Learning a Mahalanobis Distance based Dynamic Time Warping Measure for Multivariate Time Series Classification

Jiangyuan Mei, *Student Member, IEEE,* Meizhu Liu, *Member, IEEE,* Yuan-Fang Wang, *Member, IEEE* and Huijun Gao, *Fellow, IEEE*

***Abstract*—Multivariate time series (MTS) data sets broadly exist in numerous fields, including health care, multimedia, finance and biometrics. How to classify MTS accurately has become a hot research point since it is an important element in many computer vision and pattern recognition applications. In this paper, we propose a Mahalanobis distance based Dynamic Time Warping (MDDTW) measure for MTS classification. The Mahalanobis distance builds an accurate relationship between each variable and its corresponding category. It is utilized to calculate the local distance between vectors in MTS. Then we use Dynamic Time Warping (DTW) to align those MTS which are out of sync or with different lengths. After that, how to learn an accurate Mahalanobis distance function becomes another key problem. This work establishes a LogDet divergence based metric learning with triplet constraint (LDMLT) model which can learn Mahalanobis matrix with high precision and robustness. Furthermore, the proposed method is applied on 9 MTS data sets which are selected from the UCI machine learning repository and Robert T. Olszewski's homepage, and the results demonstrate the improved performance of the proposed approach.**

***Keywords*—Multivariate Time Series, Metric Learning, Dynamic Time Warping, Mahalanobis Distance**

I. Introduction

MOST machine learning and pattern recognition algorithms are constructed based on measuring the similarity of a feature space. Traditional methods based on static features which are extracted from specific points have been broadly explored by researchers. However, in some applications, the extracted features change over time. For example, we can't diagnose heart disease only by observing some static data from electrocardiograms. Unlike static data, the time series comprises dynamic features which are varying with time [1]. Thus, time series can provide further information on how subject changes, such as heart activity observed from an electrocardiogram. Time series data are of wide interest as they are used in various applications, such as complex system states prediction [2], signature verification [3], earthquake prediction [4] and action recognition [5].

There are many kinds of time series data. In [1], the distinctions of time series are made according to the data types. First, there are discrete-valued and continuous-valued time series. In most applications, the continuous-valued sequences are always sampled by numerous sensors with different samplers and frequencies. Thus, we only consider the discrete-valued time series in this paper. Second, time series can be classified as uniformly or non-uniformly sampled time series. With identical samplers and frequencies, measuring the distance between uniformly time series is not complex. However, when comparing with non-uniformly sampled time series, there is no one-to-one correspondence because of different frequencies and lengths. The main problem is how to align the non-uniformly sampled time series, which makes the distance measuring difficult. Third, according to the number of variables, the time series can be distinguished as univariate and multivariate ones. The similarity measurement for univariate time series (UTS) has been widely researched [6]–[9]. However, UTS can only represent one property of instances and it is not sufficient for some applications. In these applications, MTS should be utilized to represent multiple properties of instances. And how to measure the divergence between MTS has become a big challenge due to the following reasons: a) MTS instance has lots of variables. On one hand, if MTS instance is broken into several UTS, the correlations among the variables will be lost and the distance measuring will be inaccurate [10], [11]. Thus, MTS should be treated as a whole. On the other hand, the relevance of each variable in MTS to the category of an instance may be different. Among these variables, some of them have a strong correlation with the label of instances while others may be polluted by noise and have weak or no correlation. Thus, we shouldn't consider MTS as a whole completely. b) For the non-uniformly sampled MTS, the MTS instances may be with different lengths and phases. There is no one-to-one correspondence between two MTS when measuring their distance. We should consider the synchronization when measuring the different variables of MTS at the same time. Therefore, how to align two MTS is another challenge.

This paper mainly aims at accurate MTS classification, which is one of the bases in various computer vision and pattern recognition applications. Our objective is to propose a supervised learning algorithm to label multivariate sequences

J. Mei is with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, 150001 Harbin, China, and also with the Department of Computer Science, University of California, Santa Barbara, CA 93106. (e-mail: meijiangyuan@hit.edu.cn; meijiangyuan@cs.ucsb.edu.)

M. Liu is with Yahoo Labs, Yahoo Inc., NYC, NY, 10018, USA. (e-mail: liufkmc@gmail.com.)

Y. F. Wang is with the Department of Computer Science, University of California, Santa Barbara, CA 93106. (e-mail: yfwang@cs.ucsb.edu.)

H. Gao is with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, 150001 Harbin, China.(e-mail: hjgao@hit.edu.cn.)

with variable lengths and phases. One of the key problems is to find an efficient measure to compare MTS. Various distance measures might be used for MTS comparison, including Euclidean distance, short time series distance [12], dynamic time warping distance, probability-based distance function [13], Kullback-Liebler distance [1], J divergence and symmetric Chernoff information divergence [14], 2-dimensional singular value decomposition ($2d$SVD) [10], and locality preserving projections (LPP) [15]. Among these methods, Euclidean distance and short time series distance all require that the time series has the same phases, which are not suitable for comparing those non-uniform MTS. Probability-based distance function and Kullback-Liebler distance all regard time series as probability distributions. However, non-linear warps between two MTS would result in large difference between two probability distributions. Meanwhile, Kullback-Liebler distance requires that two time series should have the same lengthes, which is not applicative in many real situations. J divergence and symmetric Chernoff information divergence can be utilized to measure the distance among spectral matrix estimators for stationary MTS. However, these two methods can only deal with linearly warped non-uniform MTS. If there is non-linear warps between two MTS, these two divergences will lose efficiency. $2d$SVD captures of eigenvectors of row-row and column-column covariance matrices as features of MTS, and the distance between two MTS is computed by measuring the distance of these features. The LPP method is an extension of $2d$SVD. The main idea of LPP is to project the feature vectors extracted using $2d$SVD into a lower-dimensional feature speace, in which the MTS samples related to the same class are close to each other. The main problem of these two methods is that they treat the MTS as a whole completely. They are not robust because they are sensitive to noise and outliers.

Compared with above methods, DTW distance has lots of advantages. DTW was first introduced into measuring time series by Berndt [16] to overcome the phase aberration problem in time series matching. The main idea of DTW algorithm to reduce the problem of time series comparing to a static problem by suitably transforming the set of input sequences into a rectangular table composed by a fixed length [17]. The DTW algorithm is good at finding the optimal alignment between two non-uniform time series [18], [19]. It uses a dynamic programming technique to find the minimum distance by stretching or shrinking the linearly or non-linearly warped time series [19]. However, traditional DTW method can only process UTS. To overcome this problem, the DTW algorithm is extended to multiple dimensions and a multidimensional DTW algorithm was proposed [20]. The multidimensional DTW algorithm regards a time point in DTW as a vector and the corresponding local distance measure is chosen as the Euclidean distance, and the aligning process is the same to the traditional DTW algorithm. The weak point of this method is that it assigns the same weight to each variable, which is not practical in the real applications. The Euclidean distance cannot accurately measure the distances among these local vectors. As mentioned above, variables have different correlation with the label of subjects and there might also be

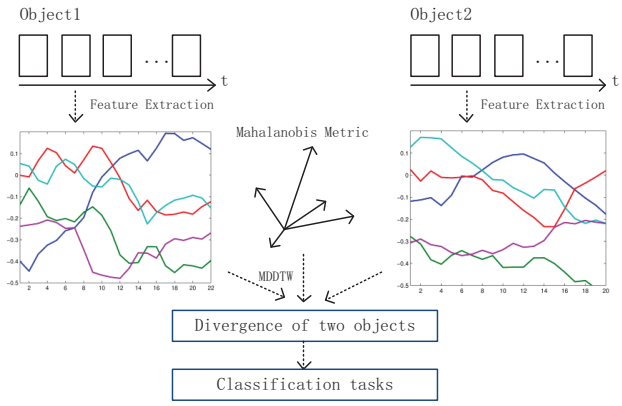


Fig. 1. The framework of MDDWT measure. Multivariate features are extracted from time series objects using the state-of-the-art feature extraction methods. Then, these MTS are compared using the Mahalanobis distance based dynamic time warping measure.

correlations among the different variables. A feasible strategy is to use the Mahalanobis distance function to measure the local distance of vectors in MTS.

The Mahalanobis distance is a unitless measure parameterized by a positive semi definite (PSD) matrix. Compared with other various metrics, Mahalanobis distance has numerous advantages. First, the Mahalanobis metric takes into account the correlations of different variables and a more accurate relationship between variables and labels of MTS can be established. Secondly, Mahalanobis distance has a multivariate effect size. It means that the scale of the Mahalanobis distance has no effect on the performance of classification or clustering of MTS. All these advantages make Mahalanobis distance a good local distance metric [21] for MTS comparison. However, how to learn a Mahalanobis metric from the training samples is a complex process. And the process is named as metric learning. Traditional metric learning algorithms, such as Probabilistic Global Distance Metric Learning (PGDM) [22], BoostMetric [23], MetricBoost [24] and Information-Theoretic Metric Learning (ITML) [25] can only deal with the metric learning process with static features. How to extend metric learning algorithm to process MTS data is another key problem.

In this paper, based on the above analysis, we propose a novel framework for accurate MTS classification. In this framework, we firstly propose a novel MDDWT method to measure the divergence among MTS, as shown in Fig. 1. The Mahalanobis distance over the feature space is utilized to compute the distance of local vectors in MTS. After applying the DWT algorithm, the non-uniform MTS can be aligned as MTS with the same phase and length. In this framework, the selection of Mahalanobis distance is very important. Therefore, we also propose a LogDet divergence based metric learning algorithm for aligned MTS to learn the Mahalanobis distance. After that, basic classification methods, including k-nearest neighbors (k-NN) and support vector machine (SVM) algorithm along with MDDWT measure can be utilized for MTS classification. Furthermore, the comparison experiments

TABLE I. SOME NOTATIONS USED IN THIS PAPER

Symbol	Definition
x	a $1 \times m$ matrix representing a UTS
X	a $d \times m$ matrix representing a MTS
x_i	a $1 \times m$ matrix representing the i^{th} row of X
X^i	a $d \times 1$ matrix representing the i^{th} column of X
W	a $2 \times p$ matrix representing the optimal warp path between two UTS or MTS
$\bar{x}$	a $1 \times p$ matrix representing extended UTS of x with the constructed optimal warp path W
$\bar{X}$	a $d \times p$ matrix representing extended MTS of X With the constructed optimal warp path W

were conducted on several MTS data sets to demonstrate the performance of the proposed framework by comparing with the state-of-the-art methods.

The remainder of this paper is organized as follows. In Section II, some related literature and background knowledge are presented. Then, the proposed MDDWT measure is described in Section III. Section IV illustrates the LogDet divergence based metric learning algorithm for MTS. Section V gives experimental results on several MTS data sets to demonstrate the effectiveness of the proposed algorithm. Finally, we draw conclusions and point out future directions in Section VI.

II. RELATED WORKS

In this section, we give a brief review of some background knowledge, including DTW, Mahalanobis distance and metric learning. Table. I illustrates the definition of some notations which can be easily confused in the remainder of this paper.

A. Dynamic Time Warping

In time series analysis, DTW is an algorithm which can measure the divergence between two time series with different phases and lengths. The basic idea of DTW is to calculate an optimal warp path between two given time series. With the obtained warp path, the two given time series can be warped nonlinearly in the time dimension. After that, these two extended time series will be placed in one-to-one correspondence, and their similarity can be measured easily.

Given two UTS $x(i)$, $i = 1, 2, \cdots, m$ and $y(j)$, $j = 1, 2, \cdots, n$. The optimal warp path W is expressed as

$$W = \begin{pmatrix} w_x(k) \\ w_y(k) \end{pmatrix},\ k = 1, 2, \cdots, p$$

where $w_x(k)$ represents an index from time series $x(i)$, and $w_y(k)$ represents an index from time series $y(j)$. p is the length of the warp path W. $(w_x(k), w_y(k))'$ indicates that the $w_x(k)^{th}$ element in $x(i)$ is corresponded to the $w_y(k)^{th}$ element in $y(j)$.

There are two constraints when constructing the warp path W [19]. The first one is that all indices of both time series should be used in the warp path W. The second one is that the warp path W should be continuous and monotonically increasing. With these two constraints, the starting point of warp path W is restricted as $W(1) = (1,1)'$ and the ending point is restricted as $W(p) = (m,n)'$. At the same time, these constraints also require that adjacent points $W(k)$ and $W(k+1)$ should satisfy that

$$\begin{cases} w_x(k) \le w_x(k+1) \le w_x(k)+1 \\ w_y(k) \le w_y(k+1) \le w_y(k)+1 \end{cases}.$$

Therefore, there are only three choice for $W(k+1)$, that is $(w_x(k), w_y(k+1))'$, $(w_x(k+1), w_y(k))'$, and $(w_x(k+1), w_y(k+1))'$. Meanwhile, the length of W satisfy that $p \in [\max(m,n), m+n]$.

With the constructed optimal warp path W, the two given time series $x(i)$ and $y(j)$ can be extended to two new time series $\bar{x}(k)$ and $\bar{y}(k)$, expressed as

$$\begin{cases} \bar{x}(k) = x(w_x(k)) \\ \bar{y}(k) = y(w_y(k)) \end{cases} k = 1, 2, \cdots, p$$

And the warp distance between time series $x(i)$ and $y(j)$ can be represented by the Euclidean distance between these two extended time series $\bar{x}(k)$ and $\bar{y}(k)$, expressed as

$$DWT(x,y) = D(\bar{x}, \bar{y}) = \sum_{k=1}^{p} D(x(w_x(k)), y(w_y(k))).$$

The dynamic time warping can be summarized as following steps [19], [26]. First of all, a cost distance matrix $Dist(i,j)$, $i = 1, 2, \cdots, l$, $j = 1, 2, \cdots, m$, is constructed. In this matrix, each element $Dist(i,j)$ represents the minimum warp distance of sub time series x of length i and sub time series y of length j. The corresponding path is named as W_{ij}. Then, as mentioned above, the warp path W_{ij} includes $(i,j)'$ and one of the following choice: $(i-1,j)'$, $(i,j-1)'$ or $(i-1,j-1)'$, which can construct the relationship between $Dist(i,j)$ and $Dist(i-1,j-1)$, $Dist(i-1,j)$ or $Dist(i,j-1)$. Because $Dist(i,j)$ represents the minimum warp distance, thus the relationship is expressed as

$$Dist(i,j) = D(x(i), y(j)) + \min \begin{cases} Dist(i-1,j-1) \\ Dist(i-1,j) \\ Dist(i,j-1) \end{cases},$$

where $Dist(1,1) = d(x(1), y(1))$. After computing all the elements in the cost distance matrix $Dist(i,j)$, $Dist(m,n)$ equals to the minimum warp $DWT(x,y)$ distance between time series $x(i)$ and $y(j)$, and the corresponding warp path W is the most optimal warp path.

B. Mahalanobis Distance and Metric Learning

The Mahalanobis distance is a standard distance metric. It satisfies all the conditions of metric definition, including non-negativity, symmetry, triangle inequality and identity of indiscernibles. Given two vectors u and v, the square Mahalanobis distance parametrized by a symmetric PSD matrix M between instances u and v is defined as

$$D_M(u,v) = (u-v)^T M (u-v). \quad (1)$$

The PSD matrix M is named as Mahalanobis matrix. When $M = I$, the Mahalanobis distance degenerates to the Euclidean distance. Applying singular value decomposition, the

Mahalanobis matrix can be decomposed as $M = H\Sigma H^T$. H is a unitary matrix which satisfies $HH^T = I$. The left unitary matrix is the transpose of right unitary matrix due to the symmetry of Mahalanobis matrix M. Σ is a diagonal matrix which contains all the singular values. Thus, the square Mahalanobis distance can be rewritten as

$$\begin{aligned} D_M(u,v) &= (u-v)^T H\Sigma H^T (u-v) \\ &= \left(H^T u - H^T v\right)^T \Sigma \left(H^T u - H^T v\right) \end{aligned} \tag{2}$$

From Eqn. 2 we can see that the Mahalanobis distance has two main functions. The first one is to find the best orthogonal matrix H to remove the correlation among variates. The second one is to assign weights Σ for the new variates. Therefore, the Mahalanobis distance can measure the distance between two vectors efficiently. However, how to learn such a Mahalanobis distance is a complex procedure.

The purpose of metric learning is to learn a Mahalanobis distance which can represent the relevance of features to the labels of training instances. The obtained Mahalanobis distance should emphasize the relevant features while decrease the effect of irrelevant dimensions [27]. There are lots of metric learning algorithms in literature. The most famous one is large margin nearest neighbor (LMNN) [28] metric learning algorithm. This method applies the idea of SVM to the metric learning. The objective Mahalanobis function is to maintain consistency of data in the same class while keeping a large margin at the boundaries of different categories. The work [29] improves the LMNN algorithm by combing the hierarchical distance metric learning (HDM) with LMNN. And the so-called HLMNN achieve the better performance of multi-class data classification. In [30], the probabilistic global distance metric learning (PGDM) is proposed. In this method, the training samples are labelled as 'similar' and 'dissimilar' pairwise constraints according to the categories of instances. Then the metric learning process is formulated as a convex optimization problem with respect to these constraints. Another metric learning strategy is proposed in [23]. This algorithm regards the Mahalanobis distance as the composing of trace-one rank-one matrices. And a boosting-based technique could be applied in the metric learning process. In [23], triplet constraints which represent instances in the same category are more similar than instances in a different category are firstly used in metric learning. Triplet constraints represent the proximity relationships, which are weaker than pairwise constraints. Of these two methods, the Mahalanobis matrix is solved by iterative projection algorithms. However, the Mahalanobis matrix is updated by using all the pairs or triplets in each optimization iteration, which is not suitable for on-line applications. To solve this problem, the information theoretic metric learning (ITML) method [25] formulates metric learning problem as that of minimizing the differential relative entropy between two multivariate Gaussian distributions under pair constraints on the distance function. ITML only uses one pairwise constraint in each optimization iteration, and the efficiency is very high. In [31], the authors proposed a Mahalanobis metric learning algorithm which is based on gradient descent. The algorithm also optimized the Mahalanobis metric step by step as receiving the pairwise constraints. However, the pairwise constraints are not as weak as triplet constraints, which will lead to very conservative results.

III. Mahalanobis distance based Dynamic Time Warping

In this section, we present the MDDTW measure for MTS. Given two MTS X and Y,

$$X = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(m) \\ x_2(1) & x_2(2) & \cdots & x_2(m) \\ \vdots & \vdots & \ddots & \vdots \\ x_d(1) & x_d(2) & \cdots & x_d(m) \end{bmatrix}$$

and

$$Y = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(n) \\ y_2(1) & y_2(2) & \cdots & y_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ y_d(1) & y_d(2) & \cdots & y_d(n) \end{bmatrix},$$

where d is the number of variables. How to accurately measure the distance between X and Y is the main problem in this section.

As mentioned above, the traditional dynamic time warping algorithms can only deal with univariate time series because $D(\cdot,\cdot)$ is the distance between two data points. However, in many applications, UTS can only provide partial information, which is not sufficient for accurate time series classification. In the work [32], the traditional DTW is extended for MTS. The local distance measure $D(\cdot,\cdot)$ is defined as

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

where X^i represents the i^{th} column in X, and Y^j represents the j^{th} column in Y. Then the minimum distance warp path calculated based on all obtained $D\left(X^i, Y^j\right)$ is the optimal alignment between two time series. In this method, the function $D(\cdot,\cdot)$ is chosen as the Euclidean distance. Thus, this method is named as Euclidean distance based dynamic time warping (EDWT) measure. One deficiency of this method is that it assigns the same weight to each variable, which is not practical in many situations. First, each variable may have different units of measure in the collection process. Second, some variables may contain lots of noise and outliers, which will disturb the classification results. Third, some variables may have a coupling relationship. Noise and outliers in one variable would affect several other variables. Hence, different variables will play different roles in determining the categories of instances. Therefore, the Euclidean distance can't measure the local distance accurately.

In the work [32], another measure for comparing MTS is proposed. The so called lower-bounding measure (LBM) is also an extension of DWT, but it reduces the calculation sharply. When comparing two MTS X and Y, two time series U and L are constructed to approximate MTS Y. U is the

upper boundary in a neighbourhood which is determined by the path constraint, and L is the lower boundary. And the lower-bounding measure is expressed as

$$LBM(X,Y) = \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{d}\begin{cases}(x_i(j)-u_i(j))^2 & if\ x_i(j)>u_i(j)\\ (x_i(j)-l_i(j))^2 & if\ x_i(j)<l_i(j)\\ 0 & otherwise\end{cases}}$$

The main advantage of this method is that the computation is very cheap. However, this method is sensitive to the path constraints because the local measure is non-linear.

The main difference between the proposed MDDTW measure and the traditional DTW measure lies in the choosing of the local distance function $D(\cdot,\cdot)$. In this paper, the local distance measure $D(\cdot,\cdot)$ is chosen as Mahalanobis distance

$$D_M\left(X^i,Y^j\right)=\left(X^i-Y^j\right)^T M\left(X^i-Y^j\right).$$

And the corresponding DTW algorithm is expressed as

$$Dist_M(i,j)=D_M\left(X^i,Y^j\right)+\min\begin{cases}Dist_M(i-1,j-1)\\ Dist_M(i-1,j)\\ Dist_M(i,j-1)\end{cases},$$

The initial condition is $Dist_M(1,1)=D_M\left(X^1,Y^1\right)$. Fig. 2 illustrates the optimized warping path between two given MTS X and Y using the MDDTW algorithm.

As mentioned above, the optimal warp path W is the one which has a minimum sum of distance from $(1,1)$ to (m,n). At the same time, the optimal warp path W is also a way to find the optimal alignment between two multivariate time series. Furthermore, W also indicates that how MTS X and Y stretch or shrink along its time axis. Thus, we define two new multivariate time series sequences $\bar{X}_{d\times p}$ and $\bar{Y}_{d\times p}$ as

$$\begin{cases}\bar{X}^k=X^{(w_x(k))}\\ \bar{Y}^k=Y^{(w_y(k))}\end{cases}.$$

As shown in Fig. 3, using the information in the warp path W, the original X and Y will be mapped to $\bar{X}$ and $\bar{Y}$. And there is one-to-one correspondence between these two new MTS $\bar{X}$ and $\bar{Y}$. Therefore, the MDDTW measure $DWT_M(X,Y)=Dist_M(m,n)$ could be rewritten as

$$\begin{aligned}DWT_M(X,Y)&=\sum_{k=1}^{p}D_M\left(X^{w_x(k)},Y^{w_y(k)}\right)\\&=\sum_{k=1}^{p}D_M\left(\bar{X}^k,\bar{Y}^k\right)\\&=\sum_{k=1}^{p}\left(\bar{X}^k-\bar{Y}^k\right)^T M\left(\bar{X}^k-\bar{Y}^k\right)\\&=trace\left(P^TMP\right)\end{aligned}\tag{3}$$

where $P_{d\times p}=\bar{X}_{d\times p}-\bar{Y}_{d\times p}$.

There are several advantages when using MDDTW measure to compare MTS. First of all, the variables of MTS stretch or shrink along time axis integrally instead independently.

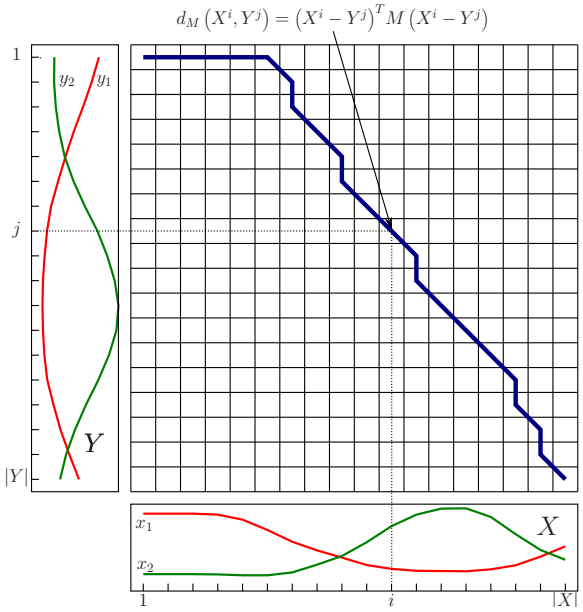
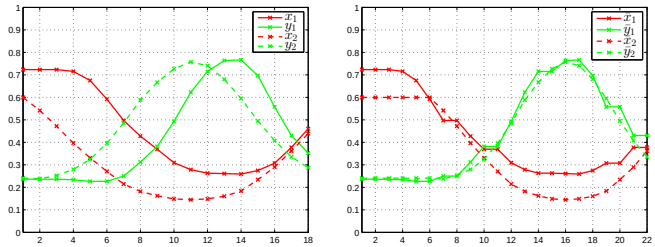


Fig. 2. Optimized warping path between two MTS X and Y.



(a) (b)

Fig. 3. The MDDTW between MTS X and Y. (a) The original MTS X and Y; (b) The extended MTS $\bar{X}$ and $\bar{Y}$.

This will not break the relationship among variables. Besides, a good Mahalanobis distance will rebuild an accurate relationship among variables. The noise and outliers in some variables will be suppressed when comparing MTS, which is benefitial for precise MTS classification. Furthermore, the MDDTW measure can be expressed as a very simple form as Eqn. 3. It will be easy to study the corresponding metric learning algorithm with that form in the following section. The computational complexity of the proposed measure is about $O(mn)$, which is similar to that of standard DTW. The main computational time is spent on finding the optimal path W. And the algorithm can be accelerated by applying techniques such as FastDTW [19] and SparseDTW [33].

IV. LogDet divergence Based Metric Learning

As mentioned above, the MDDTW measure has its own advantages compared with the traditional measures. And the main reason for these advantages is that Mahalanobis function can accurately reveal the relationships between variables and categories of instances. Therefore, how to learn an appropriate Mahalanobis metric is the key problem in MTS classification.

The goal for learning a metric is to emphasize relevant dimensions while reducing the influence of non-informative dimensions [27]. In other words, using the obtained Mahalanobis distance function, the instances which are far away in terms of Euclidean distance but within the same class should be closer. Meanwhile, the instances which are close in terms of Euclidean distance but belonging to different classes of instances should be far away. A good metric learning algorithm should have three properties. The first one is that the algorithm should be global. The algorithm should avoid some local minima and find a global minimum. As many as useful samples should be used for training. However, due to the limitation of algorithmic efficiency, only a part of the whole samples could be used in the training process in most cases. And inappropriately selecting part of the whole samples always results in the over-fitting problem. Therefore, how to select efficient training samples should be well considered in metric learning algorithms. Another property is that the algorithmic efficiency should not be too low. Metric learning algorithms should be scalable with respect to the number of the training samples. The third property is that the labels of the training samples should be as weak as possible. Training samples with strict labels are hard to obtain in most real-world applications. Thus weaker labels mean more practical.

There are three kinds of constraints for metric learning, i.e., class label [34], pairwise label [25], [35] and triplet label. The class label gives each instance a definite label, which indicates that the instance belongs to this category. The pairwise label indicates the similarity or dissimilarity of the instance pairs. If two instances belongs to the same category, the pair is labelled as similarity and their target Mahalanobis distance should be smaller than a desired superior limit; if not, the pair will be labelled as dissimilarity and their target Mahalanobis distance should be larger than a desired lower limit. Although pairwise labels are weaker than class labels [34], some of the constraints are needless [36]. The pairwise label still has some limitations in practical applications. In literature [24], [27], an even weaker representation called triplet label is introduced into metric learning algorithms. The triplet label $\{x, y, z\}$ requires that the instance x should be more similar to the instance y than the instance z using the target Mahalanobis distance function, where instances x and y are in the same category while z is in different category. The work in [24], [37] pointed out that triplet constraints can be derived from pairwise constraints, but not vice versa. In our previous work [36], we have demonstrated that triplet constraints are weaker than the pairwise constraints theoretically. Thus, triplet constraint is the weakest one as well as the most natural constraint in these three constraints. In this paper, considering the above three properties, a novel and practical metric learning model for MTS is proposed.

First of all, to make the metric learning algorithm more realizable in real applications, the triplet constaints are utilized to train the Mahalanobis matrix. The objective of metric learning is to find a PSD matrix M to ensure all triplet constraints $\{X, Y, Z\}$ satisfy that the MDDTW measure between two MTS X and Y in the same category is closer than that of X and Z in different categories, expressed as

$$DWT_M(X, Y) - DWT_M(X, Z) < -\rho, \tag{4}$$

where $\rho > 0$ represents the target margin.

In order to avoid getting in a local minimum, we should use as many triplets as possible during training. The total number of triplet constraints is the cubic of the number of the training samples. To solve this problem, we have two main strategies. One strategy is to choose part of all possible triplets randomly. Another strategy is that we choose the most useful triplets for training. This work has been done in our previous work [36], and the dynamic triplets building strategy is used in our framework for choosing triplets. How to guarantee that the metric learning process is scalable with respect to the size of the training samples is another problem. In this paper, we adopt an online metric learning framework [25], [38] to learn the Mahalanobis distance function. The model trains one constraint at a time and the Mahalanobis distance function changes step by step as constraints received. Assume M_t is a known quantity which represents the current obtained Mahalanobis distance at time step t. When received a triplet constraint (X_t, Y_t, Z_t), if the constraint satisfies the request of Eqn. 4, there is no loss when using the current M_t to measure these MTS; if not, the loss function is expressed as

$$l(M) = \rho + DWT_M(X, Y) - DWT_M(X, Z).$$

And the current M_t should be updated to a better Mahalanobis distance to reduce the loss. The optimized M will be picked as M_{t+1} at the next time step $t+1$. When the total loss function $L(M) = \sum_t \ell(M_t)$ reaches its minimum, the obtained M is the closest to the objective distance function. In metric learning algorithm, we should also focus on the stability of the learning process. A regularization term is added to the metric learning model to guarantee that the Mahalanobis matrix changes gradually and stably in the process. Thus, the regularization term should be able to measure the divergence of two matrices. There are various kinds of matrix divergence, including squared Frobenius norm, Neumann divergence [39]. Most of them are derived from Bregman matrix divergence, which is defined as

$$\begin{aligned} D_\phi(M, M_t) = \phi(M) - \phi(M_t) \\ - tr\left(\left(\nabla\phi(M_t)\right)^T (M - M_t)\right), \end{aligned} \tag{5}$$

where the function $tr()$ stands for the trace of a matrix. Properties of Bregman matrix divergence $D_\phi(M, M_t)$ are determined by the differentiable function $\phi(M)$. When the differentiable function $\phi(M) = -\sum_i \log \lambda_i = -\log(\det(M))$ is chosen as the Burg entropy of the eigenvalues λ_i, the corresponding Bregman matrix divergence is called LogDet divergence [40],

$$D_{ld}(M, M_t) = tr\left(MM_t^{-1}\right) - \log\left(\det\left(MM_t^{-1}\right)\right) - n. \tag{6}$$

where n is the dimension of M. There are several advantages when using LogDet divergence to regularize the metric learning process. First, the LogDet divergence between the covariance matrices is equivalent to the Kullback−Leibler

divergence between corresponding multivariate Gaussian distributions [41]. Second, the LogDet divergence is general linear group transformation invariant, i.e. $D_{ld}(M, M_t) = D_{ld}\left(S^T M S, S^T M_t S\right)$, where S is an invertible matrix [25]. These good properties make LogDet divergence very useful in metric learning.

Applying the LogDet divergence as the regularization term, the proposed LogDet divergence based metric learning model for MTS is to solve the following iterative minimization problem,

$$M_{t+1} = \arg\min_{M \succ 0} D_{ld}(M, M_t) + \eta_t \ell(M) \tag{7}$$

where $\eta_t > 0$ is a regularization parameter which balances the regularization function $D_{ld}(M, M_t)$ and loss function $\ell(M)$. The function $D_{ld}(M, M_t) + \eta_t \ell(M)$ reaches its minimum when its gradient is zero. Thus, we get the following equation by setting the gradient of Eqn. 7 to be zero with respect to M:

$$M_{t+1} = \left(M_t^{-1} + \eta_t \left(PP^T - QQ^T\right)\right)^{-1} \tag{8}$$

where $P_t = X_t - Y_t$ and $Q_t = X_t - Z_t$. To avoid expensive computation of matrix inverse, we apply the Woodbury matrix identity to solve Eqn. 8. The standard Woodbury matrix identity is

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1}$$

However, in our updating equation, there are two items which are the outer product of matrices. To solve this problem, we assume that $\gamma_t = \left({M_t}^{-1} + \eta_t P_t P_t^T\right)^{-1}$, and Eqn. 8 is split into two standard Woodbury matrix identity questions,

$$\begin{cases} \gamma_t = \left(M_t^{-1} + \eta_t PP^T\right)^{-1} \\ M_{t+1} = \left(\gamma_t^{-1} - \eta_t QQ^T\right)^{-1} \end{cases}$$

Applying the Woodbury matrix identity, we arrive at an analytical expression for M_{t+1}

$$\begin{cases} \gamma_t = M_t - \eta_t M_t P\left(I + \eta_t P^T M_t P\right)^{-1} P^T M_t \\ M_{t+1} = \gamma_t + \eta_t \gamma_t Q\left(I - \eta_t Q^T \gamma_t Q\right)^{-1} Q^T \gamma_t \end{cases}. \tag{9}$$

In these updating equations, the regularization parameter η_t is used to control the balance of the regularization function and the loss function. On one hand, if we choose a big η_t, the M_{t+1} will mainly be updated to minimize the loss function and satisfy the target relationship among the three instances in the current triplet, which makes the metric learning process unstable. On the other hand, if η_t is too small, the M_{t+1} will have small divergence with the current Mahalanobis matrix M_t and every iteration will have little influence on the updating of the Mahalanobis matrix. Thus, the metric learning process will be very slow and conservative. Therefore, the selection of η_t should consider the trade-off between efficiency and stability. Meanwhile, η_t should also make sure that M_{t+1} is a PSD matrix in each iteration, i.e.

$$\begin{cases} \eta_t \left(PP^T - QQ^T\right) + M_t^{-1} \geq 0 \\ \eta_t \geq 0 \end{cases}.$$

This is a standard linear matrix inequalities (LMIs). Lots of tools can be utilized to solve this LMIs, such as "LMI Solvers" in MATLAB. If the obtained result using "LMI Solvers" is $\bar{\eta}_t$, the $\eta_t \in [0, \bar{\eta}_t]$ can make sure that M_{t+1} is a PSD matrix in each iteration. In this paper, we select $\eta_t = \alpha\bar{\eta}_t$, where α is the learning rate parameter which is chosen between 0 and 1.

Using the simple form of MDDTW measure, the time series with various lengths and phases can be trained uniformly. The advantage of the proposed method is that our metric learning model can obtain the Mahalanobis metric using the updating equations Eqn. 9 efficiently. However, the computational complexity of the metric learning method is about $O\left(Nd^2mn\right)$, where N is the number of triplets. The metric learning process is time consuming. This is a weak point of the proposed method.

V. Experimental results

In this section, several experiments are conducted to evaluate the performance of the proposed algorithm. The benchmark data sets are selected from the UCI machine learning repository[1] and Robert T. Olszewski's homepage[2]. We firstly compare the proposed algorithm with the state-of-the-art methods on these benchmark data sets. After that, the computational efficiency is analyzed. Furthermore, the relationship between the performance and some parameters are analyzed through experiments. All experiments are tested in MATLAB 2012a, and all tests are implemented on a computer with Intel(R) Core(TM) i5-2400, 3.10GHz CPU, 4G RAM, and Windows 7 64-bit operating system. The code of our algorithms can be downloaded from the website[3].

A. Performance comparison with the state-of-the-art methods

In the first experiment, 9 real-world data sets were selected from the UCI machine learning repository and Robert T. Olszewski's homepage. All the data sets are listed in Table II.

The UCI machine learning repository provides 7 data sets, i.e. Japanese vowels (JapaneseVowels) data set, pen-based recognition of handwritten digits (PenDigits) data set, Libras movement (Libras) data set, Australian sign language signs (AUSLAN) data set, character trajectories (CharacterTrajectories) data set, spoken Arabic digit (ArabicDigits) data set and Robot execution failures (RobotEF) data set (including 5 subset). The JapaneseVowels data set was collected from 9 male speakers who uttered two Japanese vowels /ae/ successively. Each utterance by a speaker is processed with 12-degree linear prediction analysis. And an utterance is regarded as a MTS instance with 12 attributes. The length of these MTS ranges from 7 to 29. The total number of the MTS is 640. The PenDigits data set is created by collecting 250 digit samples from 44 writers. The (x, y) coordinate information is extracted for each digit which is written on a pressure sensitive tablet. The length of MTS is 8. And there are 10092 instances in this

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cs.cmu.edu/∼bobski/

[3] http://www.mathworks.com/matlabcentral/fileexchange/47928-ldmlt-multivariate-time-series-classification-zip

data set. The Libras data set contains 15 classes, where each class references to hand movement type in Libras. The libras movement is recorded by video. 45 frames are selected from each video and the (x, y) coordinate information of centroid pixels of the segmented objects is extracted as 2 attributes. The total number of the formed MTS in Libras data set is 360. The AUSLAN data set was collected from a volunteer native Auslan signer over a period of nine weeks. The data set contains 95 signs, 27 samples per sign. In literature [10], [15], two methods only selected the first 25 signs in the experiments. In order to compare with these two methods fairly, we also use the first 25 signs in the experiments. Thus, the total number of signs is 675. Each sign is represented by 22 channels of information. The length of each sample ranges from 45 to 136. The CharacterTrajectories data set consists of 2858 character samples. The data ware captured using a WACOM tablet. Three attributes including (x, y) coordinate information and pen tip force are recorded in the data set. There are 20 classes in the CharacterTrajectories data set and the length of each sample ranges from 109 to 205. The ArabicDigits data set was taken from 44 males and 44 females Arabic native speakers between the ages 18 and 40. Each speaker repeat ten spoken Arabic digit 10 times, and the total number of digit samples is 8800. Each digit sample is captured by 13 frequency cepstral coefficients. And the length of MTS which represents digit sample is $4 \sim 93$. The RobotEF includes 5 subsets, each of them defines a different learning problem. (a) LP1: failures in approach to grasp position; (b) LP2: failures in transfer of a part; (c) LP3: position of part after a transfer failure; (d) LP4: failures in approach to ungrasp position; (e) LP5: failures in motion with part.

The Wafer data set and ECG data set are provided by Robert T. Olszewski's homepage. The Wafer data set collects sequences of measurements recorded by six vacuum-chamber sensors during the manufacture of semiconductor microelectronics. Each wafer has an assigned category of normal or abnormal. Abnormal wafers are representative of a range of problems commonly encountered during semiconductor manufacturing. In this database, there are 327 MTS instances, among which 200 samples are normal and 127 samples are abnormal. The length of MTS sample is between 104 and 198. The ECG data set collected the sequence of measurements recorded by two electrode during a heartbeat. There are two classes in ECG data set: normal and abnormal. All abnormal heartbeats are representative of a cardiac pathology known as supraventricular premature beat. The ECG data set contains 200 MTS samples, among which 133 samples are normal and 67 samples are abnormal. The length of MTS sample is between 39 and 152.

In this experiment, the test index is chosen as the cross-validation error rates. The performance of the proposed method is evaluated according to the classification error rates using the 1-NN classification (1NN_{MDDTW}) and support vector machine (SVM_{MDDTW}) respectively. In the metric learning process, the target margin ρ is set as the difference between the 90^{th} and 10^{th} percentiles of the distribution of Euclidean distances between sample pairs in the training data. For those data sets with large size, including "PenDigits", "CharacterTrajectories", "ArabicDigits" and "Wafer", we randomly choose $10\% \sim 20\%$ triplet constraints for training. In other data sets, we adopt a dynamic triplets building strategy [36] to choose triplet constraints. In 1NN_{MDDTW} method, the distance between each test sample and all training samples are calculated using the MDDTW measure with obtained Mahalanobis function. The label of the nearest training sample is selected as the category of the test sample. In SVM_{MDDTW} method, we use the the MDDTW measure with obtained Mahalanobis function as a kernel function. After kernelization, we train the SVM classifiers and predict the labels of test samples.

TABLE II. MTS DATA SETS USED IN THE EXPERIMENTS.

Name	# Attributes	# Classes	Length	# of Instances
JapaneseVowels	12	9	$7 \sim 29$	640
PenDigits	2	10	8	10992
Libras	2	15	45	360
AUSLAN	22	25	$47 \sim 95$	675
CharacterTrajectories	3	20	$109 \sim 205$	2858
ArabicDigits	13	10	$4 \sim 93$	8800
ECG	2	2	$39 \sim 152$	200
Wafer	6	2	$104 \sim 198$	1194
RobotEF				
LP1	6	4	15	88
LP2	6	5	15	47
LP3	6	4	15	47
LP4	6	3	15	117
LP5	6	5	15	164

The proposed methods are compared with some basic classification methods and the state-of-the-art metric learning algorithms, including Euclidean distance based DTW (EDTW), lower-bounding measure (LBM) [32], 2-dimensional singular value decomposition (2dSVD) [10], locality preserving projections (LPP) [15], temporal discrete SVM (TDVM) [17], discrete SVM (DSVM) [17], SVM with dynamic time warping (SVM_{DTW}) [17] and the 1-nearest neighbour classifier (1NN_{WD}) [17]. For convenience, some parameters are chosen the same as that in [17]. In other words, we apply 5-fold cross-validation on "ECG" and "RobotEF". For other data sets, we use 10-fold cross-validation to evaluate the performance of the proposed method.

Table III presents the experimental results of different methods. We obtain the experimental results of all data set using the 1NN_{MDDTW}, SVM_{MDDTW}, EDTW and LBM. Results of 2dSVD and LPP are reported by literature [10] and [15] respectively, and the results using TDVM, DSVM, SVM_{DTW} and 1NN_{WD} are reported by the work [17]. The performance comparison reveals that the proposed methods achieves the best performance on most data sets except "ECG" and "Wafer". Some interesting conclusions can be reached from the table III. First of all, the proposed method would outperform more if the number of variable becomes larger, especially compared with the EDTW. The reason is obvious for this phenomenon. The function of the obtained Mahalanobis function is to lay stress on important variables while reducing the influence of the useless variables. If the number of the variable is too small, the Mahalanobis matrix will lose its efficiency. Thus, the performance of the proposed method will be approximate to that of EDTW. For example, the experimental results of the proposed method only improved

TABLE III. CROSS-VALIDATION ERROR RATES COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE MTS DATA SETS.

Data set	1NN$_{MDDTW}$	SVM$_{MDDTW}$	EDTW	LBM	2dSVD	LPP	TDVM	DSVM	SVM$_{DTW}$	1NN$_{WD}$
JapaneseVowels	0.014	**0.009**	0.037	0.081	0.046	0.065	0.034	0.064	0.054	0.077
ECG	0.135	0.125	0.175	0.310	0.269	0.290	**0.095**	0.145	0.140	0.100
Wafer	0.012	0.010	0.016	0.064	0.014	**0.007**				
AUSLAN	**0.041**	0.042	0.100	0.237	0.052	0.048				
PenDigits	**0.006**	**0.006**	0.007	0.122			0.037	0.054	0.066	0.055
RobotEF										
LP1	0.091	**0.079**	0.125	0.375			0.148	0.273	0.182	0.182
LP2	**0.277**	0.321	0.298	0.400			0.362	0.426	0.362	0.404
LP3	0.298	**0.255**	0.320	0.362			0.319	0.362	0.342	0.383
LP4	0.077	**0.034**	0.112	0.163			0.145	0.248	0.128	0.137
LP5	**0.275**	0.287	0.293	0.348			0.329	0.433	0.379	0.348
Libras	**0.092**	0.117	0.095	0.189						
ArabicDigits	0.031	**0.004**	0.037	0.237						
CharacterTrajectories	0.039	**0.010**	0.044	0.143						

a little when compared with EDTW on data sets "ECG", "Libras" and "PenDigits", but there is a great improvement when testing the rest data sets. The second conclusion is that the proposed framework is not good at dealing with the problem of distinguishing normal and abnormal data. Neither 1NN$_{MDDTW}$ nor SVM$_{MDDTW}$ can achieve the best performance among all methods on "ECG" and "Wafer" data sets. In general, normal data are concentrated while abnormal data are far from each other. Therefore, we can't satisfy Eqn. 4 when dealing with abnormal data. The proposed metric learning algorithm can't obtain the best Mahalanobis distance function. The third conclusion is that SVM$_{MDDTW}$ can achieve better performance than 1NN$_{MDDTW}$ on most data. If 1NN$_{MDDTW}$ can get good classification results, it requires that the samples in the same categories has the shortest Mahalanobis distance. However, SVM$_{MDDTW}$ only requires that the samples with same categories has the similar distribution when comparing with all training samples. The SVM$_{MDDTW}$ has less requirements than 1NN$_{MDDTW}$, so the performance is better in most situations.

B. Computational efficiency analysis

In this part, we'd like to analyze the computational efficiency of the proposed method. Table IV presents the processing times at different stages in 1 cross of the above cross-validation experiments. t_{metric} represents the running time at metric learning stage while t_{1NN} and t_{SVM} stands for the running time at classification stage.

There are several factors which affect the execution efficiency of the proposed method. First of all, when considering the metric learning process, the size of training data plays important roles. As is mentioned above, the time complexity of the metric learning method is about $O\left(Nd^2mn\right)$. Thus, the number of instances, length and number of attributes are three important factors. Meanwhile, the intrinsic structure of training data affect the convergence rate, which also has influence on the number of triplets N. In the triplets building process, our work has two strategies. If using the random triplets building method, the time can be neglected. However, if applying the dynamic triplets building strategy [36], the complexity of dynamic triplets building is about $O\left(L_{train}^2mn\right)$. L_{train} is the number of training instances. Sometimes, the triplets building strategy would occupy much more time than metric learning itself. Then, in the classification process, the time complexity of 1NN classifier is about $O\left(L_{train}L_{test}mn\right)$, where L_{test} is the number of testing instances. When it comes to SVM classifier, it contains the calculating Mahalanobis kernel of training and testing data $O\left(L_{train}^2mn\right) + O\left(L_{train}L_{test}mn\right)$ and training linear SVM classifier $O\left(cL_{train}^2\right)$ (c is the number of categories).

TABLE IV. THE PROCESSING TIMES IN DIFFERENT STAGES.

Data set	L_{train}	L_{test}	t_{metric}	t_{1NN}	t_{SVM}
JapaneseVowels	576	64	120.3s	3.7s	36.4s
ECG	160	40	80.2s	8.4s	43.0s
Wafer	1074	120	120.9s	15.1s	382.5s
AUSLAN	607	68	25.8s	40.5s	1605.3s
PenDigits	9892	1100	226.5s	29.2s	343.2s
RobotEF					
LP1	70	18	1.7s	0.2s	1.1s
LP2	38	9	1.1s	0.1s	0.4s
LP3	38	9	1.7s	0.1s	0.4s
LP4	93	24	2.7s	0.4s	2.0s
LP5	131	33	4.6s	0.7s	3.8s
Libras	324	36	132.4s	6.5s	65.0s
ArabicDigits	7920	880	300.0s	144.0s	1504.1s
CharacterTrajectories	2572	286	766.7s	3561.2s	4638.5s

In the whole algorithm, the memory complexity is not a main issue. The online algorithm can deal with received triplets one by one. The variables which occupy the most storage space are the Mahalanobis kernels of training and testing data, which are about $O\left(L_{train}^2\right) + O\left(L_{train}L_{test}\right)$.

C. Influence of parameters on performance

In the proposed algorithm, several parameters may have influence on the MTS classification performance. The first one is the learning rate parameter α. As mentioned above, the α will determine the learning rate of the metric learning process. On one hand, if α is too large, each triplet constraint will have significant influence on the updating of Mahalanobis matrix. And the Mahalanobis matrix would not be stable in the learning process. Thus the obtained Mahalanobis matrix would have a low classification accuracy. On the other hand, if the learning rate α is very small, each iteration will have too little influence on the changing of the Mahalanobis matrix, and the metric learning process will be very slow and insufficient. Then the obtained Mahalanobis matrix will not have a good performance because it approximates to the Eudiance distance. The Fig. 4 illustrates the relationship between the classification

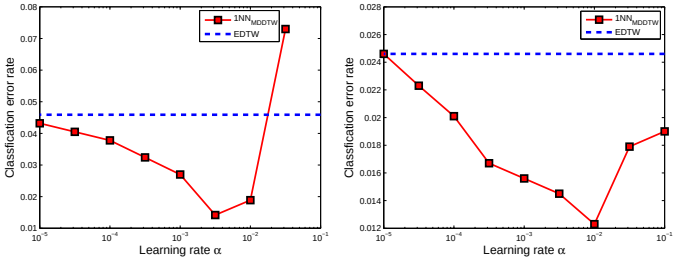


Fig. 4. The relationship between classification error rate and the parameter learning rate α. (a) The experiment results based on data set "JapaneseVowels"; (b) The experiment results based on data set "Wafer".

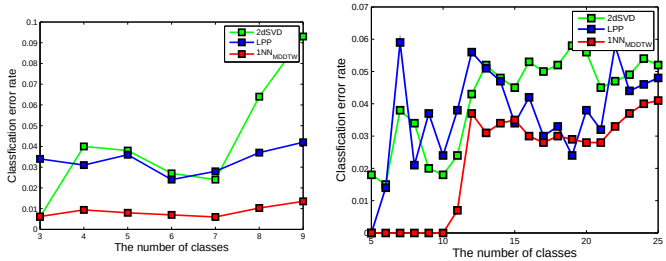


Fig. 5. The relationship between classification error rate and the number of classes. (a) The experimental results based on data set "JapaneseVowels"; (b) The experimental results based on data set "AUSLAN".

error rate and the parameter learning rate α on data set "JapaneseVowels" and "Wafer". From Fig. 4 we can see, when α is very small, the classification error rate of the proposed method is smaller than but similar to that of EDTW. When the α approximates to 1, the classification error rate will also become large, even worse than that of EDTW. The best α is determined by several factors, including the total quantity of triplet constraints N and the target margin ρ. Firstly, α and N has an inverse relationship. When the number of triplet constraints becomes larger, the α should be smaller to avoid overfitting. Secondly, approximate inversely linear dependency exists between α and ρ. If ρ is larger, more triplets can't meat the relationship in Eqn. 4. We should decrease α to avoid overfitting the higher margin ρ. Thus, in our work, an optional rule for α is recommended as $\frac{c}{N(1+\rho^{\varsigma})}$. In our experiment, we choose $\mathrm{c} = 10 \sim 50$ and $\varsigma = 0.2 \sim 0.5$.

Another factor which may affect the MTS classification performance of the proposed algorithm is the number of classes in data set. In this experiment, we increase the number of classes in the data set gradually to test the cross-validation error rates on data set "JapaneseVowels" and "AUSLAN". The main purpose is to test the relationship between the classification error rate and the number of classes, which can reveal the robustness of the algorithm to a certain degree. Theoretically, the classification error rate will increase when the number of classes becomes large. If the growth rate of the classification error rate is very low, it indicates that the performance of the algorithm is robust to the number of classes. We compare the performance of the proposed method with 2dSVD and LPP in this experiment. The experimental results of 2dSVD and LPP are reported in the literature [15]. From Fig. 5 we can see, the classification error rate of the proposed method is lower than 2dSVD and LPP in most situations. At the same time, the growth rate of classification error rate of the proposed method is also very low in Fig. 6(a). However, in Fig. 6(b), classification error rate has a jump in the proposed method. In fact, three methods all have a jump when adding the 12^{th} class, indicating that the 12^{th} class is easy to be confused with other categories. The rest stable curve of the proposed method in Fig. 6(b) demonstrates the it is more robust than 2dSVD and LPP.

In the following experiment, we also explore the relationship between the accuracy and size of the training data set. Theoretically, the size of the training data set may affect the classification accuracy from two aspects. On one hand, the number of training samples might have influence on the metric learning process. And the performance of the obtained Mahalanobis distance function then affect the final classification results. On the other hand, the classification performance of the basic classifiers, including 1NN classifier and SVM classifier also rely on the size of the training data set. Thus, we conduct 3 comparative experiments to illustrate the influence of the size of the training data set. In these experiments, we use 10-fold cross-validation to evaluate the performance of MTS classification. In the first experiment, we randomly choose 10%$\sim$100% training samples in metric learning process, and these chosen samples are also used for the following classifying process. The results are shown in Fig. 6. We can see that more training samples can improve the classification accuracy greatly. However, this experiment can't distinguish which factor plays a dominant role. The experimental condition of the second and third experiment is similar to the first experiment, the difference is that the second experiment use all training samples for metric learning while the third experiment use all training samples for classifying process. We can see that the size of the training data set has less influence on metric learning process than the classifying process. We can use a small number of training samples to train a good Mahalanobis distance function with high performance.

VI. Conclusion

In this work, we consider the problem of measuring and classifying MTS, which is one of the bases for various computer vision and pattern recognition applications. A novel measure for MTS is described. In the proposed method, the Mahalanobis distance is firstly used for measuring the local distance of vectors in MTS. Then the DTW is utilized to find the most optimized path to align MTS which are out of sync or with different length. After that, the difference between two MTS can be obtained for MTS classification and clustering. Another key problem in the proposed MDDTW measure is to learn the Mahalanobis function for MTS dataset. This work built a LDMLT model for metric learning in MTS case. In the experiments, we conducted our algorithm on several well known data sets. The results demonstrated the robustness and

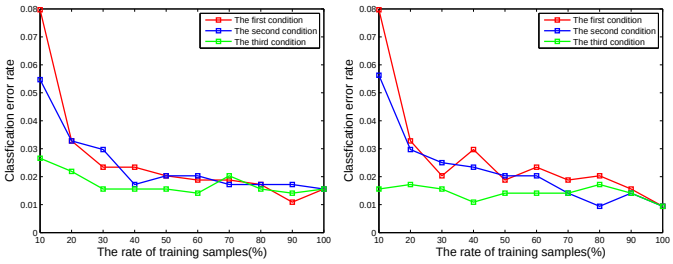


Fig. 6. The relationship between classification error rate and the size of training set on data set "JapaneseVowels". The first condition means that only a part of samples used for metric learning process and classifying process. The second condition represents that all training samples are used for metric learning and part of them are utilized in classifying process. In the third condition, all training samples are used for classifying process while part samples are used for metric learning. (a) The experimental results using 1NN classifier; (b) The experimental results using 1NN classifier.

high precision of the proposed approach. One drawback is that the proposed framework has a low computational efficiency. The MDDTW measure has a high computational cost. Thus, further research should be carried out on the computational optimization of the proposed method.

REFERENCES

[1] T. Warren Liao, "Clustering of time series data-a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[2] E. Ramasso, M. Rombaut, and N. Zerhouni, "Joint prediction of continuous and discrete states in time-series based on belief functions," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 37–50, 2013.

[3] C. Gruber, T. Gruber, S. Krinninger, and B. Sick, "Online signature verification with support vector machines based on lcss kernel functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1088–1100, 2010.

[4] W.-M. Li, F. Li, J.-W. Liu, and J.-J. Le, "Similarity search over data stream using lpc-dtw," in *International Conference on Machine Learning and Cybernetics*, vol. 3. IEEE, 2007, pp. 1631–1634.

[5] M. Lewandowski, D. Makris, S. A. Velastin, and J.-C. Nebel, "Structural laplacian eigenmaps for modeling sets of multivariate sequences," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 936 – 949, 2013.

[6] Z. Prekopcsák and D. Lemire, "Time series classification by class-specific mahalanobis distance measures," *Advances in Data Analysis and Classification*, vol. 6, no. 3, pp. 185–200, 2012.

[7] T. Górecki and M. Łuczak, "Using derivatives in time series classification," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 310–331, 2013.

[8] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.

[9] E. S. García-Treviño and J. A. Barria, "Structural generative descriptions for time series classification," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1978–1991, 2014.

[10] X. Weng and J. Shen, "Classification of multivariate time series using two-dimensional singular value decomposition," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 535–539, 2008.

[11] K. Yang and C. Shahabi, "A pca-based similarity measure for multivariate time series," in *Proceedings of the 2nd ACM international workshop on Multimedia databases*. ACM, 2004, pp. 65–74.

[12] C. S. Möller-Levet, F. Klawonn, K.-H. Cho, and O. Wolkenhauer, "Fuzzy clustering of short time-series and unevenly distributed sampling points," in *Advances in Intelligent Data Analysis V*. Springer, 2003, pp. 330–340.

[13] M. Kumar, N. R. Patel, and J. Woo, "Clustering seasonality patterns in the presence of errors," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 557–563.

[14] R. Dahlhaus, "On the kullback-leibler information divergence of locally stationary processes," *Stochastic Processes and their Applications*, vol. 62, no. 1, pp. 139–168, 1996.

[15] X. Weng and J. Shen, "Classification of multivariate time series using locality preserving projections," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 581–587, 2008.

[16] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[17] C. Orsenigo and C. Vercellis, "Combining discrete svm and fixed cardinality warping distances for multivariate time series classification," *Pattern Recognition*, vol. 43, no. 11, pp. 3787–3794, 2010.

[18] H.-S. Lim, K.-Y. Whang, and Y.-S. Moon, "Similar sequence matching supporting variable-length and variable-tolerance continuous queries on time-series data stream," *Information Sciences*, vol. 178, no. 6, pp. 1461–1478, 2008.

[19] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[20] G. Ten Holt, M. Reinders, and E. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Thirteenth annual conference of the Advanced School for Computing and Imaging*, vol. 300, 2007.

[21] J. Mei, M. Liu, H. R. Karimi, and H. Gao, "Logdet divergence based metric learning using triplet labels," *ICML Workshop on Divergences and Divergence Learning*, 2013.

[22] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems*. Citeseer, 2006.

[23] C. Shen, J. Kim, L. Wang, and A. Van Den Hengel, "Positive semidefinite metric learning with boosting." in *Advances in Neural Information Processing Systems*, vol. 22, 2009, pp. 629–633.

[24] J. Bi, D. Wu, L. Lu, M. Liu, Y. Tao, and M. Wolf, "Adaboost on low-rank psd matrices for metric learning," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 2617–2624.

[25] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 209–216.

[26] M. Müller, "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

[27] M. Liu and B. C. Vemuri, "A robust and efficient doubly regularized metric learning approach," in *Proceedings of the 12th European conference on Computer Vision*. Springer-Verlag, 2012, pp. 646–659.

[28] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[29] S. Sun and Q. Chen, "Hierarchical distance metric learning for large margin nearest neighbor classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 07, pp. 1073–1087, 2011.

[30] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," *Advances in Neural Information Processing Systems*, vol. 15, pp. 505–512, 2002.

[31] S. Sun and Q. Chen, "Kernel regression with a mahalanobis metric for

short-term traffic flow forecasting," in *Intelligent Data Engineering and Automated Learning–IDEAL 2008*. Springer, 2008, pp. 9–16.

[32] T. M. Rath and R. Manmatha, "Lower-bounding of dynamic time warping distances for multivariate time series," *University of Massachusetts Amherst Technical Report MM*, vol. 40, 2002.

[33] G. Al-Naymat, S. Chawla, and J. Taheri, "Sparsedtw: a novel approach to speed up dynamic time warping," in *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101*. Australian Computer Society, Inc., 2009, pp. 117–127.

[34] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *The Journal of Machine Learning Research*, vol. 8, pp. 1027–1061, 2007.

[35] J. Blitzer, K. Q. Weinberger, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in neural information processing systems*, 2005, pp. 1473–1480.

[36] J. Mei, M. Liu, H. R. Karimi, and H. Gao, "Logdet divergence based metric learning with triplet constraints and its applications," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4920 – 4931, 2014.

[37] Q. Wang, P. C. Yuen, and G. Feng, "Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions," *Pattern Recognition*, 2013.

[38] P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman, "Online metric learning and fast similarity search," in *Advances in Neural Information Processing Systems*, 2008, pp. 761–768.

[39] I. S. Dhillon and J. A. Tropp, "Matrix nearness problems with bregman divergences," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 4, pp. 1120–1146, 2007.

[40] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 505–512.

[41] J. Dhillon, "Differential entropic clustering of multivariate gaussians," *Advances in Neural Information Processing Systems*, vol. 19, pp. 337–344, 2007.