IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 22, 2018, accepted December 14, 2018, date of publication December 21, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889185

Multiple Kernel Fuzzy Clustering With Unsupervised Random Forests Kernel and Matrix-Induced Regularization

YIN-PING ZHAO[1], LONG CHEN[1], (Member, IEEE), MIN GAN[2], AND C. L. PHILIP CHEN[1], (Fellow, IEEE)

[1]Department of Computer and Information Science, University of Macau, Macau 999078, China
[2]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China

Corresponding author: Long Chen (longchen@umac.mo)

This work was supported in part by the University of Macau under Grant RC MYRG2018-00132-FST, in part by the Science and Technology Development Fund, Macau, under Grant 196/2017/A3, and in part by the National Nature Science Foundation of China under Grant 61673405 and Grant 61673155.

ABSTRACT Although kernel fuzzy clustering can handle non-spherical clusters by mapping data to a more separable feature space, its performance is highly determined by the setting of kernels. So, the multiple kernel fuzzy clustering (MKFC) is proposed to obtain the flexibility in designing an optimal kernel from a large set of candidates. In MKFC, many predefined general kernels like Gaussian and polynomial ones are linearly aggregated and the weights of kernels are adjusted automatically. However, the performance of MKFC is greatly hampered by two noticeable problems. First, MKFC only uses predefined general kernels and pays less attention to the inherent structure of specific data. This leads to the trouble of selecting proper base kernels for different data. The second problem is the ignorance of correlations between kernels in MKFC. It results in redundant kernels being used to define the feature space. This paper solves the two problems simultaneously by introducing a new MKFC model. Based on unsupervised random forests (RFs), some data-dependent kernels are generated and combined with others to build a more representative feature space. The correlations between kernels are also calculated and inserted into the objective function of fuzzy clustering as a matrix-induced regularization to encourage the diversity in kernels. We name the new model as MKFC with unsupervised RFs kernel and matrix-induced regularization. The optimization algorithm for the new model is derived, and the experiments on benchmark datasets demonstrate its superiority over other MKFC approaches.

INDEX TERMS Fuzzy clustering, multiple kernel, unsupervised random forests, regularization.

I. INTRODUCTION

For the task of data clustering, fuzzy c-means (FCM) and its variants have attracted much attention in the past decades [1]–[7]. But these methods are not effective in handling arbitrarily shaped clusters due to the use of squared Euclidean distance as the dissimilarity measure [8]. In order to solve this problem, kernel fuzzy clustering (KFC) was suggested and widely studied [8]–[12]. KFC maps original data points into a high even infinite dimensional feature space, where data points are more linearly separable. Unfortunately, the performance of KFC is greatly impacted by the setting of kernel or the design of feature space [8]. Therefore, multiple kernel fuzzy clustering (MKFC) was proposed to linearly combine a number of predefined general kernels like Gaussian and polynomial ones. The new ensemble kernel can acquire a more representative feature space, presenting the advantages in flexibility and adaptability compared with a single kernel [13]–[16].

However, the performance of MKFC is still hampered by two noticeable problems. The first one is using only predefined general kernels like Gaussian and polynomial ones in the linear combination. This brings the trouble of selecting proper candidate kernels for different data because there is no general kernel suitable to all [8]. For instance, in [14], one polynomial and seven Gaussian kernels are selected to cluster 20 machine learning datasets, while three carefully designed Gaussian kernels are applied for image data clustering. Of course, to avoid the selection of kernels, we may

directly include all available candidates in the aggregation. But this highlights the second problem of MKFC: the ignorance of correlations between different kernels in the framework of clustering. Using a large set of kernels in MKFC without the consideration of diversities can result in redundant kernels in feature space design and hence degrade the performance of MKFC.

To get the best combination of kernels, many solutions have been proposed [17]–[20]. By assigning $L_{2,1}$ norm, Du *et al.* [17] made multiple kernel k-means algorithm more robust to noise and outliers. Zhou *et al.* [18] utilized the maximum entropy method to obtain the optimal assignment of kernel weights. By combining kernels in a localized way, Gönen and Margolin [19] can get rid of sample-specific noise and capture sample-specific data features. Whereas, all these methods mentioned above did not consider the correlations between kernels. To select base kernels which are most representative and non-redundant, many methods were proposed with an unsupervised manner in the field of supervised learning. Gu *et al.* [21] proposed a representative multiple kernel learning (RMKL) method, i.e. kernel weights can be obtained by performing principal component analysis (PCA) on the predefined general kernels. To select significant kernels not all candidates, sparse MKL [22] method was proposed to acquire kernel by robust PCA. Furthermore, nonnegative matrix factorization (NMF) and kernel NMF (KNMF) can be used to obtain a good kernel combination [23]. Significant kernels obtained by such methods can be used for clustering as well. However, all these are two-stage methods [20] which search the base kernel combination independently of the clustering. The kernels selected by these methods may not be the ones that are useful for clustering.

To solve the two problems mentioned above simultaneously, a new MKFC model called multiple kernel fuzzy clustering with unsupervised random forests kernel and matrix-induced regularization (MKFC-URFK-MR) is proposed in this paper. At first, the unsupervised random forests kernel (URFK) is designed by transforming the conventional supervised random forests into an unsupervised one. Specifically, the URFK is generated from the proximity matrix of an unsupervised random forest, which reveals the underlying structure of data. URFK is a data-dependent kernel, it can alleviate our burden in choosing candidate kernels for different data. We will also show that MKFC with URFKs (MKFC-URFK) performs much better than the ones using only predefined general kernels. To build a more representative feature space, some URFKs and predefined general kernels are integrated linearly. Clearly, for such a combination with so many kernels, suppressing the correlations between kernels is of great importance. A matrix-induced regularization [24] is then inserted to the objective function of MKFC to encourage the diversity of kernels. As a result, the optimization algorithm for the regularized new model is derived. Moreover, the experiments on benchmark data demonstrate the good performance of the proposed new MKFC-URFK-MR model and its superiority over other MKFC approaches.

The contributions of this paper are summarized as follows.

(1) A type of data-dependent kernel named URFK is proposed and applied in clustering herein. URFK reveals the inherent structure of specific data, and therefore, possesses advantages of stability and adaptability over predefined general kernels.

(2) To balance the objective of clustering and the diversity of kernels, a matrix-induced regularization is introduced to consider the correlation of kernels in the framework of clustering.

(3) By combining the URFK and the diversity based regularization in MKFC, a novel model named MKFC-URFK-MR has been proposed. The optimization algorithm of the new model is derived. The new model shows better performance over other MKFC approaches when tested on benchmark data.

The rest of the paper is organized in following order. The related work is reviewed in Section II. Then in Section III, the URFK is introduced and the proposed MKFC-URFK-MR model is formulated. The optimization algorithm of MKFC-URFK-MR is also presented there. In Section IV, experimental results of MKFC-URFK-MR on benchmark data are discussed. Finally, we conclude this paper in Section V.

II. RELATED WORK

Fuzzy clustering algorithms are seeking to minimize the compactness of clusters:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ji}^{\alpha} D_{ij}^2 \tag{1}$$

where the membership value $u_{ji} \in [0, 1]$, with constraint $\sum_{j=1}^{C} u_{ji} = 1$, denotes the membership of data x_i belonging to center v_j. Fuzzification coefficient $\alpha > 1$ is a value that controls the fuzziness of clustering. To cluster well, D_{ij}^2, as dissimilarity measurement between the ith data and jth center, has been studied from different perspectives [25]–[29].

A. FUZZY C-MEANS AND KERNEL FUZZY C-MEANS

The dissimilarity measurement takes the form of $D_{ij}^2 = \|X_i - V_j\|^2$ in Fuzzy C-means (FCM). Specifically, FCM [3] groups given data into clusters by minimizing the weighted sum of Euclidean distances between data $X = \{x_1, x_2, ..., x_N\}$ and cluster centers $V = \{v_1, v_2, ..., v_C\}$. Now the objection function (1) is:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ji}^{\alpha} \|X_i - V_j\|^2 \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean distance. Centers and membership values are updated according to following rules

in FCM until convergence:

$$V_j = \frac{\sum_{i=1}^{N} u_{ji}^{\alpha} X_i}{\sum_{i=1}^{N} u_{ji}^{\alpha}} \tag{3}$$

$$u_{ji} = \frac{1}{\sum_{h=1}^{C} \left(\frac{\|X_i - V_j\|}{\|X_i - V_h\|} \right)^{\frac{2}{\alpha-1}}} \tag{4}$$

However, Euclidean distance is not helpful in nonlinear data clustering. Kernel Fuzzy C-means (KFCM) solves this problem by changing the form of dissimilarity measurement.

Mapping original data X_i from the data space χ to a high even infinite dimensional space (the feature space) H by a transform function $\phi : \chi \rightarrow H$, KFCM can acquire a satisfactory clustering result from transformed data $\phi(X_i)$ that becomes more linearly separable in the feature space. Using kernel trick [30], we only need to know kernel function $K(x, x') = \phi(x)\phi(x')^T$ explicitly. The predefined general kernels like Gaussian kernel $K(x, x') = exp(-\|x - x'\|^2/\delta^2)$ and polynomial kernel $K(x, x') = (x \cdot x' + d)^2$ are widely used in KFCM.

The loss function of KFCM is defined in the feature space:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ji}^{\alpha} \|\phi(X_i) - V_j\|^2 \tag{5}$$

Then the membership values and the clusters are updated alternatively until convergence as follows. More details can be found in [8].

$$V_j = \frac{\sum_{i=1}^{N} u_{ji}^{\alpha} \phi(X_i)}{\sum_{i=1}^{N} u_{ji}^{\alpha}} \tag{6}$$

$$u_{ji} = \frac{1}{\sum_{h=1}^{C} \left(\frac{\|D_{ij}^2\|}{\|D_{ih}^2\|} \right)^{\frac{1}{\alpha-1}}} \tag{7}$$

where

$$D_{ij}^2 = K(X_i, X_i) - \frac{2 \sum_{p=1}^{N} u_{jp}^{\alpha} K(X_p, X_i)}{\sum_{p=1}^{N} u_{jp}^{\alpha}} + \frac{\sum_{p=1}^{N} \sum_{q=1}^{N} K(X_p, X_q)}{(\sum_{p=1}^{N} u_{jp}^{\alpha})^2} \tag{8}$$

B. MULTIPLE KERNEL FUZZY CLUSTERING

MKFC offers more flexible representation space by a composite kernel which linearly combines several base kernels. The loss function of MKFC is similar to KFCM, i.e.,

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ji}^{\alpha} \|\Psi(X_i) - V_j\|^2 \tag{9}$$

The only difference between (5) and (9) is that the mapping ϕ are replaced by Ψ, which is derived from a composite kernel $K_\Psi(x, x') =< \Psi(x, x'), \Psi(x, x') >$. The composite kernel is the linear combination of several predefined base kernels, i.e. $K_\Psi = \omega_1 K_1 + \omega_2 K_2 + \ldots + \omega_m K_m$ [14], [31], where $[\omega_1, \omega_2, \cdots, \omega_m]$ are the weights of kernels.

For MKFC, the updating rules of centers and memberships are similar to the ones in (6) and (7). The difference is that the kernel function K in these equations is replaced by the combined kernel K_Ψ. The updating rule for kernel weights has different expressions depending on specific algorithms [14], [18].

III. MULTIPLE KERNEL FUZZY CLUSTERING WITH UNSUPERVISED RANDOM FORESTS KERNEL AND MATRIX-INDUCED REGULARIZATION

A. UNSUPERVISED RANDOM FORESTS KERNEL

As mentioned in the introduction section, defining a data-dependent kernel benefits the design of multiple kernel fuzzy clustering. In this section, we will build such a kernel based on random forests. But random forest (RF) is a supervised ensemble method that is constructed by growing a set of decision trees [32]–[34]. In order to apply RF on the unlabeled data, we need to transform it into an unsupervised method which is called unsupervised random forest (URF) [35].

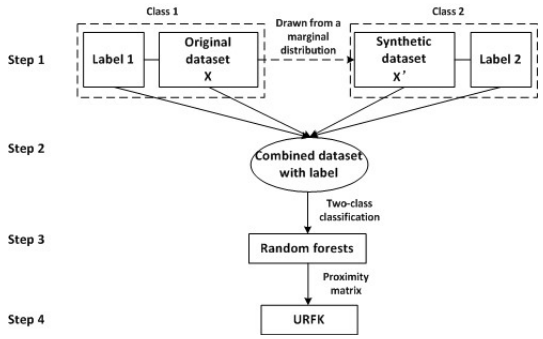


FIGURE 1. The diagram of URFK generation.

As shown in Fig. 1, the general process of URF and the production of URF kernel (URFK) include following steps:

Step 1: Generate a synthetic dataset X' from the original dataset X. Assuming that the original dataset X is a matrix with size $N \times M$, where N is the number of data and M is the number of dimensions for each data, and the synthetic dataset X' is also a matrix with size $N \times M$. The k^{th} ($k \epsilon [1, N]$) synthetic data X'_k is randomly sampled from the product of marginal distribution of variables in the original dataset X.

Step 2: Impose two different labels to the original data X and the synthetic data X', respectively. Then we concatenate these two data into a single one with size $2N \times M$.

Step 3: Classify the combined data as a binary classification problem by a random forest, taking the original data X as one class, the synthetic data X' as another class.

Step 4: A proximity matrix [36] with size $2N \times 2N$ can be produced from the random forest derived in step 3.

This proximity matrix represents the similarities between data. In the matrix, by choosing the elements belonged to the original data, a new sub-proximity matrix with size $N \times N$ can be obtained. According to the lemma below, the sub-proximity matrix is a valid kernel matrix, which is denoted as the unsupervised random forests kernel (URFK) herein. The specific procedure to generate a proximity matrix from the unsupervised random forest is also detailed in the proof.

Lemma 1: Sub-proximity matrix is a valid kernel matrix.

Proof: According to [30], to prove a matrix is a kernel one, we only need to show it is symmetrical and positive semi-definite, i.e. a function $K : \chi \times \chi \rightarrow \mathbb{R}$ on a set χ that is symmetric: $\forall(x, x') \in \chi^2, K(x, x') = K(x', x)$ and which satisfies, for all $N \in \mathbb{N}$, $(x_1, x_2, \cdots, x_N)$ and $(a_1, a_2, \cdots, a_N) \in \mathbb{R}^N : \sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j K(x_i, x_j) \geq 0$.

Sub-proximity matrix is a type of similarity matrix with size $N \times N$. For the ith and jth samples in the original data, starting from 0, if the two samples follow the same routes and terminate at the same node along one decision tree predictor in the forest, the similarity between sample i and j, i.e. $s(i, j)$ will increase by 1. The final $N \times N$ similarity values are acquired by looping all samples and all trees, and then normalized through dividing the total number of trees in the forest. In other words, $s(i, j)$ could be expressed as $s(i, j) = \frac{1}{t} v(i)v(j)^T$, where t is the number of trees in the forest and $v(i)$ is a binary vector which uses a one-hot encoding of all the leaves for the i^{th} sample. Considering all samples in the data, V is a matrix consisting of all the binary vectors. Then the sub-proximity matrix S could be expressed as $S = \frac{1}{t} VV^T$. For any $x \in \mathbb{R}^n$, we have

$$x^T Sx = \frac{1}{t} x^T VV^T x = \frac{1}{t}(V^T x)^T V^T x = \frac{1}{t}\|V^T x\|^2 \geq 0 \tag{10}$$

So the sub-proximity matrix S is symmetrical and positive semi-definite. For this reason, it is a valid kernel matrix.

As URFK is produced by comparing the original data with the synthetic one, it is a data-dependent kernel that can reveal the underlying structure of the data. In contrast to traditional predefined general kernels, URFK shows better adaptability to different data. Later, experiments on machine learning data and image data will demonstrate this point.

Combining URFK with other kernels can provide more information in feature space representation and improve the performance of MKFC. In this paper, Gaussian kernels, polynomial kernel and the proposed URFKs are linearly integrated in MKFC for all data. The specific settings of different kernels are detailed in the experiments part.

B. MKFC-URFK-MR

In previous subsection, many predefined general kernels and data-dependent kernels are combined in MKFC. To subtly adjust the weights of kernels according to the correlation information between them, a matrix-induced regularization is proposed into the objective function of multiple kernel fuzzy clustering. The novel objective function is expressed as:

$$J = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ji}^{\alpha}\|\Psi(X_i) - V_j\|^2 + \gamma \sum_{p=1}^{L}\sum_{q=1}^{L} \omega_p \omega_q M_{qp} \tag{11}$$

subject to

$$\sum_{j=1}^{C} u_{ji} = 1, \quad u_{ji} \in [0, 1];$$

$$\sum_{p=1}^{L} \omega_p = 1, \quad \omega_p \in [0, 1];$$

where $\Psi(X_i) = \sum_{p=1}^{L} \omega_p \Phi_p$, Φ_p is the mapping represented by the p^{th} kernel. ω_p is the weight of the p^{th} kernel. L is the kernel number. N is data number. C is center number. γ is a trade-off parameter which allocates the weights of kernels. u_{ji} represents the extent of ith data belonging to jth cluster. M_{qp} denotes the correlation between K_p and K_q.

In the new objective function. The first term is to encourage homogeneous elements grouped into the same cluster, while the second term is to optimize weights of kernels by a matrix-induced regularization to reduce the redundancy and enhance the diversity of selected kernels.

It is necessary to understand the importance of the second term. Let's study a case of linear combination of three kernels (K_1 K_2 K_3) in MKFC. Here we assume that high correlation M_{12} appears between K_1 and K_2 and low correlation M_{13} presents between K_1 and K_3. The high redundancy is produced when large weights are assigned to K_1 and K_2 at the same time. But this issue can be suppressed by the proposed regularization because $\omega_1\omega_2 M_{12}$ should be minimized. Similarly, low diversity may be produced when the weights of K_1 and K_3 are suppressed by constraint $\sum_{p=1}^{L} \omega_p = 1$. But the small value of M_{13} alleviates this problem because minimizing $\omega_1\omega_3 M_{13}$ in the regularization offers K_1 and K_3 the chance to take higher values.

Particularly, M_{qp} or the correlation between two kernel matrices can be defined in many ways, such as by KL divergence [37], by maximum mean discrepancy [38], by Hilbert-Schmidt independence criteria [39], to name a few. As a loose approximation of the correlation between matrices, the inner product of K_p and K_q, $Tr(K_p^{\top} K_q)$ [24], is used in the proposed model as M_{qp}.

C. OPTIMIZATION ALGORITHM

Considering the dissimilar measurement takes the form of $D_{ij}^2 = \|\Psi(X_i) - V_j\|^2$, the objective function (11) can be rewritten as follows:

$$J = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ji}^{\alpha} D_{ij}^2 + \gamma \sum_{p=1}^{L}\sum_{q=1}^{L} \omega_p \omega_q M_{qp} \tag{12}$$

We can minimize this objective function in an alternatively way.

1) OPTIMIZING MEMBERSHIPS AND CLUSTER CENTERS

At first, fixing the kernel weights W and cluster centers V, loss function (12) with Lagrange multiplier λ for the constraint $\sum_{j=1}^{C} u_{ji} = 1$ can be written as

$$J_\lambda = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ji}^{\alpha} D_{ij}^2 + \gamma \sum_{p=1}^{L}\sum_{q=1}^{L} \omega_p \omega_q M_{qp} + \lambda \sum_{j=1}^{C} u_{ji} \quad (13)$$

Take partial derivatives to memberships u_{ji} and set them to zero:

$$\frac{\partial J}{\partial u_{ji}} = \alpha D_{ij}^2 u_{ji}^{\alpha-1} + \lambda = 0$$

then,

$$u_{ji} = (-\frac{\lambda}{\alpha})^{1/(\alpha-1)} \frac{1}{D_{ij}^{2/(\alpha-1)}}$$

Considering $\sum_{j=1}^{C} u_{ji} = 1$, we have following updating rule of memberships:

$$u_{ji} = 1/\sum_{h=1}^{C} (D_{ij}^2/D_{ih}^2)^{1/(\alpha-1)} \quad (14)$$

Similarly, with fixed weights W and membership values U, taking derivatives over centers V and setting them to zero, we have

$$\frac{\partial J}{\partial V_j} = -2\sum_{i=1}^{N} u_{ji}^{\alpha}(\Psi(X_i) - V_j) = 0$$

then the updating rule of cluster centers is:

$$V_j = \frac{\sum_{i=1}^{N} u_{ji}^{\alpha}\Psi(X_i)}{\sum_{i=1}^{N} u_{ji}^{\alpha}} = \sum_{i=1}^{N} \hat{u}_{ji}\Psi(X_i) \quad (15)$$

2) OPTIMIZING WEIGHTS OF KERNELS

From equation (15), we can derive

$$\begin{aligned} D_{ij}^2 &= \|\Psi(X_i) - V_j\|^2 = (\Psi(X_i) - V_j)^T(\Psi(X_i) - V_j) \\ &= \Psi(X_i)^T\Psi(X_i) - 2\Psi(X_i)^T(\sum_{i'=1}^{N} \hat{u}_{ji'}\Psi(X_{i'})) \\ &\quad + (\sum_{i'=1}^{N} \hat{u}_{ji'}\Psi(X_{i'}))^T(\sum_{i''=1}^{N} \hat{u}_{ji''}\Psi(X_{i''})) \\ &= \sum_{p=1}^{L} \omega_p^2 K_p(X_i, X_i) - 2\sum_{i'=1}^{N}\sum_{p=1}^{L} \hat{u}_{i'j}\omega_p^2 K_p(X_i, X_{i'}) \\ &\quad + \sum_{i'=1}^{N}\sum_{i''=1}^{N}\sum_{p=1}^{L} \hat{u}_{i'j}\hat{u}_{i''j}\omega_p^2 K_p(X_{i'}, X_{i''}) \end{aligned} \quad (16)$$

Let

$$\xi_{ijp} = K_p(X_i, X_i) - 2\sum_{i'=1}^{N} \hat{u}_{i'j}K_p(X_i, X_{i'}) + \sum_{i'=1}^{N}\sum_{i''=1}^{N} \hat{u}_{i'j}\hat{u}_{i''j}K_p(X_{i'}, X_{i''}) \quad (17)$$

The objective function (12) can be rewritten as

$$J = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ji}^{\alpha}\sum_{p=1}^{L} \xi_{ijp}\omega_p^2 + \gamma\sum_{p=1}^{L}\sum_{q=1}^{L} \omega_p\omega_q M_{qp} \quad (18)$$

Let

$$\beta_p = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ji}^{\alpha}\xi_{ijp} \quad (19)$$

we have

$$J = \sum_{p=1}^{L} \beta_p\omega_p^2 + \gamma\sum_{p=1}^{L}\sum_{q=1}^{L} \omega_p\omega_q M_{qp} \quad (20)$$

with constraint

$$\sum_{p=1}^{L} \omega_p = 1, \quad \omega_p \in [0, 1]$$

Then, the weights of kernels can be acquired by solving the following quadratic programming (QP) problem with linear constraints $W^T I_L = 1$

$$\min_W J = \min_W (W^T B W + \gamma W^T M W) \quad (21)$$

where I_L is the unit vector with dimension L, $W^T = [\omega_1\ \omega_2\ \cdots \omega_L]$, $B = diag([\beta_1\ \beta_2\ \cdots \beta_p])$. Such a QP problem can be easily handled by calling corresponding solvers in software packages like Matlab.

With the new kernel weights derived from (21), the distance (16) is updated by

$$D_{ij}^2 = \sum_{p=1}^{L} \xi_{ijp}\omega_p^2 \quad (22)$$

3) FINAL ALGORITHM

Combining all the updating steps detailed above and applying URFK and other predefined general kernels as the base kernels, the final MKFC-URFK-MR algorithm is illustrated in Algorithm 1. In this algorithm, given a random membership matrix U, the related parameters in the iteration, including ξ_{ijp}, β_p, ω_p, D_{ij}^2 and U are updated according to equations (17), (19), (21), (22) and (14), respectively. The algorithm will terminate until when either the iteration counter reaches t_{max}, or the variation of U between two iterations is less than the threshold ϵ.

TABLE 1. Summary of UCI datasets.

ID	Name	Instances	Feature	Classes	Comment
S1	Breast_tissue	106	9	6	
S2	Wine	178	13	3	
S3	Glass	214	9	6	
S5	Libras	360	90	15	
S5	Australian	690	14	2	
S6	Handwritten Digits(1,7)	361	64	2	digit 1 and 7
S7	Handwritten Digits(0,6,8,9)	713	64	4	digit 0,6,8 and 9
S8	Handwritten Digits(1,2,7,9)	718	64	4	digit 1,2,7 and 9
S9	Vehicle	846	18	4	
S10	German_credit	1000	24	2	
S11	Yeast	1484	8	10	

Algorithm 1 Multiple Kernel Fuzzy Clustering With Unsupervised Random Forests Kernel and Matrix-Induced Regularization

Require:

Dataset $X = \{x_1, x_2, ..., x_N\}$; parameter α; maximum iteration number t_{max}; iteration counter $t = 1$;threshold ϵ; random membership matrix U; A set of kernel functions $\{K_p\}_{p=1}^{L}$ produced by unsupervised random forests, Gaussian kernels and Polynomial kernels.

repeat

update coefficients ξ_{ijp} according to (17);

update coefficients β_p according to (19);

update weights of kernels ω_p by solving (21);

update dissimilar measurement D_{ij}^2 according to (22);

update membership matrix U according to (14);

until $t > t_{max}$ or $\|U^t - U^{t-1}\| < \epsilon$.

return U

IV. EXPERIMENTS

In this section, a variety of experiments are conducted on UCI machine learning data [40], face image data and some noisy data. For comparison, the proposed method MKFC-URFK-MR and several typical algorithms, including Kmeans [41], FCM [3], KFCM [8] with different kernels, MKFC [14] with Gaussian kernels, termed as MKFC-G, MKFC [14] with Gaussian and Polynomial kernels, termed as MKFC, a composite kernel is predetermined by the first stage of RMKL [21] with Gaussian and Polynomial kernels, termed as RMKL, are all tested on these data. For multiple kernel based methods, the value of each kernel function is normalized to the range of [0.0001,1]. The used UCI datasets are summarized in Table 1, and the face image data contains Yale and ORL datasets [42]. For all experiments, the fuzzification degree is set to $\alpha = 1.08$; the number of clusters C equals to the ground truth of dataset; the stop threshold is given as $\epsilon = 0.0001$. Due to clustering methods depending on the initialization, each experiment runs 50 times and reports the average results. All algorithms are implemented in MATLAB on a computer with 3.5 GHz CPU and 64 GB RAM.

A. PERFORMANCE INDEXES

To evaluate the performance of clustering, two performance measures are calculated by comparing generated clusters and ground truth. The first performance measure is the clustering accuracy (ACC) which reveals the overlapping between the resulting clusters and ground truth. Let s be the total number of data points. For a data point x_i, C_i is the clustering result and G_i is the ground truth. The ACC is computed as follows:

$$ACC = \frac{1}{s}\sum_{i=1}^{s}\delta(G_i, map(C_i)) \tag{23}$$

where $\delta(s_1, s_2)$ equals 1 if $s_1 = s_2$ and $\delta(s_1, s_2)$ equals 0 otherwise. The mapping function $map(\cdot)$ is used to find the best match of generated clusters and ground truth. The Kuhn-Munkres [43] method is widely used here. If the value of ACC is closer to 1, it means that the clustering performance is better.

Another evaluation metric is the normalized mutual information (NMI). Let s be the total number of data samples. Cluster i has s_i data samples. Cluster j has s_j data samples. $s_{i,j}$ means data samples in both cluster i and j. NMI is computed as follows:

$$NMI = \frac{\sum^{i,j} s_{i,j} log(\frac{s \cdot s_{i,j}}{s_i \cdot s_j})}{\sqrt{(\sum_i s_i log\frac{s_i}{s})(\sum_j log\frac{s_j}{s})}} \tag{24}$$

The greater clustering NMI means the better clustering performance.

B. SELECTION OF BASE KERNELS

In fuzzy clustering using multiple kernels, the distances between the data points and cluster centers are evaluated by base kernel functions. To determine the optimal distance, a set of predefined general kernels frequently used by kernel methods and the proposed URFK are selected for our experiments. In this paper, we select seven Gaussian kernels for MKFC-G. An additional Polynomial kernel is added for both MKFC and RMKL. To include more data-specific information, five URFKs are added to build MKFC-URFK and MKFC-URFK-MR. For these URFKs, the number of trees T is set to 200,

TABLE 2. Description of the basic kernels.

ID	Type of kernel	Comment
K1	Gaussian $K(x_i, x_j) = exp(-\|x_i - x_j\|^2/\delta)$	$\nu = 0.1$
K2	where $\delta = min_{i,j} - \|x_i - x_j\|^2/log(\nu)$	$\nu = 0.05$
K3		$\nu = 0.01$
K4		$\nu = 0.005$
K5		$\nu = 0.001$
K6		$\nu = 0.0005$
K7		$\nu = 0.0001$
K8	Polynomial $K(x_i, x_j) = (x_i^T x_j + \vartheta)^p$	$\vartheta = 1$,$p = 2$
K9	Unsupervised random forests trees T	$T = 200$
K10		$T = 400$
K11		$T = 600$
K12		$T = 800$
K13		$T = 1000$

400, 600, 800, and 1000 respectively. For the Polynomial kernel, the parameters are set to $\vartheta = 1$ and $p = 2$. For the Gaussian kernels, the parameters ν are set to $\{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$. Details are listed in Table 2.

C. EXPERIMENTS ON UCI DATASETS

The comparison of results obtained by different algorithms on UCI machine learning data are presented in Table 3-6. ACC and NMI are used to evaluate the clustering results of different algorithms. Bold numbers denote the best results.

1) EXPERIMENTS ON URFK

The ACC comparison results of KFCM with different base kernels are illustrated in Table 3. For each dataset, there are two results produced by each algorithm, and they are the value of mean and standard deviation of ACC, respectively. The numbers in parentheses are the rankings of different algorithms. For example, the first element in the table 0.547, and 0.000 are the mean and standard deviation of ACC of KFCM with basic kernel K1, ranking the 12th among all algorithms. It is noticed that the last two rows in the table show the average ACC (mACC) and the average ranking (mRank) on all the 11 UCI datasets for each algorithm.

In general, KFCM with URFK namely K9-K13 perform better than those with predefined general kernels including K1-K8. It could be seen from Table 3 that the mACC of K9-K13 rank the 4th, 2th, 2th, 1th and 3th respectively, while that of K1-K8 are the 6th, 7th, 8th, 9th, 10th, 11th, 12th and 5th respectively. Especially, the best clustering result is generated by URFK K12, mACC and mRANK of which are both in the first place among all the kernels.

Furthermore, URFKs are more stable than predefined general kernels. Specifically, the rankings of KFCM-K11 on S6 and S11 are 1th and 6th, respectively, showing the maximum ranking range is from 1 to 6. However, that of KFCM-K1 is from 1 to 12. The comparison shows that as URFK is a data-dependent kernel, KFCM with URFK has a stable and good performance when the dataset is changed. Similarly, the observation can be found between KFCM-K5 and KFCM-K12.

A similar conclusion can be drawn in terms of NMI from Table 4. To be more intuitive, mACC and mNMI of KFCM with different basic kernels are plotted in Fig. 2. It can be shown that mACC and mNMI of K9-K13 are stable showing similar heights. Besides, mACC and mNMI of K9-K13 are larger than that of K1-K8. That is, in general, KFCM with URFK namely K9-K13 perform better than those with predefined general kernels including K1-K8.

As a short summary, URFK is a type of data-dependent kernel, so it can perform relative well on all data. On the other

TABLE 3. ACC comparison results of KFCM with different kind of single kernel on UCI datasets.

	Predefined general kernels								URFKs				
ID	KFCM-K1	KFCM-K2	KFCM-K3	KFCM-K4	KFCM-K5	KFCM-K6	KFCM-K7	KFCM-K8	KFCM-K9	KFCM-K10	KFCM-K11	KFCM-K12	KFCM-K13
S1	0.547(12)	0.550(11)	0.558(10)	0.559(9)	0.561(8)	0.568(6)	0.566(7)	0.547(12)	**0.647(1)**	0.603(4)	0.598(5)	0.609(3)	0.617(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S2	0.972(2)	0.972(2)	**0.977(1)**	0.970(4)	0.955(6)	0.955(6)	0.949(7)	0.966(5)	0.972(2)	0.966(5)	0.972(2)	0.971(3)	0.972(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S3	0.412(8)	0.412(8)	0.412(8)	0.414(7)	0.414(7)	0.417(5)	0.412(8)	0.415(6)	0.453(4)	**0.491(1)**	0.480(3)	0.481(2)	0.481(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S4	0.367(6)	0.281(7)	0.233(11)	0.241(8)	0.240(9)	0.236(10)	0.224(12)	0.456(2)	**0.463(1)**	0.429(5)	0.455(3)	0.453(4)	0.455(3)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S5	0.842(2)	**0.843(1)**	0.843(1)	**0.843(1)**	0.842(2)	0.841(3)	0.830(4)	0.842(2)	0.798(7)	0.792(9)	0.811(5)	0.806(6)	0.796(8)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S6	0.962(3)	0.752(4)	0.693(5)	0.680(6)	0.639(8)	0.640(7)	0.616(9)	**1.000(1)**	0.997(2)	**1.000(1)**	**1.000(1)**	**1.000(1)**	**1.000(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S7	0.905(3)	0.877(7)	0.629(8)	0.626(9)	0.589(10)	0.571(11)	0.515(12)	0.891(6)	0.899(5)	0.906(2)	0.906(2)	0.902(4)	**0.907(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S8	0.855(5)	0.816(7)	0.574(8)	0.530(9)	0.458(10)	0.438(11)	0.390(12)	0.855(5)	0.835(6)	0.859(4)	0.864(3)	**0.870(1)**	0.865(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S9	0.379(4)	0.379(4)	0.372(5)	0.370(6)	0.370(6)	0.367(7)	0.370(6)	0.379(4)	0.385(3)	0.393(2)	0.393(2)	**0.405(1)**	0.393(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S10	0.523(9)	0.510(10)	0.553(7)	0.553(7)	0.553(7)	0.553(7)	0.554(6)	0.534(8)	0.583(2)	**0.607(1)**	0.571(4)	0.579(3)	0.563(5)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S11	**0.350(1)**	0.349(2)	0.348(3)	0.347(4)	0.345(5)	0.345(5)	0.345(5)	0.349(2)	0.336(8)	0.342(7)	0.344(6)	0.347(4)	0.334(9)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
mACC	0.647(6)	0.613(7)	0.563(8)	0.558(9)	0.542(10)	0.539(11)	0.525(12)	0.658(5)	0.670(4)	0.672(2)	0.672(2)	**0.675(1)**	0.671(3)
mRANK	5.083(7)	5.833(8)	6.250(9)	6.583(10)	7.333(11)	7.417(12)	8.333(13)	4.833(6)	3.750(5)	3.583(4)	3.167(2)	**2.750(1)**	3.333(3)

TABLE 4. NMI comparison results of KFCM with different kind of single kernel on UCI datasets.

	Predefined general kernels								URFKs				
ID	KFCM-K1	KFCM-K2	KFCM-K3	KFCM-K4	KFCM-K5	KFCM-K6	KFCM-K7	KFCM-K8	KFCM-K9	KFCM-K10	KFCM-K11	KFCM-K12	KFCM-K13
S1	0.538(9)	0.539(8)	0.542(5)	0.540(7)	0.536(11)	0.537(10)	0.528(12)	0.541(6)	**0.566(1)**	0.551(3)	0.550(4)	0.555(2)	0.555(2)
	0.026	0.027	0.026	0.025	0.022	0.022	0.020	0.026	0.023	0.018	0.019	0.010	0.014
S2	0.893(2)	0.893(2)	**0.908(1)**	0.886(3)	0.842(9)	0.842(9)	0.823(10)	0.878(5)	0.883(4)	0.856(8)	0.883(4)	0.873(7)	0.875(6)
	0.000	0.000	0.009	0.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.000
S3	0.323(12)	0.331(10)	0.347(9)	0.353(8)	0.354(7)	0.361(6)	0.354(7)	0.324(11)	0.412(4)	0.427(2)	**0.438(1)**	0.408(5)	0.421(3)
	0.028	0.025	0.032	0.031	0.025	0.019	0.015	0.030	0.000	0.000	0.000	0.000	0.000
S4	0.498(7)	0.407(8)	0.395(10)	0.405(9)	0.392(11)	0.387(12)	0.371(13)	**0.613(1)**	0.587(4)	0.577(6)	0.583(5)	0.588(3)	0.593(2)
	0.000	0.000	0.002	0.003	0.003	0.003	0.003	0.000	0.002	0.001	0.002	0.003	0.003
S5	0.366(3)	0.369(2)	**0.370(1)**	**0.370(1)**	**0.370(1)**	0.369(2)	0.350(4)	0.366(3)	0.296(8)	0.292(9)	0.339(5)	0.335(6)	0.298(7)
	0.002	0.002	0.000	0.001	0.001	0.000	0.000	0.000	0.056	0.055	0.073	0.030	0.058
S6	0.806(3)	0.348(4)	0.262(5)	0.237(6)	0.175(7)	0.168(8)	0.129(9)	**1.000(1)**	0.975(2)	**1.000(1)**	**1.000(1)**	**1.000(1)**	**1.000(1)**
	0.000	0.101	0.092	0.081	0.074	0.077	0.074	0.000	0.000	0.000	0.000	0.000	0.000
S7	0.779(2)	0.734(6)	0.622(8)	0.629(7)	0.569(9)	0.513(10)	0.359(11)	**0.801(1)**	0.768(5)	0.779(2)	0.774(4)	0.779(2)	0.778(3)
	0.000	0.011	0.109	0.107	0.102	0.089	0.083	0.019	0.000	0.002	0.001	0.002	0.000
S8	0.672(3)	0.616(7)	0.388(8)	0.373(9)	0.331(10)	0.317(11)	0.261(12)	**0.687(1)**	0.665(6)	0.665(6)	0.671(4)	0.685(2)	0.669(5)
	0.029	0.019	0.075	0.075	0.071	0.067	0.040	0.027	0.035	0.001	0.001	0.001	0.000
S9	0.119(4)	0.118(5)	0.104(9)	0.102(11)	0.103(10)	0.102(11)	0.099(12)	0.122(3)	0.113(8)	0.116(7)	0.126(2)	**0.129(1)**	0.117(6)
	0.014	0.014	0.006	0.009	0.005	0.000	0.009	0.017	0.000	0.023	0.024	0.021	0.018
S10	0.003(6)	0.001(7)	0.010(2)	0.009(3)	0.006(4)	0.006(4)	0.005(5)	0.005(5)	**0.011(1)**	0.009(3)	0.010(2)	**0.011(1)**	0.010(2)
	0.000	0.001	0.019	0.018	0.015	0.012	0.009	0.005	0.014	0.009	0.012	0.015	0.014
S11	0.255(2)	**0.256(1)**	0.255(2)	0.254(3)	0.252(4)	0.252(4)	0.252(4)	0.255(2)	0.224(5)	0.206(7)	0.213(6)	0.201(8)	0.213(6)
	0.008	0.009	0.007	0.007	0.007	0.006	0.005	0.006	0.005	0.007	0.016	0.016	0.019
mNMI	0.477(6)	0.419(7)	0.382(8)	0.378(9)	0.357(10)	0.350(11)	0.321(12)	**0.508(1)**	0.500(4)	0.498(5)	**0.508(1)**	0.506(2)	0.503(3)
mRANK	4.917(5)	5.583(6)	5.667(7)	6.333(8)	7.750(9)	8.167(10)	9.250(11)	3.333(2)	4.333(4)	4.917(5)	**3.250(1)**	3.333(2)	3.833(3)

TABLE 5. ACC comparison results of different MKFC-based clustering algorithms on UCI datasets.

ID	Kmeans	FCM	MKFC-G	MKFC	RMKL	MKFC-URFK	MKFC-URFK-MR
S1	0.541(7)	0.542(6)	0.560(2)	0.557(4)	0.545(5)	0.558(3)	**0.617(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S2	0.961(3)	0.961(3)	0.972(2)	0.972(2)	0.961(3)	0.983(1)	**0.983(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S3	0.421(2)	0.420(3)	0.415(5)	0.405(6)	0.418(4)	0.415(5)	**0.488(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S4	0.442(4)	0.446(3)	0.260(6)	0.464(2)	0.437(5)	**0.466(1)**	0.464(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S5	0.834(4)	0.838(3)	**0.845(1)**	0.844(2)	0.838(3)	0.833(5)	0.833(5)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S6	0.941(3)	**1.000(1)**	0.743(5)	0.909(4)	**1.000(1)**	0.985(2)	0.985(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S7	0.879(3)	**0.911(1)**	0.656(5)	0.823(4)	**0.911(1)**	0.902(2)	0.902(2)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S8	0.825(5)	0.845(3)	0.626(6)	0.844(4)	0.855(2)	**0.870(1)**	**0.870(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S9	0.377(3)	0.376(4)	0.377(3)	0.377(3)	0.378(2)	**0.385(1)**	**0.385(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S10	**0.573(1)**	0.560(3)	0.561(2)	0.557(5)	0.543(6)	0.559(4)	0.559(4)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S11	**0.351(1)**	**0.351(1)**	0.346(5)	0.348(4)	0.350(2)	0.349(3)	0.348(4)
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
mACC	0.650(5)	0.659(3)	0.578(7)	0.646(6)	0.658(4)	0.664(2)	**0.676(1)**
mRANK	3.417(5)	2.833(3)	4.083(7)	3.833(6)	3.167(4)	2.500(2)	**2.083(1)**

hand, some predefined kernels such as K1 and K8, obtain better results only on certain data such as S6 and S11. This inspires us to combine URFKs and predefined general kernels in MKFC to further improve the performance of clustering. The superiority of these improved MKFCs are demonstrated in the next subsection.

2) EXPERIMENTAL RESULTS OF DIFFERENT MKFCS

The ACC comparison results of different MKFC-based algorithms on UCI datasets are illustrated in Table 5. For completeness, the results of FCM and Kmeans are also included.

Firstly, comparing the rankings of mACC and mRANK of MKFC and MKFC-G, we can find that both the two rankings go up from seventh to the sixth. Hence, the introduction of a Polynomial kernel improves the clustering performance. Secondly, RMKL performs better than Kmeans, MKFC-G and MKFC, showing its mACC and mRANK rankings are both forth. It should be noticed that RMKL is inferior to FCM, mainly because RMKL is a two-stage method which only considers the diversity and non-redundance of predefined general kernels independently of the clustering. Interestingly, the mACC and mRANK of MKFC are both only in the

TABLE 6. NMI comparison results of different MKFC-based clustering algorithms on UCI datasets.

ID	Kmeans	FCM	MKFC-G	MKFC	RMKL	MKFC-URFK	MKFC-URFK-MR
S1	0.538(5)	0.537(6)	0.542(3)	0.542(3)	0.540(4)	0.543(2)	**0.555(1)**
	0.015	0.014	0.024	0.025	0.026	0.022	0.020
S2	0.865(4)	0.865(4)	0.882(3)	0.893(2)	0.865(4)	**0.928(1)**	**0.928(1)**
	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S3	0.325(7)	0.330(5)	0.348(2)	0.331(4)	0.329(6)	0.342(3)	**0.442(1)**
	0.028	0.024	0.028	0.022	0.029	0.024	0.039
S4	0.578(5)	0.582(3)	0.405(6)	**0.609(1)**	0.579(4)	0.607(2)	0.607(2)
	0.001	0.001	0.036	0.001	0.015	0.001	0.002
S5	0.367(3)	0.356(4)	**0.374(1)**	0.373(2)	0.356(4)	0.353(5)	0.352(6)
	0.056	0.000	0.001	0.002	0.000	0.051	0.051
S6	0.880(3)	**1.000(1)**	0.317(5)	0.660(4)	**1.000(1)**	0.954(2)	**1.000(1)**
	0.141	0.000	0.129	0.199	0.000	0.000	0.000
S7	0.796(2)	**0.806(1)**	0.600(5)	0.730(4)	**0.806(1)**	0.787(3)	0.787(3)
	0.036	0.007	0.130	0.118	0.005	0.030	0.030
S8	0.670(4)	0.684(3)	0.433(6)	0.645(5)	**0.692(1)**	0.686(2)	0.686(2)
	0.057	0.046	0.075	0.064	0.040	0.020	0.020
S9	0.117(3)	0.114(5)	0.109(6)	0.115(4)	**0.122(1)**	0.118(2)	0.118(2)
	0.020	0.019	0.015	0.011	0.024	0.005	0.000
S10	0.009(3)	0.008(4)	**0.014(1)**	0.013(2)	0.006(5)	0.013(2)	0.013(2)
	0.009	0.006	0.016	0.021	0.005	0.021	0.021
S11	0.253(2)	0.253(2)	0.253(2)	**0.255(1)**	0.252(3)	0.252(3)	0.253(2)
	0.008	0.009	0.006	0.007	0.010	0.006	0.006
mNMI	0.491(5)	0.503(4)	0.389(7)	0.470(6)	0.504(3)	0.508(2)	**0.522(1)**
mRANK	3.833(6)	3.500(5)	3.917(7)	3.167(4)	3.083(3)	2.417(2)	**2.000(1)**

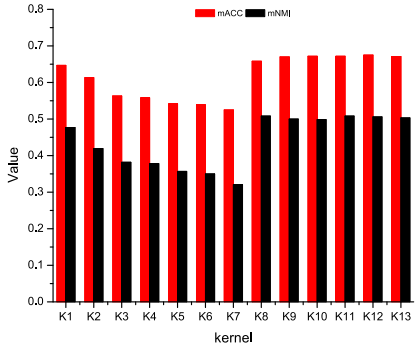


FIGURE 2. Performance of KFCM with different basic kernels.

sixth place in Table 5. This poor result is mainly due to the existence of a few poor-performed predefined general kernels in MKFC, i.e. K5, K6 and K7, the mRANKs of which are the bottom three in Table 3 and Table 4. There is no regularization in MKFC to attenuate the impact of these poor-performed predefined general kernels. However, with the help of URFKs, performance of MKFC-URFK has been greatly improved, proving the effectiveness of URFK in clustering.

Moreover, both the mACC and mRANK of MKFC-URFK-MR go up to the first place among all the algorithms on most datasets with the help of the introduced regularization adjusting the weights of kernels in MKFC-URFK. For instance, the ACC and RANK of MKFC-URFK on S1 dataset are 0.558 and 3th, respectively. MKFC-URFK-MR goes up to the first place among all the algorithms as its ACC increases to 0.617. What's more, the changes of mACC and mRANK between MKFC-URFK and MKFC-URFK-MR clearly demonstrate the effectiveness of matrix-induced regularization.

In addition, the NMI comparison results of different clustering algorithms on UCI datasets are illustrated in Table 6. Again, we can draw similar conclusions in terms of NMI. The effectiveness of URFK and matrix-induced regularization makes that the MKFC-URFK-MR performs best on average.

D. EXPERIMENTS ON FACE IMAGE DATASETS

We also evaluated the algorithms with the face image clustering problem. Yale and ORL datasets [42], the two most commonly used face image datasets, are adopted herein. Fig. 3 and Fig. 4 give sample images of a subject from Yale and ORL datasets, respectively. All images of these two datasets are normalized and cropped to 32×32 pixels.

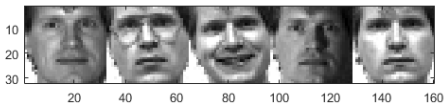


FIGURE 3. Sample images of a subject from Yale dataset.

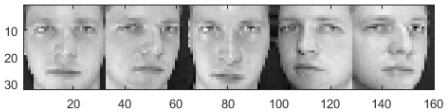


FIGURE 4. Sample images of a subject from ORL dataset.

Yale contains 165 grayscale images of 15 individuals which means each subject has 11 different images. Facial expression or configuration of different person is illustrated on per image.

ORL contains 400 images of 40 individuals which means each subject has 10 different images. Facial details,

facial expressions or the change of photographing time and light are illustrated on per image.

The experimental results on face image datasets, namely ACC, NMI and computational time of each algorithm, are tabulated in Table 7 for comparison. In Table 7, the parameter settings of K1-K13 are the same as the ones in UCI experiment. First, it can be seen that the ACCs of these predefined general kernels on Yale dataset are about 0.45. Especially, ACC of kernel method based on K8 is only 0.177. The results of predefined general kernels in Table 7 are poor. This can be attributed to the fact that the parameter settings of kernels are fixed, while the datasets are now in a different domain. This illustrates the poor adaptability of predefined general kernels to different datasets.

TABLE 7. ACC, NMI and computational time comparison results of different clustering on face datasets.

	Yale			ORL		
	ACC	NMI	Time(s)	ACC	NMI	Time(s)
Kmeans	0.409	0.496	26.44	0.560	0.775	67.11
FCM	0.413	0.501	107.62	0.565	0.778	386.49
KFCM-K1	0.444	0.528	07.61	0.572	0.785	06.41
KFCM-K2	0.443	0.525	08.32	0.577	0.786	06.92
KFCM-K3	0.455	0.527	06.48	0.576	0.787	05.96
KFCM-K4	0.462	0.538	05.94	0.576	0.788	05.52
KFCM-K5	0.463	0.536	05.89	0.587	0.791	05.79
KFCM-K6	0.458	0.536	06.34	0.590	0.793	05.86
KFCM-K7	0.466	0.537	06.40	0.591	0.794	05.59
KFCM-K8	0.177	0.211	05.50	0.092	0.310	10.33
KFCM-K9	0.644	0.669	20.27	0.566	0.770	34.74
KFCM-K10	0.649	0.666	35.10	0.556	0.775	57.88
KFCM-K11	0.642	0.670	46.98	0.566	0.783	75.98
KFCM-K12	0.626	0.651	59.22	0.575	0.784	97.65
KFCM-K13	0.629	0.653	72.13	0.560	0.778	114.52
MKFC-G	0.473	0.543	11.28	0.578	0.786	10.35
MKFC	0.450	0.529	11.29	0.586	0.790	10.36
RMKL	0.584	0.607	11.44	0.554	0.776	10.72
MKFC-URFK	0.489	0.561	208.57	0.585	0.790	356.88
MKFC-URFK-MR	**0.651**	**0.677**	241.64	**0.598**	**0.796**	429.32

On the contrary, the URFKs in Table 7, i.e K9-K13, could achieve much better performance as their ACC and NMI of Yale and ORL datasets reach about 0.65 and 0.66, 0.57 and 0.78 respectively. These experimental results show that when URFKs are applied to different datasets, it still could achieve relative good results, showing the advantage of using a data-dependent kernel.

Then, when matrix-induced regularization is added into MKFC-URFK, MKFC-URFK-MR could achieve the best results among all these algorithms, as shown in the last row of Table 7, proving the effectiveness of matrix-induced regularization.

Besides, Table 7 also shows the total computational time in seconds over 50 runs of different clustering on Yale and ORL datasets. It can be observed that KFCM with URFKs, i.e. K9-K13 require more time than that with predefined kernels, i.e. K1-K8. Specifically, computational time of KFCM with predefined general kernels is about 6 seconds while that of KFCM with URFKs is in the interval [20 73]. Furthermore, as the increase of trees of random forests, the computational time increase linearly. Clearly, MKFC-URFK and MKFC-URFK-MR cost more time than other algorithms since there are five URFKs in their candidate kernels.

E. EXPERIMENTS ON NOISY DATASET

In this section, experiments are conducted on noisy datasets to test the efficiency of the different algorithms. In the experiment, the noise is introduced to Wine dataset. The noise meets the uniformly distribution of a certain interval $[-\tau,\tau]$, where τ takes values between 0 and 1 with interval of 0.2. The efficiency of the algorithms against the increasing noisy datasets is shown in Fig. 5 and Fig. 6, which represent the ACC and NMI performance results, respectively.

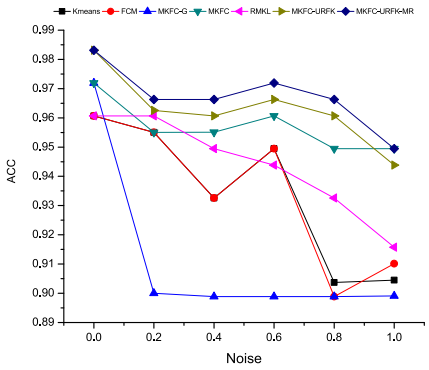


FIGURE 5. ACC performance of clustering algorithms on noisy data.

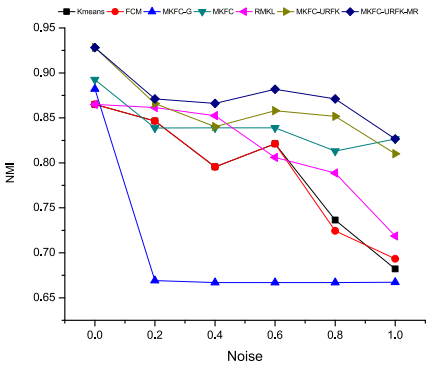


FIGURE 6. NMI performance of clustering algorithms on noisy data.

It can be observed from Fig. 5 and Fig. 6, both ACC and NMI of each algorithm decrease when the amount of noise added to data increases, showing that the clustering quality of algorithms degrades. Nonetheless, the performance of the proposed URFK based methods, i.e. MKFC-URFK and MKFC-URFK-MR always perform better than the other algorithms without URFK. What's more, MKFC-URFK-MR could achieve the best performance among these algorithms. For example, in Fig. 5, ACC of MKFC-URFK-MR, shown in pink line, varies from about 0.985 to 0.955 while that of MKFC varies from about 0.975 to 0.955, when τ takes value from 0 to 1. MKFC-URFK-MR achieves better results at each point in the figure. The similar phenomenon could be observed from Fig. 6, demonstrating the efficiency of URFK and the effectiveness of matrix-induced regularization.

F. HYPER-PARAMETER SELECTION

The proposed MKFC-URFK-MR introduces a hyper-parameter γ to balance the contributions of MKFC and the matrix-induced regularization. As a sensitivity analysis of the hyper-parameter, ACC and NMI of the proposed algorithm on UCI Breast tissue data are plotted in Fig. 7 by varying the γ from 2^{-30} to 2^{40}. As observed, ACC and NMI have similar trend that both of them first increase to its highest value and then decrease to a certain value with the increasing of γ. Besides, the proposed MKFC-URFK-MR performs stably in a certain range of hyper-parameter. Similar to most authors' approach in the clustering area [31], [44], the regularization parameter γ of the proposed model in this paper is tuned from a wide range $[2^{-30}, 2^{-18}, ..., 2^{40}]$ by grid search.

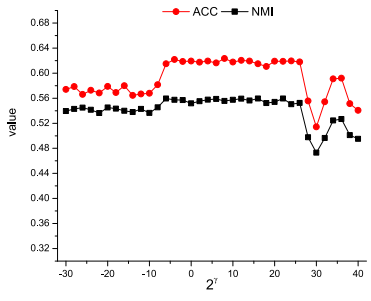


FIGURE 7. The effect of the hyper-parameter γ to the proposed MKFC-URFK-MR algorithm.

G. CONVERGENCE AND COMPUTATIONAL COMPLEXITY

With the updating of membership matrix U and kernel weight W, the objective function (11) is minimized interactively. In other words, in the ith iteration where partial minimization is achieved, the following relationship holds

$$J(U^{i+1}, W^{i+1}) \leq J(U^{i+1}, W^{i}) \leq J(U^{i}, W^{i}) \qquad (25)$$

It implies that $J(U, W)$ is a decreasing function with respect to the iteration number i. Therefore, the proposed MKFC-URFK-MR converges to either a saddle point of the objective function or a local optimal solution.

Fig. 8 shows the objective value of our proposed algorithm at each iteration on face image datasets. From the figure, we can observe that the objective value decreases monotonically. Usually, the algorithm converges in around fifty iterations.

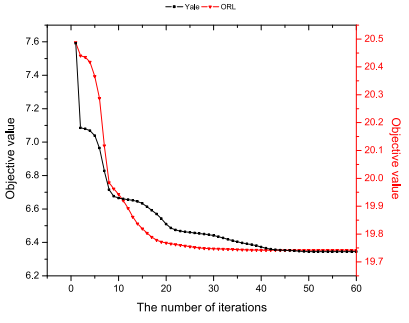


FIGURE 8. The objective value of MKFC-URFK-MR algorithm at each iteration on face image datasets.

The computational complexity of MKFC-URFK-MR per iteration is $O(N^2L)$. To store the kernel matrices, kernel weight matrix and the partition matrix, the space required by the algorithm is $O(N^2L)$, $O(L)$ and $O(NC)$, respectively.

V. CONCLUSION

In this paper, a data-dependent kernel, URFK, is introduced by transforming the supervised random forests into an unsupervised one. Unlike the traditional predefined general kernels, such as Gaussian and Polynomial ones, URFK is constructed based on inherent features of datasets. As a result, URFK shows good adaptability to various datasets. By combing URFK, Gaussian and Polynomial kernels, MKFC-URFK is formed and matrix-induced regularization is introduced to reduce redundancy between kernels in MKFC-URFK. Finally, the proposed novel algorithm, MKFC-URFK-MR, shows advantages in adaptability, performance and efficiency in clustering, which have been demonstrated by experiments on UCI data, face image data and some noisy datasets.

In the future, the study would focus on experiments on large datasets and incomplete datasets. Kernel-based methods tend to be unattainable to large datasets [45]. To solve the problem, kernel-based methods in an approximated kernel space [46] can be considered. For incomplete datasets [47], to obtain important features, feature selection [48], [49] methods in approximated kernel space could be taken into account. In other words, it is worthy studying the influence of feature weights in approximated kernel space.

ACKNOWLEDGMENT

The authors would like to thank the editors and the reviewers for their valuable and constructive comments and suggestions.

REFERENCES

[1] J. Gu, L. Jiao, S. Yang, and F. Liu, "Fuzzy double C-means clustering based on sparse self-representation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 612–626, Apr. 2018.

[2] J. Zhou, L. Chen, C. L. P. Chen, Y. Zhang, and H.-X. Li, "Fuzzy clustering with the entropy of attribute weights," *Neurocomputing*, vol. 198, pp. 125–134, Jul. 2016.

[3] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer, 1981.

[4] X. Zhu, W. Pedrycz, and Z. Li, "Fuzzy clustering with nonlinearly transformed data," *Appl. Soft Comput.*, vol. 61, pp. 364–376, Dec. 2017.

[5] K. Honda, S. Ubukata, and A. Notsu, "A novel approach to noise clustering in multivariate fuzzy c-means," in *Proc. 17th World Congr. Int. Fuzzy Syst. Assoc. (SCIS)*, Jun. 2017, pp. 1–4.

[6] M. Zarinbal, M. H. F. Zarandi, and I. B. Turksen, "Relative entropy fuzzy C-means clustering," *Inf. Sci.*, vol. 260, pp. 74–97, Mar. 2014.

[7] L. Guo, L. Chen, C. L. P. Chen, and J. Zhou, "Integrating guided filter into fuzzy clustering for noisy image segmentation," *Digit. Signal Process.*, vol. 83, pp. 235–248, Dec. 2018.

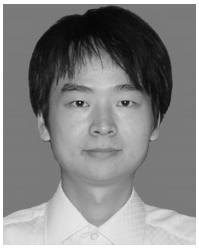
[8] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," *Fuzzy Sets Syst.*, vol. 161, no. 4, pp. 522–543, Feb. 2010.

[9] L. H. Son, "A novel kernel fuzzy clustering algorithm for geo-demographic analysis," *Inf. Sci.*, vol. 317, pp. 202–223, Oct. 2015.

[10] S. Das and S. Sil, "Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm," *Inf. Sci.*, vol. 180, no. 8, pp. 1237–1256, Apr. 2010.
[11] L. Chen, J. Zou, and C. L. P. Chen, "Kernel spatial shadowed C-means for image segmentation," *Int. J. Fuzzy Syst.*, vol. 16, no. 1, pp. 46–56, Mar. 2014.
[12] O. Bchir, H. Frigui, and M. M. B. Ismail, "Fuzzy clustering with learnable cluster-dependent kernels," *Pattern Anal. Appl.*, vol. 19, no. 4, pp. 919–937, Nov. 2016.
[13] L. Chen, C. L. P. Chen, and M. Lu, "A multiple-kernel fuzzy C-means algorithm for image segmentation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 5, pp. 1263–1274, Oct. 2011.
[14] 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.
[15] B. Zamani, A. Akbari, and B. Nasersharif, "Evolutionary combination of kernels for nonlinear feature transformation," *Inf. Sci.*, vol. 274, pp. 95–107, Aug. 2014.
[16] T. H. Dang, L. T. Ngo, and W. Pedrycz, "Multiple kernel based collaborative fuzzy clustering algorithm," in *Intelligent Information and Database Systems*, N. T. Nguyen, B. Trawiński, H. Fujita, and T. P. Hong, Eds. Berlin, Germany: Springer, 2016, pp. 585–594.
[17] L. Du *et al.*, "Robust multiple kernel k-means using l21-norm," in *Proc. 24th Int. Conf. Artif. Intell.*, Jun. 2015, pp. 3476–3482.
[18] J. Zhou, C. L. P. Chen, and L. Chen, "Maximum-entropy-based multiple kernel fuzzy C-means clustering algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 1198–1203.
[19] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran, 2014, pp. 1305–1313.
[20] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple kernel learning for hyperspectral image classification: A review," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6547–6565, Nov. 2017.
[21] Y. Gu, C. Wang, D. You, Y. Zhang, S. Wang, and Y. Zhang, "Representative multiple kernel learning for classification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 7, pp. 2852–2865, Jul. 2012.
[22] Y. Gu, G. Gao, D. Zuo, and D. You, "Model selection and classification with multiple kernel learning for hyperspectral images via sparsity," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2119–2130, Jun. 2014.
[23] Y. Gu, Q. Wang, H. Wang, D. You, and Y. Zhang, "Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2739–2751, Jun. 2015.
[24] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1888–1894.
[25] K. H. Memon and D.-H. Lee, "Generalised kernel weighted fuzzy C-means clustering algorithm with local information," *Fuzzy Sets Syst.*, vol. 340, pp. 91–108, Jun. 2018.
[26] I. Dagher, "Fuzzy clustering with optimized-parameters multiple Gaussian kernels," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Aug. 2015, pp. 1–6.
[27] K.-P. Lin, "A novel evolutionary kernel intuitionistic fuzzy *C*-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1074–1087, Oct. 2014.
[28] L. Liu, C.-F. Li, Y.-M. Lei, J.-J. Zhao, J.-Y. Yin, and X.-K. Sun, "A new fuzzy clustering method with neighborhood distance constraint for volcanic ash cloud," *IEEE Access*, vol. 4, pp. 7005–7013, 2016.
[29] C. Liu, J. Liu, D. Peng, and C. Wu, "A general multiobjective clustering approach based on multiple distance measures," *IEEE Access*, vol. 6, pp. 41706–41719, 2018.
[30] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge Univ., 2004.
[31] J. Wang *et al.*, "Distance metric learning for soft subspace clustering in composite kernel space," *Pattern Recognit.*, vol. 52, pp. 113–134, Apr. 2016.
[32] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[33] Y. Pei, L. Kou, and H. Zha, "Anatomical structure similarity estimation by random forest," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2941–2945.
[34] M. B. Pouyan, J. Birjandtalab, and M. Nourani, "Distance metric learning using random forest for cytometry data," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, p. 2590.
[35] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *J. Comput. Graph. Statist.*, vol. 15, no. 1, pp. 118–138, Mar. 2006.
[36] L. Breiman and A. Cutler, "Random forests manual v4.0," Dept. Statist., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep., 2003. [Online]. Available: https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf
[37] F. Topsøe, "Some inequalities for information divergence and related measures of discrimination," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1602–1609, Jul. 2000.
[38] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 513–520.
[39] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert–Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Berlin, Germany: Springer, 2005, pp. 63–77.
[40] C. B. C. M. D. J. Newman and S. Hettich, "UCI machine learning repository," Dept. School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 1987. [Online]. Available: https://archive.ics.uci.edu/ml/index.php
[41] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
[42] D. Cai, "Codes and datasets for feature learning," Dept. College Comput. Sci., Zhejiang Univ., Hangzhou, China. [Online]. Available: http://www.cad.zju.edu.cn/home/dengcai/Data/data.html
[43] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: SIAM, 2012.
[44] X. Liu *et al.*, "Optimal neighborhood kernel clustering with multiple kernels," in *Proc. AAAI*, Feb. 2017, pp. 2266–2272.
[45] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Red Hook, NY, USA, Curran, 2008, pp. 1177–1184.
[46] Y.-P. Zhao, L. Chen, and C. L. P. Chen, "Multiple kernel shadowed clustering in approximated feature space," in *Data Mining Big Data*, Y. Tan, Y. Shi, and Q. Tang, Eds. Berlin, Germany: Springer, 2018, pp. 265–275.
[47] X. Liu *et al.*, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2018.2879108.
[48] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2013.
[49] 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.

YIN-PING ZHAO received the B.S. and M.S. degrees from the School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China, in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, University of Macau, Macau, China. Her research interests include artificial intelligence, machine learning, pattern recognition, and image processing.

LONG CHEN received the B.S. degree in information sciences from Peking University, Beijing, China, in 2000, the M.S.E. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2003, the M.S. degree in computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2005, and the Ph.D. degree in electrical engineering from The University of Texas at San Antonio, San Antonio, TX, USA, in 2010. From 2010 to 2011, he was a Postdoctoral Fellow with The University of Texas at San Antonio. He is currently an Associate Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His current research interests include computational intelligence, Bayesian methods, and other machine learning techniques and their applications.



MIN GAN received the B.S. degree in computer science and engineering from the Hubei University of Technology, Wuhan, China, in 2004, and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2010. He is currently a Professor with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His current research interests include statistical learning, system identification, and nonlinear time series analysis.

C. L. PHILIP CHEN (S'88–M'88–SM'94–F'07) received the degree from the University of Michigan, Ann Arbor, MI, USA. He is currently the Chair Professor of the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. The University of Macau's Engineering and Computer Science programs receiving Hong Kong Institute of Engineers' (HKIE) accreditation and Washington/Seoul Accord is his utmost contribution to engineering/computer science education for Macau as the former Dean of the Faculty of Science and Technology. His current research interests include systems, cybernetics, and computational intelligence. He is a Fellow of the IEEE, AAAS, IAPR, Chinese Association of Automation (CAA), and HKIE. He received the 2016 Outstanding Electrical and Computer Engineers Award from Purdue University. He was the President of the IEEE SMC Society, from 2012 to 2013. He is a Vice President of CAA. He is the Editor-in-Chief of the IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and an Associate Editor of several IEEE TRANSACTIONS. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control, from 2015 to 2017, and also a Program Evaluator of the Accreditation Board of Engineering and Technology Education, USA, for computer engineering, electrical engineering, and software engineering programs.