MLAT: Metric Learning for kNN in Streaming Time Series

Dongmin Park, Susik Yoon, Hwanjun Song, Jae-Gil Lee*
Graduate School of Knowledge Service Engineering, KAIST
{dongminpark,susikyoon,songhwanjun,jaegil}@kaist.ac.kr

ABSTRACT

Learning a good distance measure for distance-based classification in time series leads to significant performance improvement in many tasks. Specifically, it is critical to effectively deal with *variations* and *temporal dependencies* in time series. However, existing metric learning approaches focus on tackling variations mainly using a strict alignment of two sequences, thereby being not able to capture temporal dependencies. To overcome this limitation, we propose **MLAT**, which covers both alignment and temporal dependencies at the same time. *MLAT* achieves the alignment effect as well as preserves temporal dependencies by augmenting a given time series using a sliding window. Furthermore, *MLAT* employs time-invariant metric learning to derive the most appropriate distance measure from the augmented samples which can also capture the temporal dependencies among them well. We show that *MLAT* outperforms other existing algorithms in the extensive experiments on various real-world data sets.

CCS CONCEPTS

• **Mathematics of computing** → **Time series analysis**; • **Theory of computation** → *Machine learning theory*; • **Computing methodologies** → Supervised learning by classification.

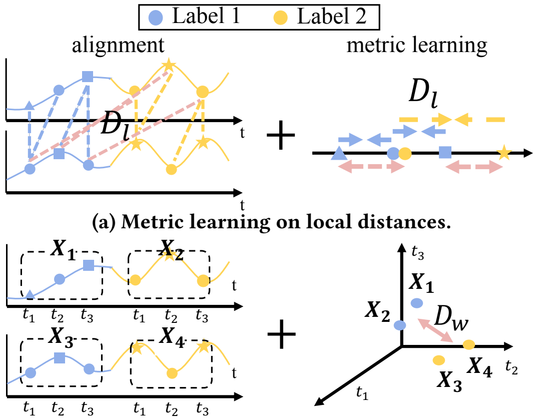
KEYWORDS

Streaming Time Series, Metric Learning, Optimization

ACM Reference Format:
Dongmin Park, Susik Yoon, Hwanjun Song, Jae-Gil Lee. 2018. MLAT: Metric Learning for kNN in Streaming Time Series. In *MileTS '19: 5th KDD Workshop on Mining and Learning from Time Series, August 5th, 2019, Anchorage, Alaska, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Streaming time series classifications play an increasingly important role in activity recognition [15] and fraud detection [12]. However, since the number of labels in streaming time series data is often insufficient to build a high-quality classifier [22], k-nearest neighbor (kNN), a non-parametric method, is widely used and it empirically results in high accuracy in several applications [6].



(a) Metric learning on local distances.

(b) Metric learning on window distances.

Figure 1: The effect of alignment to temporal dependency.

In accordance with this trend, many studies have been focused on improving the performance of kNN by designing more appropriate distance measures for a given data set. Recently, various time series metric learning approaches [3, 14, 20] have been developed to achieve this goal. Because variations in time series data, such as sequence shifting and scaling, is one of the main challenges, the existing algorithms consist of the two phases: *(i) alignment* and *(ii) metric learning*. As shown in Figure 1(a), the subsequences of the time series data are aligned in pairs to match the optimal time steps (dotted line), rendering them robust to variation. Metric learning is subsequently conducted on the *local distance* D_l computed from the matched time steps, which minimizes the distance of pairs with the same label and maximizes for pairs with different labels.

However, since the local distance only considers the difference between features at a *single* matched time step, metric learning on the local distance does not capture *temporal dependencies* appearing across consecutive time steps. Temporal dependency is known to greatly enhance the performance of time series classifications [10]. For example, when we try to distinguish between forwards and backwards from a set of motion images of a walking person, an image of a single time step gives very limited information on the direction, whereas a set of images of consecutive time steps provides clear clues. It is, therefore, more effective to distinguish samples with different labels by applying metric learning on the *window distance* D_w computed from consecutive time steps (e.g., $\{t_1, t_2, t_3\}$), as shown in Figure 1(b). The existing algorithms miss out this opportunity as they focus on alignment at the expense of the temporal dependencies. This calls for a new method that can achieve both alignment and temporal dependencies.

In this paper, we propose a novel metric learning algorithm for kNN in streaming time series, called **MLAT** (*M̲etric L̲earning considering A̲lignment and T̲emporal dependencies*). To account for both alignment and temporal dependency in learning a distance measure, *MLAT* consists of the two main phases:

[*] Jae-Gil Lee is the corresponding author.

MileTS '19, August 5th, 2019, Anchorage, Alaska, USA

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

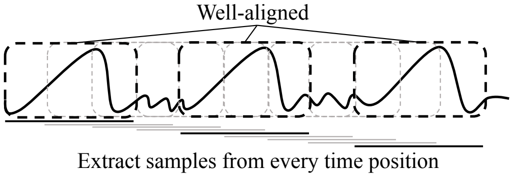


Figure 2: Sliding window augmentation.

1. **Sliding Window Augmentation**: *MLAT* augments a given time series using sliding window sampling [24] to achieve the alignment effect while preserving the temporal dependencies. As shown in Figure 2, by extracting all possible samples as sets of consecutive time steps with a fixed length, the samples having similar patterns can be well-aligned even without explicit time step matching. At the same time, because consecutive time steps are well preserved in a sample, *MLAT* can exploit the temporal dependencies between them.
2. **Time-Invariant Metric Learning**: The temporal dependencies between features of augmented samples are *time-invariant*, meaning that the dependency of two features in a fixed time difference is consistent regardless of their absolute temporal locations; i.e., the dependency of two features at time t and $t+1$ is equal to that of the two features at time $t+1$ and $t+2$. Thus, *MLAT* tries to fulfill this inherent characteristic in learning distance metric using the large margin nearest neighbor (LMNN) [13], constraining a Mahalanobis matrix to be a *Block Toeplitz* [8] structure.

Extensive experiments on *four* real-world streaming time series data sets indicate that *MLAT* results in promising kNN performance.

2 RELATED WORK

We briefly review several existing studies to find better distance measures in time series data with respect to the following aspects: *(i)* alignment to appropriately handle variation, and *(ii)* metric learning to learn a better distance function.

2.1 Alignment

Dynamic time warping (DTW) [1] is the most represensatice method to match the time steps of two samples in order to find the best warping path. Numerous variants of DTW have been studied, whose focus is on resolving the inefficiencies in time step matching [15, 17] and the invalidity of the triangle inequality in the learned distance [4, 11]. However, DTW-based alignment methods fail to deal with the temporal dependencies because they explicitly match the time steps.

2.2 Metric Learning

Most time series metric learning algorithms in the literature apply metric learning to the local distance computed using DTW based alignment. LDML-DTW [14] and LMNN-DTW [20] first match time steps by multivariate dynamic time warping (MDTW) [1], and subsequently learn the Mahalanobis matrix of the local distance using LogDet divergence and Large Margin Nearest Neighbor (LMNN) [23], respectively. DECADE [3] devised a new alignment method to learn the valid distance measure while using deep networks to capture the complex dependencies in the local distance. There have been a few studies for learning the distance representation in the embedded space obtained from the last hidden layer of LSTM [16]. However, to the best of our knowledge, no studies yet simultaneously consider alignment and temporal dependencies in time series metric learning.

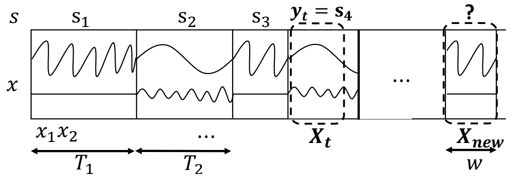


Figure 3: Stream kNN settings.

3 PRELIMINARY

3.1 Problem Setting

We introduce the main concepts in the kNN classification problem on streaming time series in the following definitions, as also illustrated in Figure 3.

Definition 3.1. (Streaming time series) A *multivariate streaming time series* $x = [x_1, x_2, \cdots, x_{\mathbb{T}}]$ is a sequential observation of $x_i \in \mathbb{R}^d$ with d features, where $\mathbb{T}$ is the length of x.

Definition 3.2. (State) A *state* $s_j \in \{1, 2, \cdots, S\}$ denotes the label of a sequence in a time series x during the time period T_j ($\subset \mathbb{T}$), where S is the number of states in x and j is the index of the state. The time series x consists of various sizes of sequences with different states; for example, a time series x collected from wearable devices for 15 min consists of the two sequences: a "walk" state of the first 10 min and a "run" state of the last 5 min.

Definition 3.3. (Sample) A *sample* $X_t = [x_t, \cdots, x_{t+w-1}] \in \mathbb{R}^{d \times w}$ is a subsequence of size w extracted from a certain sequence in time period T_j of a time series x. It consists of the consecutive time steps from time t to time $t+w-1$. The state s_j of the sequence becomes the label y_t of X_t.

Definition 3.4. (kNN classification problem) Let $\mathbb{D}$ be a set of samples from a time series x. The kNN classification problem in streaming time series classifies the label of a sample X_{new} by referring to that of the k nearest neighbors of X_{new} in $\mathbb{D}$.

3.2 Large Margin Nearest Neighbor [23]

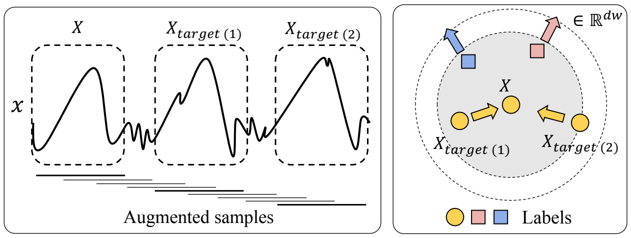
Most metric learning methods aim at learning a Mahalanobis distance matrix $\mathbf{M}$ [5, 23], where the distance is defined as:

$$D_{\mathbf{M}}(X_i, X_j) = (X_i - X_j)^T \mathbf{M} (X_i - X_j), \tag{1}$$

where X_i and $X_j \in \mathbb{R}^{dw}$ are two samples and $\mathbf{M} \in \mathbb{R}^{dw \times dw}$ is a positive semidefinite matrix. Given a set of samples $\mathbb{D}$, for each sample $X_i \in \mathbb{D}$, LMNN first finds the k nearest neighbors of X_i based on the Euclidean distance and calls them *target* neighbors. Subsequently, LMNN learns the optimal Mahalanobis distance matrix $\mathbf{M}$ which minimizes the following loss function:

$$\begin{aligned} \min_{\mathbf{M} \succeq 0} \mathcal{E}^{lmnn}(\mathbf{M}) &= (1-c) \sum_{ij} \eta_{ij} D_{\mathbf{M}}(X_i, X_j) \\ &+ c \sum_{ijl} \eta_{ij} (1 - y_{il}) \left[1 + D_{\mathbf{M}}(X_i, X_j) + D_{\mathbf{M}}(X_i, X_l) \right]_+ , \end{aligned} \tag{2}$$

where $c \in [0, 1]$ controls the weights of the two penalizing terms, $\eta_{ij} \in \{0, 1\}$ indicates whether X_j is a target neighbor of X_i, $y_{il} \in \{0, 1\}$ indicates whether X_i and X_l have the same label or not.



(a) Phase I. **(b) Phase II.**

Figure 4: Overall procedure of *MLAT*.

Algorithm 1 *MLAT*

```
INPUT: A multivariate streaming time series $x$;
OUTPUT: The optimal Mahalanobis matrix $M$;
/* PHASE I: SLIDING WINDOW AUGMENTATION */
$X \leftarrow$ Extract augmented samples from $x$;
/* PHASE II: TIME-INVARIANT METRIC LEARNING */
$\mathbf{L} \leftarrow I$, $\mathbf{M}^0 \leftarrow I$, $\mathbf{Z}^0 \leftarrow 0$, $\mathbf{U}^0 \leftarrow 0$; /* Initialize parameters. */
for $k = 1, 2, \cdots, K$ (until convergence) do /* Update parameters. */
    repeat
        Calculate $\triangledown\mathbf{M}^{k+1} \leftarrow \partial\mathcal{E}^{lmnn}/\partial\mathbf{M}^{k+1} + \rho(\mathbf{M}^{k+1} - \mathbf{Z}^k + \mathbf{U}^k)$;
        Calculate $\triangledown\mathbf{L} \leftarrow 2\mathbf{L} \times \triangledown\mathbf{M}^{k+1}$;
        Update $\mathbf{L} \leftarrow (\mathbf{L} - \alpha \triangledown \mathbf{L})$ and $\mathbf{M}^{k+1} \leftarrow \mathbf{L}^T\mathbf{L}$;
    until Converge;
    $\mathbf{Z}^{k+1}_{B^{(m)}_{ij}} \leftarrow \sum_{l=1}^{R^{(m)}} (\mathbf{M}^{k+1} + \mathbf{U}^k)_{B^{(m)}_{ij,l}} / R^{(m)}$;
    $\mathbf{U}^{k+1} \leftarrow \mathbf{U}^k + (\mathbf{M}^{k+1} - \mathbf{Z}^{k+1})$;
return $\mathbf{M}^K$;
```

Here, the first term penalizes the large distance from each sample to its k target neighbors, and the second term penalizes the small distance from each sample to all other samples that do not share the same label.

4 METHODOLOGY

MLAT consists of two phases: (1) Sliding window augmentation, and (2) Time-invariant metric learning. Algorithm 1 describes the overall procedure of *MLAT*.

4.1 Phase I: Sliding Window Augmentation

Phase I enables metric learning to consider alignment and temporal dependencies concurrently. Specifically, all possible samples in x can be extracted by sliding a fixed sized window from the beginning of x. Some parts of x where the state switches are exempted from the augmentation. These augmented samples become the training set for metric learning and help to achieve the alignment effect. As in Figure 4(a), a sample X has the most well-aligned training sample $X_{target(i)}$ as its neighbor with euclidean distance. Therefore, *MLAT*, which is based on LMNN, exploits the well-aligned samples $X_{target(i)}$ as the target neighbor of X if they share the same label and consequently learns the distance that makes them close as in Figure 4(b). Temporal dependencies can also be considered because metric learning is applied on window distance ($\mathbf{M} \in \mathbb{R}^{dw \times dw}$).

4.2 Phase II: Time-Invariant Metric Learning

Phase II allows metric learning to capture the time-invariant property of the augmented samples, making it more effective for kNN.

4.2.1 Block Toeplitz. As a result of Phase I, the consecutive samples are mostly overlapped. This leads to the covariance matrix of augmented samples in the form of *Block Toeplitz* [8] as stated in Definition 4.1, which is proven by Theorem 4.2.

Definition 4.1. (BLOCK TOEPLITZ [8]) *A $dw \times dw$ block toeplitz* matrix $\mathbf{A}$ with $d \times d$ sub-blocks $\mathbf{A}^{(m)}$ has the following form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(0)} & (\mathbf{A}^{(1)})^T & (\mathbf{A}^{(2)})^T & \cdots & (\mathbf{A}^{(w-1)})^T \\ \mathbf{A}^{(1)} & \mathbf{A}^{(0)} & (\mathbf{A}^{(1)})^T & \ddots & \vdots \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \ddots & (\mathbf{A}^{(1)})^T & (\mathbf{A}^{(2)})^T \\ \vdots & \ddots & \mathbf{A}^{(1)} & \mathbf{A}^{(0)} & (\mathbf{A}^{(1)})^T \\ \mathbf{A}^{(w-1)} & \cdots & \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \mathbf{A}^{(0)} \end{bmatrix}, \tag{3}$$

where the sub-block $\mathbf{A}^{(m)}$ appears $w - m$ times in $\mathbf{A}$ and has the same value at all occurrences.

THEOREM 4.2. *Let $\mathbb{D}_s = [X_1, X_2, \cdots, X_L] \in \mathbb{R}^{dw \times L}$ be a set of samples obtained by sliding window augmentation, where $X_t \in \mathbb{R}^{dw}$ and $L \gg w$. The covariance matrix $\Sigma \in \mathbb{R}^{dw \times dw}$ of $\mathbb{D}_s$ is in the form of Block Toeplitz.*

PROOF. $\mathbb{D}_s$ is represented with w row partitions as follows:

$$\mathbb{D}_s = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{L-w+1} \\ x_2 & x_3 & x_4 & \cdots & x_{L-w+2} \\ \vdots & \vdots & \vdots & & \vdots \\ x_w & x_{w+1} & x_{w+2} & \cdots & x_L \end{bmatrix}, \tag{4}$$

where $x_i \in \mathbb{R}^d$. Since most portions in each row partition overlap, the mean $\mu_p \in \mathbb{R}^d$ of each row partition is approximately the same. Then, by the definition of the covariance matrix, $\Sigma = 1/L \cdot (\mathbb{D}_s - \mu)(\mathbb{D}_s - \mu)^T$ approximately takes Block Toeplitz form, where $\mu = [\mu_p, \cdots, \mu_p]^T \cdot 1_L^T \in \mathbb{R}^{dw \times L}$. □

The (i, j)-th element of the covariance matrix corresponds to the dependency between the i-th and j-th features, and if the matrix is Block Toeplitz, the dependency between the two features is time-invariant [9]. Note that the time-invariant dependencies of the augmented samples should be preserved when learning distance.

4.2.2 Time-Invariant Constraints. Here, we first analyze how the Mahalanobis matrix handles the dependency between two features. Eq. (1) can be represented as follows:

$$\begin{aligned} D_{\mathbf{M}}(X_i, X_j) &= tr(\mathbf{M}(X_1 - X_2)^T(X_1 - X_2)) \\ &= \sum_{ij} \mathbf{M}_{ij}(X_1^{(i)} - X_2^{(i)})(X_1^{(j)} - X_2^{(j)}), \end{aligned} \tag{5}$$

where $X_1^{(i)}, i \in \{1, \cdots, dw\}$, is the i-th feature scala value of a sample X_1, and $\mathbf{M}_{ij}$ is the (i, j)-th element of $\mathbf{M}$. $\mathbf{M}_{ij}$ decides how much the product of the distance between i-th features and that of j-th features affects the total distance, meaning that $\mathbf{M}_{ij}$ handles the dependency between the i-th and j-th features.

To preserve the time-invariant dependencies, we propose the following time-invariant metric learning framework:

$$\begin{aligned} &\min_{\mathbf{M}} \quad \mathcal{E}^{lmnn}(\mathbf{M}) \\ &\text{subject to :} \ \ \mathbf{M} \geq 0, \ \ \mathbf{M} \in \mathcal{T}, \end{aligned} \tag{6}$$

where $\mathcal{T}$ is a set of $dw \times dw$ symmetric Block Toeplitz matrices. By constraining $\mathbf{M}$ to be Block Toeplitz, the distances between two

Table 1: Summary statistics of the UCI dataset.

Data Set	Description	Dims	# Lables
SCMA [7]	3D wearable accelerometer sensors	3	7
AReM [21]	Wireless Sensor Network	6	6
EMG [18]	Eight Electromyogram(EMG) sensors	8	5
Vicon [19]	Nine Vicon 3D trackers	27	5

features in the same time difference should be regarded equally, with the same values as in **M**. Thus, the optimal matrix can effectively capture the time-invariant dependencies structure.

4.2.3 Optimization. We solve the proposed optimization problem using the alternating direction method of multipliers (ADMM) [2]. We transform Eq. (6) into an ADMM-friendly form as follows:

$$\mathcal{L}_{\rho}(\mathbf{M},\mathbf{Z},\mathbf{U}) = \mathcal{E}^{lmnn}(\mathbf{M}) + \frac{\rho}{2} \parallel \mathbf{M}-\mathbf{Z}+\mathbf{U} \parallel_F^2 \qquad (7)$$
$$\text{subject to} : \mathbf{Z} \in \mathcal{T},\ \mathbf{Z}=\mathbf{M},\ \mathbf{M} \geq 0.$$

Then, the following three steps are repeated until convergence.

$$(a)\ \mathbf{M}^{k+1} = \underset{\mathbf{M}}{argmin}\ \mathcal{L}_{\rho}(\mathbf{M},\mathbf{Z}^k,\mathbf{U}^k)$$
$$(b)\ \mathbf{Z}^{k+1} = \underset{\mathbf{Z}}{argmin}\ \mathcal{L}_{\rho}(\mathbf{M}^{k+1},\mathbf{Z},\mathbf{U}^k) \qquad (8)$$
$$(c)\ \mathbf{U}^{k+1} = \mathbf{U}^k + (\mathbf{M}^{k+1}-\mathbf{Z}^{k+1}).$$

M-update: The **M**-update can be written as:

$$\mathbf{M}^{k+1} = \underset{\mathbf{M}\geq 0}{argmin}\ \mathcal{E}^{lmnn}(\mathbf{M}) + \frac{\rho}{2} \parallel \mathbf{M}-\mathbf{Z}^{\mathbf{k}}+\mathbf{U}^{\mathbf{k}} \parallel_F^2 . \qquad (9)$$

This problem can be solved by the gradient descent method. To ensure a positive semi-definite **M**, *MLAT* factorizes **M** as $\mathbf{M}=\mathbf{L}^T\mathbf{L}$ and updates **L** through sub-gradient descent. Here, the sub-gradient of Eq. (9) with respect to **L** is $\frac{\partial\mathcal{E}^{lmnn}}{\partial\mathbf{L}} = 2\mathbf{L}(\frac{\partial\mathcal{E}^{lmnn}}{\partial\mathbf{M}} + \rho(\mathbf{M}-\mathbf{Z}^k+\mathbf{U}^k))$ by the chain rule.

Z-update: The closed form solution of the **Z**-update is as follows:

$$\mathbf{Z}^{k+1}_{B^{(m)}_{ij}} = \sum_{l=1}^{R^{(m)}} (\mathbf{M}^{k+1}+\mathbf{U}^k)_{B^{(m)}_{ij,l}} / R^{(m)}, \qquad (10)$$

where $B^{(m)}_{ij,l}$ is the index (x,y) at **Z** of the (i,j)-th element of the l-th occurrence sub-matrix $\mathbf{Z}^{(m)}$, and $R^{(m)}$ is the number of occurrences of $\mathbf{Z}^{(m)}$ in **Z**. Refer to [9] for details.

5 EXPERIMENTS

To validate the superiority of *MLAT*, we performed the kNN classification task on *four* real-world streaming time series data sets. The experimental results confirmed that *MLAT* maintains its dominance over other algorithms.

5.1 Experiment Setup

5.1.1 Data Sets. The statistics of the four benchmark data sets are summarized in Table 1. Note that we only used 5 ambiguous labels defined as normal activities in EMG and Vicon data sets for a more difficult kNN task.

5.1.2 Algorithms. We compared *MLAT* with *four* existing algorithms for measuring distance in time series:

- ED : Euclidean distance (baseline).
- MDTW [1] : Multivariate dynamic time warping.
- LDMLT [14] : Learning local distance by DTW alignment.
- LMNN [13] : Large Margin Nearest Neighbor.
- ***MLAT*** : Our proposed method.

Table 2: Accuracy (%) of kNN with standard deviation (k = 1).

Data Set	ED	DTW	LDMLT	LMNN	*MLAT*
SCMA	97.5±0.6	98.5±0.5	97.4±0.5	99.1±0.4	**99.5±0.1**
AReM	75.3±1.2	76.4±1.9	**84.4±2.1**	73.3±1.5	80.8±0.8
EMG	61.8±2.3	65.4±1.5	65.2±2.8	65.9±2.1	**69.3±1.7**
Vicon	73.3±1.9	75.8±2.1	72.2±3.3	81.5±2.6	**84.9±1.8**

Table 3: Accuracy improvement (%) on augmented samples.

Data Set	ED	DTW	LDMLT	LMNN
SCMA	1.02	0.40	1.33	0.20
AReM	-0.21	0.52	1.30	0.41
EMG	5.34	3.52	-1.4	2.43
Vicon	3.27	-0.13	-2.2	2.33

5.1.3 Settings. We assumed that 510 consecutive observations ($|T_j|$ = 510) from each label are given for each data set, except AReM where $|T_j|$ is 480. The training samples, where the size w is set to 10, were extracted from the given observations. For existing algorithms, we randomly extracted 100 samples from each label. We set k to 1 for kNN because 1-NN is widely accepted as the most accurate for many tasks [6]. We randomly selected the starting time of the consecutive observations and reported the average results of the five experiments for each dataset.

5.2 Experimental Results

5.2.1 Overall Accuracy. Table 2 shows the kNN accuracy of the five algorithms. In the SCMA, EMG, and Vicon data sets, *MLAT* achieved the highest accuracy compared with other algorithms. LDMLT produced the highest accuracy for the AReM data set, but *MLAT* also achieved a significant improvement compared with the remaining algorithms. This emphasizes the need to consider both alignment and temporal dependencies.

5.2.2 Effect of Phases I and II. All the existing algorithms were also evaluated on augmented samples obtained in Phase I. As shown in Table 3, the accuracy of the algorithms were significantly improved by up to 5.3% compared with the accuracy from randomly extracted samples. This shows that augmenting well-aligned samples is evidently beneficial for the time series classification.

In addition, *MLAT* still outperformed LMNN by up to 9.8%, though LMNN performed on augmented samples. The performance of metric learning for time series can thus be further boosted by preserving the time-invariant characteristic.

6 CONCLUSION

In this paper, we proposed *MLAT*, a novel metric learning algorithm that considers the two main characteristics in time series, i.e., variation and temporal dependencies, by using sliding window augmentation and time-invariant metric learning, respectively. Using four real-world data sets, we showed that *MLAT* outperforms the existing algorithms in most cases. For future work, we will tackle more challenging settings where more complex variations and temporal dependencies exist.

ACKNOWLEDGMENTS

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (No. 2017R1E1A1A01075927) and the MOLIT (The Ministry of Land, Infrastructure and Transport), Korea, under the national spatial information research program supervised by the KAIA (Korea Agency for Infrastructure Technology Advancement) (19NSIP-B081011-06).

REFERENCES

[1] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*, Vol. 10. Seattle, WA, 359–370.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

[3] Zhengping Che, Xinran He, Ke Xu, and Yan Liu. 2017. DECADE: a deep metric learning model for multivariate time series. In *KDD workshop on mining and learning from time series*.

[4] Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. 2007. A kernel for time series based on global alignments. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, Vol. 2. IEEE, II–413.

[5] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*. ACM, 209–216.

[6] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment* 1, 2 (2008), 1542–1552.

[7] Activity Recognition from Single Chest-Mounted Accelerometer (SCMA) Data Set. 2014. https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer. Accessed: 2019-03-01.

[8] Robert M Gray et al. 2006. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory* 2, 3 (2006), 155–239.

[9] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. 2017. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 215–223.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Paulien Hogeweg and Ben Hesper. 1984. The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *Journal of molecular evolution* 20, 2 (1984), 175–186.

[12] Enric Junqué de Fortuny, Marija Stankova, Julie Moeyersoms, Bart Minnaert, Foster Provost, and David Martens. 2014. Corporate residence fraud detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1650–1659.

[13] Jiwen Lu, Gang Wang, Weihong Deng, and Kui Jia. 2015. Reconstruction-based metric learning for unconstrained face verification. *IEEE Transactions on Information Forensics and Security* 10, 1 (2015), 79–89.

[14] Jiangyuan Mei, Meizhu Liu, Yuan-Fang Wang, and Huijun Gao. 2016. Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification. *IEEE transactions on Cybernetics* 46, 6 (2016), 1363–1374.

[15] Abdullah Mueen and Eamonn Keogh. 2016. Extracting optimal performance from dynamic time warping. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2129–2130.

[16] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[17] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 262–270.

[18] EMG Physical Action Data Set Data Set. 2011. https://archive.ics.uci.edu/ml/datasets/EMG+Physical+Action+Data+Set. Accessed: 2019-03-01.

[19] Vicon Physical Action Data Set Data Set. 2011. https://archive.ics.uci.edu/ml/datasets/Vicon+Physical+Action+Data+Set. Accessed: 2019-03-01.

[20] Jingyi Shen, Weiping Huang, Dongyang Zhu, and Jun Liang. 2017. A novel similarity measure model for multivariate time series based on LMNN and DTW. *Neural Processing Letters* 45, 3 (2017), 925–937.

[21] Activity Recognition system based on Multisensor data fusion (AReM) Data Set. 2016. https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+system+based+on+Multisensor+data+fusion+(AReM). Accessed: 2019-03-01.

[22] Li Wei and Eamonn Keogh. 2006. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 748–753.

[23] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, Feb (2009), 207–244.

[24] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 947–956.