Tensor Multi-Elastic Kernel Self-Paced Learning for Time Series Clustering

Article *in* IEEE Transactions on Knowledge and Data Engineering · August 2019

DOI: 10.1109/TKDE.2019.2937027

5 authors, including:

Yongqiang Tang
Chinese Academy of Sciences
18 PUBLICATIONS **101** CITATIONS
SEE PROFILE

Yuan Xie
East China Normal University
109 PUBLICATIONS **2,062** CITATIONS
SEE PROFILE

Yang Xuebing
Chinese Academy of Sciences
19 PUBLICATIONS **88** CITATIONS
SEE PROFILE

Jinghao Niu
Chinese Academy of Sciences
8 PUBLICATIONS **106** CITATIONS
SEE PROFILE

Tensor Multi-Elastic Kernel Self-Paced Learning for Time Series Clustering

Yongqiang Tang, Yuan Xie, *Member, IEEE,* Xuebing Yang, Jinghao Niu, and Wensheng Zhang

Abstract—Time series clustering has attracted growing attention due to the abundant data accessible and extensive value in various applications. The unique characteristics of time series, including high-dimension, warping and the integration of multiple elastic measures, pose challenges for the present clustering algorithms, most of which take into account only part of these difficulties. In this paper, we make an effort to simultaneously address all aforementioned issues in time series clustering under a unified multiple kernels clustering (MKC) framework. Specifically, we first implicitly map the raw time series space into multiple kernel spaces via elastic distance measure functions. In such high-dimensional spaces, we resort to the tensor constraint based self-representation subspace clustering approach, involving in the self-paced learning paradigm, to explore the essential low-dimensional structure of the data, as well as the high-order complementary information from different elastic kernels. The proposed approach can be extended to more challenging multivariate time series clustering scenario in a direct but elegant way. Extensive experiments on 85 univariate and 10 multivariate time series datasets demonstrate the significant superiority of the proposed approach beyond the baseline and several state-of-the-art MKC methods.

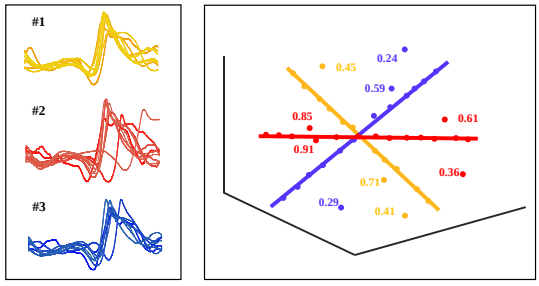
Index Terms—time series clustering, multiple kernels clustering, self-paced learning, tensor optimization.

1 Introduction

The rapid development of sensor collecting devices has resulted in a large amount of computational requirement for time series data in various domains. Among all time series analysis techniques, time series clustering is one of the most widely used methods, as it can identify interesting patterns in the absence of supervision, and facilitate other data analysis tasks, such as classification, anomaly detection, and indexing [1], [2], [3]. As a kind of uniquely high-dimensional data, time series data poses several challenges for existing clustering algorithms. First of all, time series shares the same issues with other high-dimensional data, such as the increased computational complexity and the performance degradation owing to the noise and the "curse of dimensionality" [4]. Additionally, the warping of time series data, which are common in real applications, dramatically affect the performance of many clustering algorithms [2].

Most of the previous works have been invested for overcoming only part of these two challenges. For the high dimension issue, numerous researches have focused on either representing raw data in a lower dimension compatible with conventional clustering algorithms [2], or directly using clustering algorithms suitable for high-dimensional data, e.g., subspace clustering, projected clustering [4]. To solve the warping issue, massive efforts have been taken on designing appropriate distance measures, namely elastic distance measure, e.g., the dynamic time warping (DTW) [16], the edit distance with real penalty (ERP) [23], and the shape-based distance (SBD) [20]. Nevertheless, experiments [5] suggest that these distance measures have different performances on different datasets, due to the specific characteristics of each dataset [6], [8]. Instead of pursuing a single powerful one, it has been proven that ensembling multiple elastic measures is effective in the time series classification task [9]. Motivated by this, we aim to introduce the analogical methodology into unsupervised setting, namely the time series clustering task, which is more challenging due to the absence of ground-truth.

Recently, multiple kernels clustering (MKC) approaches have achieved remarkable progress in the field of computer vision [24], [26], [29], [30]. The core idea of MKC framework is to handle the linearly non-separable problem via kernel



(a) (b)

Figure 1. Illustration of our motivation in distinguishing samples with different confidence. (a) shows a few instances of three classes in dataset *InlineSkate*. (b) illustrates how the samples distribute in the feature space, where samples of high confidence distribute compactly in the latent subspace region whereas low confidence samples are apart from the corresponding subspace.

- *Y. Tang, J. Niu and W. Zhang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China, and University of Chinese Academy of Sciences, Beijing, 101408, China. E-mail:{tangyongqiang2014, niujinghao2015}@ia.ac.cn, zhangwenshengia@hotmail.com.*
- *Y. Xie is with the School of Computer Science and Technology, East China Normal University, Shanghai, China; E-mail: yxie@sei.ecnu.edu.cn.*
- *X. Yang is with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. E-mail: yangxuebing2013@ia.ac.cn.*

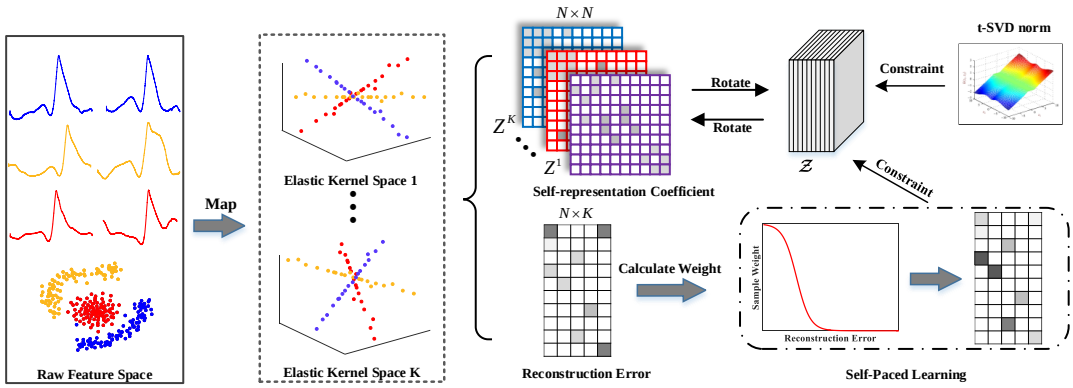


Figure 2. The flowchart of the proposed approach. The pipeline includes stages of feature space mapping (implicitly calculating the elastic kernel); subspace clustering performed in multiple elastic kernel spaces; self-representation coefficient matrices stacking and then rotating; samples confidence calculating by the SPL; the rotated tensor $\mathcal{Z}$ updating with the constraint of t-SVD norm and samples confidence.

functions, as well as exploit the complementary information from multiple kernel candidates. By mapping the raw time series space into multiple elastic kernel spaces, the MKC provides a promising solution for handling the non-linear issue caused by the time series warping, as well as the issue of multiple elastic kernels integrating.

However, the performance of directly applying the present MKC to time series clustering is far from satisfactory. In particular, there are three limitations to existing MKC approaches. First, most current MKC approaches are typically based on the k-means algorithm [26], [29], [30], while the high dimensionality characteristic of time series in elastic kernel spaces could severely degrade the performance of such kind of approach. Second, the vast majority of existing MKC approaches lack the ability of thoroughly exploring the complementary information among multiple elastic measures. Third, all samples are generally assigned equal weights, which may not be a good idea due to sample-specific characteristics or noise present in the time series datasets.

For solving the first two problems, we resort to the self-representation subspace clustering approach, specifically the low-rank representation (LRR) approach [47], which constructs affinity matrix through reconstruction coefficients, as well as emphasizes the essential structure of the data. Several recently proposed studies of subspace clustering have extended LRR to multi-view setting [10], [11], [12], [13]. Among them, by constraining the rotated subspace coefficient tensor with tensor multi-rank, t-SVD-MSC [12] can thoroughly explore the high-order complementary information from different views. In this study, we make the first effort to map the original time series space to multiple high-dimensional elastic kernel spaces, and then perform t-SVD-MSC in the new feature space to effectively capture the consensus and the complementary information among multiple elastic kernels.

As for the third problem, we propose a robust learning strategy to effectively handle the issue. Inspired by the recently proposed self-paced learning (SPL) theory [35], we introduce the SPL paradigm into our proposal by a so called easy to hard learning strategy to alleviate the chaotic subspace caused by the ambiguous samples. Specifically, as Fig. 1 (b) illustrates, at the early age of the whole clustering process, the low confidence samples that are apart from the latent low-dimensional subspace will be suppressed, while the high confidence ones that distribute compactly in the latent subspace region will be emphasized. In such a case, the proposed approach can obtain more stable and discriminative subspace by gradually involving the faithful samples.

According to above discussion, a novel approach, named Tensor constraint based Multiple Elastic Kernels integration coupled with Self-Paced Learning mechanism (T-MEK-SPL), is naturally constructed for simultaneously addressing three challenges in time series clustering, including high-dimension, warping and the integration of multiple elastic measures. The flowchart of the proposal is shown in Fig. 2. It is worth noting that the need for analysis of multivariate time series (MTS) is growing in modern society as data is increasingly collected simultaneously from multiple sources over time. While univariate time series (UTS) clustering methods are well established, most of them cannot be naturally applied to the multivariate scenario [14]. Nevertheless, we will show that the proposed T-MEK-SPL can be extended to MTS clustering in a direct but elegant way. In summary, the contributions of this paper are mainly four-fold:

- We simultaneously consider several challenges of time series clustering, including UTS and MTS, under a single unified MKC framework. To the best of our knowledge, this is the first work to adopt MKC for time series clustering.
- We propose a novel T-MEK-SPL approach, which can adequately capture the complementary information among multiple elastic kernels via high-order tensor constraint, as well as gradually achieve more stable and discriminative subspace structure in a pure self-learning way.
- We present an efficient optimization algorithm to solve the objective function of the proposed model, with relatively fast convergence empirically.
- We conduct an extensive evaluation of our proposal on 85 UTS datasets and 10 MTS datasets. The results confirm the significant advantage of our proposal.

The rest of this paper is organized as follows. Section 2 presents a brief review of related works. Section 3 introduces the notations and the preliminaries used through this paper. Section 4 describes the details of the proposed T-MEK-SPL

approach. Experimental analysis and completion results are shown in Section 5. Finally, conclusions are drawn in Section 6.

2 Related Work

This section reviews some of the previous work closely related to this paper. We first briefly review the elastic distance measure for time series. Next, related studies of multiple kernel clustering are reviewed. Finally, we introduce the self-paced learning paradigm.

2.1 Elastic Distance Measure

Time series clustering critically depends on the choice of distance measure [6]. One key issue when computing the distance or similarity between time series lies in handing time warping [53]. Conventional distance measures such as L_p-norms (e.g., Euclidean distance) fail in measuring similarity when time warping exists. In the past few decades, dozens of elastic distance measures have been proposed to overcome the limitation of non-elastic measures.

Elastic distance measures fall basically into two categories [15]: (1) elastic distance nonmetric measures, (2) elastic distance metric measures. As a representative of the first category, dynamic time warping (DTW) [16] is effective and widely used for distance measure between time series. Nevertheless, DTW is time consuming due to the dynamic programming when calculating an optimal match between two given sequences. To improve the efficiency of DTW, several variants were proposed, including Pruned-DTW [17], SparseDTW [18], and the MultiscaleDTW [19]. Despite reducing the computational complexity, most of the fast techniques based on DTW need careful tuning of the parameters. Recently, Paparrizos and Gravano introduced a new shape-based distance (SBD) [20], which are parameter-free and can be efficiently computed by Fourier transform algorithms. Unfortunately, the work described above is non-metric as they violate the principle of triangular inequality, which render most indexing structures inapplicable. By combining ℓ_p-norm with the edit distance, Chen et al. [23] proposed the edit distance with real penalty (ERP), which is a metric while supporting local time warping. Inspired by ERP, Marteau proposed an alignment-based distance metric, called time warp edit distance [15], which defines the edit operations via the paradigm of a graphical editing process. Despite their respective advantages, no significant difference between the elastic distance measures was found via extensive experiments [5].

2.2 Multiple Kernel Clustering

The key idea of kernel methods is to handle the linearly non-separable problem by mapping the original feature space into a reproducing kernel Hilbert spaces, and resorting to a nonlinear kernel function to define the similarity [24]. Performance of kernel methods highly depends on the choice of kernels, and it is especially a challenge for kernel-based clustering task due to the unsupervised setting. More recently, as a promising solution, there have been considerable interests in MKC research, which aims at exploiting the complementary information and learning an optimal kernel from multiple kernel candidates.

Existing research in MKC can roughly be divided into two categories: (1) consensus matrix optimization, (2) combination coefficient optimization. The first line optimizes a consensus matrix from multiple predefined kernel matrices with low-rank constraint. Xia et al. [25] firstly recovered a shared low-rank matrix from multiple transition probability matrices, and then put the learned matrix into the standard Markov chain model for clustering. Zhou et al. [26] utilized the specific structures of noise, and integrated them into a robust model to acquire a low-rank consensus matrix. Qu et al. [27] proposed a kernelized multi-view self-representation subspace clustering approach, which firstly optimizes the coefficient matrices with tensor multi-rank minimization constraint, and then operates spectral clustering on the affinity matrix derived from the learned coefficient matrices. The second line usually learns the optimal kernel by linearly combining multiple kernels candidates. Yu et al. [28] initialized multiple kernels k-means (MKKM) clustering algorithm, which optimizes the weights in a conic sum of kernels. Since then, efforts have been made to further improve the performance of MKKM. The work in [30] proposed a localized fusion approach that combines kernels with sample-specific weights. Du et al. [29] replaced the squared error in MKKM with $\ell_{2,1}$-norm such that the model can be robust to noise. Liu et al. [31] and Zhu et al. [32] addressed the situation where base kernels are incomplete due to the missing samples.

Among all aforementioned MKC research, the study most related to this paper is [27]. Note that, our proposal is significantly different from [27], in that their formulation uses the same weights for all kernels whereas we update sample weights in the clustering process, such that our proposal can better capture the sample-adaptive characteristics.

2.3 Self-Paced Learning

Self-paced learning shares the core idea with curriculum learning [34], in which a model is learned by gradually involving samples for training from easy to complex. The significant differences lie in that curriculum learning requires a predetermined priori of easy and hard samples in a given training dataset, whereas self-paced learning can automatically choose the order from the data themselves. The SPL model consists of a weighted loss term and a regularization term, which can be formulated as a general optimization problem:

$$\min_{\boldsymbol{\theta},\mathbf{w}} \sum_{i=1}^{n} w_i L(\boldsymbol{\theta}; x_i, y_i) + f(\mathbf{w}; \beta),\ s.t.\ \mathbf{w} \in [0,1]^n, \quad (1)$$

where $\boldsymbol{\theta}$ denotes the model parameters of specific problems, $L(\boldsymbol{\theta}; x_i, y_i)$ is the loss function, $f(\mathbf{w}, \beta)$ corresponds to a self-paced regularizer, $\mathbf{w} = [w_1, w_2, \ldots, w_n]$ represents the weight variable reflecting the complexity of samples, β is a parameter, called learning pace, for controlling the "model age" which gradually increases in order to explore more samples.

The regularizer $f(\mathbf{w}; \beta)$ is independent of loss functions, and can be defined in various forms in terms of the learning

pace. In [35], it is initially defined based on the ℓ_1-norm of $\mathbf{w} \in [0,1]^n$:

$$f(\mathbf{w};\beta) = -\beta \sum_i^n w_i. \tag{2}$$

Jiang et al. [36] proposed a regularizer consisting of $\|\mathbf{w}\|_1$ and $\|\mathbf{w}\|_{2,1}$, which can take into account the preference for both easy and diverse samples. In [37], the soft weighting schemes and three necessary conditions the self-paced function should satisfy were proposed. More recently, Li et al. [38] provided a more general way to find the desired self-paced function. By far, SPL has been considered in various tasks and models, such as image clustering [39], face identification [40], multi-instance learning [41], person re-identification [42], etc. These applications confirmed that SPL is beneficial in avoiding bad local minima and improving the generalization performance.

3 Notations and Preliminaries

In this section, we will introduce the notations and basic concepts used throughout the paper.

3.1 Notation

We use lower case letters x_{ij} to denote entries of matrix, bold lower case letters $\mathbf{x}$ to denote vector and bold upper case letters $\mathbf{X}$ to denote matrix. The i-th column of matrix $\mathbf{X}$ is denoted as $\mathbf{x}_i$. The ℓ_1-norm of $\mathbf{x}$ is $\|\mathbf{x}\|_1 = \sum_i |x_i|$, where $|\,|$ is the absolute operator. $\|\mathbf{X}\|_{2,1} = \sum_i \|\mathbf{x}_i\|$ is the $\ell_{2,1}$-norm of matrix $\mathbf{X}$, and the notation $\|\mathbf{X}\|_F := (\sum_{i,j} |x_{ij}|^2)^{\frac{1}{2}}$ is the Frobenius norm. The transpose of matrix $\mathbf{X}$ and vector $\mathbf{x}$ is denoted as $\mathbf{X}^{'}$ and $\mathbf{x}^{'}$, respectively. $\|\mathbf{X}\|_* := \sum_i \sigma_i(\mathbf{X})$ is the matrix nuclear norm, where $\sigma_i(\mathbf{X})$ denotes the i-th largest singular value of a matrix. We denote tensors by boldface calligraphy script letters, e.g., $\boldsymbol{\mathcal{X}} \in \mathcal{R}^{n_1 \times n_2 \times n_3}$ is a three-order tensor, where the order means the number of ways of the tensor and is fixed at 3 in this paper. For tensor $\boldsymbol{\mathcal{X}}$, the 2D section $\boldsymbol{\mathcal{X}}(i,:,:)$, $\boldsymbol{\mathcal{X}}(:,i,:)$ and $\boldsymbol{\mathcal{X}}(:,:,i)$ (Matlab notation is used for better understanding) denote the i-th horizontal, lateral and frontal slices, respectively. Analogously, the 1D section $\boldsymbol{\mathcal{X}}(i,j,:)$, $\boldsymbol{\mathcal{X}}(i,:,j)$ and $\boldsymbol{\mathcal{X}}(:,i,j)$ are the mode-1, mode-2 and mode-3 fibers of tensor. Specifically, $\boldsymbol{\mathcal{X}}^{(k)}$ is used to represent k-th frontal slice $\boldsymbol{\mathcal{X}}(:,:,k)$ for convenience. And $\boldsymbol{\mathcal{X}}_f$ denotes the tensor that we apply Fourier transform to $\boldsymbol{\mathcal{X}}$ along the third dimension.

3.2 Tensor Nuclear Norm

To help understand the t-SVD based tensor nuclear norm, it is necessary to introduce some pre-definitions about the new tensor theoretical and computational framework [43], [44], [45], [46].

Definition 1 (t-product). *Let $\boldsymbol{\mathcal{X}} \in \mathcal{R}^{n_1 \times n_2 \times n_3}$ and $\boldsymbol{\mathcal{Y}} \in \mathcal{R}^{n_2 \times n_4 \times n_3}$ be tensors. Then the t-product $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{Y}}$ is an $n_1 \times n_4 \times n_3$ tensor defined as*

$$\begin{bmatrix} \boldsymbol{\mathcal{M}}^{(1)} \\ \boldsymbol{\mathcal{M}}^{(2)} \\ \vdots \\ \boldsymbol{\mathcal{M}}^{(n_3)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mathcal{X}}^{(1)} & \boldsymbol{\mathcal{X}}^{(n_3)} & \cdots & \boldsymbol{\mathcal{X}}^{(2)} \\ \boldsymbol{\mathcal{X}}^{(2)} & \boldsymbol{\mathcal{X}}^{(1)} & \cdots & \boldsymbol{\mathcal{X}}^{(3)} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{\mathcal{X}}^{(n_3)} & \boldsymbol{\mathcal{X}}^{(n_3-1)} & \cdots & \boldsymbol{\mathcal{X}}^{(1)} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\mathcal{Y}}^{(1)} \\ \boldsymbol{\mathcal{Y}}^{(2)} \\ \vdots \\ \boldsymbol{\mathcal{Y}}^{(n_3)} \end{bmatrix}. \tag{3}$$

The first and second items on the right side of (3) refer to the block circulant operation bcirc(), the block vectorizing operation bvec() on tensor $\boldsymbol{\mathcal{X}}$ and tensor $\boldsymbol{\mathcal{Y}}$, respectively. Similarly, the left side of (3) represents the block vectorizing operation bvec() on tensor $\boldsymbol{\mathcal{M}}$. The opposite operation of bvec() is the fold operation denoted as bvfold(). Hence, we have $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{Y}} := \text{bvfold}\{\text{bcirc}(\boldsymbol{\mathcal{X}})\text{bvec}(\boldsymbol{\mathcal{Y}})\}$.

Theorem 1 (t-SVD). *Let $\boldsymbol{\mathcal{X}} \in \mathcal{R}^{n_1 \times n_2 \times n_3}$ be a real-valued tensor. Then $\boldsymbol{\mathcal{X}}$ can be decomposed as:*

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^{\mathrm{T}}, \tag{4}$$

where $\boldsymbol{\mathcal{U}} \in \mathcal{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathcal{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors. $\boldsymbol{\mathcal{S}}$ is an $n_1 \times n_2 \times n_3$ tensor whose each frontal slices is diagonal matrix.

The t-SVD can be computed efficiently using the fast Fourier transform fft() and the inverse operation ifft() along the third dimension, which leads to Algorithm 1 [45].

Algorithm 1: t-SVD

Input: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$;
Output: $\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{V}}$;

$\boldsymbol{\mathcal{X}}_f = \text{fft}(\boldsymbol{\mathcal{X}}, [\,], 3)$;
for $k = 1 : n_3$ **do**
  $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\boldsymbol{\mathcal{X}}_f^{(k)})$;
  $\boldsymbol{\mathcal{U}}_f^{(k)} = \mathbf{U}, \boldsymbol{\mathcal{S}}_f^{(k)} = \mathbf{\Sigma}, \boldsymbol{\mathcal{V}}_f^{(k)} = \mathbf{V}$;
end
$\boldsymbol{\mathcal{U}} = \text{ifft}(\mathcal{U}_f, [\,], 3), \boldsymbol{\mathcal{S}} = \text{ifft}(\mathcal{S}_f, [\,], 3), \ \boldsymbol{\mathcal{V}} = \text{ifft}(\mathcal{V}_f, [\,], 3)$;
Return $\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{V}}$.

Then the t-SVD based tensor nuclear norm (t-TNN) is given as:

$$\begin{aligned} ||\boldsymbol{\mathcal{X}}||_{\circledast} :=& || \begin{bmatrix} \boldsymbol{\mathcal{X}}_f^{(1)} & & \\ & \ddots & \\ & & \boldsymbol{\mathcal{X}}_f^{(n_3)} \end{bmatrix} ||_* = || \begin{bmatrix} \boldsymbol{\mathcal{S}}_f^{(1)} & & \\ & \ddots & \\ & & \boldsymbol{\mathcal{S}}_f^{(n_3)} \end{bmatrix} ||_* \\ =& \sum_{i=1}^{\min(n_1,n_2)} \sum_{k=1}^{n_3} |\boldsymbol{\mathcal{S}}_f(i,i,k)|, \end{aligned} \tag{5}$$

which is proven to be a valid norm and the tightest convex relaxation to ℓ_1-norm of the tensor multi-rank in [43], [44].

3.3 t-SVD-MSC Method

Our work is motivated by a state-of-the-art multiview clustering method, i.e., the t-SVD-MSC [12], which can be formalized as the following convex optimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{Z}^v, \mathbf{E}^v} \ \|\boldsymbol{\mathcal{Z}}\|_{\circledast} + \lambda \|\mathbf{E}\|_{2,1}, \\ s.t. \quad &\mathbf{X}^v = \mathbf{X}^v \mathbf{Z}^v + \mathbf{E}^v, \ v = 1, \ldots, V, \\ &\boldsymbol{\mathcal{Z}} = \Gamma(\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^V), \\ &\mathbf{E} = [\mathbf{E}^1; \mathbf{E}^2, \ldots, \mathbf{E}^V], \end{aligned} \tag{6}$$

where $\mathbf{X}^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \ldots, \mathbf{x}_N^v] \in \mathbb{R}^{d_v \times N}$ is the data matrix of the v-th view, N is the number of samples, d_v

is the dimensionality of each sample in v-th view, $\mathbf{Z}^v = [\mathbf{z}_1^v, \mathbf{z}_2^v, \ldots, \mathbf{z}_N^v] \in \mathbb{R}^{N\times N}$ is the coefficient matrix with each $\mathbf{z}_i^v$ representing the new coding of sample $\mathbf{x}_i^v$, and $\|\cdot\|_\circledast$ is the tensor nuclear norm as the convex approximation of tensor low-rank. The function $\Gamma(\cdot)$ constructs a 3-mode tensor $\mathcal{Z}$ by merging different representation $\mathbf{Z}^v$, and then rotate its dimensionality to $N \times V \times N$. Here, the $\ell_{2,1}$-norm is adopted to model the reconstruction error $\mathbf{E}$. The affinity matrix is constructed as:

$$\mathbf{A} = \frac{1}{V}\sum_{v=1}^{V}(|\mathbf{Z}^v| + |\mathbf{Z}^{v'}|), \tag{7}$$

where $\mathbf{Z}^{v'}$ is the transpose of matrix $\mathbf{Z}^v$. After obtaining the affinity matrix $\mathbf{A}$, spectral clustering is performed on it to generate the ultimate clustering result.

4 Formulation and Optimization

In this section, we first describe the details of our approach. Then an efficient algorithm is designed to solve the proposed optimization problem. Finally, we show that our proposal can be easily extended to multivariate time series clustering scenario.

4.1 The Proposed T-MEK-SPL

In this study, each view is represented by a kernel matrix. Following the existing kernel-based method, for the k-th kernel, let $\pi^k(\mathbf{x}_i) : \mathbb{R}^d \to \mathcal{H}^k$ represent the mapping from original space to the k-th higher-dimensional space $\mathcal{H}^k$ (usually implicitly defined). Let's define $\mathbf{\Pi}^k(\mathbf{X}) = [\pi^k(\mathbf{x}_1), \ldots, \pi^k(\mathbf{x}_N)]$. Thus, we can transform (6) to the nonlinear version in the kernel spaces $\mathcal{H}^k$ as:

$$\begin{aligned} &\arg\min_{\mathbf{Z}^k} \|\mathcal{Z}\|_\circledast + \lambda\sum_{k=1}^{K}\|\mathbf{E}^k\|_{2,1}, \\ s.t.\quad &\mathbf{E}^k = \mathbf{\Pi}^k(\mathbf{X}) - \mathbf{\Pi}^k(\mathbf{X})\mathbf{Z}^k, k = 1, \ldots, K \\ &\mathcal{Z} = \Gamma(\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^K). \end{aligned} \tag{8}$$

In the objective function (8), the algorithm treats all the samples and all the kernels equally in clustering process, which may cause the chaotic subspace derived from ambiguous samples or kernel-specific characteristics of the data. Inspired by the learning paradigm of SPL, which learns gradually from easy samples to hard samples, we further propose the following objective function by considering the complexities of samples in different kernel spaces:

$$\begin{aligned} &\arg\min_{\mathbf{Z}^k,\mathbf{W}} \|\mathcal{Z}\|_\circledast + \lambda\left(\sum_{k=1}^{K}\|\mathbf{E}^k\mathrm{diag}(\mathbf{w}^k)\|_{2,1} + f(\mathbf{W};\beta)\right), \\ &s.t.\quad \mathbf{E}^k = \mathbf{\Pi}^k(\mathbf{X}) - \mathbf{\Pi}^k(\mathbf{X})\mathbf{Z}^k, k = 1, \ldots, K \\ &\qquad \mathcal{Z} = \Gamma(\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^K), \end{aligned} \tag{9}$$

where $\mathbf{w}^k = [w_1^k, \ldots, w_N^k]$ denotes the weights of N examples in kernel space $\mathcal{H}^k$, $\mathbf{W}$ is the weight matrix with k-th column corresponding to samples weights in kernel space $\mathcal{H}^k$, i.e., $\mathbf{W} = [\mathbf{w}^1; \ldots; \mathbf{w}^K]$, and $f(\mathbf{W};\beta)$ represents the regularizer determining the examples and kernels to be selected during training.

Note that, explicitly determining the mapping function $\pi^k(\mathbf{x}_i)$ is nontrivial. Thanks to the kernel methods, which define a kernel $\mathbf{H}^k$ such that for any two points $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$, $\mathbf{H}^k(\mathbf{x}_i, \mathbf{x}_j)$ implicitly be equal to an inner product of vectors $\pi^k(\mathbf{x}_i)$ and $\pi^k(\mathbf{x}_j)$. Hence. $\mathbf{H}^k \in \mathbb{R}^{N\times N}$ can be calculated as $\mathbf{H}^k = \mathbf{\Pi}^k(\mathbf{X})^{'}\mathbf{\Pi}^k(\mathbf{X})$, and the weighted reconstruction error term in (9) can be rewritten as:

$$\begin{aligned} &\|\mathbf{E}^k\mathrm{diag}(\mathbf{w}^k)\|_{2,1} \\ =&\|(\mathbf{\Pi}^k(\mathbf{X}) - \mathbf{\Pi}^k(\mathbf{X})\mathbf{Z}^k)\mathrm{diag}(\mathbf{w}^k)\|_{2,1} \\ =&\|\mathbf{\Pi}^k(\mathbf{X})\mathbf{Q}^k\mathrm{diag}(\mathbf{w}^k)\|_{2,1} \\ =&\sum_{i=1}^{N} w_i^k(\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k)^{\frac{1}{2}}, \end{aligned} \tag{10}$$

where $\mathbf{Q}^k = \mathbf{I} - \mathbf{Z}^k \in \mathbb{R}^{N\times N}$ and $\mathbf{Q}^k = [\mathbf{q}_1^k, \ldots, \mathbf{q}_N^k]$. Consequently, the problem (9) can be reformulated as:

$$\begin{aligned} &\arg\min_{\mathbf{Z}^k,\mathbf{Q}^k,\mathbf{W}} \|\mathcal{Z}\|_\circledast + \lambda\left(\sum_{k=1}^{K}\sum_{i=1}^{N} w_i^k(\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k)^{\frac{1}{2}} + f(\mathbf{W};\beta)\right), \\ &s.t.\quad \mathcal{Z} = \Gamma(\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^K) \\ &\qquad \mathbf{Q}^k = \mathbf{I} - \mathbf{Z}^k, k = 1, \ldots, K. \end{aligned} \tag{11}$$

4.2 Optimization Procedure

The optimization problem (11) can be solved by using the inexact augmented Lagrange multiplier [47]. To adopt alternating direction minimizing strategy, we need to make the objective function seperable. By introducing the auxiliary tensor variable $\mathcal{U}$ to replace $\mathcal{Z}$, the optimization problem can be transferred to minimize the following unconstrained problem:

$$\begin{aligned} &\mathcal{L}(\mathbf{Z}^1, \ldots, \mathbf{Z}^K; \mathbf{W}; \mathbf{Q}^1, \ldots, \mathbf{Q}^K; \mathbf{L}^1, \ldots, \mathbf{L}^K; \mathcal{M}; \mathcal{U}) \\ &= \|\mathcal{U}\|_\circledast + \lambda\left(\sum_{k=1}^{K}\sum_{i=1}^{N} w_i^k(\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k)^{\frac{1}{2}} + f(\mathbf{W};\beta)\right) \\ &+ \sum_{k=1}^{K}\left(\left\langle\mathbf{L}^k, \mathbf{I} - \mathbf{Z}^k - \mathbf{Q}^k\right\rangle + \frac{\mu}{2}\|\mathbf{I} - \mathbf{Z}^k - \mathbf{Q}^k\|_F^2\right) \\ &+ \langle\mathcal{M}, \mathcal{Z} - \mathcal{U}\rangle + \frac{\rho}{2}\|\mathcal{Z} - \mathcal{U}\|_F^2, \end{aligned} \tag{12}$$

where the matrix $\mathbf{L}^k$ and the tensor $\mathcal{M}$ represent two Lagrange multipliers, μ and ρ are actually the penalty parameter, which are adjusted by using adaptive updating strategy as suggested in [48]. The alternating direction minimizing is adopted for updating $\mathbf{Z}^k$, $\mathbf{W}$, $\mathbf{Q}^k$, $\mathcal{U}$, $\mathcal{M}$ and $\mathbf{L}^k$ $(k = 1, 2, \ldots, K)$. The detailed procedure can be partitioned into six steps alternatively.

1) Representation Matrix $\mathbf{Z}^k$ Updating: When $\mathbf{Q}$ and $\mathcal{U}$ are fixed, we can solve the following subproblem for updating the representation matrix $\mathbf{Z}^k$:

$$\begin{aligned} \min_{\mathbf{Z}^k}\ &\left\langle\mathbf{L}^k, \mathbf{I} - \mathbf{Z}^k - \mathbf{Q}^k\right\rangle + \frac{\mu}{2}\|\mathbf{I} - \mathbf{Z}^k - \mathbf{Q}^k\|_F^2 \\ &+ \left\langle\mathbf{M}^k, \mathbf{Z}^k - \mathbf{U}^k\right\rangle + \frac{\rho}{2}\|\mathbf{Z}^k - \mathbf{U}^k\|_F^2. \end{aligned} \tag{13}$$

The closed-form of $\mathbf{Z}^k$ can be obtained by setting the derivative of (13) to zero:

$$\mathbf{Z}^{k*} = (\mu\mathbf{I} + \mathbf{L}^k + \rho\mathbf{U}^k - \mu\mathbf{Q}^k - \mathbf{M}^k)/(\mu + \rho). \tag{14}$$

2) Matrix $\mathbf{Q}^k$ Updating: When other parameters are fixed, we have

$$\begin{aligned}
&\min_{\mathbf{Q}^k} \lambda\sum_{i=1}^{N} w_i^k(\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k)^{\frac{1}{2}} \\
&\quad + \left\langle \mathbf{L}^k, \mathbf{I}-\mathbf{Z}^k-\mathbf{Q}^k \right\rangle + \frac{\mu}{2}\|\mathbf{I}-\mathbf{Z}^k-\mathbf{Q}^k\|_F^2 \\
=&\min_{\mathbf{Q}^k} \lambda\sum_{i=1}^{N} w_i^k(\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k)^{\frac{1}{2}} + \frac{\mu}{2}\|\mathbf{I}-\mathbf{Z}^k-\mathbf{Q}^k+\mathbf{L}^k/\mu\|_F^2 \\
=&\min_{\mathbf{Q}^k} \lambda\sum_{i=1}^{N} w_i^k(\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k)^{\frac{1}{2}} + \frac{\mu}{2}\|\mathbf{Q}^k-\mathbf{D}^k\|_F^2,
\end{aligned} \tag{15}$$

where $\mathbf{D}^k = \mathbf{I}-\mathbf{Z}^k+\mathbf{L}^k/\mu$, and we have dropped the terms that are irrelevant to $\mathbf{Q}^k$.

Note that it is nontrivial to solve the problem in (15), where the objective function is convex but nonsmooth. Nevertheless, it can be solved analytically by rewriting the problem in (15) as follows:

$$\min_{\{\mathbf{q}_i^k\}_{i=1}^N} \sum_{i=1}^{N}\sqrt{\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k} + \sum_{i=1}^{N}\frac{\tau_i^k}{2}\|\mathbf{q}_i^k-\mathbf{d}_i^k\|^2, \tag{16}$$

where $\mathbf{d}_i^k$ (respectively, $\mathbf{q}_i^k$) is the i-th column of $\mathbf{D}^k$ (respectively, $\mathbf{Q}^k$) and $\tau_i^k = \mu/(\lambda w_i^k)$. Note that the optimization problem (16) is separable with respect to $\mathbf{q}_i^k$, hence it can be decomposed into N subproblems and each is in the following form:

$$\min_{\mathbf{q}_i^k} \sqrt{\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k} + \frac{\tau_i^k}{2}\|\mathbf{q}_i^k-\mathbf{d}_i^k\|^2. \tag{17}$$

According to [33], the optimal solution $\mathbf{q}_i^{k*}$ of problem (17) (where $\tau_i^k > 0$) is

$$\mathbf{q}_i^{k*} = \begin{cases} \hat{\mathbf{q}^k}, & \text{if } \|[1/\sigma_1^k,\ldots,1/\sigma_r^k]'\circ\mathbf{g}^k\| > 1/\tau_i^k \\ \mathbf{d}_i^k - \mathbf{V}_r^k\mathbf{g}^k, & \text{otherwise,} \end{cases} \tag{18}$$

where $\mathbf{V}^k$ is the SVD of the k-th kernel matrix, *i.e.*, $\mathbf{H}^k = \mathbf{V}^k{\mathbf{\Sigma}^k}^2{\mathbf{V}^k}'$, $\mathbf{\Sigma}^k = \mathrm{diag}(\sigma_1^k,\ldots,\sigma_r^k,0,\ldots,0)$, r is the rank of kernel matrix $\mathbf{H}^k$ and $\mathbf{V}_r^k \in \mathbb{R}^{N\times r}$ is formed by the first r column of $\mathbf{V}^k$. We define $\mathbf{g}^k = {\mathbf{V}_r^k}'\mathbf{d}_i^k \in \mathbb{R}^r$ and the vector $\hat{\mathbf{q}^k}$ is defined as:

$$\hat{\mathbf{q}^k} = \mathbf{d}_i^k - \mathbf{V}_r^k\left(\left[\frac{{\sigma_1^k}^2}{\tau_i^k\alpha+{\sigma_1^k}^2},\ldots,\frac{{\sigma_r^k}^2}{\tau_i^k\alpha+{\sigma_r^k}^2}\right]'\circ\mathbf{g}^k\right), \tag{19}$$

where α is a positive scalar, satisfying

$${\mathbf{g}^k}'\mathrm{diag}\left(\left\{\frac{{\sigma_i^k}^2}{\left(\tau_i^k\alpha+{\sigma_i^k}^2\right)^2}\right\}_{1\le i\le r}\right)\mathbf{g}^k = \frac{1}{{\tau_i^k}^2}. \tag{20}$$

In particular, when $\|[1/\sigma_1^k,\ldots,1/\sigma_r^k]'\circ\mathbf{g}^k\| > 1/\tau_i^k$, the equation in (20) (with respect to α) has a unique positive root, which can be obtained by the bisection method [49].

3) Samples Weight $\mathbf{w}^k$ Updating: (12) can be decomposed into N individual subproblems. For w_i^k, the optimization problem is:

$$\min_{w_i^k} w_i^k\sqrt{\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k} + f(w_i^k;\beta). \tag{21}$$

We define $l_i^k = \sqrt{\mathbf{q}_i^{k'}\mathbf{H}^k\mathbf{q}_i^k}$ and adopt the tanh function as soft weighting regularizer for self-paced learning [38]:

$$f(w_i^k,\beta) = \frac{1}{2}\left((1-w_i^k)\ln(1-w_i^k) + w_i^k\ln w_i^k\right) - \beta w_i^k, \tag{22}$$

where parameter β controls the pace at which the model learns new samples (kernels) , and it is usually iteratively increased during optimization. The optimal w_i^{k*} can be obtained by setting the gradient of (21) with respect to w_i^k to zero

$$w_i^{k*} = \frac{1}{1+e^{2(l_i^k-\beta)}}. \tag{23}$$

4) Tensor $\boldsymbol{\mathcal{U}}$ Updating: When $\mathbf{Z}^k(k=1,2,\ldots,K)$ are given, we update the tensor $\boldsymbol{\mathcal{U}}$ as

$$\boldsymbol{\mathcal{U}}^* = \arg\min_{\boldsymbol{\mathcal{U}}} \|\boldsymbol{\mathcal{U}}\|_{\circledast} + \frac{\rho}{2}\|\boldsymbol{\mathcal{U}}-(\boldsymbol{\mathcal{Z}}+\frac{1}{\rho}\boldsymbol{\mathcal{M}})\|_F^2. \tag{24}$$

Instead of directly solving the subproblem of (24), according to [12], we can first transform (24) to the Fourier domain in which (24) can be reformulated as:

$$\boldsymbol{\mathcal{U}}_f^* = \underset{\boldsymbol{\mathcal{U}}_f}{\mathrm{argmin}} \sum_{j=1}^{N}\gamma||\boldsymbol{\mathcal{U}}_f^{(j)}||_* + \frac{1}{2}||\boldsymbol{\mathcal{U}}_f^{(j)}-(\boldsymbol{\mathcal{Z}}+\frac{1}{\rho}\boldsymbol{\mathcal{M}})_f^{(j)}||_F^2, \tag{25}$$

where $\gamma = N/\rho$. Then the tensor optimization can be divided into N independent F-norm based nuclear norm low rank matrix approximation problem in Fourier domain, which can be solved by a soft-thresholding operation [50].

5) Lagrange Multiplier Updating: The Lagrange multiplier $\mathbf{L}^k$ and $\boldsymbol{\mathcal{M}}$ need to be updated as follows:

$$\mathbf{L}^{k*} = \mathbf{L}^k + \mu(\mathbf{I}-\mathbf{Z}^k-\mathbf{P}^k), \tag{26}$$

$$\boldsymbol{\mathcal{M}}^* = \boldsymbol{\mathcal{M}} + \rho(\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{U}}). \tag{27}$$

6) Parameters μ, ρ and Self-paced Parameter β Updating:

$$\mu = \min(\eta_1\mu, \mu_{\max}), \tag{28}$$

$$\rho = \min(\eta_1\rho, \rho_{\max}), \tag{29}$$

$$\beta = \min(\eta_2\beta, \beta_{\max}). \tag{30}$$

Finally, the optimization procedure of the proposed T-MEK-SPL approach is described in Algorithm 2.

4.3 The Extension to Multivariate Time Series Clustering

The existing methods for MTS clustering usually make the hypothesis that the data from individual variate is independent of each other. Nevertheless, such a kind of assumption ignores the collaborative and complementary information of multiple variates. Built upon our MKC clustering problem, we can further explore the shared information contained in multiple variates. Thanks to the tensor structure, the proposed model can be easily incorporated with multiple variates by stacking their corresponding kernel matrices and be formulated as:

$$\begin{aligned}
&\arg\min_{\mathbf{Z}^{vk},\mathbf{W}} \|\boldsymbol{\mathcal{Z}}\|_{\circledast} + \lambda\left(\sum_{v=1}^{V}\sum_{k=1}^{K}\|\mathbf{E}^{vk}\mathrm{diag}(\mathbf{w}^{vk})\|_{2,1} + f(\mathbf{W};\beta)\right), \\
&\quad s.t. \quad \boldsymbol{\mathcal{Z}} = \Gamma(\mathbf{Z}^{v1},\mathbf{Z}^{vk},\ldots,\mathbf{Z}^{VK}), \\
&\qquad\quad \mathbf{E}^{vk} = \mathbf{\Pi}^k(\mathbf{X}^v) - \mathbf{\Pi}^k(\mathbf{X}^v)\mathbf{Z}^{vk},
\end{aligned} \tag{31}$$

Algorithm 2: T-MEK-SPL for time series clustering

Input: Multi-view elastic kernel matrices: $\mathbf{H}^1, \mathbf{H}^2, \ldots, \mathbf{H}^K$, λ, cluster number K_c
Output: Clustering result $\mathcal{C}$

Initialize $\mathbf{Z}^k = \mathbf{L}_k = \mathbf{0}; \mathbf{Q}^k = \mathbf{I}_N; \boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{M}} = \mathbf{0}; \mu = \rho = 10^{-5}, \eta_1 = 2, \mu_{\max} = 10^{10}, \rho_{\max} = 10^{10}, \epsilon = 10^{-7}, \beta = 0.5, \eta_2 = 1.05, \beta_{\max} = 1,$
while *not converge* **do**
  // Representation matrix $\mathbf{Z}^k$ *updating*
  Update $\mathbf{Z}^k, k = 1, 2, \ldots, K$ by using (14);
  // Matrix $\mathbf{Q}_k$ *updating*
  Update $\mathbf{Q}_k, k = 1, 2, \ldots, K$ by using (18);
  // Samples weight $\mathbf{w}^k$ *updating*
  Update $\mathbf{w}^k, k = 1, 2, \ldots, K$ by using (23);
  Obtain $\boldsymbol{\mathcal{Z}} = \Gamma(\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^K)$;
  // Tensor $\boldsymbol{\mathcal{U}}$ *updating*
  Update $\boldsymbol{\mathcal{U}}$ via subproblem (24);
  Update $\mathbf{L}^k$ and $\boldsymbol{\mathcal{M}}$ by using (26)$\sim$(27);
  Update parameter μ, ρ and self-paced parameter β:
    $\mu = \min(\eta_1 \mu, \mu_{\max})$,
    $\rho = \min(\eta_1 \rho, \rho_{\max})$,
    $\beta = \min(\eta_2 \beta, \beta_{\max})$;
  $(\mathbf{Z}^1, \ldots, \mathbf{Z}^K) = \Gamma^{-1}(\boldsymbol{\mathcal{Z}})$,
  $(\mathbf{U}^1, \ldots, \mathbf{U}^K) = \Gamma^{-1}(\boldsymbol{\mathcal{U}})$;
  Check the convergence conditions:
    $\|\mathbf{I} - \mathbf{Z}^k - \mathbf{Q}^k\|_\infty < \epsilon$;
    $\|\mathbf{Z}^k - \mathbf{U}^k\|_\infty < \epsilon$;
end
Obtain the affinity matrix by
  $\mathbf{A} = \frac{1}{K} \sum_{k=1}^{K} (|\mathbf{Z}^k| + |\mathbf{Z}^{k'}|)$;
Apply the spectral clustering method with the affinity matrix $\mathbf{A}$;
Return Clustering result $\mathcal{C}$.

where V is the number of variates, the superscript of $\mathbf{E}^{vk}$ and $\mathbf{Z}^{vk}$ represents the mapping from the v-th variate to the k-th kernel space. The optimization problem can be directly solved with the above procedure.

5 Experimental Study

In this section, we first give a description of the datasets, competitors, parameters setting and evaluation metrics. Then, the experimental results and observations of the proposed methods are presented, where both UTS clustering and MTS clustering are tested. At last, we analyze the characteristics of our T-MEK-SPL approach, such as parameter sensitivity, computational complexity and convergence.

5.1 Datasets

For UTS clustering, we perform experiments on time series datasets from UCR [1] repository. Unlike several studies that carried out experiments on only a few datasets, experiments on 85 univariate datasets are implemented in this study. The sequences in each dataset have equal length, but from one dataset to another the sequence length varies from 24 to 2709, the number of samples varies from 40 to 16637, and the number of classes varies from 2 to 60. Due to space limitations, we are unable to display all the results on the 85 datasets. In [20], Paparrizos and Gravano evaluate their method on 48 datasets, which are the subset of the 85 datasets version. Fortunately, in this study, we find that the key conclusions drew from the 48 datasets are consistent with the ones from the 85 datasets. Hence, in this paper, we only show and analyze the results on the 48 datasets. The results on other 37 datasets and the analysis on all 85 datasets are detailed in the supplementary materials. Clustering performance is evaluated over the fused train and test sets of each dataset.

For MTS clustering, we evaluate the clustering performance of our proposal on 10 datasets, which are commonly used by other MTS analysis studies [56], [57], [58] and publicly available from websites [2] [3] [4] [5]. The datasets originate from various domains, including medicine, robotics, handwriting recognition, etc. The number of samples per dataset varies from 58 to 10,992, the number of classes varies from 2 to 42, and the number of variables varies from 2 to 62. The detailed information on the MTS datasets used is presented in Table 3. In order to make meaningful comparisons between two time series, we normalize each variate of MTS separately using *Z-normalization* before making a comparison [52].

5.2 Competitors and Parameters

The proposed method is compared with four other representative clustering method, three of which are MKC algorithms:

- TADPole [21][6] : Dynamic time warping is a highly competitive distance measure for most time series data mining task. Since TADPole makes dynamic time warping clustering extremely efficient and achieves competitive results , we set it as the baseline for UTS clustering.
- LMKKM [30][7]: refers to Localized Multiple Kernel K-Means. This method shares similar motivation with us, in that combines multiple kernels by assigning sample-specific weights.
- RMKKM [29][8]: refers to Robust Multiple Kernel K-Means. This method improves the robustness of multiple kernel k-means [28] by replacing the sum-of-squared loss with a $l_{2,1}$-norm one.
- RMKC [26]: refers to Robust Multiple Kernel Clustering. RMKC first learns a robust yet low-rank kernel for clustering by exploring the structure of noise in multiple kernels, and then run k-means or spectral clustering algorithms. Here, we report the best result between the two algorithms for each dataset.

1. http://www.cs.ucr.edu/ eamonn/time_series_data/
2. https://archive.ics.uci.edu/ml/index.php
3. http://mlr.cs.umass.edu/ml/datasets.html
4. http://mocap.cs.cmu.edu/search.php?subjectnumber=16
5. https://www.cs.cmu.edu/ bobski/pubs/tr01108.html
6. https://www.cs.ucr.edu/ nbegu001/SpeededClusteringDTW/
7. Matlab code: https://github.com/mehmetgonen/lmkkmeans
8. Matlab code: http://lcs.ios.ac.cn/ duliang/

In this study, it is assumed that the true number of clusters is known and is set as the true number of classes. We generalize Gaussian RBF kernel into the Gaussian elastic matching kernel by replacing the Euclidean distance with elastic distance. The widths of the Gaussian kernel are set as the mean of all pair-wise sample distance, respectively. Note that, Gaussian elastic matching kernel $\mathbf{H}^k$ may not be a positive definite symmetric (PDS) kernel. In practice, as suggested by [53], we first check whether $\mathbf{H}^k$ is PDS. If $\mathbf{H}^k$ is not PDS, then we replace the non-PDS $\mathbf{H}^k$ with the proper PDS matrices by using the spectrum clip method [54]. It is worthy noting that the purpose of this study is to design a promising clustering algorithm for time series data. In order to achieve a fair comparison, we tune the parameters for all approaches and report their optimal results. Two parameters can have effect on the performance of TADPole, i.e., the cutoff distance (d_c) and the maximum amount of warping (w) [22]. Since TADPole is not sensitive to d_c [21], we set the default value of d_c to 3. To be consistent with [22], w is tuned from 0 to 20 with step of 1. The parameters for three MKC competitors are set according to the recommendations of the corresponding authors. For the proposed T-MEK-SPL, only the reconstruction error parameter λ needs to be tuned and is chosen from a discrete set $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5\}$ by grid search. We found its empirical value is within the range [0.05,0.1]. For all UTS and MTS datasets, we fix the self-paced learning rate η_2 to 1.05, initial value β to 0.5, maximum value $\beta_{\max}$ to 1.

5.3 Evaluation Metrics

In this study, two external metrics, namely, Normalized Mutual Information (NMI) and Rand Index (RI), are used as the quality evaluation metrics, which can evaluate cluster performance from information theoretic and pair-counting perspective, respectively [60].

NMI is an information theoretic metric of how much information is shared between two clusterings, which is more reliable for measuring clustering results of imbalanced data and can be calculated as:

$$\mathrm{NMI}(U,C) = \frac{\sum_{i=1}^{R}\sum_{j=1}^{V} P(i,j)\log\frac{P(i,j)}{P(i)P(j)}}{\sqrt{\left(-\sum_{i=1}^{R} P(i)\log P(i)\right)\left(-\sum_{j=1}^{V} P(j)\log P(j)\right)}}, \tag{32}$$

where U and C represent the predicted partition and the ground-truth partition, and $P(i,j)$ denotes the probability that a sample belongs to both the cluster U_i in U and the cluster C_j in C. It is easy to check that NMI ranges from 0 to 1.

RI views the clustering as a series of decisions. A true positive (TP) decision assigns two similar samples to the same cluster, and a true negative (TN) decision assigns two dissimilar samples to different clusters. The false positive (FP) and false negative (FN) have the similar definition. RI [59] is defined as:

$$\mathrm{RI} = \frac{2(TP+TN)}{N(N-2)}. \tag{33}$$

For each of the metrics, the higher it is, the better the performance is. These two metrics favor different properties in the clustering such that a comprehensive evaluation can be achieved. In all datasets, we repeat each algorithm for 10 times and report the mean results. All experiments are implemented in Matlab on a workstation with 2.4GHz CPU, 160GB RAM, and TITANX GPU (12GB caches). Source codes accompanying this paper can be achieved at: https://drive.google.com/open?id=1KtJSBeJTo-SbKiPJyTf_DewbO4Eyu3GG

5.4 Univariate Time Series Clustering

In this subsection, the proposed T-MEK-SPL is first compared with the baseline and three state-of-the-art MKC methods for UTS clustering. Then, we compare our proposal with its two variants to prove the effectiveness of the multiple kernel fusion and self-paced learning. At last, further analysis for the proposal are carried out. Here, we use three commonly used kernels, namely, DTW, ERP and SBD.

5.4.1 Comparison of T-MEK-SPL with MKC methods and Baseline

Table 1 shows the detailed performance results for each dataset. Generally, in terms of NMI, our proposed T-MEK-SPL performs at least as well as TADPole on 46 of 48 datasets in contrast to LMKKM, RMKKM and RMKC, which perform at least as well as TADPole on 28, 31, and 28 datasets, respectively; in terms of RI, the proposal and other three MKC methods perform at least as well as TADPole on 40, 34, 34, and 31 datasets, respectively.

We further conduct the Friedman test to evaluate whether the MKC methods and the baseline are significantly different. The Friedman test is a non-parametric statistical analysis for comparison of multiple algorithms over multiple datasets. If the Friedman test rejects the null hypothesis stating that all algorithms perform equally, a post hoc Nemenyi test is conducted to evaluate the significance of the difference in average ranks. According to [55], the performance of any two algorithms is significantly different if the corresponding average ranks differ by at least the critical difference (CD):

$$\mathrm{CD} = q_\alpha\sqrt{\frac{N(N+1)}{6M}}, \tag{34}$$

where q_α is the critical value that can be found in any statistical book, N denotes the number of algorithms, and M denotes the number of datasets. In the following, statistically significant results with a 95% confidence level (i.e. $\alpha = 0.05$) is considered for all post hoc tests.

Fig. 3 shows the average rank across all 48 datasets of TADPole, LMKKM, RMKKM, RMKC and T-MEK-SPL. Through Friedman test with a significance level of 0.05, we reject the null hypothesis for both NMI and RI. On the basis of the rejection, we further conduct a post hoc Nemenyi test to evaluate the significance of the rank differences. In this case $\mathrm{CD} = 2.728\sqrt{\frac{5\times6}{6\times48}} = 0.881$. The bold line in Fig. 3 connects all methods that do not perform statistical differences according to the Nemenyi test. We can obtain two remarkable observations:

- First, the baseline TADPole and three multiple kernel clustering competitors do not present a significant difference on 48 UTS datasets.

TABLE 1
NMI and RI Results from TADPole, LMKKM, RMKKM, RMKC, M-SEK-SPL, T-MEK and T-MEK-SPL on 48 Univariate Time Series Datasets

Dataset	NMI							RI						
	TADPole	LMKKM	RMKKM	RMKC	M-SEK-SPL	T-MEK	T-MEK-SPL	TADPole	LMKKM	RMKKM	RMKC	M-SEK-SPL	T-MEK	T-MEK-SPL
Adiac	0.398	0.641	0.603	0.66	0.664	0.941	**0.97**	0.731	0.964	0.957	0.965	0.966	0.993	**0.996**
Beef	0.323	0.284	0.315	0.319	**0.344**	0.295	0.318	0.725	0.709	0.721	0.712	**0.732**	0.729	0.731
CBF	0.426	0.811	0.794	0.781	0.953	0.988	**1**	0.707	0.904	0.884	0.876	0.984	0.997	**1**
Car	0.32	0.286	0.325	0.205	0.4	**0.947**	0.929	0.672	0.704	0.712	0.617	0.724	**0.983**	0.975
ChlorineConc.	0.002	0.002	0	0	0.002	0.024	**0.847**	0.533	0.535	0.529	0.521	0.533	0.404	**0.943**
CinC-ECG-tor.	0.654	0.09	0.093	0.088	0.279	0.962	**0.993**	0.774	0.641	0.635	0.652	0.684	0.99	**0.999**
Coffee	**1**	**1**	0.573	0.544	0.693	**1**	**1**	**1**	**1**	0.786	0.778	0.865	**1**	**1**
Cricket-X	0.466	0.437	0.384	0.449	0.499	0.774	**0.997**	0.837	0.882	0.87	0.882	0.885	0.948	**1**
Cricket-Y	0.337	0.465	0.47	0.458	0.547	0.759	**0.989**	0.856	0.887	0.882	0.886	0.9	0.941	**0.998**
Cricket-Z	0.302	0.434	0.392	0.356	0.493	0.776	**0.992**	0.839	0.881	0.87	0.872	0.886	0.948	**0.999**
DiatomSizeRedu.	0.954	0.821	0.975	0.846	0.956	**1**	0.979	0.986	0.931	0.992	0.895	0.99	**1**	0.995
ECGFiveDays	0.307	0.501	0.332	0.532	0.747	0.979	**1**	0.6	0.738	0.618	0.758	0.897	0.995	**1**
FISH	0.381	0.355	0.409	0.466	0.539	**0.993**	**0.993**	0.782	0.819	0.82	0.838	0.862	**0.998**	**0.998**
FaceAll	0.716	0.819	0.781	0.793	0.845	**0.972**	**0.972**	0.918	0.959	0.928	0.949	0.96	**0.989**	**0.989**
FaceFour	0.745	0.723	0.65	0.64	0.75	**0.944**	0.925	0.906	0.878	0.805	0.797	0.888	**0.982**	0.974
FacesUCR	0.702	0.819	0.801	0.787	0.839	0.945	**0.973**	0.916	0.958	0.944	0.946	0.958	0.979	**0.994**
Gun-Point	0.044	0	0	0	0.015	0.015	**0.902**	0.499	0.497	0.497	0.498	0.507	0.503	**0.97**
Haptics	0.081	0.11	0.135	0.156	0.149	0.47	**0.914**	0.64	0.699	0.711	0.715	0.713	0.826	**0.974**
InlineSkate	0.075	0.104	0.117	0.113	0.112	0.133	**0.83**	0.751	0.762	0.757	0.751	0.732	0.762	**0.946**
InsectWingbe.	0.521	0.445	0.423	0.442	0.544	0.92	**0.979**	0.856	0.881	0.873	0.877	0.898	0.986	**0.997**
ItalyPowerDe.	0.413	0.508	0.389	0.003	0.564	0.951	**0.958**	0.697	0.776	0.724	0.501	0.819	0.989	**0.991**
Lighting2	0.174	0.149	0.173	0.149	0.142	**0.333**	0.157	0.579	0.543	0.548	0.543	0.538	**0.633**	0.548
Lighting7	0.422	0.593	0.595	0.57	0.641	**0.888**	0.83	0.75	0.837	0.836	0.836	0.846	**0.953**	0.925
MALLAT	0.901	0.923	0.966	0.944	0.974	0.98	**1**	0.949	0.977	0.99	0.978	0.994	0.995	**1**
MedicalIma.	0.228	0.294	0.288	0.304	0.306	0.66	**0.778**	0.653	0.683	0.678	0.676	0.679	0.76	**0.825**
MoteStrain	0.409	0.596	0.442	0.503	0.542	0.911	**0.928**	0.736	0.844	0.703	0.801	0.811	0.974	**0.98**
OSULeaf	0.235	0.253	0.326	0.353	0.47	0.91	**0.932**	0.752	0.759	0.753	0.767	0.799	0.977	**0.98**
OliveOil	0.529	0.634	0.685	0.68	0.656	0.868	**0.905**	0.705	0.871	0.871	0.859	0.829	0.938	**0.963**
Plane	0.958	0.931	0.928	0.961	**1**	**1**	**1**	0.987	0.975	0.965	0.984	**1**	**1**	**1**
SonyRobot	0.054	0.647	0.746	0.802	0.745	**0.984**	**0.984**	0.519	0.874	0.911	0.94	0.902	**0.997**	**0.997**
SonyRobotII	0.074	0.414	0.372	0.372	0.405	0.873	**0.914**	0.532	0.677	0.656	0.704	0.72	0.96	**0.976**
StarLightCur.	0.682	0.606	0.603	0.679	0.677	0.692	**0.748**	0.797	0.766	0.767	0.795	0.794	0.804	**0.849**
SwedishL.	0.442	0.687	0.725	0.699	0.766	0.989	**0.997**	0.723	0.937	0.932	0.925	0.938	0.998	**1**
Symbols	0.918	0.844	0.903	0.837	0.95	**0.982**	**0.982**	0.972	0.916	0.958	0.914	0.986	0.995	**0.996**
Trace	**1**	0.751	0.751	0.75	0.761	0.757	**1**	**1**	0.874	0.874	0.874	0.88	0.879	**1**
TwoLeadECG	0.121	0.087	0.106	0.36	0.186	**1**	**1**	0.512	0.559	0.571	0.709	0.616	**1**	**1**
Two Pat.	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
WordsSynon.	0.501	0.876	0.558	0.572	0.554	0.786	**0.878**	0.89	0.925	0.906	0.909	0.909	0.929	**0.951**
50words	0.673	0.741	0.728	0.733	0.708	0.895	**0.906**	0.946	0.96	0.959	0.96	0.958	0.972	**0.976**
Synt.Cont.	0.905	0.876	0.952	0.95	0.984	**1**	**1**	0.94	0.925	0.979	0.984	0.997	**1**	**1**
Yoga	0.002	0.002	0.005	0.001	0.005	0.003	**0.917**	0.504	0.501	0.501	0.502	0.505	0.503	**0.979**
uWaveGest.Lib.X	0.591	0.501	0.547	0.544	0.506	0.897	**0.97**	0.881	0.872	0.878	0.877	0.872	0.978	**0.995**
uWaveGest.Lib.Y	0.453	0.42	0.455	0.444	0.437	0.793	**0.938**	0.828	0.846	0.852	0.851	0.85	0.95	**0.988**
uWaveGest.Lib.Z	0.522	0.467	0.505	0.514	0.455	0.799	**0.943**	0.843	0.856	0.859	0.861	0.846	0.949	**0.989**
wafer	0	0	0	0	0	0	**0.61**	0.534	0.535	0.535	0.535	0.535	0.535	**0.899**
NonIn.Fat.ECG_Tho.1	0.607	0.815	0.806	0.807	0.818	0.987	**0.993**	0.851	0.981	0.98	0.98	0.981	0.998	**0.999**
NonIn.Fat.ECG_Tho.2	0.717	0.831	0.848	0.828	0.862	0.984	**0.993**	0.904	0.983	0.983	0.98	0.982	0.997	**0.999**
ECG200	0.154	0.138	0.242	0.112	0.341	0.66	**0.717**	0.633	0.604	0.649	0.584	0.649	0.852	**0.887**

The best result is in bold face.

Figure 3. Ranking of multiple clustering methods for 48 univariate time series datasets based on the average of their ranks across datasets in terms of (a) NMI and (b) RI. The bold line connects all measures that do not perform statistically differently according to the Nemenyi test.

- Second, among all five algorithms, T-MEK-SPL is the top measure, with an average rank of 1.156 and 1.104 in terms of NMI and RI, respectively, meaning that T-MEK-SPL performs the best in the majority of the datasets and significantly outperforms the baseline and three state-of-the-art MKC methods with clear large margins.

5.4.2 Comparison of T-MEK-SPL with Its Variants

In this subsection, we aim to experimentally prove the importance of two key components in our proposed approach: (1) the multiple kernels fusion, which integrates multiple elastic kernels via tensor low-rank constraint, and (2) the self-paced learning, which assigns samples specific weights in different kernel spaces during the clustering process.

TABLE 2
Results of Wilcoxon's Test after Comparing T-MEK-SPL, T-MEK and M-SEK-SPL Using Each of the Evaluation Metrics (NMI, RI)

	Comparison	R^+	R^-	p	p^*
NMI	T-MEK/M-SEK-SPL	1145.0	31.0	0	0
	T-MEK-SPL/T-MEK	990.5	137.5	0	0.000006
	T-MEK-SPL/M-SEK-SPL	1168.5	7.5	0	0
RI	T-MEK/M-SEK-SPL	1066.5	61.5	0	0
	T-MEK-SPL/T-MEK	1018.0	158.0	0	0.000006
	T-MEK-SPL/M-SEK-SPL	1171.5	4.5	0	0

p: exact p-value, p^*: asymptotic p-value.

To quantitatively analyze the effectiveness of these two key components, we conduct experiments with variants of our approach, which replace the tensor low-rank constraint with matrix low-rank constraint when only one elastic kernel is used (M-SEK-SPL) or discard the self-paced learning regularizer (T-MEX). Note that, for each dataset, M-SEK-SPL is performed on each kernel separately and the best result is reported. To evaluate the significance of the results in Table 1, following [55], we use the Wilcoxon test to statistically analyze the results via pairwise comparisons. Table 2 shows the statistical results for the 48 datasets, where the p-values, the asymptotic p-value (p^*), the sum of the ranks in favor of the first algorithm (R^+) and the sum of the ranks in favor of the second algorithm (R^-) are given. The p-values determine whether two algorithms are significantly different. In this study, we consider a difference to be significant at $p < 0.05$.

From the results in Table 2, we can obtain three key findings: 1) T-MEK-SPL outperforms M-SEK-SPL with statistical significance, which proves that the integration of various elastic kernels can achieve promising results for time series clustering; 2) Compared with T-MEK, the superior performance achieved by T-MEK-SPL verify the importance of differently treating samples during clustering process; 3) Despite treating all elastic kernels equally, T-MEK still elevates the performance compared to M-SEK-SPL, which further confirm the effectiveness of multiple elastic kernels fusion.

5.4.3 Deep Insight for T-MEK-SPL

The results presented above have mainly proven the effectiveness of two key components. Nevertheless, how the multiple elastic kernel fusion and the self-paced learning improve the performance is also of interest, and deserve further exploration and discussion. In this subsection, we will analyze our approach in more detail.

In Fig. 4, we present the kernel-specific confusion matrices and the T-MEK-SPL confusion matrix for the dataset *FacesUCR*, where the row and column numbers are true and predicted labels respectively. As we can see, three elastic kernels could provide complementary information for each other in a great way. For instance, although ERP kernel is more suitable for *FacesUCR* clustering, it does not always perform well since mistakenly confuses 10-th class into 3-th class. Under such circumstances, the additional information provided by DTW kernel could help improve the clustering performance. The same situation exists when clustering

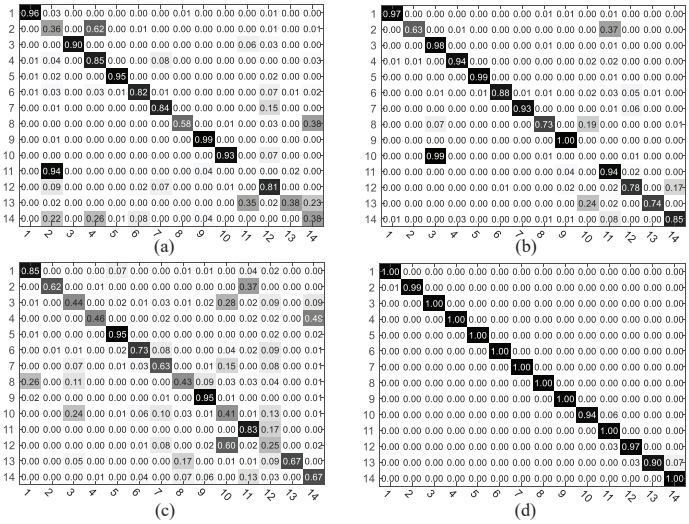


Figure 4. Comparison of the confusion matrices on *FacesUCR* among the kernel-specific M-SEK-SPL with respect to (a) DTW kernel, (b) ERP kernel, (c) SBD kernel and (d) the proposed T-MEK-SPL.

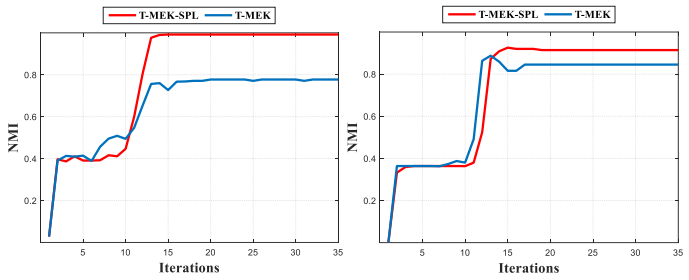


Figure 5. The iteration process comparison of T-MEK and T-MEK-SPL in terms of NMI on datasets (a) *Cricket-Z* and (b) *SonyAIBORobotSurfaceII*.

with DTW kernel which makes confusion between 11-th class and 2-th class. However, the elevated performance indicates that the complementary information provided by ERP and SBD kernels can be captured and propagated in high-order tensor space through our proposed T-MEK-SPL.

As illustrated by Table 2 and Table 1, T-MEX-SPL performs significantly better than T-MEK in terms of both NMI and RI, especially for several challenging datasets such as *ChlorineConcentration* and *Gun-Point*. The reason is that the time series datasets usually suffer from noisy samples, which may degenerate the model performance. However, by introducing the self-paced learning, T-MEK-SPL will emphasize the high-confidence fidelity samples and suppress the low-confidence noisy samples at the early stage of the whole clustering process. In such case, T-MEK-SPL can better capture the spatial structure information of samples distribution, and eventually learn a more stable and discriminative feature subspace. This effect can be shown in Fig.5, where T-MEK-SPL may be slightly worse than T-MEK at early few iterations (i.e., the model with younger age), whereas with the increasing iteration step, T-MEK-SPL would eventually obtain superior performance.

5.5 Multivariate Time Series Clustering

In this subsection, in order to test whether our proposal is also suitable to MTS clustering problem, the experiments

TABLE 3
Summary Description of the Multivariate Time Series Datasets

Data Sets	Variables	Max Length	Min Length	Classes	Size	Sources
Australian language	13	93	4	10	8800	UCI
JapaneseVowel	12	29	7	9	640	UCI
Character trajectories	3	205	109	20	2858	UCI
CMU subject 16	62	580	127	2	58	CMUMC
ECG	2	152	39	2	200	Olszewski
Libras	2	45	45	15	360	UCI
Non-invasive fetal ECG	2	750	750	42	3765	UCR
Pen digits	2	8	8	10	10992	UCI
uWaveGestureLibrary	3	315	315	8	4478	UCR
Wafer	6	198	104	2	1194	Olszewski

TABLE 4
NMI and RI Results from LMKKM, RMKC, RMKKM, M-SEK-SPL$_c$, M-SEK-SPL, T-MEK and T-MEK-SPL for the Multivariate Time Series Datasets

Dataset	NMI							RI						
	LMKKM	RMKC	RMKKM	M-SEK-SPL$_c$	M-SEK-SPL	T-MEK	T-MEK-SPL	LMKKM	RMKC	RMKKM	M-SEK-SPL$_c$	M-SEK-SPL	T-MEK	T-MEK-SPL
Charac.Traj.	0.616	0.782	0.809	0.904	0.737	0.972	**0.98**	0.945	0.962	0.965	0.985	0.959	0.996	**0.997**
CMU16	0.023	0.006	0.127	0.127	**0.385**	0.167	0.19	0.492	0.492	**0.758**	**0.758**	0.71	0.501	0.61
ECG	0.002	0.051	0.047	0.236	0.258	0.525	**0.555**	0.504	0.526	0.566	0.672	0.672	**0.78**	**0.78**
Japan.Vowel	0.105	0.151	0.131	0.15	0.256	0.888	**0.971**	0.802	0.812	0.744	0.812	0.801	0.971	**0.993**
Libras	0.315	0.326	0.309	0.554	0.489	0.723	**0.844**	0.891	0.891	0.874	0.912	0.902	0.933	**0.962**
Wafer	0.007	0.137	0.169	0.141	0.227	0.089	**0.304**	0.501	0.581	0.582	0.58	0.683	0.517	**0.689**
NonIn.Fat.ECG	0.714	0.801	0.816	0.835	0.819	0.984	**0.991**	0.971	0.979	0.978	0.983	0.979	**0.998**	**0.998**
uWaveGest.Lib.	0.63	0.713	0.788	0.795	0.519	0.961	**0.994**	0.898	0.915	0.951	0.951	0.867	0.993	**0.999**
Austra.Lang.	0.499	0.778	0.706	0.456	0.629	0.95	**0.969**	0.976	0.989	0.982	0.98	0.972	0.997	**0.998**
PenDigits	0.534	0.701	0.742	0.722	0.467	0.93	**0.991**	0.886	0.915	0.937	0.935	0.872	0.989	**0.998**

The best result is in bold face.

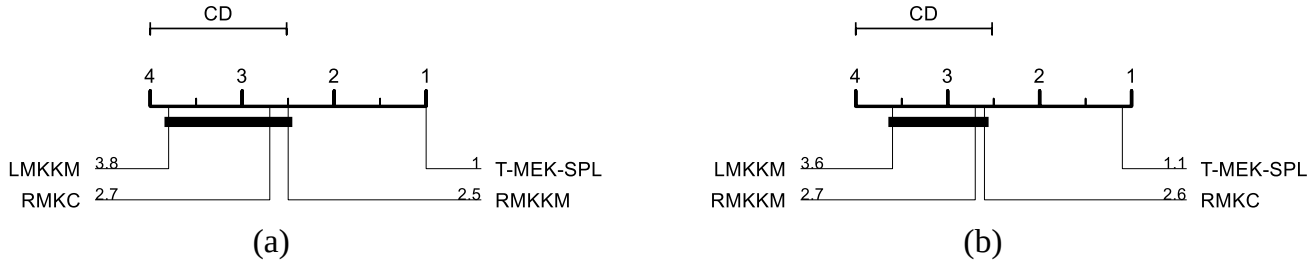


Figure 6. Ranking of multiple clustering methods for multivariate time series based on the average of their ranks across datasets in terms of (a) NMI and (b) RI. The bold line connects all measures that do not perform statistically differently according to the Nemenyi test.

are carried out on 10 multivariate time series datasets, which are widely used and publicly available. The detailed information is shown in Table 3. Since SBD is highly computationally efficient and competitive in terms of accuracy compared with DTW, we use it to acquire kernel matrix for each variate.

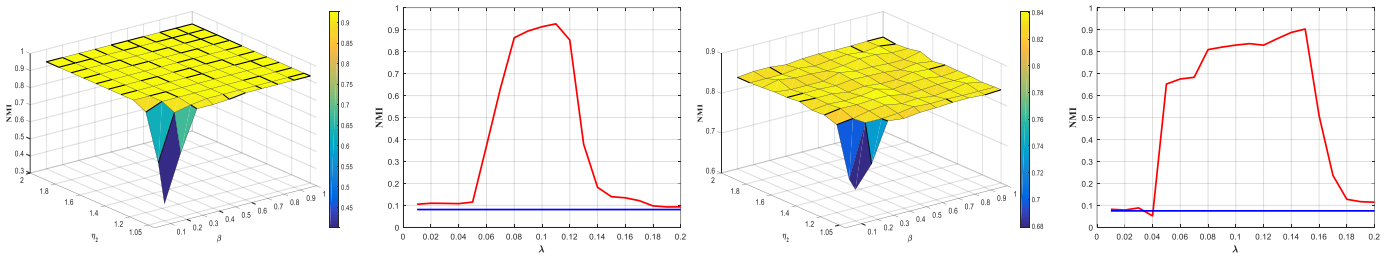
Table 4 presents the detailed experimental results of different methods. Fig. 6 shows the average rank across all 10 datasets of LMKKM, RMKKM, RMKC, and T-MEK-SPL. The null hypothesis for both NMI and RI are rejected via Friedman test with a significant level of 0.05. According to the Nemenyi test, the bold line in Fig. 6 connects all methods that do not perform statistical differences. The statistical results indicate that the proposed method significantly outperforms the MKC rivals.

In addition to M-SEK-SPL and T-MEK, for MTS clustering, we implement one additional variant denoted as M-SEK-SPL$_c$, which first concatenate all variables and then carry out M-SEK-SPL. Table 5 shows the statistical results of Wilcoxon test for the 10 datasets. We consider a difference to be significant at $p < 0.05$. Except for the conclusions similar to UTS clustering, we also find that T-MEK-SPL achieves significantly superior performance in terms of NMI compared with M-SEK-SPL$_c$. The reason behind this is that the operation of concatenating all variables corrupts the data structure, and fails in utilizing the complementary information from multiple variates.

TABLE 5
Results of Wilcoxon's Test after Comparing T-MEK-SPL with T-MEK, M-SEK-SPL and M-SEK-SPL$_c$ Using Evaluation Metrics NMI and RI

	Comparison	R^+	R^-	p	p^*
NMI	T-MEK-SPL/M-SEK-SPL	52.0	3.0	0.0098	0.0108
	T-MEK-SPL/T-MEK	55.0	0	0.0020	0.0043
	T-MEK-SPL/M-SEK-SPL$_c$	55.0	0	0.0020	0.0043
RI	T-MEK-SPL/M-SEK-SPL	49.0	6.0	0.0273	0.0249
	T-MEK-SPL/T-MEK	53.5	1.5	0.0049	0.0062
	T-MEK-SPL/M-SEK-SPL$_c$	46.0	9.0	0.0645	0.0528

p: exact p-value, p^*: asymptotic p-value.



(a) (b) (c) (d)

Figure 7. Parameter analysis (η_2, β and λ) in terms of NMI on (a)(b) *Haptics* and (c)(d) *InlineSkate* datasets.

5.6 Model Analysis

In this subsection, we conduct further analysis and experiments to understand the characteristics of our T-MEK-SPL approach better.

5.6.1 Parameter Sensitivity Analysis

In the proposed algorithm, it appears that three parameters i.e., λ for the error term, the initial value β and the learning rate η_2 for self-paced learning term, may have the influence on time series clustering performance. Actually, only one key parameter λ needs to be tuned, and we set $\beta = 0.5$ and $\eta_2 = 1.05$ for all UTS and MTS datasets.

Fig. 7 shows the NMI results on *Haptics* and *InlineSkate* datasets by using different values of the parameters. As shown in Fig. 7(a)(c), when the β is small, with the increasing η_2, the NMI firstly increases and then keeps steady. Whereas as the β being bigger than 0.3, the η_2 has little effect on the performance. Since β represents the age when the model begins to learn, and η_2 denotes the learning rate, we set $\beta = 0.5$ and $\eta_2 = 1.05$ with clear physical meaning, in that *we want the model to learn new knowledge step by step at the appropriate age for admission*.

Fig. 7(b)(d) indicate that λ plays a key role in the model. It is infeasible to determine the optimal value for λ analytically, since it highly depends on the domain prior knowledge of the datasets. However, we empirically find that, when λ is located within the range [0.05, 0.1], the proposed model can achieve significantly better performance than the baseline on most datasets. The results of varying λ on *Haptics* and *InlineSkate* datasets are present in Fig. 7(b)(d), where the blue horizontal lines denote the baseline indexes. More detailed parameter analysis for λ can be found in the supplementary materials. Despite such excellent performance, it is still a worthwhile research direction to adaptively determine the optimal parameters of the model. Recently, Dau et al [22] propose a novel semi-supervised approach to learn parameter value, which can be coupled with our proposal promisingly. Related research will be carried out in our future work.

5.6.2 Computational Complexity and Convergence

The computational costs for the proposed approach mainly include three parts: the calculation of elastic kernel matrix, the optimization of the proposed approach and the operation of spectral clustering.

Among all K elastic kernels used, we denote the highest complexity as C_1. The computational complexity of K elastic kernel matrix is $\mathcal{O}(KC_1)$. For both DTW and ERP, the time complexity of calculating the distance between two different length sequences is $\mathcal{O}(m_1^2)$, where m_1 denotes the length of the longer sequence. SBD has a relatively low complexity, and takes $\mathcal{O}(m_1 \log(m_1))$. Hence, in this study, the highest complexity C_1 for a dataset with N samples is $N^2 m_1^2$ for UTS clustering, and $N^2 m_1 \log(m_1)$ for MTS clustering. Note that the elastic kernel matrix can be calculated in advance and the procedure only performs once.

The bottleneck of the optimization lies in solving the subproblem $\boldsymbol{\mathcal{U}}$, which equals to operate $(N-1)/2$ matrix SVD decomposition separately in the Fourier domain, with each matrix dimension being $K \times N$. Due to the special structure, the optimization of subproblem $\boldsymbol{\mathcal{U}}$ can be easily parallelized. In general, in each iteration, it takes $\mathcal{O}(2N^2 K \log(N))$ for calculating FFT and the inverse, and $\mathcal{O}(\min(N^2 K, NK^2))$ for calculating each matrix SVD decomposition. Since in multiple kernel clustering setting we have $N \gg K$ and $\log(N) > K$, the complexity of optimization procedure for each iteration is:

$$\mathcal{O}(2N^2 K \log(N)), \tag{35}$$

In summary, the computational complexity of T-MEK-SPL is:

$$\mathcal{O}(N^3) + \mathcal{O}(KC_1) + \mathcal{O}(T(2N^2 K \log(N))), \tag{36}$$

where the first term is the cost of spectral clustering, T is the iteration number of convergence. The convergence of our procedure can be indicated by the match error (E_m) and reconstruction error (E_r), which are defined as:

$$E_m = \|\mathbf{Z}^k - \mathbf{U}^k\|_\infty, \tag{37}$$

$$E_r = \|\mathbf{I} - \mathbf{Z}^k - \mathbf{Q}^k\|_\infty. \tag{38}$$

Actually, as illustrated in Fig. 5, the proposed T-MEK-SPL converges within 40 iterations for most datasets.

6 Conclusions and Future Work

In this paper, a novel multiple elastic kernels clustering model named T-MEK-SPL is proposed to simultaneously address several challenges in time series clustering, i.e., high-dimension, warping, and the integration of multiple elastic measures. To overcome the high-dimensional issues and capture the complementary information from multiple elastic distance measures, the proposal constraints the rotated self-representation coefficient tensor, which is obtained from multiple elastic kernel spaces, via tensor nuclear norm.

In addition, to acquire more stable and discriminative subspace structure, the SPL theory is introduced into the proposed model to gradually involve the faithful samples from easy to hard in the clustering process. The effectiveness of the above key components are experimentally demonstrated by extensive self-comparisons. Moreover, based on the statistical analysis, the experiments results on 85 UTS datasets and 10 MTS datasets reveal that T-MEK-SPL significantly outperforms the baseline and three state-of-the-art multiple kernels clustering methods.

Future research will include the following: 1) the adaptive learning approach for the critical parameter λ; 2) more elaborate consideration of time series characteristic, e.g., the correlation between two adjacent timestamps; and 3) exploring parallelizable algorithms.

ACKNOWLEDGMENTS

The authors are thankful for the financial support in part by the National Natural Science Foundation of China (61432008, 61472423, 61602484, U1636220 and 61772524); by the Beijing Municipal Natural Science Foundation under Grant 4182067; by the Fundamental Research Funds for the Central Universities associated with Shanghai Key Laboratory of Trustworthy Computing. The authors especially want to express the gratitude to Professor Keogh and collaborators for creating the UCR time series classification archive, which has promoted rapid development in time series analysis field.

REFERENCES

[1] P. Esling and C. Agon, "Time-series data mining," *ACM Comput. Surveys*, vol. 45, no. 1, pp. 1-34, Nov. 2012.
[2] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering-A decade review," *Inf. Syst.*, vol. 53, pp. 16-38, Oct. 2015.
[3] A. Khaleghi, D. Ryabko, J. Mary, and P. Preux, "Consistent algorithm for clustering time series," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 94-125, 2016.
[4] H. P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 1, pp. 1-58, Mar. 2009.
[5] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining Knowl. Discovery*, vol. 26, no. 2 pp. 275-309, Mar. 2013.
[6] U. Mori, A. Mendiburu, and J.A. Lozano, "Similarity measure selection for clustering time series databases," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp.181-195, Jan. 2016.
[7] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: the collective of transformation-based ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2522-2535, Sep. 2015.
[8] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: the collective of transformation-based ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2522-2535, Sep. 2015.
[9] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining Knowl. Discovery*, vol. 29, no. 3, May 2015.
[10] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," *Proc. IEEE Int .Conf. Comput. Vis.*, pp. 4238-4246, 2015.
[11] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1582-1590, Dec. 2015.
[12] Y. Xie, D. Tao, W. Zhang, L. Zhang, Y. Liu, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, 2016.
[13] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-Laplacian Regularized Multilinear Multiview Self-Representations for Clustering and Semisupervised Learning," *IEEE Trans. Cybern.*, 2018.
[14] D. E. Zhuang, G. C. Li, and A. K. Wong, "Discovery of temporal associations in multivariate time series," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2969-2982, 2014.
[15] P. F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 306-318, Mar. 2009.
[16] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43-49, Feb. 1978.
[17] D. F. Silva and G. E. Batista, "Speeding up all-pairwise dynamic time warping matrix calculation," in *Proc. SIAM Int. Conf. Data Min.*, pp. 837-845, 2016.
[18] G. Al-Naymat, S. Chawla, and J. Taheri, "SparseDTW: A novel approach to speed up dynamic time warping," in *Proc. 8th Australasian Data Min. Conf.*, vol. 101, pp. 117-127, 2009.
[19] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 569-573, 2016.
[20] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, pp.1855-1870, 2015.
[21] N. Begum, L. Ulanova, J. Wang, and E. Keogh, "Accelerating dynamic time warping clustering with a novel admissible pruning strategy," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, pp. 49-58, Aug. 2015.
[22] H. A. Dau, D. F. Silva, F. Petitjean, G. Forestier, A. Bagnall, A. Mueen, and E. Keogh, "Optimizing dynamic time warping's window width for time series data mining applications," *Data Mining Knowl. Discovery*, vol. 32, no. 1, pp. 1074-1120, Jul. 2018.
[23] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proc. 30th Int. Conf. Very Large Databases*, pp. 792-803, 2004.
[24] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1354-1369, Jul. 2014.
[25] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, pp. 2149-2155, 2014.
[26] P. Zhou, L. Du, L. Shi, H. Wang, and Y. D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Int. Joint Conf. Aritificial Intelligence*, pp. 4105-4111, Jul. 2015.
[27] Y. Qu, J. Liu, Y. Xie, and W. Zhang, "Robust kernelized multi-view self-representations for clustering by tensor multi-rank minimization," *arXiv:1709.05083*, 2017.
[28] H. C. Huang, Y. Y. Chuang, and C. S. Chen, "Multiple kernel fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120-134, Feb. 2012.
[29] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y. D. Shen, "Robust multiple kernel k-means using $l_{2,1}$-norm," in *Int. Joint Conf. Aritificial Intelligence*, pp. 3476-3482, Jul. 2015.
[30] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 1305-1313, 2014.
[31] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-Means with incomplete kernels," in *Proc. AAAI Conf. Artif. Intell.*, pp. 2259-2265, Feb. 2017.
[32] X. Zhu, X. Liu, M. Li, E. Zhu, L. Liu, Z. Cai, J. Yin, and W. Gao, "Localized incomplete multiple kernel k-means," in *Int. Joint Conf. Aritificial Intelligence*, pp. 3271-3277, 2018.
[33] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268-2281, Nov. 2016.
[34] F. Khan, B. Mutlu, and X. Zhu, "How do humans teach: On curriculum learning and teaching dimension," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 1449-1457, 2011.
[35] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 1189-1197, 2010.
[36] L. Jiang, D. Meng, S. I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 2078-2086, 2014.
[37] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *Proc. ACM Int. Conf. Multimedia*, pp. 547-556, 2014.

[38] C. Li, F. Wei, J. Yan, X. Zhang, Q. Liu, and H. Zha, "A self-paced regularization framework for multilabel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2660-2666, Jun. 2018.

[39] C. Xu, D. Tao, and C. Xu, "Multi-view self-paced learning for clustering," in *Int. Joint Conf. Aritificial Intelligence*, pp. 3974-3980, 2015.

[40] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, "Active self-paced learning for cost-effective and progressive face identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 7-19, Jan. 2018.

[41] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865-878, May. 2017.

[42] S. Zhou, J. Wang, D. Meng, X. Xin, Y. Li, Y. Gong, and N. Zheng, "Deep self-paced learning for person re-identification," *Pattern Recognit.*, vol. 76, pp. 739-751, Apr. 2018.

[43] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on Tensor-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3842-3849, 2014.

[44] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678-1693, Apr. 2014.

[45] M. Kilmer, K. Braman, and N. Hao, "Third order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," SIAM J. Matrix Anal. Appl., vol. 34, no. 1, pp. 148-172, 2013.

[46] M. E. Kilmer, and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641-658, 2011.

[47] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," Technical Report UILU-ENG-09-2215, UIUC (arXiv: 1009.5055), 2009.

[48] X. Ren and Z. Lin, "Linearized alternating direction method with adaptive penalty and warm starts for fast solving transform invariant low-rank textures," *Int. J. Comput. Vis.*, vol. 104, no. 1, pp. 1-14, 2013.

[49] R. L. Burden and J. D. Faires, *Numerical Analysis*. Boston, MA, USA: Cengage Learning, 2011.

[50] J. Cai, E. Cande's, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM J. Optimization*, vol. 20, no. 4, pp. 1956-1982, 2010.

[51] J. Paparrizos, and L. Gravano, "Fast and accurate time-series clustering," *ACM Trans. Database Systems*, vol. 42, no. 2, Jun. 2017.

[52] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping," *ACM Trans. Knowl. Disc. Data*, vol. 7, no. 3, Sep. 2013.

[53] Z. Chen, W. Zuo, Q. Hu, and L. Lin, "Kernel sparse representation for time series classification," *Inf. Sci.*, vol. 292, pp. 15-26, Jan. 2015.

[54] Y. Chen, E. Garcia, M. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *J. Mach. Learning Res.*, vol. 10, pp. 747-776, 2009.

[55] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1-30, 2006.

[56] T. Górecki and M. Łuczak, "Multivariate time series classification with parametric derivative dynamic time warping," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2305-2312, Apr. 2015.

[57] M. G. Baydogan and G. Runger, "Learning a symbolic representation for multivariate time series classification," *Data Min. Knowl. Disc.*, vol. 29, no. 2, pp. 400-422, Mar. 2015.

[58] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, and R. Jenssen, "Time series cluster kernel for learning similarities between multivariate time series with missing data," *Pattern Recognit.*, vol. 76, pp. 569-581, Apr. 2018.

[59] L. Hubert, and P. Arabie, "Comparing partitions," *J. Class.*, no. 2, vol. 1, pp.193-218, 1985.

[60] M. Rezaei and P. Fränti, "Set matching measures for external cluster validity," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp.2173-2186, Apr. 2016.

Yongqiang Tang is currently a PhD candidate in the Institute of Automation, Chinese Academy of Sciences (CAS). He received the BS degree from the Department of Automation, Central South University, Changsha, Hunan, China, in 2014. His research interests include machine learning and data mining.

Yuan Xie (M'14) received the PhD degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2013.

He is currently a full professor with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include image processing, computer vision, machine learning and pattern recognition. He has published around 35 papers in major international journals and conferences including the IJCV, IEEE TPAMI, TIP, TNNLS, TCYB, TCSVT, TGRS, TMM, and NIPS, CVPR, ECCV, ACM MM, etc. He also has served as a reviewer for more than 15 journals and conferences. Dr. Xie received the Hong Kong Scholar Award from the Society of Hong Kong Scholars and the China National Postdoctoral Council in 2014.

Xuebing Yang received the PhD degree from Institute of Automation, Chinese Academy of Sciences (CAS), in 2018. He is currently an assistant professor in the Institute of Automation, CAS. His research interests include machine learning, data mining and meteorological applications.

Jinghao Niu is currently a PhD candidate in the Institute of Automation, Chinese Academy of Sciences (CAS). He received his Bachelor's degree from Shandong University in 2015. His current research focuses on processing natural language text and understanding physical signals.

Wensheng Zhang received the PhD degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2000. He joined the Institute of Software, CAS, in 2001. He is a professor of machine learning and data mining and the director of Research and Development Department, Institute of Automation, CAS. His research interests include computer vision, pattern recognition and artificial intelligence.