A High-Order Possibilistic C-Means Algorithm for Clustering Incomplete Multimedia Data

Qingchen Zhang, Laurence T. Yang, Zhikui Chen, and Feng Xia

***Abstract*—Clustering is a commonly used technique for multimedia organization, analysis, and retrieval. However, most multimedia clustering methods are difficult to capture the high-order nonlinear correlations over multimodal features, resulting in the low clustering accuracy. Furthermore, they cannot extract features from multimedia data with missing values, leading to failure in clustering incomplete multimedia data that are widespread in practical applications. In this paper, we propose a high-order possibilistic C-means algorithm (HOPCM) for clustering incomplete multimedia data. HOPCM improves the basic autoencoder model for learning features of multimedia data with missing values. Furthermore, HOPCM uses the tensor distance rather than the Euclidean distance as the distance metric to capture as much as possible the unknown high-dimensional distribution of multimedia data. Extensive experiments are carried out on three representative multimedia data sets: NUS-WIDE, CUAVE, and SNAE. The results demonstrate that HOPCM achieves significantly better clustering performance than many existing algorithms. More importantly, HOPCM is able to cluster both high-quality multimedia data and incomplete multimedia data effectively, while other existing methods can only cluster the high-quality multimedia data.**

***Index Terms*—Feature learning, incomplete multimedia data, possiblistic C-means (PCM) algorithm, tensor distance, vector outer product.**

I. Introduction

With recent development of visual computing technology and wireless communication, multimedia data over networks are proliferating at extremely high speed for many applications, for example, in smart traffic monitoring, target recognition, and customer behavior predicting [1]–[4]. Multimedia data are very complex in information, properties, and representation [5]–[7]. First, multimedia data in the real world comes from many input channels; hence, multimedia data are a typical kind of multimodal data. Second, different modalities often convey different information. For example, an image uses a lot of details such as shading, complex scene, and rich color to vividly display an object and uses a caption to show things that may not be obvious in the image, such as the name of the object [8], [9]. Moreover, different modalities have complex relations. Finally, a lot of multimedia data suffer from many missing values as a result of sensor defaults, fault measurements, and data transfer problems over networks. In other words, some multimedia data are incomplete in practical applications. These characteristics, particularly incompleteness, pose an important challenge on multimedia clustering techniques.

Clustering techniques aim to partition a large number of data into groups based on measured similarity. They are usually applied to the task of multimedia processing such as multimedia organization, analysis, communication, and retrieval [43]–[45]. Thus, multimedia clustering is a fundamental issue. Many multimedia clustering algorithms have been developed. Conventional clustering algorithms for multimedia data focus on single-modal data, such as image clustering, audio clustering, and text clustering. In recent years, the multimedia data coclustering approaches have drawn much attention from researchers [10]–[13]. Some works aim at image–text coclustering, while other methods have developed for image–audio clustering. Although existing algorithms perform their job well for clustering multimedia data, they have still several drawbacks elaborated as follows. First, most of them are generally designed for high-quality data; hence, they fail to cluster incomplete multimedia data. Second, most of them are mainly performed upon bimodal multimedia data. However, many multimedia data are multimodal, such as a chip of video containing texts, images, and audio. Finally, they combine the features of different modalities by only concatenating them linearly, making them hard to capture the high nonlinear correlations over different modalities that exist in the level of features, resulting in a low clustering accuracy.

In view of the aforementioned issues, we propose a high-order possibilistic C-means (HOPCM) algorithm based on feature learning for clustering incomplete multimedia data in this paper. HOPCM is implemented by three steps: unsupervised feature learning, feature fusion, and high-order clustering. First, we improve the basic autoencoder (BAE) model to learn features from incomplete data. Each single modality of multimedia data is separately learned by the improved autoencoder (AE) model. Next, the vector outer product is used to fuse the learned features to model the nonlinear correlations over different modalities, which aims at forming the joint representation of multimedia data. Finally, a HOPCM algorithm is implemented for clustering the multimedia data in the tensor space. HOPCM uses the tensor distance rather than Euclidean distance as the metric between two objects to capture as much as possible the unknown high-dimensional distribution of multimedia data.

To evaluate the performance of the proposed algorithm, we carry out some experiments on three representative multimedia data sets: NUS-WIDE, CUAVE, and SNAE. The results demonstrate that HOPCM achieves significantly better clustering

Manuscript received November 14, 2014; revised January 31, 2015; accepted April 9, 2015.

Q. Zhang, Z. Chen, and F. Xia are with the School of Software, Dalian University of Technology, Dalian 116620, China.

L. T. Yang is with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada, and also with Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: ltyang@gmail.com).

Digital Object Identifier 10.1109/JSYST.2015.2423499

performance than that by many existing algorithms. More importantly, HOPCM is able to cluster both high-quality multimedia data and incomplete multimedia data effectively, while other existing methods can only cluster the high-quality data.

The contributions of this paper can be summarized as the following three aspects: first, we improve the BAE model to learn features of incomplete data. Second, to capture the high nonlinear correlations over different modalities of multimedia data, we use the vector outer product to fuse the learned features of each modality to form the joint representations of multimedia data. Finally, to cluster the multimedia data in the tensor space, we design a HOPCM algorithm by using the tensor distance as the distance metric that encourages HOPCM to capture as much as possible the unknown high-dimensional distribution of multimedia data.

The rest of the paper is organized as follows: Section II presents the preliminaries related to this paper. The problem formulation of multimedia clustering is described in Section III. The proposed algorithm is illustrated in Section IV, and the performance evaluation and analysis is described in Section V. Section VI reviews related work on multimedia clustering. Finally, the whole paper is concluded in Section VII.

II. Preliminaries

A. AEs

The AEs have been successfully used in unsupervised feature learning for many application domains, such as, for example, image processing, audio recognition, and natural language analysis.

A BAE is defined by a parameter set $\theta = (W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)})$, where $(W^{(1)}, W^{(2)})$ are weight matrices, and $(b^{(1)}, b^{(2)})$ are bias vectors [15]. The BAE maps an input x to hidden representation h by an encoding function f as follows:

$$H = f_\theta\left(W^{(1)} \odot X + b^{(1)}\right) \tag{1}$$

where f is a nonlinear activation function, typically a logistic sigmoid $s_f(x) = 1/(1+e^{-x})$.

Then, the BAE reconstructs the input from hidden representation by a decoding function

$$Y = h_{W,b}(X) = g_\theta\left(W^{(2)} \odot H + b^{(2)}\right) \tag{2}$$

where g is the decoding function, which is also a sigmoid function.

Thus, the parameters are trained by minimizing the following function:

$$J_{\mathrm{AE}}(\theta) = \sum_{x \in D} L\left(x, g\left(f(x)\right)\right) \tag{3}$$

where L is the reconstruction error that can be typically defined by a squared error or a cross-entropy.

To prevent overfitting, a regularization term called weight decay is added into the reconstruction error

$$J_{\mathrm{AE+wd}}(\theta) = \left(\sum_{x \in D} L\left(x, g\left(f(x)\right)\right)\right) + \lambda \sum_{ij} W_{ij}^2 \tag{4}$$

where the λ hyperparameter controls the strength of the regularization.

The weights and biases of the AE can be trained typically by the back-propagation algorithm.

In recent years, several variants of the AE have been developed by imposing different constraints. The most well-known constraint is called sparsity regularization, which aims to achieve a sparse representation of the input. Different sparsity regularization can generate different sparse representation. The most widely used sparsity regularization is a Kullback–Liebler divergence with respect to the binomial distribution [16], [17].

Another AE model is called the denoising AE (DAE), whose goal is to learn robust representations from a noisy input [18]–[20]. The DAE simply corrupts input x, before sending it through the AE, and then reconstructs the clean version. Thus, the DAE has an objective function as follows:

$$J_{\mathrm{DAE}}(\theta) = \sum_{x \in D} E_{\tilde{x} \sim q(\tilde{x}|x)}\left[L\left(x, g\left(f(\tilde{x})\right)\right)\right] \tag{5}$$

where the expectation is over corrupted versions $\tilde{x}$ of examples x obtained from a corruption process $q(\tilde{x}|x)$.

Similar to the motivation of DAE, to learn robust representations, a contractive AE (CAE) is proposed to enhance the robustness with the training objective [21], [22]

$$J_{\mathrm{CAE}}(\theta) = \left(\sum_{x \in D} L\left(x, g\left(f(x)\right)\right)\right) + \lambda \left\| J_f(x) \right\|^2. \tag{6}$$

One drawback of CAE is that its analytic penalty only encourages robustness to infinitesimal input variations. Rifai *et al.* improved the CAE model with an objective function

$$J_{\mathrm{CAE}+H} = \sum_t L\left(x^{(t)}, g_\theta\left(x^{(t)}\right)\right) + \lambda \left\| J\left(x^{(t)}\right)\right\|_F^2 + \gamma E_\epsilon[\| J(x) - J(x+\epsilon)\|_F^2 \tag{7}$$

where $\epsilon \sim N(0, \sigma^2 I)$, and γ is the associated regularization strength hyperparameter.

AEs and their variants have been applied successfully in learning features for various data, such as text, images, and audio. However, they fail to learn features for incomplete data.

B. PCMs

A possibilistic C-means (PCM) algorithm is defined by a matrix $U = \{u_{ij}\}$, where u_{ij} denotes what degree x_j belongs to the ith cluster, with a constraint as follows:

$$u_{ij} \in [0,1] \text{ for all } i \text{ and } j, \quad 0 < \sum_{j=1}^{n} u_{ij} \le N$$
$$\text{for all } i, \quad \max{}_i u_{ij} > 0 \text{ for all } j. \tag{8}$$

PCM minimizes the following objective function [23]:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \left\| x_k - v_i \right\|^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m \tag{9}$$

where $V = (v_1, v_2, \ldots, v_c)$ is a C-tuple of prototypes, $m > 1$ is a fuzzification constant, and η_i is a suitable positive number.

Solving the minimization problem yields membership functions of the form

$$u_{ij} = \frac{1}{\left(1 + \left(\frac{d_{ij}}{\eta_i}\right)^{\frac{1}{m-1}}\right)}. \tag{10}$$

The cluster centers are updated using

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}. \tag{11}$$

The PCM algorithm can be outlined as follows:

Step 1: Choose m, c, and $\xi > 0$ and then initialize the membership matrix $U^{(0)}$;
Step 2: Update cluster centers using (11);
Step 3: Estimate η_i using the following formula:

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m}. \tag{12}$$

Step 4: Calculate the distance d_{ki} between x_k and v_i;
Step 5: Update membership matrix U using (10);
Step 6: If $\varepsilon \leq \|u_{ij} - u'_{ij}\|^2$, stop; else repeat step 2.

PCM has been used in image clustering and speech recognition; however, it still has several drawbacks in the practical applications. For example, PCM is often corrupted by noise and tends to produce coincident clusters. In recent years, many variants have been developed for improving the original PCM algorithm.

Zhang and Leung applied the fuzzy method to PCM for overcoming the noise sensitivity [24]. Yang and Lai improved the PCM algorithm to find the suitable number of clusters automatically by using a merging strategy [25]. Another improved PCM algorithm was developed by Xie *et al.* to enhance the robustness [26]. Another representative improved algorithm is the kernel-based PCM algorithm, which first maps original data into higher dimensional feature space and then clusters the data in the feature space [27]. The kernel-based PCM algorithm is performed very well in finding clusters with nonspherical shapes. In addition, Schneider has developed the weighted PCM to find homogeneous groups [28], [29]. To cluster incomplete data, Zhang and Chen presented a weighted PCM algorithm by applying the partial distance strategy to PCM [14]. Other methods for improving PCM include possibilistic fuzzy C-means (PFCM) and fuzzy possibilistic C-means (FPCM) [30].

Although these methods are performed well in the cluster process, they can only cluster single-modal data, such as relational data and image, making it hard to cluster multimedia data with multiple modalities.

III. Problem Formulation

Consider a data set with t objects $X = \{x_1, x_2, \ldots, x_t\}$. Each object is represented by m features with the form $A = \{a_1, a_2, \ldots, a_m\}$. For example, an image $R^{28\times 28}$ can be represented by 576 raw pixels, which means that each element in the feature set A is denoted as a pixel. The goal of multimedia clustering is to partition the data set into several groups based on the similarity measure, such that the objects belonging to the same group share much similarity. For example, in the web document domain, the clustering task is to identify similar documents based on the visual content, audio element, and text features.

As reviewed in the previous section, the multimedia clustering task poses a number of issues and challenges, particularly for incomplete multimedia data. We discuss the key challenges in three aspects as follows.

1) Feature Learning of Incomplete Data: Feature extraction and analysis is the fundamental step of clustering. Feature learning has been well studied in literature. Typically, many feature learning methods based on machine learning techniques, including deep learning that is an extremely active subfield of machine learning, have been successfully used in visual feature extraction, text feature analysis, and audio feature learning. Unfortunately, current techniques focus on feature learning for high-quality data. In other words, they cannot learn features of incomplete data. Hence, feature learning of incomplete data is the first problem to be solved for clustering multi media data.

2) Joint Representation of Multimedia Data: Feature fusion and joint representation of multimedia data plays an important role in the clustering task. Existing works on feature fusion rely on some global optimization methods. They are usually of high computational complexity. Furthermore, they do not address the problem of weighting the feature modalities in their objective functions, leading to a poor joint representation of multimedia data.

3) Distance Measure in the Tensor Space: Distance measure is the key challenge related to multimedia clustering. Many metrics can be used to measure the distance between different objects, such as Euclidean distance, Mahalanobis distance, and Hamming distance. However, most of them work in the vector space, making them hard to measure the distance between different objects in the tensor space. Thus, how to measure the distance is a key challenge in a multimedia clustering algorithm.

IV. Description of the Proposed Method

A general HOPCM algorithm is implemented by three stages, i.e., unsupervised feature learning, feature fusion, and high-order clustering, which is shown in Fig. 1.

In unsupervised feature learning, each single modality of multimedia data is separately learned by the improved AE model. Next, the vector outer product is used to fuse the learned features to form the joint representation of multimedia data. Finally, a HOPCM algorithm is used to cluster the multimedia data for producing the final result.

A. Improved AE Model

To learn features of incomplete multimedia data, we divide the raw data set with missing values into two different subsets C and O. In the subset O, each sample contains some missing values, whereas each sample in the other subset has no missing values.

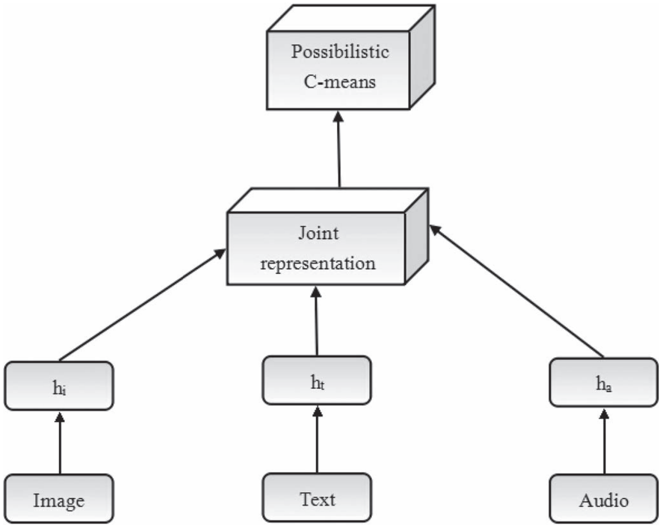


Fig. 1. Architecture of the proposed method.

TABLE I
SIMPLE EXAMPLE OF THE INCOMPLETE DATA SET

R/A	temperature	humidity	lumious	power
Room1	28	41	200	31.4
Room2	28	41	200	87.1
Room3	29	42	180	43.5
Room4	27.5	*	170	29.4
Room5	29	42	*	*

In particular, the subset C is called the complete subset, whereas the other one is called the incomplete subset. Table I shows a simple example of the incomplete data set sampled from Digital Home Lab.

In Table I, $*$ represents the missing values. In Table I, there is one missing value in the fourth record, whereas there are two missing values in the last record. The two records are called incomplete objects, whereas the others are called complete objects.

Hence, for the data set X in Table I, the incomplete subset $C = \text{room4}, \text{room5}$, whereas the other subset $O = \text{room1}, \text{room2}, \text{room3}$.

The paper improves a BAE for learning features from incomplete objects, which is performed as follows.

First, we get a stochastic instance subset $P = \{x_i | i = 1, 2, \ldots, n\}$ by choosing many samples from C. Next, we delete some attribute values from every sample in the instance subset. Finally, we replace the deleted values with random values. Therefore, we can obtain a training subset $T = \{x_i | i = 1, 2, \ldots, n\}$ using these deleted samples.

The improved AE model, which is presented in Fig. 2, maps each sample in subset T to a hidden data y via the following encoder function:

$$y = f_\theta(x') = s\left(W^{(1)}x' + b^{(1)}\right). \tag{13}$$

Next, y is mapped back to an object z as follows:

$$z = g_\theta(y) = s\left(W^{(2)}y + b^{(2)}\right). \tag{14}$$

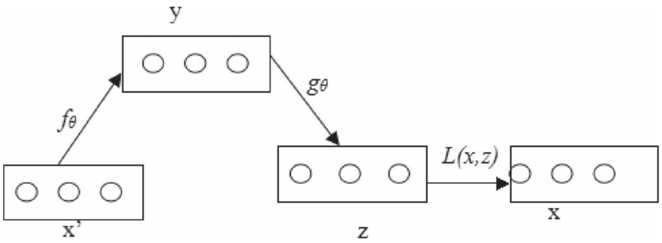


Fig. 2. Improved AE model.

The parameter $\theta = (W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)})$ is trained typically by the back-propagation algorithm [31].

After getting the improved AE model, we can use it to learn features of the samples with missing values.

B. Feature Fusion via the Vector Outer Product

Here, the learned features of each modality are fused via the vector outer product.

The outer product is one of the most common types of tensor multiplications defined in literature [32]. Given an N-order tensor $A \in R^{I_1 \times I_2 \times \cdots \times I_N}$ and an M-order tensor $B \in R^{J_1 \times J_2 \times \cdots \times J_M}$, their outer product will produce an $(N + M)$-order tensor $C \in R^{I_1 \times I_2 \times \cdots \times I_N \times J_1 \times J_2 \times \cdots \times J_M}$. Each entry in the tensor C is defined as $c_{i_1,\ldots,i_N,j_1,\ldots,j_M} = a_{i_1,\ldots,i_N} \cdot b_{j_1,\ldots,j_M}$, where $a_{i_1,\ldots,i_N}, b_{j_1,\ldots,j_M}, c_{i_1,\ldots,i_N,j_1,\ldots,j_M}$ are one entry in the tensor A, B, C, respectively. The vector outer product is a special form of the outer product. In detail, the outer product of two nonzero vectors $a \in R^I, b \in R^J$ produces a matrix $X = a \circ b = ab^T \in R^{I \times J}$, and the outer product of three nonzero vectors $a \in R^I, b \in R^J, c \in R^K$ produces a 3-order tensor $X = a \circ b \circ c \in R^{I \times J \times K}$, whose entries are $x_{ijk} = a_i \cdot b_j \cdot c_k$.

For an image, text, and an audio, we use three vectors a, b, c to represent their features learned by the improved AE model, respectively. Then, the vector outer product is used to fuse the learned features to form the joint representation of multimedia data according to the following rules.

1) For the multimedia data with an image and a text, its joint representation is denoted as $X = a \circ b = ab^T$.
2) For the multimedia data with an image and an audio, its joint representation is denoted by $X = a \circ c = ac^T$.
3) For the multimedia data with an image, a text and an audio, its joint representation is denoted by $X = av \circ b \circ c$.

C. HOPCM Algorithm

In the paper, the PCM algorithm is used to cluster the joint representation obtained in Section IV-B for the final clustering result. The conventional PCM works in the vector space; however, the joint representation of the multimedia data is represented by the high-order tensor. For example, the joint representation of the multimedia data with an image, a text, and an audio is represented by a 3-order tensor. Thus, this paper designs a HOPCM algorithm by extending PCM from the vector space to the tensor space.

The HOPCM algorithm has the similar objective function

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ki} + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1-u_{ij})^m. \quad (15)$$

In the conventional PCM algorithm, d_{ki} refers to the Euclidean distance between x_k and v_i, while it represents the tensor distance in the HOPCM algorithm.

The tensor distance is an effective tool for measuring the distance between two samples for high-order complex data by capturing the correlation in the high-order tensor space [33].

Given an N-order tensor $X \in R^{I_1 \times I_2 \times \cdots \times I_N}$, x is denoted as the vector form representation of X, and the element $X_{i_1 i_2 \ldots i_N (1 \le i_j \le I_j, 1 \le j \le N)}$ in X is corresponding to x_l, i.e., the lth element in x, where $l = i_1 + \sum_{j=2}^{N} \prod_{t=1}^{j-1} I_t$. Then, the tensor distance between two N-order tensors is defined as

$$\begin{aligned} d_{\text{TD}} &= \sqrt{\sum_{l,m=1}^{I_1 \times I_2 \times \cdots \times I_N} g_{lm}(x_l - y_l)(x_m - y_m)} \\ &= \sqrt{(x-y)^T G (x-y)} \end{aligned} \quad (16)$$

where g_{lm} is the metric coefficient used to capture the correlations between different coordinates in the tensor space, which is defined as

$$g_{lm} = \frac{1}{2\pi\delta^2} \exp\left\{ -\frac{\|p_l - p_m\|_2^2}{2\delta^2} \right\} \quad (17)$$

where $\|p_l - p_m\|_2$ is defined as

$$\|p_l - p_m\|_2 = \sqrt{(i_1 - i_1')^2 + \cdots + (i_N - i_N')^2}. \quad (18)$$

The details of the tensor distance can be referred to [33].

Solving the minimization problem yields the same form of membership functions $u_{ij} = 1/(1 + (d_{ij}/\eta_i)^{1/(m-1)})$, where d_{ki} refers to the tensor distance between x_k and v_i. The way for updating cluster centers is the same with PCM.

Therefore, the HOPCM algorithm can be described as follows:

Step 1: Choose m, c, and $\xi > 0$ and then initialize the membership matrix $U^{(0)}$;
Step 2: Update cluster centers using (11);
Step 3: Estimate η_i using (12);
Step 4: Calculate the tensor distance d_{ki} between x_k and v_i;
Step 5: Update membership matrix U;
Step 6: If $\varepsilon \le \|u_{ij} - u_{ij}'\|^2$, stop; else repeat step 2.

By comparing the steps of the PCM algorithm and the HOPCM algorithm, it can be observed that they share the same computational complexity that is dominated by the computation of the distance between x_k and v_i, which needs $O(n^2)$ operations for each cluster. Thus, the HOPCM algorithm has a total time complexity of $O(tcn^2)$. Their difference is mainly demonstrated in the fourth step. In the fourth step, PCM calculates the Euclidean distance between x_k and v_i, while HOPCM calculates the tensor distance.

V. Experiments

Here, we evaluate the performance of the proposed HOPCM algorithm on three representative data sets: NUS-WIDE, CUAVE, and SNAE.

A. Evaluation Criteria

In order to assess the effectiveness of HOPCM, two well-known evaluation criteria, i.e., $E*$ and adjusted Rand index (ARI), are used in the experiment [34], [35].

$E*$ is used to assess the error between ideal clustering centers and the clustering centers produced by a specific algorithm. $E*$ is calculated by the following:

$$E_* = \sqrt{\sum_{i=1}^{c} \left\| v_{\text{ideal}}^i - v_*^i \right\|^2} \quad (19)$$

where v_{ideal}^i represents the ith ideal cluster center, and v_*^i denotes the ith cluster center produced by a specific algorithm $*$. A lower value of $E*$ indicates that the algorithm produces more accurate clustering centers.

$\text{ARI}(U, U')$ [46], [47] is used to measure the agreement between two possibilistic partitions of a set of objects, where U represents the ground truth labels for the objects in the data set, and U' denotes a partition produced by a specific algorithm. A higher value of $\text{ARI}(U, U')$ indicates that the algorithm produces a more accurate clustering result. Note that, to calculate the ARI of PCM and HOPCM, we need to harden the possibilistic partitions by setting the maximum element in each column of U' to 1, and all else to 0.

B. Experiments on the NUS-WIDE Data Set

The NUS-WIDE data set, the largest well-annotated web image set, consists of 269 648 images [36]. Each image is annotated by filtered surrounding texts that are grouped into 81 concepts. The images of the NUS-WIDE data set are downloaded from the famous photo-sharing website Flickr.com. To verify the robustness of the proposed algorithm, we collected the representative images from the NUS-WIDE data set to generate eight different data sets. Every data set consists of 10 000 images, which fall into 14 categories. The goal of this experiment is to verify the performance of the HOPCM algorithm in clustering high-quality multimedia data sets, by comparing with K-means, spectral relational clustering (SRC) [39], and nonnegative matrix factorization (NMF) [40].

For K-means, we first perform the same preprocessing method with HOPCM, i.e., the AE model, on the NUS-WIDE data set to extract visual features and textual features and then concatenate these features to form a feature vector. Finally, we use the Euclidean distance for clustering the feature vectors by K-means.

We collected different images of the NUS-WIDE data to generate eight subsets to compare the robustness performance of the four algorithms. The clustering results are presented in Tables II and III.

TABLE II
CLUSTERING RESULT IN TERMS OF $E*$

Algorithm/dataset	1	2	3	4	5	6	7	8	Overall
K-means	4.25	3.19	2.87	4.07	2.26	6.13	4.42	3.49	4.58
SRC	4.19	2.81	3.01	2.99	2.77	5.01	3.95	4.16	4.12
NMF	3.58	4.64	3.03	2.69	4.21	3.67	3.24	3.96	3.71
HOPCM	2.04	2.57	2.91	2.63	2.12	2.91	2.99	2.08	2.93

TABLE III
CLUSTERING RESULT IN TERMS OF ARI

Algorithm/dataset	1	2	3	4	5	6	7	8	Overall
K-means	0.61	0.75	0.79	0.69	0.84	0.56	0.63	0.77	0.72
SRC	0.64	0.81	0.76	0.75	0.82	0.69	0.71	0.66	0.76
NMF	0.77	0.68	0.82	0.87	0.73	0.75	0.77	0.71	0.81
HOPCM	0.91	0.84	0.94	0.91	0.88	0.92	0.82	0.84	0.90

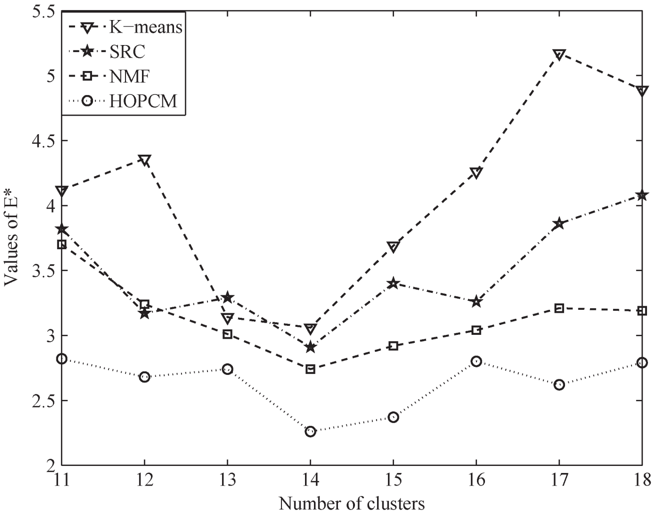


Fig. 3. Clustering result in terms of $E*$.

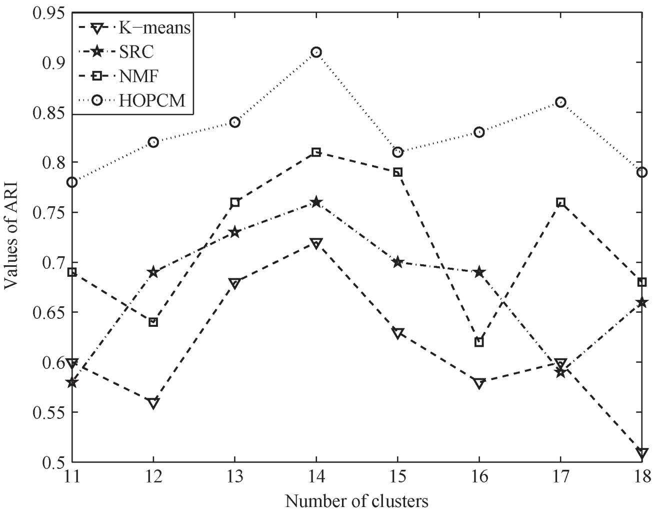


Fig. 4. Clustering result in terms of ARI.

Table II shows the clustering performance in terms of $E*$. We observe that the values of $E*$ obtained by HOPCM are the lowest in all eight data subsets and the overall data set, which demonstrates that HOPCM produces the most accurate clustering centers. K-means usually performs worst, whereas NMF achieves the better result than SRC.

In Table III, the values of ARI obtained by HOPCM are significantly higher than that obtained by the others, which demonstrates that HOPCM produces the most accurate clustering result in terms of ARI.

It is worth noting that HOPCM outperforms the others in all cases. In particular, the values of $E*$ are lower than 3.0, and the values of ARI are higher than 0.8, for all the data sets. Therefore, we can conclude that HOPCM achieves the best performance of robustness, in terms of $E*$ and ARI. Generally, the four algorithms need a prefixed number of clusters. Therefore, we compare their performance with different numbers of clusters, ranging from 11 to 18. The result is shown in Figs. 3 and 4.

Figs. 3 and 4 show that HOPCM outperforms the other three algorithms, in all cases, based on the fact that the values of $E*$ obtained by HOPCM are the lowest and the values of ARI obtained by HOPCM are significantly higher than that obtained by the others. When the number of clusters is 14, the four algorithms achieve the best performance at the same time, implying that 14 is most likely to be the correct number of clusters.

C. Experiments on the CUAVE Data Set

The CUAVE data set is composed of 36 individuals saying the digits from 0 to 9 [37]. To verify the performance of the HOPCM algorithm, we added text annotations in this data set. The goal of this experiment is to verify the performance of the HOPCM algorithm in clustering incomplete multimedia data by comparing with K-means and PCM. Note that there are no ideal clustering centers in this data set; hence, we use only ARI to evaluate the performance of HOPCM.

For the CUAVE data set, we can generate three different data subsets, which are associated with different combinations of each bimodality, i.e., an image–text subset, an image–audio subset, and a text–audio subset. Thus, we have four different data sets, including the overall CUAVE data set, in total.

For PCM and K-means, we first perform the same preprocessing method with HOPCM, i.e., the AE model, to extract visual, textual, and audio features and then concatenate these

TABLE IV
AVERAGE VALUES OF ARI ON THE IMAGE–TEXT SUBSET

Algorithm/datase	1	2	3	4	5
K-means	0.74	0.71	0.82	0.69	0.76
PCM	0.78	0.81	0.83	0.71	0.79
HOPCM	0.84	0.89	0.87	0.88	0.85

TABLE V
AVERAGE VALUES OF ARI ON THE IMAGE–AUDIO SUBSET

Algorithm/datase	1	2	3	4	5
K-means	0.75	0.81	0.66	0.72	0.73
PCM	0.73	0.79	0.86	0.75	0.72
HOPCM	0.81	0.83	0.89	0.91	0.81

TABLE VI
AVERAGE VALUES OF ARI ON THE TEXT–AUDIO SUBSET

Algorithm/datase	1	2	3	4	5
K-means	0.56	0.62	0.59	0.67	0.57
PCM	0.67	0.69	0.71	0.66	0.68
HOPCM	0.74	0.79	0.81	0.72	0.77

TABLE VII
AVERAGE VALUES OF ARI ON THE OVERALL DATA SET

Algorithm/datase	1	2	3	4	5
K-means	0.79	0.82	0.84	0.75	0.81
PCM	0.82	0.85	0.88	0.72	0.86
HOPCM	0.89	0.92	0.95	0.97	0.91

features to form a feature vector. Finally, we use the Euclidean distance for clustering the feature vectors by PCM and K-means.

We artificially create 10% missing values in the four data sets, for simulating incomplete data sets, and then cluster them using the three algorithms. For example, we randomly remove 10% pixel values to simulate an incomplete image. We generate five different incomplete data sets for each data set. Specifically, any two data sets of the five different incomplete data sets can have different missing values. Tables IV–VII present the average values of ARI obtained over ten trials on such incomplete data sets.

According to Tables IV–VII, when the missing ratio is 10%, the average values of ARI obtained by HOPCM are significantly higher than that obtained by the other two algorithms in all cases, indicating that HOPCM produces the most accurate clustering result in terms of ARI. There are two reasons for this result. On one hand, HOPCM fuses the learned features of different modalities by using the outer product to model the nonlinear correlations over multiple modalities, whereas the other two methods only concatenate the learned features, making them hard to model the nonlinear correlations over multiple modalities. On the other hand, HOPCM is able to capture the high-dimensional distribution of the multimedia data by using the tensor distance as the metric. Thus, HOPCM achieves the best performance for clustering multimedia data.

Since the clustering performance depends on the amount of missing values, we artificially create six kinds of missing ratios, which are 5%, 10%, 15%, 20%, 25%, and 30%. For every missing ratio, we perform three algorithms for ten times. Figs. 5–8 present the corresponding clustering results.

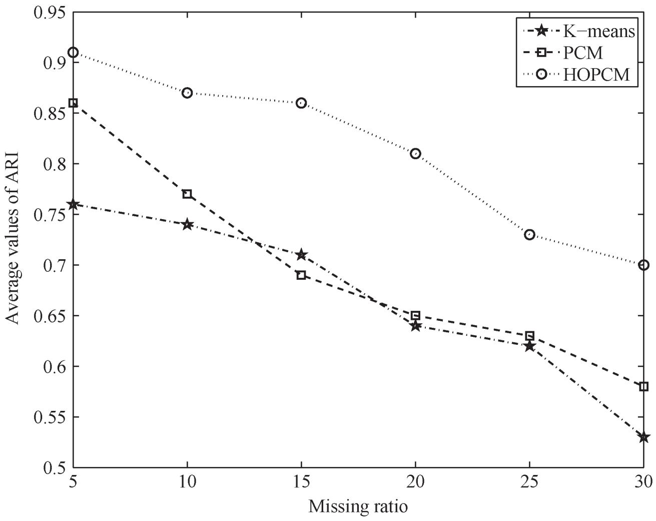


Fig. 5. Clustering result on the image–text subset.

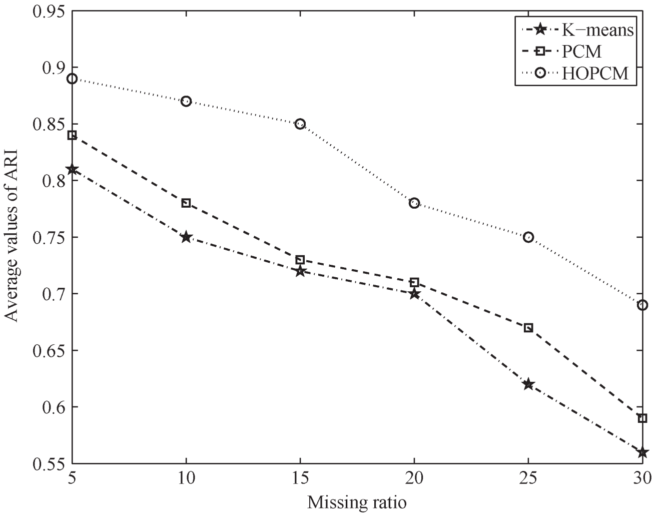


Fig. 6. Clustering result on the image–audio subset.

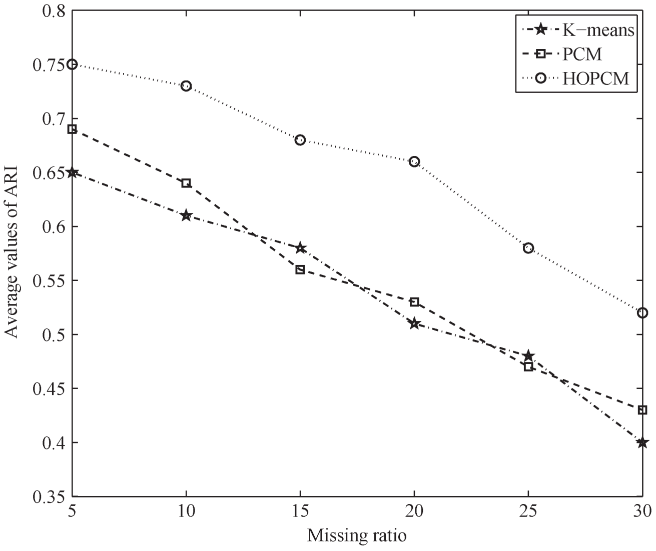


Fig. 7. Clustering result on the text–audio subset.

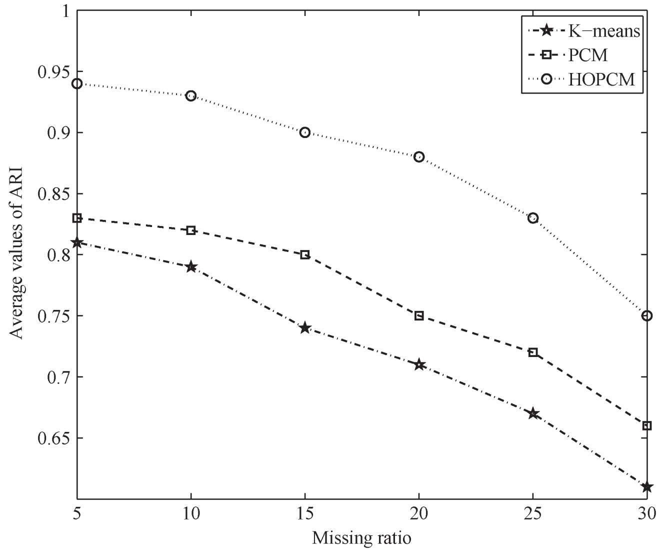


Fig. 8. Clustering result on the overall subset.

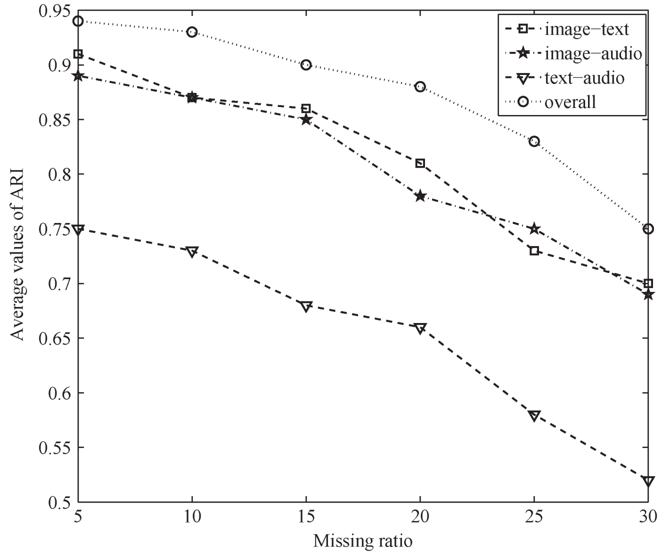


Fig. 9. Clustering result on different data sets.

According to Figs. 5–8, with the increase of the missing ratio, the average values of ARI are lower, which argues that the clustering accuracy is corrupted by missing ratios. To be more exact, the increasing missing ratio will result in the lower clustering accuracy. It is worth noting that HOPCM outperforms the others, in terms of ARI, in all cases, based on the fact that the average ARI values of HOPCM are significantly higher than that of the other two methods for six missing ratios. Hence, we can say that HOPCM produces more accurate clustering result than the other two methods, in terms of ARI, indicating that HOPCM is effective in clustering incomplete multimedia data.

Next, we investigate the relationship between the clustering result and the different combinations of modalities. We perform the HOPCM algorithm on the four different data sets under the six missing ratios. The result is shown in Fig. 9.

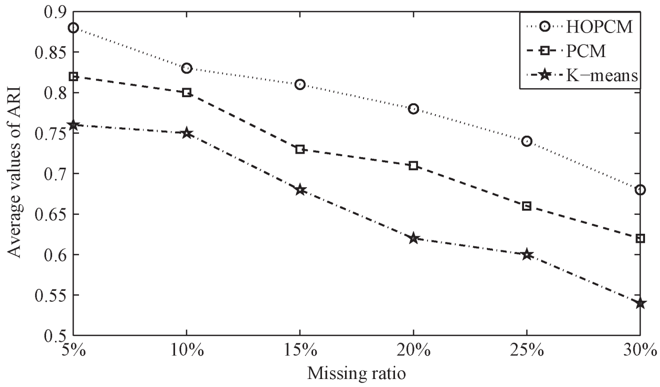


Fig. 10. Clustering result on the SNAE data set.

We observe that the best clustering performance is always obtained by performing the HOPCM algorithm on the overall data set, which demonstrates that the clustering performance of multimedia data depends on the joint features of image–text–audio modalities. In addition, the worst clustering result is obtained on the text–audio subset. The reason may be due to the fact that the text–audio modalities cannot reflect the essential features of the CUAVE data set.

D. Experiments on the SNAE Data Set

We have collected a total of 180 video clips to form the SNAE data set from YouTube [48]. The video data set is classified into four clusters: sport, new, advertisement, and entertainment. We have performed experiments to demonstrate the validity of the proposed HOPCM algorithm, by comparing with K-means and PCM, on this data set.

For PCM and K-means, we first perform the same preprocessing method with HOPCM, i.e., the AE model, to extract visual and audio features and then concatenate these features to form a feature vector. Finally, we use the Euclidean distance for clustering the feature vectors by PCM and K-means.

Similar to the experiments on the CUAVE data set, we artificially create six kinds of missing ratios, which are 5%, 10%, 15%, 20%, 25%, and 30%. For every missing ratio, we perform three algorithms for ten times. Fig. 10 presents the average values of ARI obtained over ten trials on such incomplete data sets.

According to Fig. 10, with the increase of the missing ratio, the average values of ARI are lower, which argues that the clustering accuracy is corrupted by missing ratios. It is worth noting that HOPCM outperforms the others, in terms of ARI, in all cases, based on the fact that the average ARI values of HOPCM are significantly higher than that of the other two methods for six missing ratios. Hence, HOPCM produces the best clustering result, demonstrating that HOPCM is effective in clustering incomplete video data, as well.

VI. Related Works

Multimedia clustering aims to group the multimedia data into clusters, such that the objects belonging to the same cluster should be more similar to each other than to the objects in the other clusters. Multimedia clustering is an import technique for multimedia organization, analysis, and retrieval. In recent years, many algorithms have been developed for clustering multimedia data.

For instance, Gao *et al.* developed a consistent bipartite graph copartitioning for heterogeneous data coclustering, which is the typical approach on image–text coclustering [38]. Long *et al.* presented an SRC algorithm for multitype relational data based on a graph model [39]. SRC clusters multimedia data by minimizing a reconstruction error of both an affinity matrix and a feature matrix. SRC is usually inefficient in clustering large-scale multimedia data since it needs to calculate the eigendecomposition. Xu *et al.* proposed a document clustering algorithm based on the NMF [40]. Afterward, Gu and Zhu extended the NMF method to multimedia coclustering [41]. NMF achieves a good clustering result. Chen *et al.* proposed a symmetric nonnegative matrix trifactorization algorithm, which reveals the relationship between each data item and a predefined number of clusters by deriving a latent semantic space [11]. Meng *et al.* developed a semisupervised heterogeneous fusion for multimedia data coclustering, which is called GHF-ART [13]. GHF-ART is effective in clustering multimedia data. However, it only focuses on text–image coclustering. Another type of multimedia clustering is based on information theory. For example, Bekkerman *et al.* [42] proposed the combinatorial Markov random fields (Comrafs) for the multimodal information coclustering based on the information bottleneck theory. The performance of this kind of algorithms is usually limited by the high computational complexity.

In spite of all the recent achievement in clustering multimedia data, as discussed earlier, existing methods have still some shortcomings in the following three aspects. First, they cannot cluster multimedia data with missing values since they are hard to learn features from incomplete data. Second, they do not model high nonlinear correlations over the multiple modalities of multimedia data, leading to a bad clustering result. Finally, most of them focus on only bimodalities. Therefore, their performance is limited when clustering the multimedia data with more than two modalities.

VII. Conclusion

In this paper, we propose a HOPCM algorithm for clustering incomplete multimedia data. Different from many existing techniques that can only cluster multimedia without missing values, the HOPCM algorithm is able to cluster incomplete data by designing an improved AE model for learning features of multimedia data with missing values. Another unique property of the proposed algorithm is the use of the vector outer product and the tensor distance. The vector outer product is used to fuse the learned features of different modalities to form the joint representation of multimedia data. The tensor distance encourages HOPCM to capture as much as possible the unknown high-dimensional distribution of multimedia data. The results demonstrate that HOPCM achieves significantly better clustering performance than many existing algorithms. More importantly, HOPCM is able to cluster both high-quality multimedia data and incomplete multimedia data effectively, whereas other existing methods can only cluster the high-quality multimedia data.

References

[1] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Trans. Multim.*, vol. 15, no. 4, pp. 870–883, Jun. 2013.

[2] C. Alvarez, C. Corbal, and V. Cortes, "Dynamic tolerance region computing for multimedia," *IEEE Trans. Comput.*, vol. 61, no. 5, pp. 650–665, May 2012.

[3] J. Chen, X. Xie, C. Luo, and L. Yang, "Cloud-based mobile multimedia recommendation system with user behavior information," *IEEE Syst. J.*, vol. 8, no. 1, pp. 184–193, Mar. 2014.

[4] S. Zeadally, H. Moustafa, and F. Siddiqui, "Internet Protocol Television (IPTV): Architecture, trends, and challenges," *IEEE Syst. J.*, vol. 5, no. 4, pp. 518–527, Dec. 2011.

[5] J. Ngiam *et al.*, "Multimodal deep learning," in *Proc. ICML*, 2011, pp. 689–696.

[6] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann's machines," in *Proc. NIPS*, 2012, pp. 2222–2230.

[7] S. Lew, N. Sebe, C. Djeraba, and J. Ramesh, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multim. Comput. Appl.*, vol. 2, no. 1, pp. 1–19, Feb. 2006.

[8] L. D. P Mendes, J. J. P. C. Rodrigues, J. Lloret, and S. Sendra, "Cross-layer dynamic admission control for cloud-based multimedia sensor networks," *IEEE Syst. J.*, vol. 8, no. 1, pp. 235–246, Mar. 2014.

[9] W. Wang, C. Huang, and S. Wang, "VQ applications in steganographic data hiding upon multimedia images," *IEEE Syst. J.*, vol. 5, no. 4, pp. 528–537, Dec. 2011.

[10] M. Rege, M. Dong, and J. Hua, "Graph theoretical framework for simultaneously integrating visual and textual features for efficient web image clustering," in *Proc. ACM WWW*, 2008, pp. 317–326.

[11] Y. Chen, L. Wang, and M. Dong, "Non-negative matrix factorization for semisupervised heterogeneous data coclustering," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1459–1474, Oct. 2010.

[12] R. Bekkerman and J. Jeon, "Multi-modal clustering for multimedia collections," in *Proc. IEEE CVPR*, 2007, pp. 1–8.

[13] L. Meng, A. Tan, and D. Xu, "Semi-supervised heterogeneous fusion for multimedia data co-clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2293–2306, Sep. 2014.

[14] Q. Zhang and Z. Chen, "A distributed weighted possibilistic c-means algorithm for clustering incomplete big sensor data," *Int. J. Distrib. Sens. Netw.*, vol. 2014, 2014, Art. ID. 430814.

[15] 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.

[16] B. Zhao, Y.-C. Tam, and J. Zheng, "An autoencoder with bilingual sparse features for improved statistical machine translation," in *IEEE ICASSP*, 2014, pp. 7103–7107.

[17] J. Ngiam *et al.*, "On optimization methods for deep learning," in *Proc. ACM ICML*, 2011, pp. 265–272.

[18] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, Jul. 2011.

[19] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising autoencoders as generative models," in *Proc. NIPS*, 2013, pp. 899–907.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[21] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. ACM ICML*, 2011, pp. 833–840.

[22] S. Rifai *et al.*, "Higher order contractive auto-encoder," in *Proc. ECML PKDD*, 2011, pp. 645–660.

[23] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 96–110, May 1993.

[24] J. Zhang and Y. Leung, "Improved possibilistic c-means clustering algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 2, pp. 209–217, Apr. 2004.

[25] M. Yang and C. Lai, "A robust automatic merging possibilistic clustering method," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 26–41, Feb. 2011.

[26] Z. Xie, S. Wang, and F. L. Chung, "An enhanced possibilistic C-Means clustering algorithm EPCM," *Soft Comput.*, vol. 12, no. 6, pp. 593–611, Apr. 2008.

[27] M. Filippone, F. Masulli, and S. Rovetta, "Applying the possibilistic c-means algorithm in kernel-induced spaces," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 572–584, Jun. 2010.

[28] A. Schneider, "Weighted possibilistic c-means clustering algorithms," in *Proc. IEEE FUZZ*, 2000, pp. 176–180.

[29] Q. Zhang and Z. Chen, "A weighted kernel possibilistic c-means algorithm based on cloud computing for clustering big data," *Int. J. Comm. Syst.*, vol. 27, no. 9, pp. 1378–1391, Sep. 2014.

[30] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. William. "Learning representations of back-propagation errors," *Nature*, vol. 323, pp. 533–536, 1986.

[32] A. Cichocki, "Era of big data processing: A new approach via tensor networks and tensor decompositions," *Arxiv Preprint Arxiv: 1403.2048*, 2014.

[33] Y. Liu, Y. Liu, and K. Chan, "Tensor distance based multilinear locality-preserved maximum information embedding," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1848–1854, Nov. 2010.

[34] C. T. Havens, C. J. Bezdek, C. Leckie, and O. L. Hall, "Fuzzy c-means algorithms for very large data," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 6, pp. 1130–1147, May 2012.

[35] X. Han, S. Xia, and Y. Zhou, "Kernel-based fast improved possibilistic C-means clustering method," *Comput. Eng. Appl.* vol. 47, no. 6, pp. 176–180, 2011.

[36] T. Chua *et al.*, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM CIVR*, 2009, pp. 1–9.

[37] E. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy, "CUAVE: A new audio-visual database for multimodal human-computer interface research," in *Proc. IEEE ICASSP*, 2002, pp. 2017–2020.

[38] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," in *Proc. ACM SIGKDD*, 2005, pp. 41–50.

[39] B. Long, X. Wu, Z. Zhang, and P. Yu, "Spectral clustering for multi-type relational data," in *Proc. ACM ICML*, 2006, pp. 585–592.

[40] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. ACM SIGIR*, 2003, pp. 268–273.

[41] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *Proc. ACM SIGKDD*, 2009, pp. 359–367.

[42] R. Bekkerman, M. Sahami, and E. Learned-Miller, "Combinatorial Markov random fields," in *Proc. ECML PKDD*, 2006, pp. 30–41.

[43] L. Zhou, H. Wang, and M. Guizani, "How mobility impacts video streaming over multi-hop wireless networks," *IEEE Trans. Commun.*, vol. 60, no. 7, pp. 2017–2028, Jul. 2012.

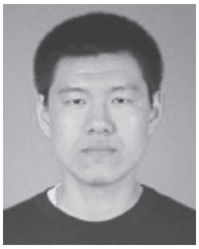
[44] L. Zhou, M. Chen, Y. Qian, and H. Chen, "Fairness resource allocation in blind wireless multimedia communications," *IEEE Trans. Multim.*, vol. 15, no. 4, pp. 946–956, Jun. 2013.

[45] L. Zhou and B. Zheng, "Joint physical-application layer security for wireless multimedia delivery," *IEEE Commun. Mag.*, vol. 52, no. 3, pp. 66–72, Amr. 2014.

[46] W. Ran, "Objective criteria for the evaluation of clustering method," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

[47] L. Hubert and P. Arabi, "Comparing partition," *J. Class.*, vol. 2, no. 1, pp. 193–218, 1985.

[48] [Online]. Available: http://www.youtube.com



Qingchen Zhang received the bachelor's degree and the master's degree from Southwest University, Chongqing, China. He is currently working toward the Ph.D. degree with the School of Software, Dalian University of Technology (DLUT), Dalian, China.

His research interests include big data and deep learning.

Laurence T. Yang received the B.SC. degree from Tsinghua University, Beijing, China and the Ph.D. degree from University of Victoria, Vancouver, Canada. He is a Professor in computer science with the Huazhong University of Science and Technology, Wuhan, China, and also with St. Francis Xavier University, Antigonish, NS, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous computing, and big data.

Zhikui Chen received the B.Sc. degree from Chongqing Normal University, Chongqing, China and the Ph.D. degree from Chongqing University, Chongqing, China. He is a Professor with Dalian University of Technology, Dalian, China, leading the Institute of Ubiquitous Networks and Computing. His research area includes Internet of things and big data processing.

Prof. Chen is a Member of the IEEE Computer Society.

Feng Xia received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China.

He is currently a Professor and Ph.D. Supervisor with the School of Software, Dalian University of Technology, Dalian, China. He has authored/coauthored one book and over 160 scientific papers in international journals and conferences [with more than 80 indexed by Thomson Reuters Scientific (ISI) Science Citation Index Expanded]. His research interests include social computing, mobile computing, and cyber-physical systems.

Dr. Xia is a Senior Member of the IEEE Computer and IEEE Systems, Man, and Cybernetics Societies and a member of ACM and ACM SIGMOBILE. He is the (Guest) Editor of several international journals. He serves as a General Chair, Program Committee (PC) Chair, Workshop Chair, Publicity Chair, or PC Member of a number of conferences.