting from the classification in the space of approximation coefficients (B). Empty squares mark the results achieved after the randomized deletion of time-points in setting (C). For comparison the original number of sampling points for each dataset is denoted.

in the UCR archive [5] (cf. Table 1). Validation data underwent the same preprocessing as the training set in each individual experiment. This includes the transformation to the space of approximation coefficients and the randomized deletion of time-points in setting (C). The z-score transformation of the data was performed with respect to the mean and variance determined from the training dataset. The results of the experiments are depicted in Figure 1.

4 Results and Discussion

In the example datasets considered here, we observe only insignificant or no increase of the classification accuracy. However, the transformation of the data to the space of approximation coefficients yields a massive reduction of input dimension. The largest reduction (99%) was achieved in the *StarLightCurves* dataset when using $n = 10$ coefficients.

The evaluation of results from setup (C), where up to 60% of the data points were disregarded, shows that the approach can compensate for missing data and irregular sampling to a very large extent. In fact, the results show that

the random removal of time-points had no impact on the overall classification performance achieved in the considered example problems.

One of the main advantages of prototype based classification is that the prototypes are determined in the domain of the original data. A GMLVQ system directly trained from time series data, yields interpretable prototypes and relevances with respect to the sampling points of the time series. In the setups (B) and (C), however, the GMLVQ system is adapted in the more abstract space of approximation coefficients. Hence, it is not obvious how to interpret prototypes and relevance matrices adequately. In previous work [11], the interpretation of prototypes and relevances in the space of coefficients was provided with respect to the characteristics of the basis functions. Since this is less intuitive than an interpretation in the original feature space, it is desirable to back-transform prototypes as well as relevance matrices to the original time series representation. In order to obtain such a transformation we can use the matrix $\boldsymbol{\Psi}$ introduced in Eq. (5): Including the transformation into the distance measure applied in GMLVQ [15] we obtain

$$d(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x} - \boldsymbol{z})^\top \boldsymbol{\Psi}^\top \boldsymbol{\Lambda}_c \boldsymbol{\Psi} (\boldsymbol{x} - \boldsymbol{z}) \tag{6}$$

where $\boldsymbol{\Lambda}_c$ denotes the relevance matrix obtained in coefficient space. This yields the relation

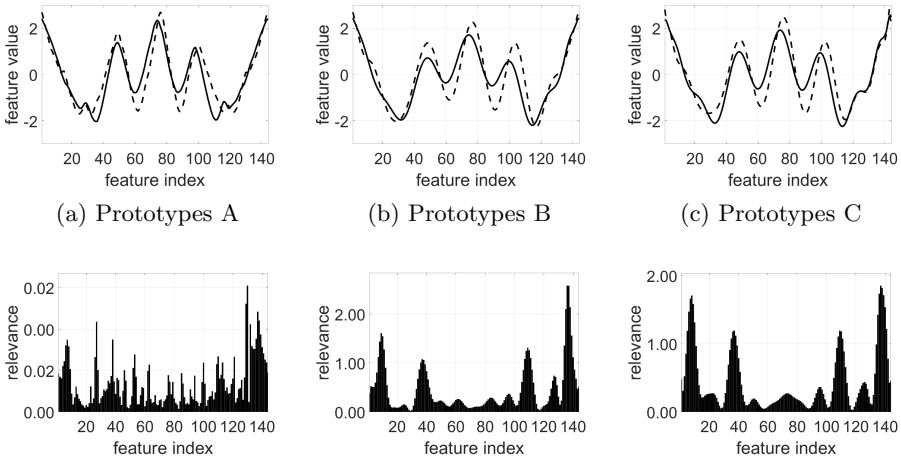
$$\boldsymbol{\Lambda} = \boldsymbol{\Psi}^\top \boldsymbol{\Lambda}_c \boldsymbol{\Psi} \tag{7}$$

which translates the obtained relevance matrix back to original feature space.

An illustrative example for the prototypes and relevance matrices obtained in settings (A), (B) and (C) for the *Plane* dataset is depicted in Fig. 2. Apart from the implicit smoothening it is evident that, both, prototypes and relevance profiles are very similar to those obtained in the original feature space. As a result of the applied normalization steps, the absolute values can be different, but the general shapes of the relevance profiles are essentially identical. The comparison of Figs. 2d, 2e, and 2f, does not reveal major differences. Note, in particular, that although there is a loss of information in experiments (C) due to the random dilution of time-points, prototypes as well as relevances can be transformed to a uniformly sampled input space. Therefore we maintain their interpretability over the complete input space.

5 Summary and Outlook

We have presented an approach for time series classification using a representation that takes the functional nature of smooth time series into account. Our computer experiments show that the approximation of the time series with a suitable set of basis functions yields a massive reduction of input dimensionality without significant loss of classification accuracy. Furthermore we studied the influence of irregular, missing data by randomly deleting up to 60% of the values in each sample. The achieved results show that the functional approximation of the data can compensate for the missing information to a very large extent. No significant decrease in classification accuracy was observed.



(a) Prototypes A (b) Prototypes B (c) Prototypes C

(d) Relevances A (e) Relevances B (f) Relevances C

Fig. 2: Prototypes and relevance profiles emerging from the different setups (A, B, C). For setting (A) prototypes and relevance profiles directly emerge from the model, in (B) and (C) they are shown after back-transformation to the original feature space. The shown results were achieved using $n = 20$ approximation coefficients. For the sake of clarity, only the prototypes for the first (solid line) and second (dashed line) class are shown.

The use of Chebyshev polynomials as basis functions in combination with a linear resampling of the data constitutes a suitable representation of time series. Furthermore the transformation of the data can be done in a single matrix multiplication and therefore has clear advantages over DTW in terms of computational effort. Finally, the linearity and invertibility of the transformation makes it possible to interpret the GMLVQ system also in the original input space. The interpretation of prototypes and relevances is maintained over the full time domain, even for time series with non-equidistant and non-uniform sampling.

Future work will concern the selection of alternative basis functions for the analysis of time series and other functional data. An interesting question concerns the choice of an optimal number of approximation coefficients corresponding to a minimum number of adaptive parameters while maintaining close to optimal accuracy. The presented approach allows for a compact representation of smooth time series, which should be very useful for the analysis of heterogeneous datasets comprising several data modalities.

Acknowledgments. F. Melchert thanks for support through an Ubbo-Emmius Sandwich Scholarship from the Faculty of Mathematics and Natural Sciences, University of Groningen.

References

1. Backhaus, A., Seiffert, U.: Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size. Neurocomputing 131, 15–22 (2014)
2. Biehl, M.: A no-nonsense beginner's tool for GMLVQ. Available online, University of Groningen, `http://www.cs.rug.nl/~biehl/gmlvq`, Ver. 2.2
3. Biehl, M., Hammer, B., Villmann, T.: Distance measures for prototype based classification. In: Grandinetti, L., Petkov, N., Lippert, T. (eds.) BrainComp 2013, Proc. International Workshop on Brain-Inspired Computing, Cetraro/Italy, 2013. Lecture Notes in Computer Science, vol. 8603, pp. 100–116. Springer (2014)
4. Biehl, M., Hammer, B., Villmann, T.: Prototype-based models in machine learning. Wiley Interdisciplinary Reviews: Cognitive Science 7, 92–111 (2016), `http://dx.doi.org/10.1002/wcs.1378`
5. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive (July 2015), `www.cs.ucr.edu/~eamonn/time_series_data/`
6. Driscoll, T.A., Hale, N., Trefethen, L.N.: Chebfun guide. Pafnuty Publ. (2014)
7. Floater, M.S., Hormann, K.: Barycentric rational interpolation with no poles and high rates of approximation. Numerische Mathematik 107(2), 315–331 (2007)
8. Geurts, P.: Pattern extraction for time series classification. In: European Conf. on Principles of Data Mining and Knowledge Discovery. pp. 115–127. Springer (2001)
9. Gil, A., Segura, J., Temme, N.M.: Numerical methods for special functions. Siam (2007)
10. Kohonen, T.: Self-organizing maps. Springer, Berlin (1995)
11. Melchert, F., Seiffert, U., Biehl, M.: Functional representation of prototypes in LVQ and relevance learning. In: Advances in Self-Organizing Maps and Learning Vector Quantization, pp. 317–327. Springer (2016)
12. Papari, G., Bunte, K., Biehl, M.: Waypoint averaging and step size control in learning by gradient descent. Machine Learning Reports MLR-06/2011, 16 (2011)
13. Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y., Keogh, E.: Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. Knowledge and Information Systems 47(1), 1–26 (2016)
14. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing 26(1), 43–49 (1978)
15. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in Learning Vector Quantization. Neural Computation 21, 3532–3561 (2009)
16. Strickert, M., Hammer, B., Villmann, T., Biehl, M.: Regularization and improved interpretation of linear data mappings and adaptive distance measures. In: Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on. pp. 10–17 (April 2013)
17. Strickert, M., Bojer, T., Hammer, B.: Generalized relevance LVQ for time series. In: International Conf. on Artificial Neural Networks. pp. 677–683. Springer (2001)
18. Tomašev, N., Radovanović, M.: Clustering evaluation in high-dimensional data. In: Unsupervised Learning Algorithms, pp. 71–107. Springer (2016)
19. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on Machine learning. pp. 1033–1040. ACM (2006)