Expert Systems with Applications 42 (2015) 2305–2312

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Multivariate time series classification with parametric derivative dynamic time warping

Tomasz Górecki [a],*, Maciej Łuczak [b]

[a] *Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Umultowska 87, 61-614 Poznań, Poland*
[b] *Department of Civil Engineering, Environmental and Geodetic Sciences, Koszalin University of Technology, Śniadeckich 2, 75-453 Koszalin, Poland*

ARTICLE INFO

Article history:
Available online 13 November 2014

Keywords:
Dynamic time warping
Derivative dynamic time warping
Multivariate time series

ABSTRACT

Multivariate time series (MTS) data are widely used in a very broad range of fields, including medicine, finance, multimedia and engineering. In this paper a new approach for MTS classification, using a parametric derivative dynamic time warping distance, is proposed. Our approach combines two distances: the DTW distance between MTS and the DTW distance between derivatives of MTS. The new distance is used in classification with the nearest neighbor rule. Experimental results performed on 18 data sets demonstrate the effectiveness of the proposed approach for MTS classification.

1. Introduction

In recent decades, time series analysis has become one of the most popular branches of statistics. Time series are currently ubiquitous, and have come to be used in many fields of science. Data sets in the form of time series occur in many areas of human life. Recent developments in computing have provided the basic infrastructure for fast access to vast amounts of online data. This is especially true for the recording of time series data, for example in the medical and financial sectors. One of the major applications is time series classification. Multivariate time series (MTS) classification is an important problem in time series data mining. MTS classification is difficult for traditional machine learning algorithms mainly because of the dozens of variables (if an MTS sample is broken into univariate time series and each processed separately, the correlations among the variables could be lost) and different lengths of MTS samples.

Several approaches have been proposed for MTS classification. Maharaj (1999) used *p*-values and hierarchical clustering to classify stationary MTS. Geurts and Wehenkel (2005) classified subsequences instead of the whole MTS sample. Hayashi, Mizuhara, and Suematsu (2005) proposed an approach involving embedding MTS samples in a vector space and classifying them in the embedded space. Kadous and Sammut (2005) proposed an approach to MTS classification using metafeatures. Rodriguez, Alonso, and Maestro (2005) proposed to select literals from MTS samples with boosting and to use these literals with SVM. Yang et al. (2005) proposed a new feature subset selection method for MTS classification, based on common principal component analysis. Li, Khan, and Prabhakaran (2006) and Li, Khan, and Prabhakaran (2007) proposed two feature vector selection approaches for MTS classification by using singular value decomposition. The first approach considers only the first singular vector and the normalized singular values, while the second takes into account the first two dominating singular vectors weighted by associated singular values. Spiegel, Gaebler, Lommatzsch, De Luca, and Albayrak (2011) separated a time series into segments using SVD and then clustered the recognized segments into groups of similar context. Ghalwash, Ramljak, and Obradović (2012) integrated the Hidden Markov model and SVM classifier to make an early classification of MTS. Ghalwash and Obradović (2012) proposed using time series segments called shapelets in classification of MTS. Finally, Prieto, Alonso-González, and Rodríguez (2014) proposed stacking for multivariate time series classification.

However, these approaches do not consider explicitly the two-dimensional nature of MTS samples (an MTS sample is in fact one kind of two-dimensional matrix data). Weng and Shen (2008a) proposed a new approach for MTS classification using two-dimensional singular value decomposition, which is an extension of standard SVD. This method captures explicitly the two-dimensional nature of objects. Weng and Shen (2008b) tried also to use locality-preserving projections in the classification process of MTS. Weng (2013) presented an extension of the previous method which preserves the within-class local structure of the MTS.

* Corresponding author.
E-mail addresses: tomasz.gorecki@amu.edu.pl (T. Górecki), mluczak@wilsig.tu.koszalin.pl (M. Łuczak).

http://dx.doi.org/10.1016/j.eswa.2014.11.007
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

This paper introduces a new shape-based similarity measure, called parametric derivative dynamic time warping DD_{DTW}, for multivariate time series data. There have been many measures proposed for univariate time series data, the most widely known being the Euclidean distance. The main problem with this measure is that the compared time series need to have the same length. A newer measure, DISSIM (Frentzos, Gratsias, & Theodoridis, 2007), provides a solution to this problem, but it is computationally costly and in general does not compare favorably with elastic measures. The family of elastic measures uses dynamic programming to align sequences with different lengths, and includes DTW (Berndt & Clifford, 1994), LCSS (Das, Gunopulos, & Mannila, 1997), edit distance with real penalty (ERP) (Chen & Ng, 2004), edit distance on real sequence (EDR) (Chen, Øzsu, & Oria, 2005), derivative dynamic time warping (DDTW) (Keogh & Pazzani, 2001), and angular metric for shape similarity (AMSS) (Nakamura, Taki, Nomiya, Seki, & Uehara, 2013). The sequence weighted alignment model (Swale) (Morse & Patel, 2007) can be regarded as another elastic measure, but without employing dynamic programming. A major difference between DTW, LCSS, ERP, EDR on the one hand, and AMSS, DTW, DD_{DTW} on the other, is that those in the first group look only at individual data points, without considering the shapes of trajectories. AMSS and LCSS are less affected by outliers, but AMSS is more sensitive to short-term oscillations, which require preprocessing. There have also been other measures proposed, such as TQuEST (Aßfalg, Kriegel, Kunath, Pryakhin, & Renz, 2006) and SpADe (Chen, Nascimento, Ooi, & Tung, 2007). SpADe is similar to AMSS, DTW and our DD_{DTW} in the sense that it looks at the shapes of data. A critical difference, however, is that SpADe requires many parameters, which must be tuned for each data set, whereas AMSS and DDTW have no parameter to tune, and our DD_{DTW} has only one parameter.

The simple method combining the nearest neighbor (1NN) classifier and some form of dynamic time warping (DTW) distance has been shown to be one of the best-performing univariate time series classification techniques (Ding, Trajcevski, Scheuermann, Wang, & Keogh, 2008). The expansion of DTW to multiple dimensions is only rarely found in the literature. There exist a few works which describe extensions of the DTW algorithm to include multiple dimensions. Gavrila and Davis (1995) described a type of multivariate DTW, but used it only for the recognition of human movement. An extension of DTW into two dimensions was proposed by Vlachos, Hadjieleftheriou, Gunopulos, and Keogh (2003) and Vlachos, Hadjieleftheriou, Gunopulos, and Keogh (2006), but not systematically tested. An extension of the method of Vlachos et al. (2003) was proposed by ten Holt, Reinders, and Hendriks (2007). They also used derivatives, but calculated DTW separately on feature derivatives and on feature values, and finally added these values. In these works the term multidimensional refers to the size of the feature vectors coming from the same modality. Consequently, these approaches use the conventional two-dimensional distance matrix, whose entries are calculated from multidimensional feature vectors. Also Mello and Gondra (2008) measured the similarity between two multidimensional (but not multimodal) series. Wöllmer, Al-Hames, Eyben, Schuller, and Rigoll (2009) introduced multidimensional dynamic time warping for multimodal data streams (they assumed bimodal data streams). Finally, Banko and Abonyi (2012) proposed algorithm called correlation based dynamic time warping (CBDTW) which combines DTW and PCA for highly correlated multivariate time series.

Our previous work (Górecki & Łuczak, 2013) contains the results of research on DTW for univariate time series where the derivative is added, and where parameterization involves both function and derivative. As was shown, our method outperforms classical DTW and DDTW. Because the addition of the first derivative gave such good results in the classification of univariate time series, we decided to research further and to use our technique to classify MTS. Our approach is therefore similar to the method of Vlachos et al. (2003). In contrast to that algorithm, however, we used the parametric approach, which allows us to choose the impact of each distance on the final distance measure between the MTS, and consequently on the quality of the classification. The new distance functions so constructed are used in the nearest neighbor classification method.

The main difference between the method proposed here and other methods which use DTW and other distance measures is the use of a combined approach. We use information from regular DTW and from its derivative version DDTW in one parametric distance measure DD_{DTW}. The parametric approach means that we can choose the size of the contributions from component distance measures for different data sets. Another advantage of our algorithm is that the parameter is not located within the distance DTW (as in some works by other authors), but outside that distance. This significantly reduces the computation time. An appropriate choice of the parameter in the new method means that the error from both components of the distance on the test data sets can be made even smaller. In spite of the need to tune a parameter in the training phase, the computational complexity does not depend on the number of parameters to search. However, in the testing phase of classification the method preserves the computational complexity of the component methods (DTW or DDTW). For all these reasons in combination, our method appears to be a universal method for the classification of MTS, able to identify for which data sets the impact of the derivative is helpful, and to what extent. At the same time, the parametric approach in the new method is a disadvantage as regards computation time. An algorithm (cross-validation in this paper) is required to seek the best value of the parameter on the training data, which unfortunately increases the computation time in the learning phase. We can, however, use the standard lower bound technique to reduce the computation time for the nearest neighbor method. For the DD_{DTW} distance measure the lower bound is a combination of lower bounds for the component distances (DTW and DDTW). It is also possible to construct a special algorithm (described in this paper) which accelerates the calculations on training data in the learning phase.

The remainder of the paper is organized as follows. We first (Section 2) review the concept of MTS and the dynamic time warping algorithm for MTS data. In the same section we introduce our parametric distance based on derivatives, and explain the optimization process and properties of the new distance measure. In Section 3 the MTS data sets used in the empirical comparison of methods are described, and we explain the experimental setup. Later in that section we present the results of our experiments on the described MTS, as well as statistical analysis of the examined methods. We conclude in Section 4 with discussion of possible future extensions of the work.

2. Methods

A (one-dimensional, univariate) time series is a sequence of observations ordered in time (or space) (Box, Jenkins, & Reinsel, 2008), where time is the independent variable. For simplicity and without any loss of generality, we assume that time is discrete. Formally, a time series x is defined as a sequence of real numbers in the form:

$$x = \{x(i) \in \mathbb{R} : \ i = 1, 2, \ldots, n\}.$$

The number n of data points in a given time series is called its length.

We define a multivariate (multi-dimensional) time series X as a finite sequence of univariate time series:

$$X = (x_1, x_2, \dots, x_m),$$

where each x_j is a univariate time series of length n:

$$x_j = \{x_j(i) \in \mathbb{R} : \ i = 1, 2, \dots, n\} \qquad (j = 1, 2, \dots, m).$$

The number of measurements (variables) m is the dimensionality of the series X, and the number of time instances n is its length.

2.1. Dynamic time warping

Let us first introduce standard one-dimensional dynamic time warping (DTW) for two time series with the same length $n \in \mathbb{N}$:

$$x = \{x(i) \in \mathbb{R} : \ i = 1, 2, \dots, n\} \quad \text{and} \quad y = \{y(i) \in \mathbb{R} : \ i = 1, 2, \dots, n\}.$$

We construct an $n \times n$ matrix D with distances $d(x(i), y(j))$ between two points $x(i)$ and $y(j)$ of the time series x and y. We will call the function d the local cost function. In fact d can be any function of two variables. In standard DTW d is usually a distance between two real values, e.g. $d(a, b) = |a - b|$ or $d(a, b) = (a - b)^2$. The matrix element $D(i,j)$ corresponds to the alignment between values $x(j)$ and $y(i)$ of the time series. Then we construct a warping path $W = \{w_1, w_2, \dots, w_K\}$ of matrix elements $D(i,j)$. The warping path W must satisfy three conditions:

(1) boundary conditions: $w_1 = D(1,1), w_K = D(n,n)$;
(2) continuity: for $w_k = D(i_k, j_k)$ and $w_{k+1} = D(i_{k+1}, j_{k+1})$, $i_{k+1} - i_k \leqslant 1$ and $j_{k+1} - j_k \leqslant 1$;
(3) monotonicity: $i_{k+1} - i_k \geqslant 0$ and $j_{k+1} - j_k \geqslant 0$.

Hence to form a warping path we start at element $D(1,1)$ and then move at most one index right or up until ending at $D(n,n)$ – Fig. 1.

The path which minimizes the warping cost gives us the value of the DTW distance:

$$\mathrm{DTW}(x, y) = \min_{W} \left\{ \sum_{k=1}^{K} w_k \right\}.$$

A practical way to compute the DTW distance between two time series is to build so-called cumulative distance matrix Γ. To compute the matrix we use dynamic programming with the following recurrence

$$\Gamma(i,j) = d(x(i), y(j)) + \min\{\Gamma(i-1, j-1), \Gamma(i-1, j), \Gamma(i, j-1)\}$$

and start conditions

$$\Gamma(0,0) = 0, \quad \Gamma(0,i) = \infty, \quad \Gamma(i,0) = \infty \quad (i = 1, 2, \dots, n).$$

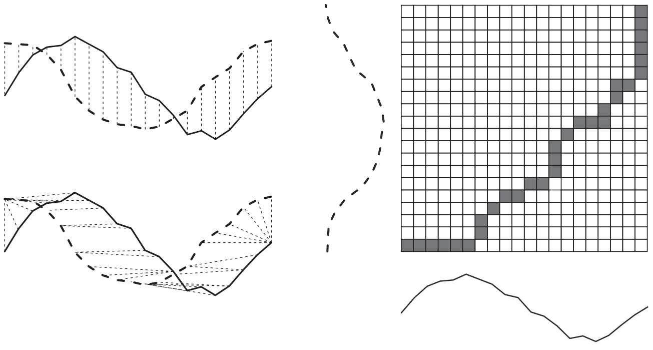

Fig. 1. Top left: two time series which are similar but out of phase produce a large Euclidean distance. Bottom left: this can be corrected by DTW's nonlinear alignment. Right: to align the signals we construct a warping matrix, and search for the optimal warping path.

We fill in the matrix column by column (or row by row); then in the cell at (n, n) we obtain the value of the DTW distance:

$$\mathrm{DTW}(x, y) = \Gamma(n, n).$$

The DTW distance is not a metric (the triangular inequality does not hold), but it is the case that $\mathrm{DTW}(x, x) = 0$ and $\mathrm{DTW}(x, y) = DTW(y, x)$ (if the local cost function d is the same).

If we define any form of derivative x' of a time series x, for example

$$x'(i) = x(i+1) - x(i) \quad (i = 1, 2, \dots, n-1), \tag{1}$$

we can introduce the derivative dynamic time warping (DDTW) distance of two time series x, y as the DTW between their derivatives:

$$\mathrm{DDTW}(x, y) = \mathrm{DTW}(x', y').$$

In this paper we will use the DDTW distance with the derivative defined in (1).

2.2. Multivariate dynamic time warping

We can assume that a MTS is a one-dimensional trajectory in an m-dimensional Euclidean space:

$$X = \{X(i) = (x_1(i), x_2(i), \dots, x_m(i)) \in \mathbb{R}^m : \ i = 1, 2, \dots, n\}. \tag{2}$$

With this notation we can define the DTW distance between MTS X and Y in the same way as in the univariate case, one introducing a local cost function d by

$$d(X(i), Y(j)) = \sum_{k=1}^{m} (x_k(i) - y_k(j))^2,$$

i.e. the square of the Euclidean distance between points $X(i)$ and $Y(j)$.

Defining a the derivative of an MTS X as

$$X' = (x'_1, x'_2, \dots, x'_m)$$

we can introduce the derivative dynamic time series distance between two MTS X and Y in the same way as for univariate time series, i.e.

$$\mathrm{DDTW}(X, Y) = \mathrm{DTW}(X', Y').$$

2.3. Parametric derivative distance

For two distance measures $dist_1$ and $dist_2$ of MTS X and Y, we define a new distance measure $dist_{ab}$ as a linear combination:

$$\mathrm{dist}_{ab}(X, Y) := a\,\mathrm{dist}_1(X, Y) + b\,\mathrm{dist}_2(X, Y), \tag{3}$$

depending on two real parameters $a, b \in [0, 1]$.

The distance function dist_{ab} can be used with the nearest neighbor method in the classification process. Note that then we do not have to pass through all values of $a, b \in [0, 1]$. If $a_1 = ca_2$ and $b_1 = cb_2$, where $c > 0$ is a constant (i.e. the points (a_1, b_1), (a_2, b_2) are linearly dependent), we have

$$\mathrm{dist}_{a_1 b_1}(X_1, Y_1) \overset{=}{\underset{>}{<}} \mathrm{dist}_{a_1 b_1}(X_2, Y_2) \Longleftrightarrow \mathrm{dist}_{a_2 b_2}(X_1, Y_1) \overset{=}{\underset{>}{<}} \mathrm{dist}_{a_2 b_2}(X_2, Y_2)$$

so we can choose points (a, b) on any continuous line between the points $(0, 1)$ and $(1, 0)$. We take the simplest case – the straight line (Fig. 2) with equations:

$$a = (1 - \alpha), \quad b = \alpha, \quad \alpha \in [0, 1].$$

If the subset of the parameters α is dense enough, the choice of parameterization should not be critical. In the next part of the paper we will use one parameter α instead of a, b.

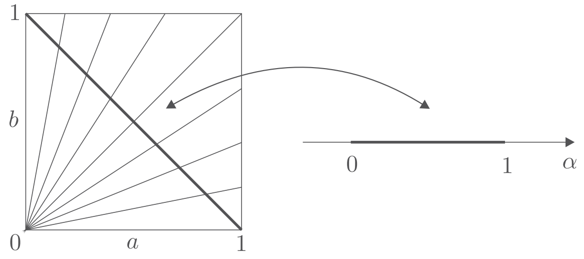

Fig. 2. Dependence of parameters a, b and α.

In this paper we take the distances as $\mathrm{dist}_1 = \mathrm{DTW}$, $\mathrm{dist}_2 = \mathrm{DDTW}$ and define a new distance $\mathrm{DD}_{\mathrm{DTW}}$ between MTS X and Y. Here $\mathrm{DD}_{\mathrm{DTW}}$ is a convex combination of the distances DTW and DDTW:

$$\mathrm{DD}_{\mathrm{DTW}}(X,Y) = (1-\alpha)\mathrm{DTW}(X,Y) + \alpha\mathrm{DDTW}(X,Y) \tag{4}$$

for $\alpha \in [0,1]$. The parameter α is chosen in the learning phase. In this paper we use the cross-validation (leave-one-out) method on the learning data set in the process of parameter tuning.

2.4. Lower bound and triangular inequality

To speed up the calculation we can use a lower bound function. It is easy to find a lower bound for our new distance measure dist_{ab}. If LB_1 is a lower bound for distance dist_1 and LB_2 is a lower bound for distance dist_2, then

$$\mathrm{LB}_{ab}(X,Y) := a\,\mathrm{LB}_1(X,Y) + b\,\mathrm{LB}_2(X,Y)$$

is a lower bound for the distance dist_{ab}.

There are many good lower bounds for the univariate DTW distance, for example LB_Keogh (Keogh, 2002) or LB_Improved (Lemire, 2009), which by the notation (2) can be transformed to multivariate DTW lower bounds. Therefore we can find a good bound for $\mathrm{DD}_{\mathrm{DTW}}$ as well.

If the distances dist_1 and dist_2 are metrics, then the new distance dist_{ab} is also a metric. If the distances obey the triangle inequality, then the distance dist_{ab} also obeys the triangle inequality:

$$\mathrm{dist}_{ab}(X,Y) \leqslant \mathrm{dist}_{ab}(X,Y) + \mathrm{dist}_{ab}(X,Y).$$

Notice that DTW (both univariate and multivariate) is not a metric and does not satisfy the triangle inequality, so the distance $\mathrm{DD}_{\mathrm{DTW}}$ does not have these properties either.

2.5. Optimization

In the training phase we have to tune the parameter α. Let $A \subset [0,1]$ be a finite subset of k parameters. We have to compute the cross-validation (leave-one-out) error rate on the learning data set for every $\alpha \in A$ and choose the parameter for the smallest value of the error (1NN method). However, we can do this in several ways, differing in computational and memory complexity.

In standard procedures, the parameter is tuned as follows. We fix a parameter α and then (by cross-validation) for every element of a training data set we calculate an appropriate number $(n-1)$ of distance functions. We repeat this for every parameter α. However, for our distance function $\mathrm{DD}_{\mathrm{DTW}}$, we can proceed in the opposite direction.

First, we fix one element E of the training data set. Then we take an element $E_1 \neq E$, and for every parameter α we compute the distance function $d_1 = \mathrm{DD}_{\mathrm{DTW}}(E_1, E)$ and put its value into a vector D (with k elements). Note that we need to compute the two distance measures $\mathrm{DTW}(E_1,E)$ and $\mathrm{DDTW}(E_1,E)$ only once. Then for the next element $E_2 \neq E$ from the training set, we repeat the procedure and obtain a new distance vector d_2. Now we compare the elements in corresponding positions of the vectors D and d_2. In every position of vector D we insert the smaller one. We create a vector L (with k elements) which we fill with the labels of elements E_1, E_2 corresponding to values of the vector D. We repeat the procedure for the next elements of the training set ($E_i \neq E$, $i = 3, 4, \ldots, n$). As a result we obtain a vector L with the labels of the nearest neighbors of the element E for every parameter α. This is one step of the cross-validation process. We performed the classification of element E for all parameters α. Note that we computed the distance functions DTW and DDTW only $n-1$ times each. Repeating the procedure for all elements from the training data set, we obtain the cross-validation error rate for all parameters. The code of the algorithm (Matlab code) is given in Fig. 3.

For a fixed parameter α, the computation of the distance $\mathrm{DD}_{\mathrm{DTW}}$ has a complexity of $O(n^2)$. Because calculations of the distance functions DTW and DDTW are the most time-consuming part of the leave-one-out algorithm, the computation time depends only

```
% E - list of MTS in the learning data set (cell vector of vectors)
% labels - vector of labels of elements of list E

alpha = 0 : 0.01 : 1;
n = length(E);
k = length(alpha);
mistakes(1 : k) = 0; % vector of numbers of misclassified elements

for i = 1 : n
    D(1 : k) = inf;  % vector of minimal distances
    L(1 : k) = 0;    % vector of 'minimal' labels
    for j = [1 : i-1, i+1 : n] % leave-one-out
        d = (1 - alpha) * DTW(E{j}, E{i}) + alpha * DDTW(E{j},E{i});
        D(d < D) = d(d < D);
        L(d < D) = labels(j);
    end
    mistakes = mistakes + (L ~= labels(i));
end
errors = mistakes / n; % error rates for every parameter alpha
```

Fig. 3. Optimized leave-one-out cross-validation algorithm for $\mathrm{DD}_{\mathrm{DTW}}$ distance.

Table 1
Summary of data sets.

Data sets	Variables	Max length	Min length	Classes	Size	Source
Arabic digits	13	93	4	10	8800	UCI
Australian language	22	136	45	95	2565	UCI
BCI	28	500	500	2	416	Blankertz
Character trajectories	3	205	109	20	2858	UCI
CMU subject 16	62	580	127	2	58	CMUMC
ECG	2	152	39	2	200	Olszewski
Graz	3	1152	1152	3	140	Leeb
Japanese vowels	12	29	7	9	640	UCI
Libras	2	45	45	15	360	UCI
Non-invasive fetal ECG	2	750	750	42	3765	UCR
Pen digits	2	8	8	10	10992	UCI
Robot failure LP1	6	15	15	4	88	UCI
Robot failure LP2	6	15	15	5	47	UCI
Robot failure LP3	6	15	15	4	47	UCI
Robot failure LP4	6	15	15	3	117	UCI
Robot failure LP5	6	15	15	5	164	UCI
uWaveGestureLibrary	3	315	315	8	4478	UCR
Wafer	6	198	104	2	1194	Olszewski

Table 2
10CV error rates (in %).

Data set	DTW	DDTW	DD_{DTW}	$\frac{DD_{DTW}-DTW}{DTW}$
Arabic digits	0.19	10.50	0.19	0.00
Australian language	18.05	27.33	18.48	2.37
BCI	44.89	52.92	40.15	−10.55
Character trajectories	1.36	1.78	0.91	−33.32
CMU subject 16	3.67	10.67	3.67	0.00
ECG	18.50	14.00	14.50	−21.62
Graz	37.14	40.00	30.71	−17.31
Japanese vowels	2.03	39.06	2.03	0.00
Libras	8.61	4.17	5.00	−41.94
Non-invasive fetal ECG	9.99	16.84	9.54	−4.52
Pen digits	0.65	0.61	0.50	−22.53
Robot failure LP1	12.64	22.50	14.86	17.58
Robot failure LP2	32.00	38.00	32.00	0.00
Robot failure LP3	29.00	29.00	25.00	−13.79
Robot failure LP4	10.08	20.45	10.08	0.00
Robot failure LP5	29.30	37.32	28.75	−1.88
uWaveGestureLibrary	1.90	3.55	1.50	−21.16
Wafer	2.01	9.21	1.92	−4.15
MEAN				−9.60

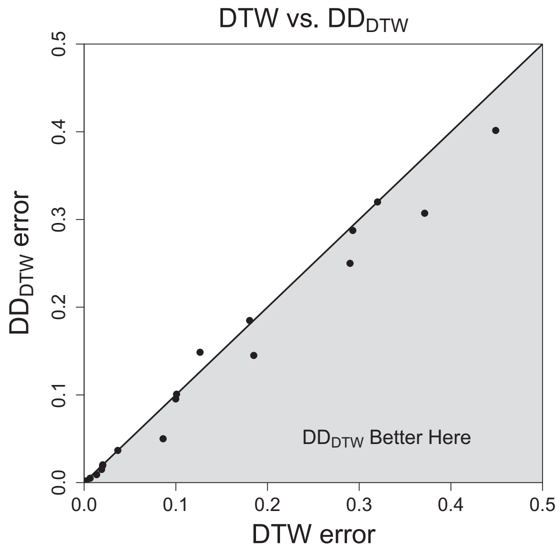

Fig. 4. Comparison of error rates (DTW vs. DD_{DTW}).

to a small degree on the number of parameters (especially for large values of n). Because of this, we can choose a large subset of parameters in the cross-validation process without increasing the computation time of the parameter tuning phase.

3. Experimental results

3.1. Experimental setup

The experiments are carried out on 18 data sets, which are all labeled data sets whose labels are given. The data sets originate from a plethora of different domains, including medicine, robotics, handwriting recognition, etc. Information on the time series used is presented in Table 1 (UCI – Bache & Lichman, 2013, UCR – Keogh et al., 2011; Carnegie Mellon University Motion Capture Database, 2014; Blankertz, Curio, & Müller, 2002; Leeb et al., 2007; Olszewski, 2001). The number of time series per data set varies from 47 to 10,992, the number of classes varies from 2 to 95, and the number of variables varies from 2 to 62.

The MTS samples in each data set are of different lengths. For each data set, the MTS samples are extended to the length of the longest MTS sample in the data set (Rodriguez et al., 2005). We extend all variables of MTS to the same length. For a short TS instance x with length n we enlarge it to a long instance y with length $n_{\max}$ by

$$y(j) = x(i), \quad \text{for} \quad i = \left\lfloor \frac{j-1}{n_{\max}-1}(n-1)+0.5 \right\rfloor \quad (j = 1, 2, \ldots, n_{\max}).$$

Some of the values in an MTS sample are duplicated in order to extend the sample. For instance, if we want to extend an MTS sample of length 75 to a length of 100, one out of every three values would be duplicated. In this way, all of the values in the original MTS sample are contained in the extended MTS sample.

For the classification process, the nearest neighbor method (1NN) is used for all of the distances being compared: DD_{DTW}, DTW and DDTW. We use the cross-validation (leave-one-out) method to find the best parameter α in our classifier DD_{DTW} on a training subset. If the minimum error rate is the same for more than one value of α, we choose the smallest of those values. A finite subset of parameters α is taken, from 0 to 1 with fixed step 0.01. For each data set we calculated the classification error rate using the 10-fold cross-validation method (1NN classifier).

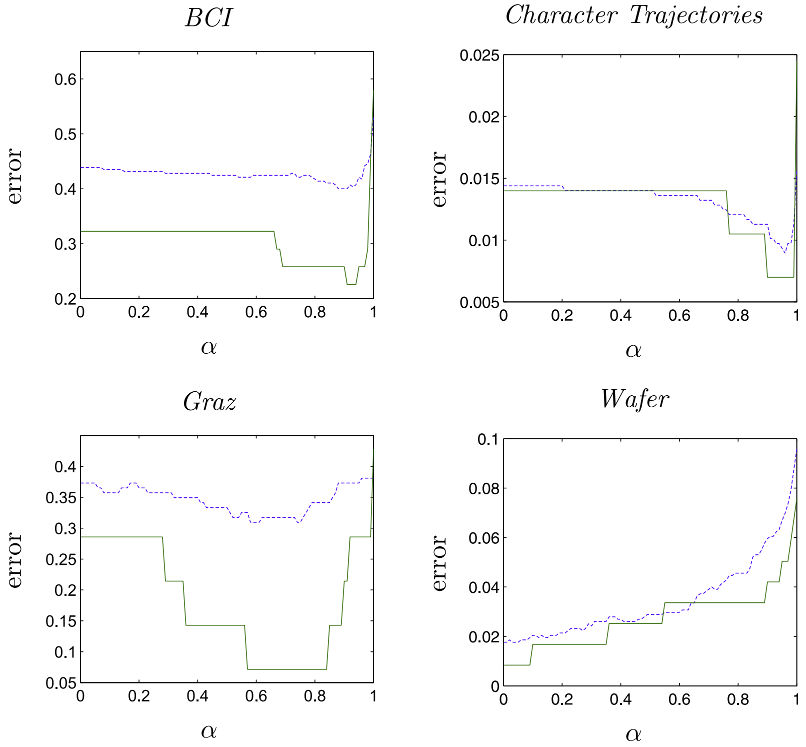

Fig. 5. Dependence of classification error rates on the parameter α for example data sets. Dashed line: CV error, solid line: test error.

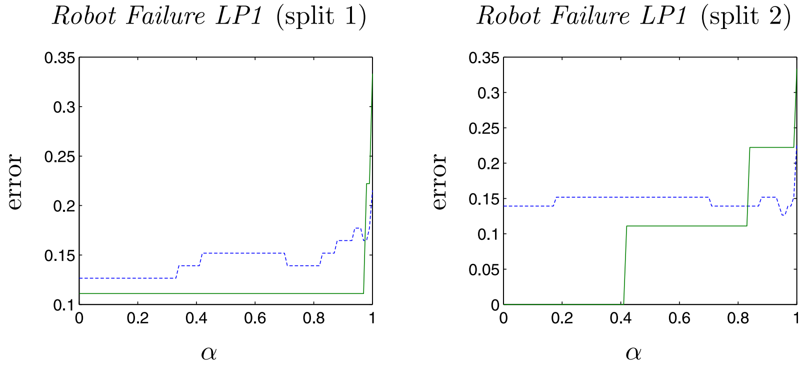

Fig. 6. Difference in prediction of the best value of the parameter α for two splits of the same data set. Dashed line: CV error, solid line: test error.

3.2. Results

Table 2 shows the main results. The columns DTW and DDTW contain (absolute) error rates with the 1NN method for the DTW and DDTW distances. In the column DD_{DTW} we have absolute error rates for the new parametric derivative method, and in the next column we have relative error rates. We may use the mean ratio of relative error rates across data sets as a measure of relative performance (Bauer & Kohavi, 1999; Quinlan, 1996). For two methods A and B with errors ε_A and ε_B the relative error rate is $\frac{\varepsilon_B - \varepsilon_A}{\varepsilon_A}$. In our situation we are comparing everything with DTW, hence the method A is DTW. The mean ratio of relative error rates is the average (over all data sets) of the relative errors between the pair of methods being compared. A value of the mean ratio of relative error rates less than 0 represents an improvement relative to the base method (DTW). This information is given in the last row of Table 2.

It can be clearly seen that we obtain a significant reduction in the average relative error of classification. The reduction amounts to 9.60. Comparing the methods DTW and DD_{DTW} our approach is better in the case of 11 data sets, while for 5 there is no difference and for 2 our approach is slightly worse.

A graphical comparison of the DD_{DTW} method with DTW is presented in Fig. 4. We see that the new method is clearly superior to the DTW distance on most of the examined data sets.

In Figs. 5 and 6 we can see the contribution of the component distances DTW and DDTW to the final DD_{DTW} distance. For each data set we choose one from the 10-fold splits on the training and testing subset to illustrate the contribution and correspondence of the CV (leave-one-out) and test error rate. It is clear that there is no one universal best value of the parameter α for all data sets. The α corresponding to the minimal error rate is different for each data set. On the other hand we can see that the minimum of error is well positioned – there is only one minimum for each error curve. The test error rate curve generally corresponds to the CV error rate curve, so we can predict quite well the best value of the parameter α. Only for two data sets does the minimum of the test error differ slightly from the CV error for some of the 10-fold splits (Fig. 6). For this reason DD_{DTW} is slightly worse than DTW on those data sets.

Finally, to confirm that the classifier based on DD_{DTW} is superior to the classifier based on DTW, we present a statistical comparison of their 10CV error rates on all 18 data sets. To statistically compare two classifiers over multiple data sets, Demšar (2006) recommends the Wilcoxon signed-ranks test. The Wilcoxon signed-ranks test is a non-parametric alternative to the paired t-test, which ranks the differences in the performances of two classifiers for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences. In our case we obtain a p-value equal to 0.02106. We see that the classifier based on DD_{DTW} is significantly better than the classifier based on DTW at a significance level of $\alpha = 0.05$.

4. Conclusions and future work

The introduction of a new distance measure for multivariate time series classification, as described here, would appear very promising. Comparison with other leading distances for MTS classification using the 1NN method clearly shows the advantage of the new measure. The parametric approach used in the DD_{DTW} method makes it possible to combine the advantages and avoid the disadvantages of the component methods DTW and DDTW. The new method adapts well to different data sets without showing signs of overfitting. The complexity of the construction and the running time of the new method are at a similar level as for its predecessor, constructed for univariate time series classification.

The parametric approach of the new method is a disadvantage with regard to computation time. Cross-validation is required to seek the best value of the parameter on the training data, which unfortunately increases the computation time in the learning phase. We can, however, use the standard lower bound technique to reduce the computation time for the nearest neighbor method. For the DD_{DTW} distance measure the lower bound is a combination of lower bounds for the component distances (DTW and DDTW). Moreover, the algorithm described in Section 2.5 accelerates the calculations on the training data in the cross-validation phase.

The future development of the method may involve transferring other similar parametric distances from univariate time series to the multivariate domain, for example by using additionally the second derivative (Górecki & Łuczak, 2014a) and higher or more general transforms (Górecki & Łuczak, 2014b), or other distances (Górecki, 2014) or other representations of the MTS.

It would be beneficial to focus on methods of reducing limitations, particularly in terms of the computational complexity of the distance, as well as reducing the time of the learning phase (cross-validation). Efforts may be made to develop more sophisticated lower bound techniques, using specific properties of the new method. On the other hand, we can use some approximate algorithms for calculating the DTW distance measure (and consequently DDTW and DD_{DTW}). One of the possibilities for further research may involve limiting the time of the learning phase by reducing the cross-validation computation. This might be achieved by decreasing the size of the training set, for example by pre-editing the training data or using other methods than cross-validation to find the parameter.

It is also possible to transfer the techniques presented here to the domain of unsupervised classification. The new distance measure can be applied in processes of cluster analysis. It would be necessary to develop methods to select the necessary parameter for the algorithm and to study its influence on the real quality of cluster analysis on the data sets. It appears that the combination approach could bring similar benefits there as in the case of supervised learning.

References

Aßfalg, J., Kriegel, H. P., Kröger P., Kunath, P., Pryakhin, A., & Renz, M. (2006) Similarity search on time series based on threshold queries. In *Proceedings of the 10th international conference on extending database technology* (pp. 276–294).

Bache, K., & Lichman, M. (2013). UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>.

Banko, Z., & Abonyi, J. (2012). Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications, 39*, 12814–12823.

Bauer, E., & Kohavi, R. (1999). An experimental comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning, 36*, 105–139.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *AAAI workshop on knowledge discovery in databases* (pp. 229–248).

Blankertz, B., Curio, G., & Müller, K. R. (2002). Classifying single trial EEG: towards brain computer interfacing. In T.G. Diettrich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural inf. proc. systems 14 (NIPS 01)*. Available from: <http://www.bbci.de/competition/ii/>.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time series analysis: Forecasting and control*. Wiley.

Carnegie Mellon University Motion Capture Database. (2014). Available from: http://mocap.cs.cmu.edu/.

Chen, L., & Ng, R. (2004). On the marriage of lp-norms and edit distance. In *Proceedings of the 30th international conference on very large data bases* (pp. 792–803).

Chen, L., Øzsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on management of data* (pp. 491–502).

Chen, Y., Nascimento, M. A., Ooi, B. C., & Tung, A. K. H. (2007). Spade: On shape-based pattern detection in streaming time series. In *Proceedings of the IEEE 23rd international conference on data engineering* (pp. 786–795).

Das, G., Gunopulos, D., & Mannila, H. (1997). Finding similar time series. In *Proceedings of the first european symposium on principles of data mining and knowledge discovery* (pp. 88–100).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. In *Proc. 34th int. conf. on very large data bases* (pp. 1542–1552).

Frentzos, E., Gratsias, K., & Theodoridis, Y. (2007). Index-based most similar trajectory search. In *Proceedings of the IEEE 23rd international conference on data engineering* (pp. 816–825).

Gavrila, D. M., & Davis, L. S. (1995). Towards 3-d model-based tracking and recognition of human movement: A multi-view approach. In *IEEE international workshop on automatic face and gesture recognition* (pp. 272–277).

Geurts, P., & Wehenkel, L. (2005). Segment and combine approach for non-parametric time-series classification. In *Proceedings of PKDD. Vol. 3721: LNAI 05* (pp. 478–485).

Ghalwash, M. F., Ramljak, D., & Obradović, Z. (2012). Early classification of multivariate time series using a hybrid HMM/SVM model. In *Proceedings of the IEEE international conference on bioinformatics and biomedicine (BIBM '12)* (pp. 1–6).

Ghalwash, D., & Obradović, Z. (2012). Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC Bioinformatics, 13*, 195.

Górecki, T., & Łuczak, M. (2013). Using derivatives in time series classification. *Data Mining and Knowledge Discovery, 26*(2), 310–331.

Górecki, T., & Łuczak, M. (2014a). First and second derivative in time series classification using DTW. *Communications in Statistics-Simulation and Computation, 43*(9), 2081–2092.

Górecki, T., & Łuczak, M. (2014b). Non-isometric transforms in time series classification using DTW. *Knowledge-Based Systems, 61*, 98–108.

Górecki, T. (2014). Using derivatives in a longest common subsequence dissimilarity measure for time series classification. *Pattern Recognition Letters, 45C*, 99–105.

Hayashi, A., Mizuhara, Y., & Suematsu, N. (2005). Embedding time series data for classification. In *Proceedings of MLDM. Vol. 3587: LNAI 05* (pp. 356–365).

Kadous, M. W., & Sammut, C. (2005). Classification of multivariate time series and structured data using constructive induction. *Machine Learning, 58*(1–2), 179–216.

Keogh, E. (2002). Exact indexing of dynamic time warping. In *28th International conference on very large data bases* (pp. 406–417).

Keogh, E., & Pazzani, M. (2001). Dynamic time warping with higher order features. In *First SIAM international conference on data mining (SDM'2001), Chicago, USA*.

Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., et al. (2011). *The UCR time series classification/clustering homepage*. <http://www.cs.ucr.edu/eamonn/time_series_data/>.

Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 15*, 473–482<http://www.bbci.de/competition/iv/>.

Lemire, D. (2009). Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition, 42*(9), 2169–2180.

Li, C., Khan, L., & Prabhakaran, B. (2006). Real-time classification of variable length multiattribute motion data. *International Journal of Knowledge and Information Systems, 10*(2), 163–183.

Li, C., Khan, L., & Prabhakaran, B. (2007). Feature selection for classification of variable length multi-attribute motions. In V. A. Petrushin & L. Khan (Eds.), *Multimedia data mining and knowledge discovery* (pp. 116–137). Springer-Verlag.

Maharaj, E. A. (1999). Comparison and classification of stationary multivariate time series. *Pattern Recognition, 32*(7), 1129–1138.

Mello, R. F., & Gondra, I. (2008). Multi-dimensional dynamic time warping for image texture similarity. In G. Zaverucha & A. L. Costa (Eds.), *Proceedings of the 19th Brazilian symposium on artificial intelligence: Advances in artificial intelligence (SBIA '08)* (pp. 23–32). Springer-Verlag.

Morse, M. D., & Patel J. M. (2007). An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM SIGMOD international conference on management of data* (pp. 569–580).

Nakamura, T., Taki, K., Nomiya, H., Seki, K., & Uehara, K. (2013). A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis and Applications, 16*, 535–548.

Olszewski, R. T. (2001). *Generalized feature extraction for structural pattern recognition in time-series data* (Ph.D. thesis). Carnegie Mellon University, Pittsburgh. Available from: <http://www.cs.cmu.edu/bobski>.

Prieto, O. J., Alonso-González, C. J., & Rodríguez, J. J. (2014). Stacking for multivariate time series classification. *Pattern Analysis and Applications*. http://dx.doi.org/10.1007/s10044-013-0351-9.

Quinlan, R. (1996). Bagging, boosting, and C4.5. In *AAAI/IAAI* (Vol. 1, pp. 725–730).

Rodriguez, J. J., Alonso, C. J., & Maestro, J. A. (2005). Support vector machines of interval based features for time series classification. *Knowledge-Based Systems, 18*, 171–178.

Spiegel, S., Gaebler, J., Lommatzsch, A., De Luca, E., & Albayrak, S. (2011). Pattern recognition and classification for multivariate time series. In *Proceedings of the fifth international workshop on knowledge discovery from sensor data (SensorKDD '11)* (pp. 34–42). New York: ACM.

ten Holt, G. A., Reinders, M. J. T., & Hendriks, E. A. (2007). Multi-dimensional dynamic time warping for gesture recognition. In *Proceedings of the 13th annual conference of the advanced school for computing and imaging*.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 216–225).

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2006). Indexing multi-dimensional time-series. *The VLDB Journal, 15*(1), 1–20.

Weng, X. (2013). Classification of multivariate time series using supervised locality preserving projection. In *Proceedings of the third international conference on intelligent system design and engineering applications (ISDEA '13)* (pp. 428–431).

Weng, X., & Shen, J. (2008a). Classification of multivariate time series using two-dimensional singular value decomposition. *Knowledge-Based Systems, 21*(7), 535–539.

Weng, X., & Shen, J. (2008b). Classification of multivariate time series using locality preserving projections. *Knowledge-Based Systems, 21*(7), 581–587.

Wöllmer, M., Al-Hames, M., Eyben, F., Schuller, B., & Rigoll, G. (2009). A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing, 73*(1–3), 366–380.

Yang, K., Yoon, H., & Shahabi C. (2005). CLeVer: A feature subset selection technique for multivariate time series. In *Proceedings of PAKDD. Vol: 3518. LNAI 05* (pp. 516–522).