Neural Process Lett (2017) 45:925–937
DOI 10.1007/s11063-016-9555-5

A Novel Similarity Measure Model for Multivariate Time Series Based on LMNN and DTW

Jingyi Shen[1] · **Weiping Huang**[1] · **Dongyang Zhu**[1] · **Jun Liang**[1]

Published online: 26 September 2016

Abstract In this paper, a novel model is proposed to measure the similarity of multivariate time series by combining large margin nearest neighbor (LMNN) and dynamic time warping (DTW). Firstly we use a Mahalanobis distance-based DTW measure for multivariable time series, which considers the relations among variables through the Mahalanobis matrix. Secondly, the LMNN algorithm is applied to learn the Mahalanobis matrix by minimizing a renewed cost function. As the cost function is non-differentiable, the minimization problem is solved from a perspective of k-means by coordinate descent method. We empirically compare the proposed model with other techniques and demonstrate its convergence and superiority in similarity measure for multivariate time series.

Keywords Multivariate time series · Similarity measure · Large margin near neighbor · Dynamic time warping

1 Introduction

Time series is a widely existing data form which relates to the time feature greatly. It exists in various fields, such as finance [1], medical science [2], process engineering [3], and bioinformatics [4]. The knowledge discovery and data mining tasks for time series have attracted a large amount of attention and research efforts in recent years.

Most data mining applications on time series, from simple clustering and classification tasks to complex decision-making systems, are highly dependent on similarity measure, such as k-nearest neighbors [5], k-means [6], support vector machine (SVM) [7]. There are a lot of algorithms that could be used to measure the similarity between two time series. Euclidean distance (ED) is the most widely used method because of its simple calculation. However, the ED measure can only deal with the kind of time series with fixed length. Dynamic time

✉ Jun Liang
jliang@iipc.zju.edu.cn

[1] Zhejiang University, Hangzhou, China

warping (DTW), which was firstly introduced in the field of speech recognition [8], is able to measure similarity between time series with different lengths. Moreover, DTW can also deal with the offset or scaling problems in time axis by allowing for time warping. The flexibility and effectiveness makes DTW widely used in domains outside speech recognition [4,9], such as gesture recognition [10], robotics [11], data mining [12], etc.

Most recent studies on similarity measure methods like DTW are aiming at univariate time series, which only represent one feature varying with time. However, multivariate time series is much more wildly existing than univariate time series in practical applications, for which the property of a system is usually described by a number of variables simultaneously rather than single one variable. The multivariate time series can be seen as a group of same-length univariate time series. Each univariate time series represents the varying of a feature. A modified DTW algorithm for multivariate time series similarity measure was proposed in [13], we call it EDDTW here for convenience. EDDTW algorithm extends the DTW algorithm from univariate time series to multivariate time series, by computing the local distance at each time point using Euclidean distance. [14] proposed a model for multivariate time series classification combining discrete SVM and EDDTW. However, the Euclidean distance puts the same weight on each variable, ignoring the correlations among variables. [15] proposed a Mahalanobis distance based DTW measure for multivariate time series, called MDDTW. Differently to EDDTW, MDDTW computes the local distance by Mahalanobis distance, which considers the relations among variables. Now the problem is changing to how to determine the Mahalanobis matrix of Mahalanobis distance.

In recent years, metric learning has becoming a more and more popular study on similarity measure for multivariate applications [16]. It learns a distance metric from the training dataset, which can represent more advisable relations among variables. Most metric learning algorithms are based on Mahalanobis distance, by learning a symmetric positive semi-definite (PSD) matrix under a certain cost function. The PSD matrix is called Mahalanobis matrix. A lot of metric learning algorithms have been proposed, such as probabilistic global distance metric learning [17], relevant components analysis (RCA) [18], neighborhood components analysis (NCA) [19], large margin nearest neighbor (LMNN) [20], and a LogDet divergence-based metric learning [21]. LMNN is one of the most famous metric learning algorithms, which applies the idea of large margin similarly to SVM [22].

However, as all the metric learning algorithms are proposed for similarity measure of non-time series data, we can't just apply an existing metric learning algorithm such as LMNN to multivariate time series directly. [15] established a LogDet divergence-based metric learning (LDML) with triplet constraint model to learn the Mahalanobis matrix of MDDTW, we call the approach LDML–DTW in this paper for convenience. [15] showed improved performances of the proposed approach through several experiments. However, the proposed LDML algorithm is much more complex to comprehend and solve than LMNN, which has a straightforward meaning.

In this paper, we propose a similarity measure model for multivariate time series based on LMNN metric learning and DTW, called LMNN–DTW. Firstly we use the MDDTW to measure the similarity between multivariate time series by applying a Mahalanobis-based local distance. Secondly, we apply the LMNN algorithm to learn an advisable Mahalanobis matrix under a specific cost function. To demonstrate the performance of the proposed measure model, we compared it with LDML–DTW method [15], TDVM [14], statistic Mahalanobis distance based DTW (SMDDTW), Euclidean distance based DTW (EDDTW), Euclidean distance (ED) by classification experiments on seven multivariate time series datasets.

The rest of the paper is organized as follows. Section 2 reviews related work on DTW, metric learning and LMNN. Section 3 presents the proposed method LMNN–DTW. In Sect. 4,

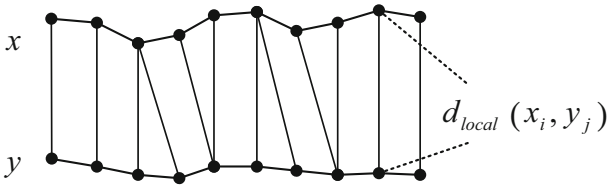

Fig. 1 An example of DTW for univariate time series

we perform experiments to evaluate the performance of the proposed method. The conclusion is provided in Sect. 5.

2 Related Work

2.1 Dynamic Time Warping

Dynamic time warping (DTW) computes the best possible mapping between two time series by allowing for time warping. Firstly we introduce the original DTW for univariate time series. Suppose we have two univariate time series, $x(i), i = 1, 2, \ldots, m$ and $y(j), j = 1, 2, \ldots, n$. An m-by-n matrix D is computed with the (i, j) th element being $d_{local}(i, j) = (x(i) - y(j))^2$. We call $d_{local}(i, j)$ the local distance between elements $x(i)$ and $y(j)$. Figure 1 shows an example of DTW.

The warping path W is an alignment between x and y, which could be expressed as

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

where $w_x(k)$ and $w_y(k)$ represent the indexes in time series x and $y$respectively, p is the length of the warping path W. $\left(w_x(k), w_y(k)\right)'$ indicates that the $w_x(k)$ th element in time series x is mapping to the $w_y(k)$ th element in time series y. The warping path must be subject to several constraints as follows:

Boundary Condition: The warping path should start at $W(1) = (1, 1)'$ and end up at $W(p) = (m, n)'$.

Continuity: The adjacent elements of path W, $W(k)$ and $W(k + 1)$ must be subject to $w_x(k + 1) - w_x(k) \leq 1$ and $w_y(k + 1) - w_y(k) \leq 1$.

Monotonicity: The adjacent elements of path W, $W(k)$ and $W(k + 1)$ must be subject to $w_x(k + 1) - w_x(k) \geq 0$ and $w_y(k + 1) - w_y(k) \geq 0$.

Obviously there are many warping paths satisfying the above constraints. We use $DTW(x, y)$ to represents the minimal distance between time series x and y, which corresponds to the optimal warping path. The minimal distance and optimal warping path could be found via a dynamic programming algorithm:

$$\begin{cases} r(i, j) = d(i, j) + \min\{r(i-1, j-1), r(i-1, j), r(i, j-1)\} \\ \text{DTW}(x, y) = \min\{r(m, n)\} \end{cases}. \tag{2}$$

$r(i, j)$ represents the minimal cumulative distance from $(0, 0)$ to (i, j) in matrix D. Figure 2 shows an example of the warping path.

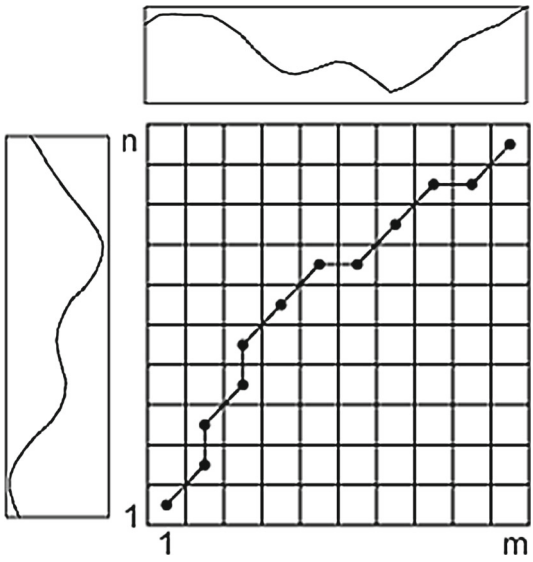

Fig. 2 The warping of dynamic time warping

Now suppose we have acquired the optimal warping path W by solving the above dynamic programming problem, then the given time series x and y can be extended to another two time series $\bar{x}$ and $\bar{y}$:

$$\begin{cases} \bar{x}(k) = x(w_x(k)) \\ \bar{y}(k) = y(w_y(k)) \end{cases} k = 1, 2, \ldots, p. \tag{3}$$

The DTW distance between x and y could be represented by the Euclidean distance between the two new time series $\bar{x}$ and $\bar{y}$:

$$\mathrm{DTW}(x, y) = \sum_{k=1}^{p} (\bar{x}(k) - \bar{y}(k))^2. \tag{4}$$

2.2 Metric Learning and Large Margin Nearest Neighbor

Metric learning is to learn a distance metric which can represent the distance relations among the training data. Most metric learning algorithms are based on Mahalanobis distance. Given two vectors of multivariable instances (non-time series) x_i and x_j, the Mahalanobis distance between x_i and x_j parametrized by a symmetric PSD matrix is defined as

$$D_M(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j). \tag{5}$$

The PSD matrix M is named as Mahalanobis matrix, which can be decomposed into $M = L^T L$. Then, the Mahalanobis distance can be rewritten as

$$\begin{aligned} D_{\mathrm{M}}(x_i, x_j) &= (x_i - x_j)^T L^T L (x_i - x_j) \\ &= (Lx_i - Lx_j)^T (Lx_i - Lx_j). \end{aligned} \tag{6}$$

Thus, the calculation of Mahalanobis distance can be considered as two steps: (1) a linear transformation from the original instances space to a new instances space by the matrix L; (2) calculation of Euclidean distance between two new instances.

The Mahalanobis distance based metric learning algorithm learns the Mahalanobis matrix through optimizing a certain cost function. In recent years, many metric learning algorithms with different cost functions have been proposed, e.g., RPDM, RCA, NCA, LMNN.

Large margin nearest neighbor (LMNN) [20] is one of the most popular metric learning methods, which learns a Mahalanobis matrix basing on the concept of large margin. The cost function of LMNN is based on two intuitions: (1) bring instances closer to their target neighbors; (2) separate instances with different labels wildly. Here the target neighbors of an instance are identified as its k nearest neighbors. Specifically, let $\{(x_i, y_i)\}_{i=1}^{n}$ denotes a training set with multivariate instances $x_i \in R^d$ and class labels y_i, the cost function is given by:

$$\varepsilon(M) = (1-c)\sum_{ij}\eta_{ij} D_M\left(x_i, x_j\right) + c\sum_{ijl}\eta_{ij}\left(1-y_{il}\right)\left[1 + D_M\left(x_i, x_j\right) - D_M\left(x_i, x_l\right)\right]_+, \tag{7}$$

where $[z]_+ = \max(z, 0)$,$c \in [0, 1]$ is the weighting parameter to balance two penalizing terms, $y_{ij} \in \{0, 1\}$ indicates whether instances x_i and x_j share the same label or not, $\eta_{ij} \in \{0, 1\}$ indicates whether instance x_j is a target neighbor of instance x_i.

The first term in the cost function penalizes large distances between each instance and its target neighbors. The second term indicates the idea of a margin, which penalizes small distances among instances with different labels. The optimization of (7) can be cast as an instance of convex programming [23],which can be solved efficiently on modern computers.

However, all of the existing metric learning methods are proposed for non-time series applications so that cannot be used to deal with multivariate time series applications directly. In this paper, we integrate LMNN and DTW to propose a similarity measure model aiming at multivariate time series. We call the model LMNN–DTW.

3 LMNN–DTW

In this section, we present the LMNN and DTW based similarity measure model, called LMNN–DTW, for multivariate time series. Firstly, we use the MDDTW measure, which was firstly proposed in [15], to calculate similarity between multivariate time series. The main difference between DTW measure for univariate time series and multivariate time series lies in the calculation of local distance. Fig 3a shows an example of DTW measure for multivariate time series. By comparing Fig 3a with Fig 1, we can learn the difference more intuitively. In the DTW measure for univariate time series, the local distance represents the distance between two scalars, mostly ED is used. For multivariate time series, however, the local distance is the distance between two vectors. Given two multivariate time series X and Y

$$X = \begin{bmatrix} x_{11}x_{12}\ldots x_{1m} \\ x_{21}x_{22}\ldots x_{2m} \\ \vdots\ \vdots\ \ddots\ \vdots \\ x_{d1}x_{d2}\ldots x_{dm} \end{bmatrix} \text{ and } Y = \begin{bmatrix} y_{11}y_{12}\ldots y_{1n} \\ y_{21}y_{22}\ldots y_{2n} \\ \vdots\ \vdots\ \ddots\ \vdots \\ y_{d1}y_{d2}\ldots y_{dn} \end{bmatrix},$$

where d is the number of variables, m and n are the lengths of X and Y respectively. We define $X^i = (x_{1i}, x_{2i}, \ldots, x_{di})'$ to represent the i th column in X and $Y^j = \left(y_{1j}, y_{2j}, \ldots, y_{dj}\right)'$ to represent the jth column in Y. The local distance $d_{local}(i, j)$ here is defined as the distance between X^i and Y^j. For example, the ED-based local distance, firstly proposed in [13], is defined as

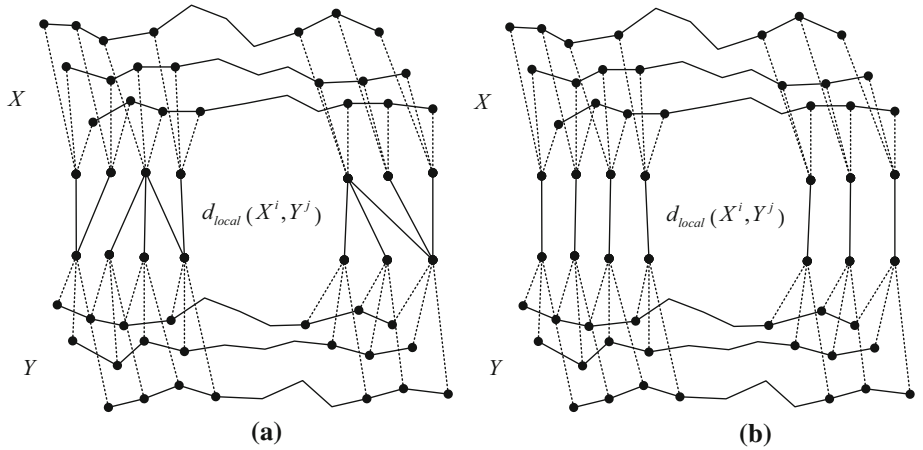

Fig. 3 Examples of DTW and ED for multivariate time series. **a** MDDTW measure **b** ED measure

$$d_{ED}\left(X^i, Y^j\right) = \left(X^i - Y^j\right)^T \left(X^i - Y^j\right) = \sum_{k=1}^{d} \left(x_{ki} - y_{kj}\right)^2, 1 \le i \le m, 1 \le j \le n. \tag{8}$$

However, the Euclidean distance measure assigns the same weight to all variables, which is not proper in most applications. In this paper, we use Mahanalobis distance to calculate the local distance, which is defined as

$$d_M\left(X^i, Y^j\right) = \left(X^i - Y^j\right)^T M \left(X^i - Y^j\right), 1 \le i \le m, 1 \le j \le n, \tag{9}$$

where M is a symmetric PSD matrix, called Mahanalobis matrix. Then the DTW distance and optimal warping path between X and Y can be achieved by a dynamic programming algorithm

$$\begin{cases} r(i, j) = d_M\left(X^i, Y^j\right) + \min\left\{r(i-1, j-1), r(i-1, j), r(i, j-1)\right\} \\ \mathrm{DTW}_M(X, Y) = \min\left\{r(m, n)\right\} \end{cases}. \tag{10}$$

Similarly to the DTW method for univariate time series, suppose we have got the optimal warping path W by solving the above dynamic programming problem, two new multivariate time series $\bar{X}_{d\times p}$ and $\bar{Y}_{d\times p}$ could be acquired as

$$\begin{cases} \bar{X}^k = X^{(w_x(k))} \\ \bar{Y}^k = Y^{(w_y(k))} \end{cases}, k = 1, 2, \ldots, p. \tag{11}$$

Subsequently the DTW distance $\mathrm{DTW}_M(X, Y)$ can be rewritten as

$$\begin{aligned} \mathrm{DTW}_M(X, Y) &= \sum_{k=1}^{p} d_M\left(\bar{X}^k, \bar{Y}^k\right) \\ &= \sum_{k=1}^{p} \left(\bar{X}^k - \bar{Y}^k\right)^T M \left(\bar{X}^k - \bar{Y}^k\right). \end{aligned} \tag{12}$$

After the construction of the Mahanalobis distance-based DTW measure for multivariate time series, we apply LMNN to learn the Mahanalobis matrix M. Like function (7), we define the

cost function as

$$\varepsilon(M) = (1-c)\sum_{ij}\eta_{ij}DTW_M\left(X_i, X_j\right) + c\sum_{ijl}\eta_{ij}\left(1-y_{il}\right)\left[1+\mathrm{DTW}_M\left(X_i, X_j\right)-\mathrm{DTW}_M\left(X_i, X_l\right)\right]_+ . \quad (13)$$

By introducing (12) into (13), we can rewrite the cost function as

$$\varepsilon(M) = (1-c)\sum_{ij}\eta_{ij}\sum_{k=1}^{p}\left(\bar{X}_i^k-\bar{Y}_j^k\right)^T M\left(\bar{X}_i^k-\bar{Y}_j^k\right) + c\sum_{ijl}\eta_{ij}\left(1-y_{il}\right)\left[1+\sum_{k=1}^{p}\left(\bar{X}_i^k-\bar{Y}_j^k\right)^T M\left(\bar{X}_i^k-\bar{Y}_j^k\right) - \sum_{k=1}^{p}\left(\bar{X}_i^k-\bar{Y}_j^k\right)^T M\left(\bar{X}_i^k-\bar{Y}_l^k\right)\right]_+ . \quad (14)$$

The main difference between (7) and (13) lies in the distance measure between two instances. Here in (13), the distance measure is based on the MDDTW method. The computation of MDDTW measure is a dynamic programming procedure, which makes the cost function non-differentiable. Consequently the minimization of cost function (13) cannot be solved by gradient-based methods like Newton's method [24] or conjugate gradient [25]. However, we can review this minimization problem from a perspective of K-means algorithm as a two-step procedure: firstly, we compute the best warping path for each pair of multivariate time series by dynamic programming, keeping M fixed; secondly, we minimize cost function (14) with respect to M by a gradient method such as Newton's method, keeping the warping paths fixed. Specifically, the algorithm can be expressed as follows:

1. Initialize Mahalanobis matrix M as diagonal, with values following normal distribution $N(0, 1)$.
2. Keeping M fixed, find the best warping path for each pair of multivariable time series by (10), and generate new multivariable time series by (11).
3. Keeping the warping paths fixed, minimize cost function (14) with respect to M by Newton's method.
4. Repeat step (2) and (3) until convergence.

The optimization in step (2) is easy to achieve by a dynamic programming procedure. In step (3), the optimization problem is same as the problem in original LMNN algorithm with cost function (7). We use Limited-memory BFGS (L-BFGS) [26,27], an optimization algorithm in the family of quasi-Newton methods, to solve the optimization. We set two stop criterions for both step (3) and the whole iterative algorithm: convergence or meeting the maximum iterations.

The algorithm described above is exactly the coordinate descent method [28] for optimization. The algorithm is guaranteed to converge theoretically. In simple terms, both steps (2) and (3) minimize the cost function repeatedly, making the cost function decrease monotonically and converge finally. We will demonstrate the convergence of the algorithm through experiment in next section but not in theory. However, the algorithm cannot be guaranteed to converge to the global minimum because the cost function is non-convex. In other words, the cost function optimization by coordinate descent can be converge to local optima. Actually,

the algorithm works fine and comes up with very good results in most applications. To alleviate the local optima problem, we initialize Mahalanobis matrix M as diagonal, with values following normal distribution N (0, 1).

The time complexity of LMNN–DTW is mainly depend on the iterations of the optimizing procedure, the time consumption of DTW, and the number of instances. The iterations of the optimizing procedure is mainly depend on the number of variables. The time complexity is O (mn), where m and n are the lengths of two series. From this we can approximately acquire the time complexity of LMNN–DTW as O $(Ndmn)$, where N is the number of instances, d is the number of variables.

4 Experiment

To evaluate the performance of the proposed method, several experiments are conducted in this section. We choose seven multivariate time series datasets from the University of California Irvine (UCI) machine learning repository[1], i.e., Japanese vowels dataset (JapaneseVowels), Libras movement dataset (Libras), spoken Arabic digit dataset (ArabicDigits), activities of daily living recognition dataset (ADLs), pen-based recognition of handwritten digits dataset (PenDigits), LP4 and LP5 included in Robot execution dataset (RobotEF). The JapaneseVowels dataset was collected from nine male speakers who uttered two Japanese vowels /ae/ successively. The Libras dataset is composed of frames information from videos which record the hand movement types in official Brazilian signal language. The ArabicDigits dataset contains time series of mel-frequency cepstrum coefficients (MFCCs) corresponding to spoken Arabic digits from 44 male and 44 female native Arabic speakers. The ADLs dataset comprises information regarding the activities of daily living performed by two users on a daily basis in their own homes. PenDigits dataset is created by collecting 250 digit samples from 44 writers. The (x, y) coordinate information is extracted for each digit. The RobotEF contains force and torque measurements on a robot after failure detection, including five kinds of failures. We only choose two of them for experiments according to the numbers of instances and classes, i.e. failures in approach to ungrasp position named LP4 and failures in motion with part named LP5. Information of the datasets is presented in Table 1.

The proposed method is compared with some basic distance measures including statistic Mahalanobis distance based DTW (SMDDTW), Euclidean distance based DTW (EDDTW) [13], Euclidean distance (ED), and the state-of-art ones, including LDML based DTW (LDML–DTW) [15] and TDVM [14]. Specifically, LMNN–DTW and LDML–DTW learn the Mahalanobis matrix from the training datasets by a certain metric learning algorithm while SMDDTW sets the Mahalanobis matrix as the inverse of sample covariance matrix [29] without a learning procedure. EDDTW computes the local distance using Euclidean distance as (8). ED can only deal with the multivariate time series with fixed length because it does not allow time warping. Fig 3b shows an example of ED measure for multivariate time series as long as setting the local distance as Euclidean distance. As LMNN–DTW, LDML–DTW, SMDDTW, EDDTW and ED are distance measure algorithms, we combine them with 1NN for classification in the experiments. TDVM is a two-stage model combining EDDTW and discrete SVM for multivariate time series classification. In the first stage, TDVM uses EDDTW measure to transform the set of multivariate input sequences into a fixed number of columns. In the second stage, the discrete SVM is used for classification. We use the result in [14] directly here instead of implementing the TDVM by ourselves. The code

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

Table 1 Summary of datasets

Dataset	Variables	Classes	Instances	Length	k	c
JapaneseVowels	12	9	640	7–29	31	0.6
Libras	2	15	360	45	9	0.7
ArabicDigits	13	10	8800	4–93	44	0.5
ADLS	3	10	960	250	31	0.4
PenDigits	2	10	10, 992	8	700	0.7
LP4	6	3	117	15	13	0.6
LP5	6	5	164	15	13	0.7

Table 2 Testing error rates

Dataset	LMNN–DTW	LDML–DTW	SMDDTW	TDVM	EDDTW	ED
JapaneseVowels	1.32	2.10	4.32	3.40	5.14	–
Libras	11.33	9.67	26.67	–	25.33	25.33
ArabicDigits	7.30	5.04	15.56	–	15.33	–
ADLS	8.20	12.33	21.56	–	16.39	–
PenDigits	1.08	1.28	4.34	3.70	4.12	5.28
LP4	11.93	9.08	33.33	14.50	28.21	33.33
LP5	23.81	27.06	47.37	32.90	30.32	37.58

of LDML–DTW algorithm was downloaded from the website[2], which was provided by the author of [15]. We use cross-validation to measure the classification error rates. Due to the restricted number of available instances, for LP4 and LP5 datasets fivefold cross-validation was applied. For other five datasets we used tenfold cross-validation.

The number of target neighbors k was set to be the smallest number of samples of each class. The weighting parameter c was selected from 0 to 1 with interval 0.1 by cross-validation. The values of k and c for each dataset are shown in the last two columns in Table 1. Figure 4 shows the error rates of classification with different values of weighting parameter c from 0 to 1 with interval 0.1 on seven datasets.

Finally, all the experiments were performed on a PC with Intel(R) Core(TM) i3 CPU, 4G RAM memory, on a MATLAB 8.4.0 platform.

The classification error rates of different methods are presented in Table 2. We can see that the proposed method performs much better than SMDDTW, TDVM, EDDTW and ED on all seven datasets. Meantime LMNN–DTW performs better than LDML–DTW on four datasets, and a bit worse on other three datasets, which demonstrates that LMNN–DTW can achieve the performance of LDML–DTW, sometimes even better. The LDML–DTW performs better than SMDDTW, TDVM, EDDTW and ED. The bad performance of SMDDTW demonstrates that the statistic Mahalanobis matrix cannot reveal an appropriate relations among variables. The ED can only deal with the datasets whose time series shares the same length, i.e., Libras, LP4 and LP5. It is predictable that ED performs worse than the others because it does not

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

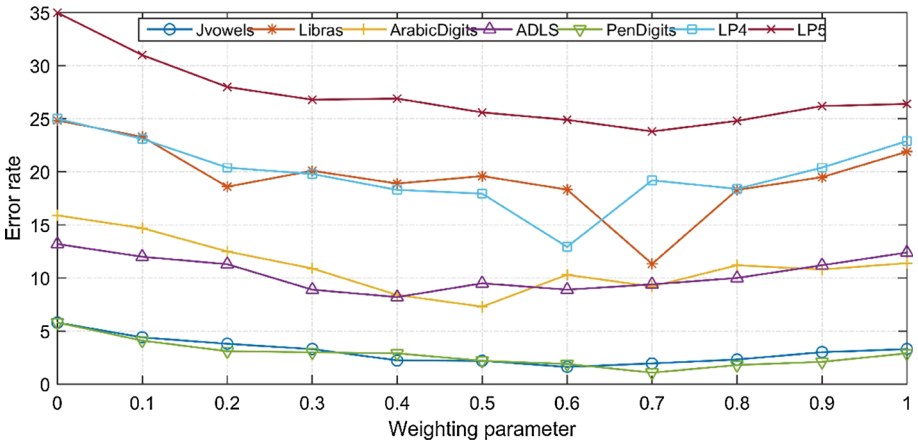

Fig. 4 Classification results with different values of weighting parameter

Table 3 Time consumption

Dataset	LMNN–DTW	LDML–DTW
JapaneseVowels	350.4s	210.3s
Libras	170.2s	142.7s
ArabicDigits	750.8s	452.2s
ADLS	859.6s	531.1s
PenDigits	1257.3s	701.1s
LP4	22.3s	15.9s
LP5	45.4s	31.3s

consider neither the offset and scaling problems that may exist in time axis nor the relations among variables.

As mentioned above, the time complexity of LMNN–DTW is about $O\ (Ndmn)$. However, the time complexity of LDML–DTW is about $O\ (Nmn)$ [15,21], smaller than LMNN–DTW. Table 3 shows the time consumption of LMNN–DTW and LDML–DTW of each folder cross-validation tests. We can tell that LMNN–DTW does take more time than LDML–DTW. Clearly that the most serious issue of LMNN–DTW in the future is to shorten the time consumption.

From Fig. 4 we can see that neither a too small nor too large weighting parameter c can lead to a good result. If the weighting parameter is too small, the first term in the cost function which penalizes large distances within same class will take main effect while the second term which penalizes small distances among different classes will lose effect. On the contrary, if the weighting parameter becomes too large, the second term will take main effect and the first term will lose effect. Both these two situations result in bad performance. The last column in Table 1 shows that the best values of weighting parameter c are commonly a bit larger than 0.5. From this we can conclude that the second term which reveals the idea of large margin plays a more important role than the first term.

Fig. 5a–g show the convergence steps of minimizing the cost function (14) by coordinate descent method on seven datasets, respectively. We can see that the minimizing procedure converges after certain iterations for all the seven datasets. Moreover, the procedures of

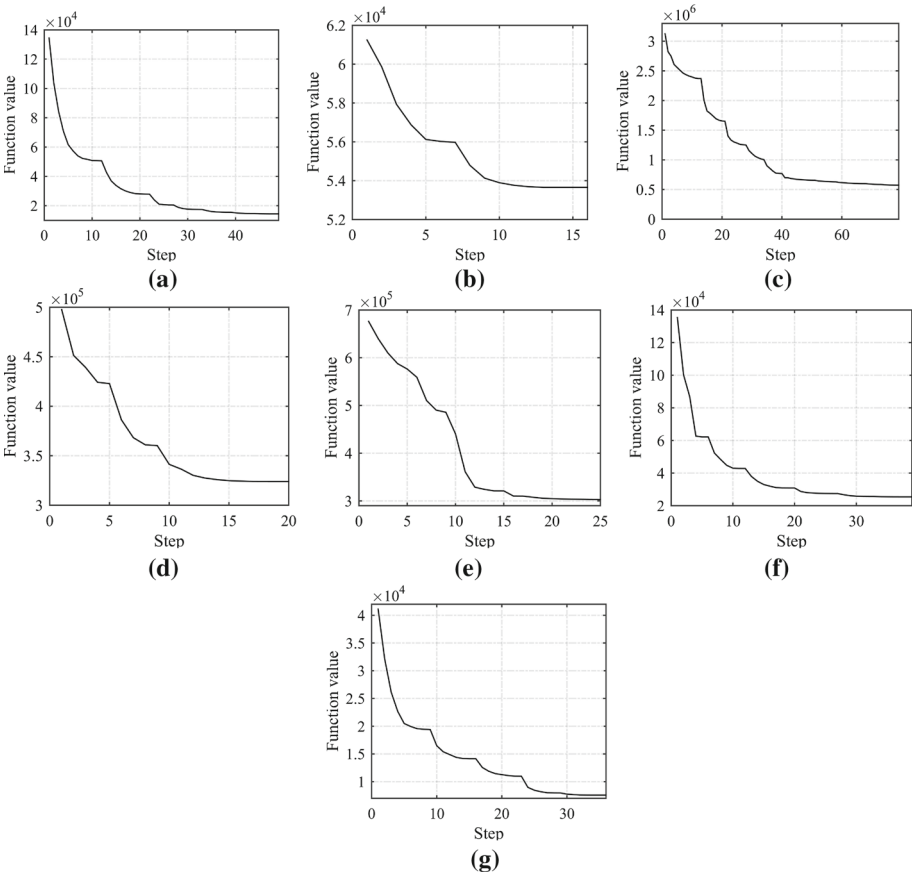

Fig. 5 Convergence steps of minimization. **a** JapaneseVowels **b** Libras **c** ArabicDigits **d** ADLS **e** PenDigits **f** LP4 **g** LP5

Table 4 Iterations of outer and inner optimization

Dataset	Outer iterations	Inner iterations
JapaneseVowels	6	9
Libras	3	5
ArabicDigits	8	10
ADLS	3	6
PenDigits	4	6
LP4	5	7
LP5	5	7

datasets with small number of variables like Libras, ADLs, PenDigits, LP4 and LP5 shown in Fig. 5b, d, e, f and g respectively will converge faster than others with large number of variables like JapaneseVowels and ArabicDigits shown in Fig. 5a, c. The reason is obvious for this phenomenon that the dataset with larger number of variables corresponds to larger size of the Mahalanobis matrix. It will take more iterations to find the optimum for a Mahalanobis

matrix with larger size. Table 4 shows the iterations that the minimizing procedure needs to converge. Outer iteration represents the whole two-step optimization iteration, while inner iteration represents the iteration in the second step minimized by L-BFGS. As there are several inner iteration procedures in each outer iteration, the Inner Iterations in Table 4 records the mean iterations.

5 Conclusion

In this paper, we propose a novel model to measure the similarity between two multivariate time series. Firstly we use a Mahalanobis distance-based DTW measure for multivariable time series, which considers the relations among variables by the Mahalanobis matrix. Secondly, the LMNN metric learning algorithm is applied to learn the best Mahalanobis matrix by rebuilding the cost function. The cost function cannot be minimized by common gradient-based methods because of the non-differentiable solving procedure of DTW measure. Consequently we review the optimization problem, from a perspective of K-means algorithm, as a two-step iterating procedure which is exactly the coordinate descent method. We compared the proposed model with other existing similarity measure methods, i.e. LDML–DTW, SDDTW, TDVM, EDTW, and ED, by testing 1NN classification on seven multivariate time series datasets. Three conclusions can be reached from the results. Firstly, from the view of classification accuracy, the proposed model performs as well as the state-of-art method LDML–DTW, and much better than other four methods, which demonstrates the validity and superiority of the proposed model. Secondly, the optimum weighting parameter c is commonly a bit larger than 0.5, which means in the cost function, the second term with the idea of large margin plays a more important role than the first term. Thirdly, the convergence curves of the iterating processes demonstrate that the minimization of cost function by coordinate descent method can converge after several iterations. Moreover, the larger the number of variables is, the more iterations need.

Acknowledgements This study was supported by the National Natural Science Foundation of China (61174114), the National Natural Science Foundation of China (U1509203), the Research Fund for the Doctoral Program of Higher Education in China (20120101130016) and the Zhejiang Provincial Science and Technology Planning Projects of China (2014C31019).

Funding This study was supported by the National Natural Science Foundation of China (61174114), the Research Fund for the Doctoral Program of Higher Education in China (20120101130016) and the Zhejiang Provincial Science and Technology Planning Projects of China (2014C31019).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals per-formed by any of the authors.

References

1. Chan NH (2002) Time series: applications to finance. Wiley, Hoboken
2. Tormene P, Giorgino T, Quaglini S, Stefanelli M (2008) Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artif Intell Med 45(1):11–34
3. Seborg DE, Singhal A (2005) Clustering multivariate time-series data. J Chemom 19(8):427–438

4. Aach J, Church GM (2001) Aligning gene expression time series with time warping algorithms. Bioinformatics 17(6):495–508
5. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27
6. Hartigan JA, Wong MA (1979) A K-means clustering algorithm. Appl Stat 28(1):100–108
7. Adankon MM, Cheriet M (2002) Support vector machine. Comput Sci 1(3):1303–1308
8. Berndt DJ, Clifford J (1994) Using dynamic time warping to find patterns in time series. In: Advances in knowledge discovery in databases papers from the Aaai workshop
9. Assent I, Kremer H (2009) Robust adaptable video copy detection., Lecture notes in computer science-Springer, New York, pp 380–385
10. Gavrila DM, Davis LS (1995) 3-D model-based tracking of human upper body movement: a multi-viewapproach. In: IEEE, pp 253–258
11. Schmill MD, Oates T, Cohen PR, Schmill MD (2000) Learned models for continuous planning. In: Proceedings of uncertainty the seventh international workshop on artificial intelligence & statistics, pp. 278–282
12. Keogh EJ, Pazzani MJ (1999) Scaling up dynamic time warping to massive dataset. In: Proceedings of the third european conference on principles of data mining and knowledge discovery, pp 1–11
13. ten Holt GA, Reinders MJ, Hendriks E (2007) Multi-dimensional dynamic time warping for gesture recognition. In: Thirteenth annual conference of the advanced school for computing and imaging
14. Orsenigo C, Vercellis C (2010) Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. Pattern Recognit 43(11):3787–3794
15. Mei J, Liu M, Wang YF, Gao H (2015) Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification. IEEE Trans Cybern, p 1
16. Yang L (2006) Distance metric learning: a comprehensive survey. Michigan State Universiy, East Lansing
17. Xing EP, Ng AY, Jordan MI, Russell S (2002) Distance metric learning, with application to clustering with side-information. Adv Neural Inf Process Syst 15:505–512
18. Barhillel BA, Hertz T Shental N, Weinshall D (2003) Learning distance functions using equivalence relations. In: Proceedings of the 20th international conference on machine learning, 2012
19. Goldberger J, Roweis ST, Hinton GE, Salakhutdinov R (2004) Neighbourhood components analysis. Adv Neural Inf Process Syst 83(6):513–520
20. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10(1):207–244
21. Mei J, Liu M, Karimi HR, Gao H (2014) LogDet divergence-based metric learning with triplet constraints and its applications. IEEE Trans Image Process A Publ IEEE Signal Process Soc 23(11):4920–4931
22. Do H, Kalousis A, Wang J, Woznica A (2012) A metric learning perspective of SVM: on the relation of SVM and LMNN. Eprint arXiv:308-317
23. Marasovic T, Papic V, Zanchi V (2015) LMNN metric learning and fuzzy nearest neighbour classifier for hand gesture recognition. J Multimodal User Interfaces 9:1–11
24. Nocedal J, Wright S (2006) Numerical optimization. Springer, New York
25. Hestenes MR, Stiefel E (1952) Methods of conjugate gradients for solving Linear systems. J Res Natl Bureau Stand 49(6):409–436
26. Byrd RH, Lu P, Nocedal J, Zhu C (1996) A limited memory algorithm for bound constrained optimization. Office of Scientific & Technical Information Technical Reports
27. Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. Math Program 45:503–528
28. Wright SJ (2015) Coordinate descent algorithms. Math Program 151(1):3–34
29. Mahalanobis PC (1936) On the generalized distance in statistics. Proc Natl Inst Sci 2:49–55