IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 22, 2018, accepted June 10, 2018, date of publication June 13, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2847037

TUMK-ELM: A Fast Unsupervised Heterogeneous Data Learning Approach

LINGYUN XIANG[1,2], GUOHAN ZHAO[2], QIAN LI[3], WEI HAO[4], AND FENG LI[1,2]

[1]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha 410114, China
[2]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China
[3]Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia
[4]School of Traffic and Transportation Engineering, Changsha University of Science and Technology, Changsha 410114, China

Corresponding author: Wei Hao (haowei@csust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61202439, in part by the Scientific Research Foundation of the Hunan Provincial Education Department of China under Grant 16A008, and in part by the Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems under Grant 2017TP1016.

ABSTRACT Advanced unsupervised learning techniques are an emerging challenge in the big data era due to the increasing requirements of extracting knowledge from a large amount of unlabeled heterogeneous data. Recently, many efforts of unsupervised learning have been done to effectively capture information from heterogeneous data. However, most of them are with huge time consumption, which obstructs their further application in the big data analytics scenarios, where an enormous amount of heterogeneous data are provided but real-time learning are strongly demanded. In this paper, we address this problem by proposing a fast unsupervised heterogeneous data learning algorithm, namely two-stage unsupervised multiple kernel extreme learning machine (TUMK-ELM). TUMK-ELM alternatively extracts information from multiple sources and learns the heterogeneous data representation with closed-form solutions, which enables its extremely fast speed. As justified by theoretical evidence, TUMK-ELM has low computational complexity at each stage, and the iteration of its two stages can be converged within finite steps. As experimentally demonstrated on 13 real-life data sets, TUMK-ELM gains a large efficiency improvement compared with three state-of-the-art unsupervised heterogeneous data learning methods (up to 140 000 times) while it achieves a comparable performance in terms of effectiveness.

INDEX TERMS Unsupervised learning, heterogeneous data, clustering, extreme learning machine, multiple kernel learning.

I. INTRODUCTION

In most real-world data analytics problems, a huge amount of data are collected from multiple sources without label information, which is often with different types, structures, and distributions, namely heterogeneous data [1], [2]. For example, in a sentiment analysis task, the data may contain texts, images, and videos from Twitters, Facebook, and YouTube. For extracting knowledge from such big unlabeled heterogeneous data, advanced unsupervised learning techniques are required to (1) have a large model capacity/complexity, (2) have the ability to integrating information from multiple sources and (3) have a high learning speed.

Recently, many researchers enhance model capacity by combining unsupervised learning with deep learning to propose deep unsupervised learning models [3], [4]. These models inherit the powerful model capacity from deep neural networks that can reveal highly complex patterns and extremely nonlinear relations. However, most of them fail to learn from multiple sources. They are challenged by types, relations and distributions of the heterogeneous data because of the deep neural networks they used. Without strong supervised information, the deep neural networks may arbitrarily fit complex heterogeneous data that leads to meaningless solutions.

One promising way to reveal information from multiple sources is using multiple kernel learning (MKL, for short) [5], [6]. MKL first adopts multiple kernels to capture heterogeneous data characteristics from different sources. It then learns optimal combination coefficients for these kernels guided by a specific learning task. In this way, MKL can effectively capture different complex distributions by different kernels, and reveal the relations between

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

these different distributions by the kernel combination coefficients [7]–[9]. Despite the advantages of MKL, it requires supervised label information to learn the optimal kernel combination coefficients. However, label information is often not available or very costly in real big data analytics task, which limits the application of MKL.

More recently, unsupervised MKL [10]–[12] has been studied to tackle the heterogeneous data learning without supervised labels. Similar to MKL, unsupervised MKL also uses multiple kernels to distill information from various sources. To enable the learning without supervised labels, it introduces a kernel-based unsupervised learning objective, e.g. kernel k-means [13], to learn the optimal kernel combination coefficients. Although unsupervised MKL achieves remarkable performance in unsupervised heterogeneous data learning, most of the current unsupervised MKL methods are with a slow learning speed. The slow learning speed is mainly caused by the iterative numerical solution, which is adopted by these methods for optimizing the kernel combination coefficients. It does not satisfy the requirements of (1) handling a large amount of data and (2) real-time learning.

To address the above issues, we here propose a fast unsupervised heterogeneous data learning approach, namely Two-stage Unsupervised Multiple Kernel Extreme Learning Machine (TUMK-ELM, for short). TUMK-ELM iteratively extracts information from multiple sources and learns the heterogeneous data representation with closed-form solutions. It adopts multiple kernels to capture information in heterogeneous data and learns an optimal kernel for heterogeneous data representation. Different from current unsupervised multiple kernel learning methods, it seamlessly integrates a much more efficient kernel combination coefficients optimization method with an effective unsupervised learning objective that simultaneously guarantees a fast learning speed and a high learning quality. Specifically, TUMK-ELM uses the kernel k-means [13] objective function to guide the unsupervised learning process and adopts the distance-based multiple kernel extreme learning machine (DBMK-ELM, for short) [9] to learn the kernel combination coefficients. TUMK-ELM can be split into two iterative stages. At the first stage, TUMK-ELM assigns a cluster for each object in a given dataset via the kernel k-means algorithm based on multiple kernels with a set of combination coefficients. It treats the assigned cluster as the pseudo-label for each object. At the second stage, TUMK-ELM learns optimal kernel combination coefficients based on the learned pseudo-label by an analytic solution. This set of coefficients will be further used at the first stage of TUMK-ELM in the next iteration. TUMK-ELM iteratively repeats these two stages until the kernel k-means objective function is converged. Since the time complexity of each stage is small, TUMK-ELM enjoys a high speed of learning from multiple source information.

The key contributions of this work include:

- *A novel unsupervised learning method from multiple sources.* The proposed method provides an effective and efficient way to analyze multiple sources in an unsupervised fashion. It breaks out the obstacle of low learning speed for high-performance big data analytics.
- *The first fast unsupervised multiple kernel learning method.* As far as we know, the proposed method is the first fast unsupervised multiple kernel learning method. It shows a promising paradigm for the multiple kernel learning community to efficiently handling large-scale unlabeled data.
- *We prove that the proposed method is with a low time complexity and can be converged within finite steps.* The theoretical evidence guarantees the high learning speed of the proposed method in real large-scale multiple sources learning applications.

We present comprehensive experiments on 13 real-life data sets, *Haberman*, *Biodeg*, *Seeds*, *Wine*, *Iris*, *Glass*, *Image-Segment*, *Libras-Movement*, *Frogs*, *Wine-Quality*, *Statlog*, *Isolet* and *Shuttle* to evaluate our proposed TUMK-ELM method. We show that: (1) Our proposed TUMK-ELM can efficiently learn from multiple sources, which is up to 140, 000 times faster compared with the state-of-the-art methods; (2) Our proposed TUMK-ELM well captures local and global relations of objects (reflected by retrieval task), producing results substantially better than previous unsupervised multiple kernel learning methods (up to 9.71% in terms of accuracy, 12.6% in terms of NMI and 15% in terms of Purity); (3) Our proposed TUMK-ELM converges very fast (within 2 or 3 iterations); and (4) Our proposed TUMK-ELM is quite stable regarding its key parameters. The above strong evidence shows that the proposed TUMK-ELM is fit for the fast unsupervised learning from multiple sources, and we expect that it can be adopted in other unsupervised big data analytics scenarios that enable better performance.

The rest of paper is organized as follows: Section II briefly introduces the current work related to this paper. Section III explain heterogeneous data learning clearly. Section IV gives the details of the proposed TUMK-ELM. Then, Section V presents the theoretical analysis of the TUMK-ELM properties. Section VI demonstrates the performance of TUMK-ELM by comparing it with existing unsupervised multiple kernel learning algorithms. Lastly, Section VII concludes the paper and discusses future prospects.

II. RELATED WORK

This work is most related to two learning paradigms. The one is unsupervised deep learning that utilizes deep models to handle large data complexities. The other one is unsupervised multiple view learning that leverages heterogeneous information from multiple views/modes.

A. UNSUPERVISED DEEP LEARNING

Recently, lots of efforts have been done for unsupervised deep learning [14], which aims to reveal complex relations/patterns/knowledge in huge amount of data [15]–[17]. Typically, the unsupervised deep learning method combines an unsupervised objective and deep neural networks to learn a

powerful data representation [18]. For example, the methods in [19]–[21] adopt the input reconstruction as the unsupervised objective to learn an insight representation of data. To link the representation more related to analytics tasks, some methods use clustering objective and/or distribution divergence as the learning objective [22]–[24], because such objectives may induce a representation with a clearer structure. More recently, many efforts try to learn unsupervised data representation in adversarial approaches [25]–[27], which simultaneously take the advantages of both deep generator and deep discriminator. Although such unsupervised deep learning methods can capture highly complex patterns and extremely non-linear relations, they cannot learn heterogeneous data well in an unsupervised fashion. The key reason is that heterogeneous data may have much higher complexity and cause the learning methods converge at a local optimum. Without strong supervised information, the deep network may arbitrarily fit the complex heterogeneous data that leads to meaningless solution.

B. UNSUPERVISED MULTIPLE VIEW LEARNING

Unsupervised multiple view learning aims to learn heterogeneous data without supervised information [28], [29]. Among various unsupervised multiple view learning methods, unsupervised multiple kernel learning methods attract the most attention because of their ability to represent highly complex data with multimodality. The unsupervised multiple kernel learning is first proposed in [30]. After that, the work in [11] adaptively changes multiple kernel combination coefficients to better capture localized data characteristics. To enhance the robustness of the unsupervised multiple kernel learning, the work in [10] introduces a $\ell_{2,1}$-norm to regularize the space of kernel combination coefficients. More recently, [12] proposes a local kernel alignment methods to focus on local data relationships. Although the above methods achieve remarkable performance in terms of heterogeneous data representation, all of them fail to apply in big data analytics tasks due to lack of efficiency.

III. HETEROGENEOUS DATA LEARNING

In this section, we first formalize the problem and objective of heterogeneous data learning. Then, we discuss the key challenges and requirements for achieving the learning objective.

A. PROBLEM STATEMENT AND LEARNING OBJECTIVE

Let's denote a heterogeneous data set as $X = \{X_1, X_2, \cdots, X_s\}$.[1] The X contains s data sets from multiple sources and/or with multiple structures/distributions, the i-th of which is denoted as X_i. Given a heterogeneous data set X that contains n objects, the heterogeneous data learning aims to learn a representation $\hat{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} := r(X)$ for such data set, where $\mathbf{x}_i$ refers to the representation of the i-th object, and $r(\cdot)$ is the representation learning function from multiple sources. It further takes several analytics tasks, e.g. clustering, upon the representation $\hat{X}$. Typically, the representation $\hat{X}$ should contain both consensus and complementary information from multiple sources for a better analytics performance.

Without loss of generality, given a specific learning task with objective $l(\cdot)$, the objective function of heterogeneous data learning can be formalized as follows,

$$\begin{aligned} &\underset{\hat{X}}{\text{minimize}} \sum_{i=1}^{n} l(\mathbf{x}_i) \\ &\text{subject to } \mathbf{x}_i \in \hat{X} \\ &\qquad \hat{X} = r(\{X_1, X_2, \cdots, X_s\}), \end{aligned} \tag{1}$$

where n is the number of objects, and s is the number of data sources. The Eq. (1) indicates the key components of heterogeneous data learning involve a representation task and a specific learning task. To achieve a better learning performance, a heterogeneous data learning always couples the representation learning with the specific task. For example, the MKL methods use the label information of a specific task to guide the kernel combination coefficients learning.

B. LEARNING CHALLENGES AND REQUIREMENTS

Heterogeneous data learning faces several challenges in real application. These challenges may include but not limited to high dimension, complex relations, multiple structures, heterogeneous distributions, a large number of objects, and lacking supervised information. Currently, many efforts have been made on handling complex dimensional heterogeneous distributed data, e.g. using kernel methods. However, the challenges brought by a large number of objects and of lacking supervised information are not well analyzed and solved.

Heterogeneous data learning requires fast learning speed when the number of objects is large. However, leveraging heterogeneous information is an NP-hard problem [5]. Although many efforts have been done to reduce the time complexity to polynomial time [5], [7], [8], [31], these methods still need large time cost due to the iterative numerical solution. Therefore, heterogeneous data learning expects a faster optimization method for learning from heterogeneous information to handle the large complex samples in big data scenarios.

Heterogeneous data learning requires an unsupervised objective function. In most real cases, supervised label information for heterogeneous data learning is not available or with high time/human consumption. In these cases, most of current heterogeneous data learning methods do not work well since the guidelines for heterogeneous information integration is missing. How to define an unsupervised objective function that can benefit general analytics tasks is critical yet challenge.

[1]The meaning of symbol styles in this paper are as follows. Value: lowercase; vector: lowercase with bold font; matrix: uppercase with bold font; set: uppercase; function: lowercase with parentheses; space: uppercase with Calligraphic font; value index: subscript.

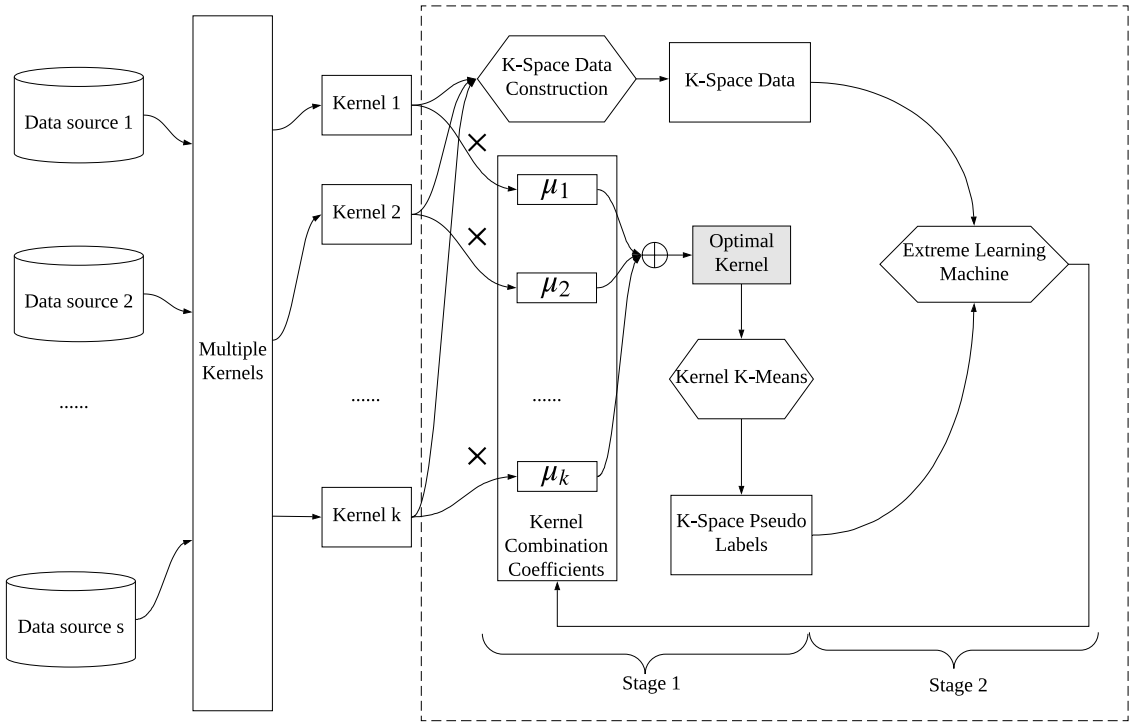


FIGURE 1. The TUMK-ELM framework. TUMK-ELM first projects heterogeneous data into kernel spaces by multiple kernels. It then adopts an iterative two stages approach to integrate heterogeneous information. At the first stage, TUMK-ELM generates a K-Space, in which the data is constructed from multiple kernel spaces and the pseudo-labels are assigned according to the learned optimal kernel. At the second stage, TUMK-ELM learns optimal kernel combination coefficients based on the generated K-Space. After convergence, the optimal kernel contains the integrated information from heterogeneous data that suits for the subsequent analytics tasks.

IV. TWO-STAGE UNSUPERVISED MULTIPLE KERNEL EXTREME LEARNING MACHINE

A. TUMK-ELM FRAMEWORK

We propose a two-stage unsupervised multiple kernel extreme learning machine (TUMK-ELM, for short) for the fast unsupervised heterogeneous data learning. TUMK-ELM captures the heterogeneous information from different sources via multiple kernels and integrates the heterogeneous information into an optimal kernel through an iterative two-stage approach guided by a general unsupervised objective. The framework of TUMK-ELM is shown in Fig. 1.

At the first stage, TUMK-ELM constructs a new data space, namely K-Space. In the K-Space, data is constructed from the multiple kernels, and pseudo-labels are assigned via kernel k-means algorithm according to a learned optimal kernel, which is built by a linear combination of the multiple kernels with learned optimal combination coefficients.

At the second stage, TUMK-ELM learns the optimal coefficients for the combination of multiple kernels. These coefficients are learned via an extreme learning machine on the data and pseudo-labels in the K-Space that constructed at the first stage. TUMK-ELM iteratively conducts these two stages until a convergence condition is satisfied. After convergence, the optimal kernel contains the integrated information from heterogeneous data that suits for following analytics tasks.

The intuitions behind TUMK-ELM are two-fold. On one hand, kernel k-means is a good unsupervised learning objective, which can induce a representation with a clear clustering structure. Specifically, kernel k-means divides data into several clusters with a maximum cut in a given kernel space. Such divide theoretically guarantees the unsupervised learning performance of TUMK-ELM. On the other hand, the extreme learning machine can effectively learn a good kernel combination with an extremely fast speed in the K-Space, which is demonstrated in [9]. It efficiently captures information from multiple sources with a closed-form solution that provides a more comprehensive description of a data set. TUMK-ELM enjoys the advantages of both kernel k-means and extreme learning machine that gains its superior performance for unsupervised heterogeneous data learning in terms of both effectiveness and efficiency.

B. FIRST STAGE OF TUMK-ELM: K-SPACE DATA CONSTRUCTION

TUMK-ELM extracts heterogeneous information from multiple sources by p kernel functions $\{k_1(\cdot), k_2(\cdot), \cdots, k_p(\cdot)\}$. These kernel functions can be designed according to prior knowledge and data characteristics. Typical functions include linear, polynomial and Gaussian kernels etc. After the kernel projection, TUMK-ELM gets a set of k base kernel matrices

$K = \{\mathbf{K}_1, \mathbf{K}_2, \cdots, \mathbf{K}_k\}$, which is used for the optimal kernel generation and K-Space data construction.

TUMK-ELM constructs K-Space data from k base kernel matrices directly. Denoting the data set in a K-Space as Z, the transformation from K to Z of a given data set X is formalized as follows,

$$\mathbf{z}_{x_i x_j} = (\mathbf{K}_{1,(x_i,x_j)}, \mathbf{K}_{2,(x_i,x_j)}, \cdots, \mathbf{K}_{k,(x_i,x_j)}), \quad \forall x_i, x_j \in X, \tag{2}$$

where $\mathbf{z}_{x_i x_j}$ is an element in Z corresponding to object x_i and x_j in X, and $\mathbf{K}_{q,(x_i,x_j)}$ refers to the (i,j)-th entry in the q-th kernel matrix.

TUMK-ELM assigns K-Space pseudo-label via kernel k-means based on an optimal kernel. The optimal kernel $\hat{\mathbf{K}}$ is generated by a linear combination of the k base kernel matrices according to a set of combination coefficients $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_k]^\top$, where μ_i is the coefficient of the i-th kernel matrix. Formally,

$$\hat{\mathbf{K}} = \sum_{i=1}^{k} \mu_i \mathbf{K}_i. \tag{3}$$

At the first iteration, TUMK-ELM initializes all combination coefficients as $\frac{1}{k}$ to generate the optimal kernel. From the second iteration, it uses the coefficients learned at the second stage to generate the optimal kernel according to Eq. (7). With the optimal kernel $\hat{\mathbf{K}}$, the kernel k-means objective function is as follows,

$$\begin{aligned} &\underset{\mathbf{C}\in\{0,1\}^{n\times n_c}}{\text{minimize}} \ \mathrm{Tr}(\hat{\mathbf{K}}) - \mathrm{Tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{C}^\top \hat{\mathbf{K}}\mathbf{C}\mathbf{L}^{\frac{1}{2}}) \\ &\text{subject to } \mathbf{C}\mathbf{1}_{n_c} = \mathbf{1}_n, \end{aligned} \tag{4}$$

where $\mathbf{C} = [c_{11}, \cdots, c_{1n_c}; \cdots; c_{n1}, \cdots, c_{nn_c}] \in \{0,1\}^{n\times n_c}$ is the indicator matrix that indicates which cluster an object belongs to, n_c is the number of clusters, $\mathrm{Tr}(\cdot)$ calculates the trace of a matrix, $\mathbf{L} = \mathrm{diag}([n_{c1}^{-1}, n_{c2}^{-1}, \cdots, n_{cn_c}^{-1}])$, $n_{cj} = \sum_{i=1}^{n} c_{ij}$ is the number of objects in the j-th clusters, and $\mathbf{1}_\ell \in \{1\}^\ell$ is a column vector with all elements being 1. Directly solving Eq. (4) is difficult since the values of $\mathbf{C}$ are limited to either 0 or 1. Alternatively, Eq. (4) can be relaxed by allowing $\mathbf{C}$ takes real values. Denoting $\mathbf{H} = \mathbf{C}\mathbf{L}^{\frac{1}{2}}$, the Eq. (4) can be reduced as

$$\begin{aligned} &\underset{\mathbf{H}\in\mathcal{R}^{n\times n_c}}{\text{minimize}} \ \mathrm{Tr}(\hat{\mathbf{K}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \\ &\text{subject to } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_{n_c}, \end{aligned} \tag{5}$$

where, $\mathcal{R}$ is a real value space, $\mathbf{I}_{n_c}$ is an identity matrix with size $n \times n_c$. The optimal $\mathbf{H}$ for Eq. (5) can be obtained by taking the n_c eigenvectors that have the n_c largest eigenvalues of $\hat{\mathbf{K}}$ [32]. The cluster of the i-th object c_i is set as the $\arg\max_j \mathbf{h}_{ij}$, where $\mathbf{h}_i$ is the i-th row of $\mathbf{H}$. After the kernel k-means clustering, TUMK-ELM assigns the pseudo-labels $t_{x_i x_j}$ for the data $\mathbf{z}_{x_i x_j}$ in the K-Space as follows.

$$t_{x_i,x_j} = \begin{cases} 0, & c_i = c_j \\ 1, & c_i \neq c_j. \end{cases} \tag{6}$$

C. SECOND STAGE OF TUMK-ELM: MULTIPLE KERNEL LEARNING

TUMK-ELM formulates the multiple kernel combination coefficients learning as a binary classification problem in the K-Space, and solves it via an extreme learning machine. For n_k K-Space data and pseudo-labels, TUMK-ELM optimizes the following objective function to calculate the optimal kernel combination coefficients.

$$\begin{aligned} &\underset{\boldsymbol{\mu}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{\mu}\|^2 + \frac{1}{2}C\sum_{i=1}^{n_k}\xi_i^2 \\ &\text{subject to } \xi_i = \mathbf{z}_i\boldsymbol{\mu} - t_i, \quad i = 1, 2, \cdots, n_k, \end{aligned} \tag{7}$$

where C is a trade-off parameter of ℓ_2 regularization and the empirical learning error, and $\mathbf{z}_i$ and t_i refer to the i-th data and pseudo-label in the K-Space, respectively.

TUMK-ELM calculates the optimal solution of Eq. (7) in a closed-form as:

$$\boldsymbol{\mu} = \left(\frac{\mathbf{I}}{C} + \mathbf{Z}^T\mathbf{Z}\right)^{-1}\mathbf{Z}^\top\mathbf{T} \tag{8}$$

or:

$$\boldsymbol{\mu} = \mathbf{Z}^\top\left(\frac{\mathbf{I}}{C} + \mathbf{Z}\mathbf{Z}^\top\right)^{-1}\mathbf{T}, \tag{9}$$

where $\mathbf{T} = [t_1, \cdots, t_{n_k}]^\top$ and $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_{n_k}]^\top$. For a large data set, TUMK-ELM can use Eq.(8) to quickly obtain the optimal solution. For data from a lot of multiple sources, TUMK-ELM prefers the Eq. (9) to calculate the optimal solution in a faster way. The learned $\boldsymbol{\mu}$ will be further used in (3) for the optimal kernel generation at the stage 1.

D. TUMK-ELM ALGORITHM

TUMK-ELM iteratively conducts the first and second stages to solve the following objective function,

$$\begin{aligned} &\underset{\mathbf{C}\in\{0,1\}^{n\times n_c},\hat{\mathbf{K}}}{\text{minimize}} \ \mathrm{Tr}(\hat{\mathbf{K}}) - \mathrm{Tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{C}^\top\hat{\mathbf{K}}\mathbf{C}\mathbf{L}^{\frac{1}{2}}) \\ &\text{subject to } \mathbf{C}\mathbf{1}_{n_c} = \mathbf{1}_n. \end{aligned} \tag{10}$$

The objective function Eq. (10) implements Eq. (1) by using kernel k-means as the unsupervised objective $l(\cdot)$ and using the extreme learning-based multiple kernel learning as the representation learning function $r(\cdot)$. It can be solved by alternatively updating $\mathbf{C}$ and $\hat{\mathbf{K}}$: (1) **Optimizing C given $\hat{\mathbf{K}}$.** By fixing $\hat{\mathbf{K}}$, $\mathbf{C}$ can be obtained via a kernel k-means clustering algorithm as shown in Eq. (5) by an eigenvalue decomposition of $\hat{\mathbf{K}}$; (2) **Optimizing $\hat{\mathbf{K}}$ given C.** With $\mathbf{C}$ fixed, $\hat{\mathbf{K}}$ can be generated by a linear combination of base kernel matrices with a set of coefficients that learned by extreme learning machine as shown in Eq. (7). TUMK-ELM adopts the change of loss value of objective function Eq. (10), denoted as Δ, as a convergence criteria. When Δ is closing to 0, i.e. is smaller than a given small threshold δ, TUMK-ELM stops the two-stage iteration and outputs the optimal kernel. If the learning task is clustering, TUMK-ELM can also output the kernel k-means clustering result directly.

Algorithm 1 explains the whole process of TUMK-ELM.

Algorithm 1 TUMK-ELM

Input: A set of heterogeneous data X, a set of kernel functions $\{k_1(\cdot), k_2(\cdot), \cdots, k_p(\cdot)\}$, the number of clusters n_c, the regularization trade-off parameter C, the convergence rate δ.

Output: An optimal kernel $\hat{\mathbf{K}}$, a cluster assignment $\mathbf{C}$.

1: Constructing base kernel matrices $K = \{\mathbf{K}_1, \mathbf{K}_2, \cdots, \mathbf{K}_k\}$ by using the input kernels to project the heterogeneous data.
2: Initializing the kernel combination coefficients $\boldsymbol{\mu}$, the loss value l' and the loss change Δ. Setting $\mu_i = \frac{1}{k}, \forall \mu_i \in \boldsymbol{\mu}, l' = +\infty$ and $\Delta = +\infty$.
3: **while** $\Delta > \delta$ **do**
4: Constructing K-Space data $\mathbf{Z}$ via Eq. (2) based on K.
5: Generating the optimal kernel $\hat{\mathbf{K}}$ via Eq. (3) based on $\boldsymbol{\mu}$ and K.
6: Assigning the kernel k-means clustering $\mathbf{C}$ generation via calculating the n_c largest eigenvalue of $\hat{\mathbf{K}}$.
7: Calculating $n_{cj} = \sum_{i=1}^{n} c_{ij}$ and $\mathbf{L} = \text{diag}([n_{c1}^{-1}, n_{c2}^{-1}, \cdots, n_{cn_c}^{-1}])$.
8: Constructing K-Space pseudo-labels $\mathbf{T}$ via Eq. (6) based on $\mathbf{C}$.
9: Learning the kernel combination coefficients $\boldsymbol{\mu}$ by Eq. (8) or (9) based on $\mathbf{Z}$, $\mathbf{T}$ and C.
10: Calculating loss value as $l = \text{Tr}(\hat{\mathbf{K}}) - \text{Tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{C}^{\top}\hat{\mathbf{K}}\mathbf{C}\mathbf{L}^{\frac{1}{2}})$.
11: Calculating loss change as $\Delta = |l' - l|$.
12: Setting $l' = l$.
13: **return** $\hat{\mathbf{K}}$, $\mathbf{C}$

V. THEORETICAL ANALYSIS OF TUMK-ELM PROPERTIES

We here theoretically analyze the learning speed of TUMK-ELM since having fast learning speed is one of the most important properties of TUMK-ELM. We first analyze the time complexity of TUMK-ELM, and then, discuss its convergence property.

A. TIME COMPLEXITY ANALYSIS

The time complexity of TUMK-ELM is mainly determined by three parts: the K-Space construction, the optimal kernel learning, and the number of iterations. For the K-Space construction, the main cost is from the kernel k-means clustering. Although the time complexity of kernel k-means is up to $O(n^3)$, it can be reduced to $O(n^2)$ by considering distributed kernel k-means [33] and $O(n)$ by considering cluster shifting [34]. For the optimal kernel learning, the time cost is determined by the calculation of Eq. (8) or Eq. (9). If Eq. (8) is adopted, the time complexity is $O(k^3 + k^2 n + kn)$. If Eq. (9) is adopted, the time complexity is $O(n^3 + kn^2 + kn)$.

Denoting the number of iterations as n_i, the time complexity of TUMK-ELM is $O(k^3 n_i + k^2 n n_i + k n n_i)$ or $O(n^3 n_i + n^2 k n_i + k n n_i)$. These two time complexities indicate Eq. (8) should be adopted when the number of objects is large, and Eq. (9) should be used when the number of sources is large for a faster learning speed. These time complexities are linear to the number of objects or the number of data sources, which theoretically guarantees the extremely fast learning speed of TUMK-ELM for heterogeneous data learning. It should be noted that n_i also affects the efficiency of TUMK-ELM. Actually, TUMK-ELM can theoretically converge within a finite step as shown in Section V-B and empirically converge within very small iterations (2-3 iterations) as demonstrated in Section VI-E.1. This fast convergence speed further supports the high efficiency of TUMK-ELM.

B. CONVERGENCE ANALYSIS

The convergence of TUMK-ELM algorithm is guaranteed by the Theorem 1.

Theorem 1: The TUMK-ELM algorithm described in Algorithm 1 can converge to a local optimal in finite steps.

Proof: Let y be the number of all possible partitions on a heterogeneous data set X. Each partition can be represented by a indicator matrix $\mathbf{C} \in \{0, 1\}^{n \times n_c}$. If two partitions are different, their indicator matrices are also different. Otherwise, they are identical. In addition, y is finite given the heterogeneous data set X and the number of cluster n_c. Therefore, there are a finite number of $\mathbf{C}$ on X. For n_i iterations, TUMK-ELM generates a series of $\mathbf{C}$, denoted as $\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_{n_i}$, and a series of $\hat{\mathbf{K}}$, denoted as $\hat{\mathbf{K}}_1, \hat{\mathbf{K}}_2, \cdots, \hat{\mathbf{K}}_{n_i}$. Given an indicator matrix $\mathbf{C}$ and an optimal kernel $\hat{\mathbf{K}}$, we denote the loss value of TUMK-ELM objective function Eq.(10) as $l_{\mathbf{C},\hat{\mathbf{K}}}$. Since kernel k-means and extreme learning machine all converge to minimal solutions, $l_{\mathbf{C},\hat{\mathbf{K}}}$ is strictly decreasing, i.e. $l_{\mathbf{C}_1,\hat{\mathbf{K}}_1} > l_{\mathbf{C}_2,\hat{\mathbf{K}}_2} > \cdots > l_{\mathbf{C}_{n_i},\hat{\mathbf{K}}_{n_i}}$. We assume that the number of iterations n_i is more than $y+1$. That indicates there are at least two same indicator matrices in the sequence, i.e., $\mathbf{C}_i = \mathbf{C}_j$, $1 \le i \ne j \le n_i$. For $\mathbf{C}_i$ and $\mathbf{C}_j$, we have the optimal kernel $\hat{\mathbf{K}}_i$ and $\hat{\mathbf{K}}_j$, respectively. It is clear that $\hat{\mathbf{K}}_i = \hat{\mathbf{K}}_j$ since $\mathbf{C}_i = \mathbf{C}_j$. Therefore, we obtain $l_{\mathbf{C}_i,\hat{\mathbf{K}}_i} = l_{\mathbf{C}_j,\hat{\mathbf{K}}_i} = l_{\mathbf{C}_j,\hat{\mathbf{K}}_j}$, i.e. the value of objective function is not change, and $\Delta = 0$. In other word, $\Delta < \delta, \forall \delta > 0$. In this case, the convergence criteria of TUMK-ELM is satisfied and the TUMK-ELM algorithm stops. Therefore, n_i is not more than $y + 1$. Hence, TUMK-ELM algorithm converges to a local minimal solution in a finite number of iterations. ■

VI. EXPERIMENTS

In this section, we compare TUMK-ELM to several state-of-the-art heterogeneous data learning methods to evaluate TUMK-ELM's performance in terms of both learning performance and learning speed. The experimental results support our above analysis that TUMK-ELM extremely improves the learning speed while achieving the better or compatible clustering accuracy compared with current multiple kernel clustering methods.

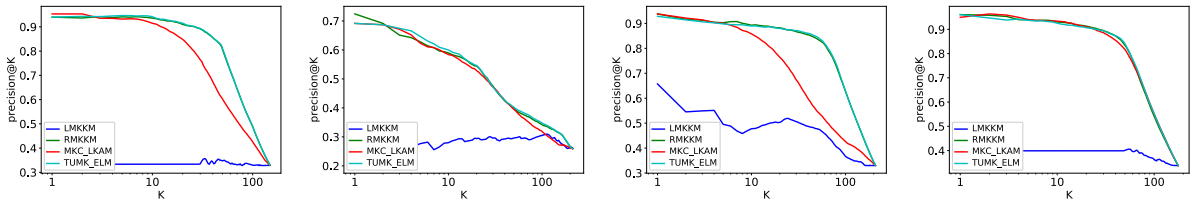


FIGURE 2. The precision@k-curve of different heterogeneous data learning methods: A better metric yields a higher curve. (a) Curve on iris data set. (b) Curve on glass data set. (c) Curve on seeds data set. (d) Curve on wine data set.

A. BENCHMARK DATA SETS

In the experiment, the benchmark data sets include *Haberman*, *Biodeg*, *Seeds*, *Wine*, *Iris*, *Glass*, *Image-Segment*, *Libras-Movement*, *Frogs*, *Wine-Quality*, *Statlog*, *Isolet* and *Shuttle*. These data sets can be accessed from UCI Machine Learning Repository [35]. The data characteristics of them are shown in Table1, in which the number of instance, classes, attributes of these sets are reported. These data sets are collected from different scenarios in real life with different characteristics, which can comprehensively evaluate our proposed TUMK-ELM from different aspects. In addition, it can also demonstrate that our algorithm can be applied in a variety of problem in practice.

TABLE 1. Summary of data sets.

Data Set	Number of Instance	Number of Attributes	Number of Class
Haberman	360	4	2
Biodeg	1055	41	2
Seeds	210	7	3
Wine	178	13	3
Iris	150	4	3
Glass	214	9	6
Image-Segment	210	18	7
Libra-Movement	360	90	15
Frogs	7195	22	4
Wine-Quality	4892	12	6
Statlog	4435	37	6
Isolet	6238	618	26
Shuttle	14500	10	6

B. EXPERIMENTS SETTING

1) PARAMETERS SETTING

For each benchmark data set, 15 kernels are adopted. These kernels include a linear kernel, three polynomial kernels of degrees $\{2, 3, 4\}$, eleven Gaussian kernels with kernel width $\{10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 1, 10^{2}, 10^{4}, 10^{6}, 10^{8}, 10^{10}\}$. For all algorithms, the regulation parameter C is set as 1. All parameters of compared algorithms are set as the recommended/default values in their corresponding papers.

2) EVALUATION CRITERIA

We evaluate the performance of TUMK-ELM in terms of information retrieval, clustering effectiveness, computational efficiency, algorithm stability and heterogeneous data representation quality.

For information retrieval, all objects are used as queries, and their k-closest objects are retrieved per learned optimal kernel. The precision@k and recall@k, i.e. the fraction of the retrieved k objects are the same-class neighbors, and the fraction of same-class neighbors are retrieved in the k objects, are reported.

The clustering effectiveness is measured by three criteria: accuracy, normalized mutual information (NMI), and purity. These criteria use the label of data in UCI repository [35] as clustering ground truth. They demonstrate the clustering effectiveness from different aspects. Specifically, accuracy measures whether the clustering results are similar to the ground truth, NMI reflects the correlation between the clustering results and the ground truth, and purity measures the percentage of samples that govern a given cluster.

The computational efficiency is measured by two criteria: algorithm time cost and convergence speed. While the algorithm time cost reflects the absolute efficiency of TUMK-ELM, the convergence speed demonstrates to what extent the efficiency of TUMK-ELM is affected by its iterative learning process.

The stability is measured by the clustering effectiveness of TUMK-ELM while varying its key parameters.

The heterogeneous representation quality is qualitatively measured by the visualization of data distributions in the learned optimal kernel space. A better representation will show a clearer structure, i.e. contains more information and with lower entropy.

C. TESTING TUMK-ELM ENABLED INFORMATION RETRIEVAL PERFORMANCE

We evaluate the heterogeneous data learning performance of TUMK-ELM in object retrieval, which is a task that heavily depends on data representation. This performance reflects whether TUMK-ELM can well integrate heterogeneous information from multiple sources. Four data sets, i.e. Iris, Glass, Seeds and Wine, are tested for TUMK-ELM-enabled retrieval performance evaluation.

The precision@k and recall@k of retrieval are used as evaluation metrics. They can demonstrate the quality of learned representation from local (when k is small) to global (when k is large). The results are shown in Fig. 2 and Fig. 3, in which the precision and recall of TUMK-ELM-enabled retrieval consistently outperform or are compatible with the others. They reflect that TUMK-ELM is able to learn heterogeneous data for analytics task.

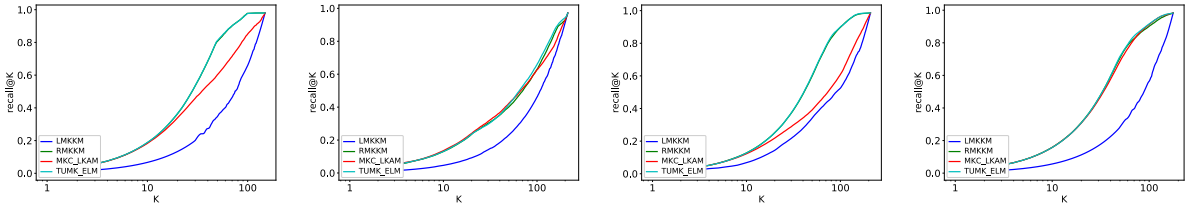


FIGURE 3. The recall@k-curve of different heterogeneous data learning methods: A better metric yields a higher curve. (a) Curve on iris data set. (b) Curve on glass data set. (c) Curve on seeds data set. (d) Curve on Wine Data Set.

TABLE 2. Accuracy of unsupervised clustering algorithms. The best results are highlighted in bold-face.

Data Set	RMKKM [10]	LMKKM [11]	MKC-LKAM [12]	TUMK-ELM
Haberman	0.5065	0.5098	0.5163	**0.5196**
Biodeg	0.5280	0.5469	0.5553	**0.5848**
Seeds	0.8905	**0.9333**	0.8575	0.9190
Wine	**0.9719**	0.9663	0.8876	**0.9719**
Iris	0.8867	0.8467	0.8187	**0.8933**
Glass	**0.4393**	0.4065	0.3464	0.4252
Image-Segment	**0.6641**	0.6619	0.6491	0.5571
Libras-Movement	0.4417	**0.5167**	0.4945	0.5028
Frogs	**0.6860**	0.5676	0.6132	0.6687
Wine-Quality	0.2178	**0.4100**	0.3739	0.3322
Statlog	0.7121	0.6808	**0.7301**	0.7061
Isolet	0.5631	NA	0.5910	**0.6186**
Shuttle	0.4491	0.3938	**0.4817**	0.3410

D. TESTING TUMK-ELM CLUSTERING EFFECTIVENESS

We evaluate the clustering effectiveness in terms of accuracy, NMI and purity, and report the results of them in Table 2, 3, and 4, respectively.

The results indicate that the clustering effectiveness of TUMK-ELM is comparable with its competitors. Although TUMK-ELM does not always achieve the best result on the benchmark data sets in terms of one metric, it can achieve the best one in terms of another metric. For example, TUMK-ELM achieves the first rank on *Biodeg* data set in terms of accuracy, but it is not the best one under the measure of NMI and purity. In addition, the difference between the performance of TUMK-ELM and its competitors is not significant. Such comparable results demonstrate our proposed TUMK-ELM can effectively capture the information from multiple sources and can enable a good clustering result.

These results are consistent with the design of TUMK-ELM and other multiple kernel clustering methods. All of these methods learn a linear combination of base kernels in an iterative way, in which the clusters are generated by the kernel k-means. Therefore, the information leveraged by these methods is similar, which leads to their similar clustering performance.

E. TESTING TUMK-ELM EFFICIENCY

We evaluate the TUMK-ELM efficiency in two aspects: convergence speed and time cost.

1) TUMK-ELM CONVERGENCE SPEED

We perform experiments on four data sets to evaluate the convergence speed of the proposed TUMK-ELM method.

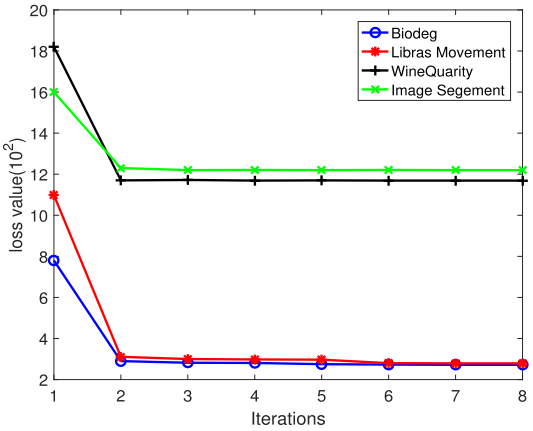


FIGURE 4. The clustering loss value of TUMK-ELM per iteration.

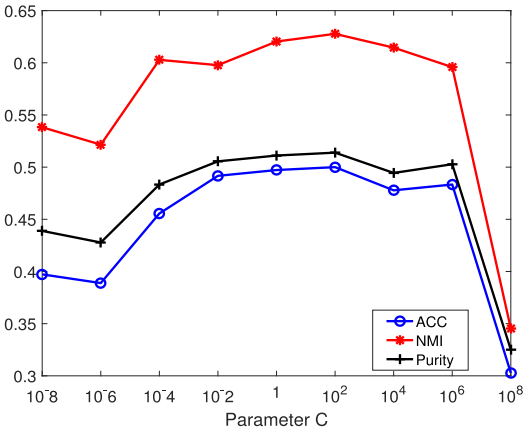


FIGURE 5. The stability of TUMK-ELM in terms of parameter C.

Specifically, we calculate the clustering loss value at step 10 of Algorithm 1 in each iteration. The convergence speed of the loss value metric reflects the convergence speed of the TUMK-ELM algorithm, which is illustrated in Fig. 4.

As shown in Fig. 4, the loss value tends to converge within four iterations on these four data sets. It demonstrates our proposed TUMK-ELM has a fast converge speed. Since TUMK-ELM can converge within few iterations, and the cost in each iteration is small (due to the closed-form solution), TUMK-ELM enjoys a good performance in terms of efficiency.

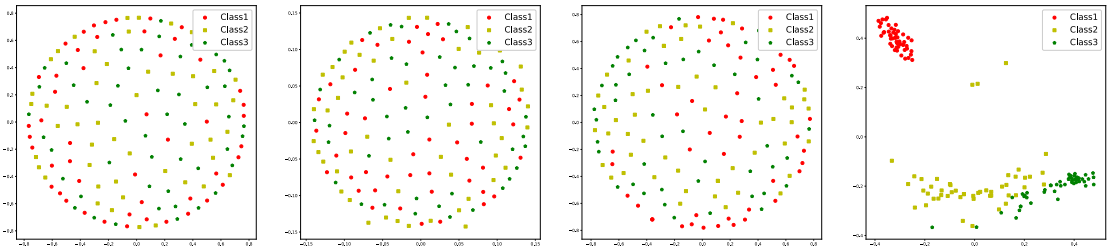


FIGURE 6. The visualization of data distribution in the optimal kernel space learned by different unsupervised heterogeneous learning methods on iris data set. These figures illustrate the data distribution in the TUMK-ELM learned optimal kernel has clearer boundaries between different clusters. The plotted two-dimensional embedding is converted from the optimal kernel by multidimensional scaling [36]. Different symbols refer to different data clusters per ground truth. (a) Distribution in the RMKKM learned optimal kernel space. (b) Distribution in the LMKKM learned optimal kernel space. (c) Distribution in the MKC-LKAM learned optimal kernel space. (d) Distribution in the TUMK-ELM learned optimal kernel space.

TABLE 3. NMI of unsupervised clustering algorithms. The best results are highlighted in bold-face.

Data Set	RMKKM [10]	LMKKM [11]	MKC-LKAM [12]	TUMK-ELM
Haberman	0.0006	**0.0018**	**0.0018**	**0.0018**
Biodeg	0.0649	0.1055	**0.1168**	0.0559
Seeds	0.6743	0.7683	0.6833	**0.7719**
Wine	**0.8897**	0.8748	0.8636	**0.8897**
Iris	0.7364	0.6504	**0.7450**	0.5693
Glass	0.2983	0.3084	0.3239	**0.3384**
Image-Segment	0.5846	0.5997	**0.6260**	0.5181
Libras-Movement	0.5300	0.5925	**0.6319**	0.6275
Frogs	0.4112	**0.4331**	0.3731	0.3826
Wine-Quality	0.0127	**0.1791**	0.0817	0.0671
Statlog	0.6121	0.4359	**0.7001**	0.5988
Isolet	**0.6676**	NA	0.5498	0.6004
Shuttle	0.1502	0.2415	**0.3419**	0.2951

TABLE 4. Purity of unsupervised clustering algorithms. The best results are highlighted in bold-face.

Data Set	RMKKM [10]	LMKKM [11]	MKC-LKAM [12]	TUMK-ELM
Haberman	**0.7353**	**0.7353**	**0.7353**	**0.7353**
Biodeg	0.6626	**0.6628**	0.6626	0.6626
Seeds	0.8905	**0.9333**	0.8952	0.8575
Wine	**0.9719**	0.9663	0.9663	**0.9719**
Iris	0.8867	0.8467	**0.8933**	0.8876
Glass	0.5234	**0.5654**	0.5327	0.5651
Image-Segment	0.6333	0.6857	**0.6871**	0.5810
Libras-Movement	0.4306	0.4889	0.5121	**0.5222**
Frogs	0.7717	0.6161	0.6920	**0.8295**
Wine-Quality	0.3772	0.3912	0.4104	**0.4891**
Statlog	0.6789	0.5410	0.6118	**0.7594**
Isolet	0.5631	NA	**0.7102**	0.6343
Shuttle	0.4111	0.6710	0.7087	**0.8912**

2) TUMK-ELM TIME COST

We further evaluate the time cost of TUMK-ELM and its competitors. We report the results in Table 5. The Table 5 indicates our proposed TUMK-ELM is much faster than its competitors on all data sets. Specifically, the learning speed of TUMK-ELM achieves as much as $1,000$ times faster than RMKKM, $140,000$ times faster than LMKKM, and $8,500$ times faster than MKC-LKAM.

TUMK-ELM can gain such low time cost because it adopts distance-based multiple kernel extreme learning machine to learn the kernel combination at step 9 of Algorithm 1. Different from other methods, which use iterative numerical solution for multiple kernel learning, TUMK-ELM inherits the efficiency of an analytic solution of DBMK-ELM. Its low time cost enables the applications on much larger data sets with real-time learning requirements. With such high learning speed, TUMK-ELM can also achieve the same clustering performance level as other multiple kernel clustering methods as demonstrated in Section VI-D, which shows the essential superiority of TUMK-ELM.

TABLE 5. Time cost of unsupervised clustering algorithms. The most efficient results are highlighted in bold-face.

Data Set	RMKKM [10]	LMKKM [11]	MKC-LKAM [12]	TUMK-ELM
Haberman	4.9566	467.4184	5.7162	**0.0233**
Biodeg	85.751	11,699.7	712.369	**0.0832**
Seeds	3.8916	168.387	1.7207	**0.0308**
Wine	2.9938	97.5415	0.9480	**0.0229**
Iris	2.0941	61.5084	0.8600	**0.0471**
Glass	4.6956	171.485	1.8842	**0.0948**
Image-Segment	3.6576	166.446	1.7355	**0.0486**
Libras-Movement	11.5414	787.7498	10.3473	**0.0956**
Frogs	165,4.4	367,12	671.93	**1.5549**
Wine-Quality	2,170.1	20,719	3,271.7	**1.2148**
Statlog	2,976.3	27,981	4,173.2	**0.8967**
Isolet	4,397.1	NA	5,192.1	**5.9381**
Shuttle	9,861.2	71,368	32,814	**6.7917**

F. TESTING TUMK-ELM STABILITY

We further evaluate the TUMK-ELM stability in terms of its key parameter C in Eq. (8) and Eq. (9). We measure the clustering performance of TUMK-ELM when varying the value of C, which is set as a value in set $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 1, 10^{2}, 10^{4}, 10^{6}, 10^{8}\}$. We illustrate the TUMK-ELM clustering performance changes on *Seeds* data set in Fig. 5.

As can be seen from Fig. 5, our proposed TUMK-ELM is relatively stable with the parameter C. In practice, we recommend setting this value between 1 to 10^2, which can enable the best results with the highest probability.

G. TESTING TUMK-ELM REPRESENTATION QUALITY

We illustrate the visualization of TUMK-ELM-represented heterogeneous data by converting it from the optimal kernel representation to a two-dimensional embedding by multidimensional scaling [36]. Fig. 6 shows the visualization of

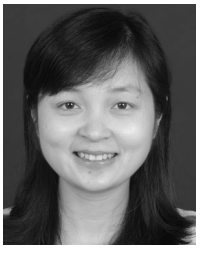
different representation methods on Iris data set. As seen in this figure, the TUNK-ELM-represented heterogeneous data has clearer boundaries between different clusters, compared with that from other methods. It qualitatively demonstrates that the TUMK-ELM-represented data is more suitable for analytics tasks, e.g. classification and clustering. This is because TUMK-ELM learns the heterogeneous data by optimizing the objective function Eq. (10), which induces a larger inter-cluster distance and intra-cluster similarity.

VII. CONCLUSION

In this paper, we have proposed a Two-Stage Unsupervised multiple kernel Extreme Learning Machines (TUMK-ELM), a more flexible algorithm for fast unsupervised heterogeneous data learning. According to the experiments, the learning speed can achieve as much as $1,000$ times faster than RMKKM, $140,000$ times faster than LMKKM, and $8,500$ times faster than MKC-LKAM. Meanwhile, the clustering accuracy of our proposed TUMK-ELM is comparable with its competitors. Experimental results clearly demonstrate the superiority of TUMK-ELM. In the future, how to adaptive adjust the base kernels to fit the dynamic heterogeneous data distributions will be considered.

REFERENCES

[1] L. Cao, "Non-IIDness learning in behavioral and social data," *Comput. J.*, vol. 57, no. 9, pp. 1358–1370, 2013.
[2] C. Zhu, L. Cao, Q. Liu, J. Yin, and V. Kumar, "Heterogeneous metric learning of categorical data with hierarchical couplings," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1254–1267, Jul. 2018.
[3] P. Bojanowski and A. Joulin. (2017). "Unsupervised learning by predicting noise." [Online]. Available: https://arxiv.org/abs/1704.05310
[4] A. Ghaderi and V. Athitsos, "Selective unsupervised feature learning with convolutional neural network (S-CNN)," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 2486–2490.
[5] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Feb. 2011.
[6] A. Kumar, A. Niculescu-Mizil, K. Kavukcoglu, and H. Daumé, "A binary classification framework for two-stage multiple kernel learning," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, U.K., 2012, pp. 1295–1302. [Online]. Available: https://arxiv.org/html/1207.4676v1
[7] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 557–569, Apr. 2013.
[8] X. Liu, L. Wang, J. Zhang, and J. Yin, "Sample-adaptive multiple kernel learning," in *Proc. AAAI*, 2014, pp. 1975–1981.
[9] C. Zhu, X. Liu, Q. Liu, Y. Ming, and J. Yin, "Distance based multiple kernel ELM: A fast multiple kernel learning approach," *Math. Problems Eng.*, vol. 2015, Nov. 2014, Art. no. 372748. [Online]. Available: https://www.hindawi.com/journals/mpe/2015/372748/
[10] L. Du *et al.*, "Robust multiple kernel K-means using $\ell_{2,1}$-norm," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 3476–3482.
[11] M. Gönen and A. A. Margolin, "Localized data fusion for kernel K-means clustering with application to cancer biology," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2014, pp. 1305–1313.
[12] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1704–1710.
[13] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel K-means: Spectral clustering and normalized cuts," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 551–556.
[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[15] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1096–1104.
[16] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 8595–8598.
[17] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.
[18] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[19] L. Deng, M. L. Seltzer, D. Yu, A. Acero, A.-R. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1692–1695.
[20] S. Chandar *et al.*, "An autoencoder approach to learning bilingual word representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1853–1861.
[21] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2360–2368.
[22] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 31–35.
[23] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
[24] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3861–3870.
[25] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[26] A. Radford, L. Metz, and S. Chintala. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434
[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[28] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 822–833.
[29] C. Xu, D. Tao, and C. Xu. (2013). "A survey on multi-view learning." [Online]. Available: https://arxiv.org/abs/1304.5634
[30] S. Yu *et al.*, "Optimized data fusion for kernel K-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.
[31] X. Liu, L. Wang, G.-B. Huang, J. Zhang, and J. Yin, "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, pp. 253–264, Feb. 2015.
[32] S. Jegelka, A. Gretton, B. Schölkopf, B. K. Sriperumbudur, and U. von Luxburg, "Generalized clustering via kernel embeddings," in *Proc. Annu. Conf. Artif. Intell.* Berlin, Germany: Springer, 2009, pp. 144–152.
[33] M. J. Ferrarotti, S. Decherchi, and W. Rocchia. (2017). "Distributed kernel K-means for large scale clustering." [Online]. Available: https://arxiv.org/abs/1710.03013
[34] M. K. Pakhira, "A linear time-complexity K-means algorithm using cluster shifting," in *Proc. Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Nov. 2014, pp. 1047–1051.
[35] M. K. Bache. (2017). *UCI Machine Learning Respository*. [Online]. Available: http://archive.ics.uci.edu/ml/index.php
[36] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. New York, NY, USA: Springer, 2005.



LINGYUN XIANG received the B.E. degree in computer science and technology and the Ph.D. degree in computer application from Hunan University, Hunan, China, in 2005 and 2011, respectively. She is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology. Her research interests include information security, steganography, steganalysis, machine learning, and pattern recognition.

GUOHAN ZHAO received the B.E. degree in electronic information engineering from Hunan Normal University, Hunan, China, in 2012. He is currently pursuing the M.S. degree in computer technology from the Changsha University of Science and Technology. His research interests include machine learning, pattern recognition, feature engineering, and convex optimization.

QIAN LI received the B.S. degree in management from the University of Science and Technology Beijing, China, in 2013, and the M.S. degree in finance from the Minzu University of China in 2016. She is currently pursuing the Ph.D. degree with the Faculty of Engineering and IT, University of Technology Sydney, Australia. Her research interests include sentiment analysis, fraud detection, social media analysis, and text mining.

WEI HAO received the Ph.D. degree from the Department of Civil and Environmental Engineering, New Jersey Institute of Technology. He is currently a Professor with the Changsha University of Science and Technology, China. His areas of expertise include traffic operations, ITS, planning for operations, traffic modeling and simulation, connected automated vehicles, and travel demand forecasting.

FENG LI received the B.E. degree from Hunan Normal University, China, in 1984, the M.E. degree from Zhejiang University, China, in 1988, and the Ph.D. degree from Sun Yat-sen University, China, in 2003. He is currently a Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. His main research interests lie in the areas of human pose estimation, computer vision, pattern recognition, and information security.