Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 4356127, 8 pages
http://dx.doi.org/10.1155/2016/4356127

Research Article

A High-Order CFS Algorithm for Clustering Big Data

Fanyu Bu,[1,2] Zhikui Chen,[1] Peng Li,[1] Tong Tang,[3] and Ying Zhang[4]

[1]*School of Software Technology, Dalian University of Technology, Dalian 116620, China*
[2]*School of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot 010070, China*
[3]*Department of Student Work, Southwest University, Chongqing 400715, China*
[4]*College of Business Administration, Dalian University of Finance and Economics, Dalian 116622, China*

Correspondence should be addressed to Fanyu Bu; bufanyu@imufe.edu.cn

Received 6 May 2016; Accepted 26 June 2016

Academic Editor: Beniamino Di Martino

With the development of Internet of Everything such as Internet of Things, Internet of People, and Industrial Internet, big data is being generated. Clustering is a widely used technique for big data analytics and mining. However, most of current algorithms are not effective to cluster heterogeneous data which is prevalent in big data. In this paper, we propose a high-order CFS algorithm (HOCFS) to cluster heterogeneous data by combining the CFS clustering algorithm and the dropout deep learning model, whose functionality rests on three pillars: (i) an adaptive dropout deep learning model to learn features from each type of data, (ii) a feature tensor model to capture the correlations of heterogeneous data, and (iii) a tensor distance-based high-order CFS algorithm to cluster heterogeneous data. Furthermore, we verify our proposed algorithm on different datasets, by comparison with other two clustering schemes, that is, HOPCM and CFS. Results confirm the effectiveness of the proposed algorithm in clustering heterogeneous data.

1. Introduction

With the rapid development of the Internet of Things, Internet of People, and Industrial Internet, big data analytics and mining have become a hot topic [1]. One widely used technique of big data analytics and mining is clustering that aims to group data into several clusters according to similarities between the data objects [2]. In 2014, Laio and Rodriguez proposed a novel clustering algorithm by fast search and finding of density peaks (CFS) published in Science Magazine [3]. CFS is the most potential clustering technique because of its efficiency and high accuracy. However, CFS is limited in clustering big data because it cannot cluster heterogeneous data which is prevalent in big data.

Heterogeneous data, different from the homogeneous data containing only one type of objects, involves multiple interrelated types of objects [4]. Moreover, a heterogeneous data object is usually of complex correlations among different modalities. Therefore, heterogeneous data poses important challenges on clustering techniques. Recently, researchers have proposed some algorithms to cluster heterogeneous data [5]. One of this type is based on the graph partition, for instance, the bipartite spectral algorithm, which clusters heterogeneous data by optimizing a unified objective function. However, this kind of methods is usually of low efficiency for clustering big datasets since they need to solve an eigen-decomposition procedure. Another typical algorithm based on the nonnegative matrix factorization, such as SS-NMF, clusters heterogeneous data by revealing the relationships between different objects in a semantic space. In addition, Comrafs is developed for clustering heterogeneous data by constructing the Markov Rand Fields. Since this method is of high computational complexity, it is limited for large-scale heterogeneous data clustering. These algorithms could cluster heterogeneous data; however, they are hard to achieve desired clustering results since they do not model the high nonlinear correlations over multiple types of heterogeneous data objects effectively. Moreover, they are of high time complexity, leading to low efficiency in clustering heterogeneous data.

In this paper, we propose a high-order CFS algorithm (HOCFS) for clustering heterogeneous data based on the dropout deep learning model. The dropout deep learning model was proposed by Hinton to prevent overfitting [6]. It is especially useful in training large networks with small

amount of samples. However, the dropout sets the same omitting probability with 0.5 in each hidden layer of the deep learning model, resulting in its ineffectiveness. Aiming at this problem, we propose an adaptive dropout deep learning model, which sets the omitting probability of each hidden layer according to the relationship between the omitting probability and the layer opposition. Then, we applied the proposed adaptive dropout deep learning model in feature learning for each type of data of every heterogeneous data object. Next, the algorithm uses the vector outer product to fuse the learned features to form a feature tensor for each heterogeneous data object. Finally, since the tensor distance can not only measure the distance between every two heterogeneous samples but also reveal the intrinsic correlations between different coordinates in the high-order tensor space, the tensor distance is applied to the CFS algorithm for clustering heterogeneous data represented by fused features.

Finally, we compare our proposed algorithm with two representative data clustering techniques, namely, HOPCM and CFS, on two datasets, namely, NUS-WIDE and CUAVE in terms of E^* and Rand Index (RI).

Therefore, the contributions of the paper are summarized as the following three aspects:

(i) Current dropout deep learning models are of low effectiveness and efficiency in learning features for heterogeneous data. To tackle this problem, the paper proposes an adaptive dropout deep learning model to learn features for each type of data and then fuses the learned features to form a feature tensor for each heterogeneous data object.

(ii) To measure the similarity between heterogeneous data objects in high-order tensor space, the paper applies the tensor distance in the clustering process.

(iii) Conventional CFS algorithm cannot cluster heterogeneous data directly because it works in the vector space. The paper extends the CFS algorithm from the vector space to the tensor space for clustering heterogeneous data represented by the feature tensors.

2. Preliminaries

This section presents the technique preliminaries about our scheme, including the stacked autoencoder, dropout, and the CFS clustering algorithm. The stacked autoencoder is presented first, followed by the CFS clustering algorithm.

2.1. Stacked Autoencoder (SAE) and Dropout. The stacked autoencoder (SAE) that is one important example of deep learning models has been widely employed in supervised feature learning for many applications [7]. SAE is built to learn hierarchical features of data by stacking multiple basic autoencoders (BAEs) as shown in Figure 1.

As the typical module of a stacked autoencoder, a basic autoencoder (BAE) [8] learns a hidden representation h of the input data x by an encoding function f:

$$h = f_\theta\left(W^{(1)}x + b^{(1)}\right). \tag{1}$$

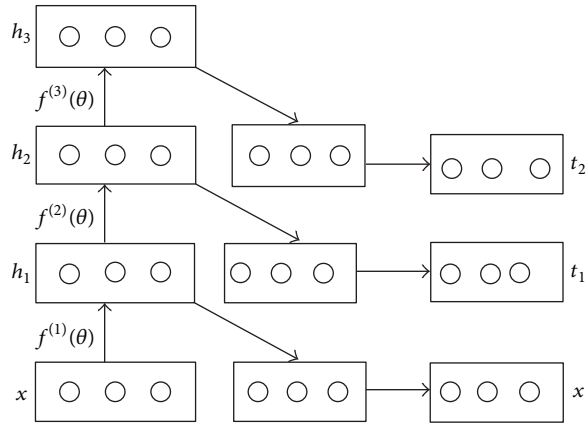

Figure 1: The architecture of the stacked autoencoder.

Then, BAE reconstructs the input from the hidden representation h to a reconstruction y by a decoding function s:

$$y = s_\theta\left(W^{(2)}x + b^{(2)}\right), \tag{2}$$

where $\theta = (W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)})$ denotes the parameter of the autoencoder and the functions f and s typically adopt the sigmoid function: $f(x) = 1/(1 + e^{-x})$.

To train the parameter of the autoencoder, an objective function with a weight-decay that is used to prevent overfitting is defined as follows:

$$J(\theta) = \left(\sum_{x\in D} L\left(x, s\left(f\left(x\right)\right)\right)\right) + \lambda\sum_{ij} W_{ij}^2, \tag{3}$$

where L is the reconstruction error and λ is a hyperparameter used to control the strength of the regularization.

The stacked autoencoder is a full-connected model and it involves many redundant connections. Therefore, it usually produces overfitting in the real applications. Aiming at this problem, Hinton proposed dropout to reduce the overfitting by preventing coadaption of feature detectors in deep learning models. It randomly omits half of the feature detectors on each training sample to prevent a hidden unit from relying on other hidden units being present. Dropout was proved to be especially effective and efficient in training a large neural network with a small training set.

2.2. Clustering by Fast Search and Finding of Density Peaks (CFS). CFS is the latest clustering algorithm proposed by Laio and Rodriguez in Science Magazine in 2014 [3]. It is highly robust and efficient. More importantly, it can find clusters of arbitrary shape and determine the number of clusters automatically. Several experiments have demonstrated its superiority in the efficiency and effectiveness over the previous algorithms for clustering large amounts of data. Therefore, it has become the most potential algorithm for clustering big data.

The key of the CFS algorithm lies in the characterization of cluster centers. Particularly, the algorithm basically assumes that cluster centers should be surrounded by neighbor objects with lower local density and be more far away

from other objects with a higher local density. Based on this assumption, CFS defines two quantities for every data object x_i, the local density ρ_i and the minimum distance δ_i from any other object with higher density, in

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right)$$

$$\chi(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} \left(d_{ij}\right),$$

where d_c represents a cutoff distance. According to Laio and Rodriguez, d_c can be set to the biggest 2% of all the distances between every two objects to get a good clustering result. For the object x_i with the highest density, its distance δ_i is taken as $\delta_i = \max_j(d_{ij})$.

In the CFS algorithm, cluster centers are recognized as the objects with the large value of γ that is defined in

$$\gamma_i = \rho_i \times \delta_i. \tag{5}$$

3. Problem Statement

Consider a dataset with n heterogeneous data objects $X = \{x_1, x_2, \ldots, x_n\}$ and assume that each object can be represented by a feature tensor. The task of heterogeneous data clustering is to classify the dataset into groups according to their similarity such that the objects belonging to the same cluster share similarity as much as possible. Based on the analysis in the previous parts, heterogeneous data poses a large number of challenges on the clustering techniques. We discuss the key issues in three following aspects:

(1) *Feature Learning of Heterogeneous Data*. Feature learning is the fundamental step for heterogeneous data clustering. In fact, many feature learning algorithms, especially some methods based on deep learning, have been well studied in recent years. However, most of them are hard to learn features for heterogeneous data. Although the deep computation model can learn features for heterogeneous data, it is of low accuracy and efficiency since it cannot avoid overfitting.

(2) *Similarity Measurement for Heterogeneous Data*. Similarity measurement is the key to one clustering technique. There are a lot of metrics for measuring the similarity between two objects. However, they can only measure the distance between homogeneous objects represented by feature vectors because they work in the vector space. A heterogeneous object is typically represented by a feature tensor, making most of current metrics hard to calculate the similarity for heterogeneous data objects.

(3) *Clustering Technique for Heterogeneous Data*. Typically, a heterogeneous object is represented by a feature tensor. However, most of clustering techniques including the CFS algorithm work only in the vector space, resulting in failure to cluster heterogeneous data in the high-order tensor space.

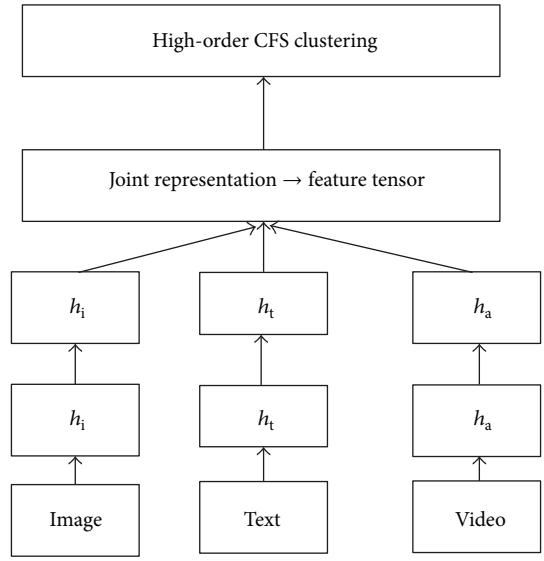

Figure 2: The architecture of the proposed scheme.

4. High-Order CFS Algorithm Based on Dropout Deep Computation Model

In this section, we describe the details of the proposed high-order CFS algorithm based on the dropout deep learning model for clustering heterogeneous data. The proposed algorithm works in three stages: unsupervised feature learning, feature fusion, and high-order clustering, which is shown in Figure 2.

In the first stage, each type of data in the heterogeneous dataset is separately learned by the proposed adaptive dropout deep learning model. In the second stage, the proposed algorithm uses the vector outer product to fuse the learned futures to form a feature tensor as the joint representation of each object. Finally, the proposed algorithm extends the conventional CFS technique from the vector space to the tensor space for clustering the heterogeneous dataset.

4.1. Feature Learning Based on the Adaptive Dropout Deep Learning Model. In the dropout deep learning model, each hidden unit is randomly omitted from the network always with a constant probability of 0.5. This way will ignore the relationship between the omitting probability and the layer opposition, resulting in a low effectiveness of deep learning models in heterogeneous data feature learning. A large number of studies demonstrate that the fundamental layers of a deep architecture share many common characters, implying that the dropout in the lower layers has more generalization function than that in higher layers. Therefore, the omitting probability of the dropout should decay with the layers becoming higher.

Based on the above analysis, we propose an adaptive dropout deep learning model by defining a distribution model of the omitting probability y of dropout as the following function:

$$y = f(l) = \begin{cases} -0.1l + 0.05n + 0.5 & n = 2k \ \ (k = 1, 2, \ldots) \\ -0.1l + 0.05n + 0.55 & n = 2k - 1 \ \ (k = 1, 2, \ldots), \end{cases} \tag{6}$$

where $n \leq 9$ denotes the number of hidden layers in the deep learning model and l represents the position of the layer.

Function (6) has the following properties:

(1) it is monotonically decreasing.

(2) The omitting probability is 0.5 for the middle hidden layer.

(3) The omitting probability is always in $(0, 1)$ for $x = 1, 2, \ldots, n$.

Proof. (1) By the assumption, function $f(l)$ is continuously differentiable and we may write

$$f'(l) = -0.1 < 0, \tag{7}$$

which implies that (6) is a strictly decreasing function. Particularly, the omitting probability of the dropout should decay with the layers becoming higher.

(2) When $n = 2k$ $(k = 1, 2, \ldots)$,

$$f\left(\frac{n}{2}\right) = -0.1 \times \frac{n}{2} + 0.05 \times n + 0.5 = 0.5. \tag{8}$$

When $n = 2k - 1$ $(k = 1, 2, \ldots)$,

$$f\left(\frac{n+1}{2}\right) = -0.1 \times \frac{n+1}{2} + 0.05 \times n + 0.55 = 0.5, \tag{9}$$

which proves that the omitting probability is 0.5 for the middle hidden layer.

(3) Based on property (1),

$$f(n) \leq f(l) \leq f(1) \quad (1 \leq n \leq 9). \tag{10}$$

Then,

$$f(n) = \begin{cases} -0.05n + 0.5 \geq -0.05 \times 8 + 0.5 = 0.1 > 0 \\ -0.05n + 0.55 \geq -0.05 \times 9 + 0.55 = 0.1 > 0 \end{cases} \tag{11}$$

$$f(1) = 0.05n + 0.45 \leq 0.05 \times 9 + 0.45 = 0.9 < 1.$$

Therefore, the omitting probability is always in $(0, 1)$ for $l = 1, 2, \ldots, n$. □

We can get the adaptive dropout deep learning model by applying the distribution function of the omitting probability to the deep learning model outlined in Algorithm 1.

In the proposed high-order CFS algorithm, the adaptive dropout deep learning model is used to learn features of each type of data of the heterogeneous data.

Input: $\{(X^{(i)}, Y^{(i)})\}$, $1 \leq i \leq N$, $\text{iterater}_{\max}$, η, threshold
Output: $\theta = \{W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)}\}$

(1) Randomly initialize all $\theta = \{W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)}\}$;
(2) $y = f(l)$;
(3) **for** iteration $= 1, 2, \ldots, \text{iterater}_{\max}$ **do**
(4) **for** example $= 1, 2, \ldots, N$ **do**
(5) **for** $j = 1, 2, \ldots, m$ **do**
(6) $z_j^{(2)} = w_{ji}^{(1)} \cdot x_i + b_j^{(1)}$;
(7) $a_j^{(2)} = f(z_j^{(2)})$;
(8) $\text{mark}\{i\} = \text{rand}(\text{size}(a^{(2)}) > y)$;
(9) $a^{(2)} = a^{(2)} \cdot * \text{mark}\{i\}$;
(10) **for** $i = 1, 2, \ldots, n$ **do**
(11) $z_i^{(3)} = w_{ij}^{(2)} \cdot a_j^{(2)} + b_i^{(2)}$;
(12) $a_i^{(3)} = f(z_i^{(3)})$;
(13) **for** $i = 1, 2, \ldots, n$ **do**
(14) $\sigma_i^{(3)} = -(y - a_i^{(3)}) \cdot f'(z_i^{(3)})$;
(15) **for** $j = 1, 2, \ldots, m$ **do**
(16) $\sigma_j^{(2)} = (\sum_{i=1}^{n} w_{ij}^{(2)} \cdot \sigma_i^{(3)}) \cdot f'(z_j^{(2)})$;
(17) $\sigma^{(2)} = \sigma^{(2)} \cdot * [\text{ones}(\text{size}(\sigma^{(2)}, 1), 1) \text{mark}\{i\}]$;
(18) **for** $i = 1, 2, \ldots, n$ **do**
(19) $b_i^{(2)} = \sigma_i^{(3)}$;
(20) **for** $j = 1, 2, \ldots, m$ **do**
(21) $\Delta w_{ij}^{(2)} = a_j^{(2)} \cdot \sigma_i^{(3)}$;
(22) **for** $j = 1, 2, \ldots, m$ **do**
(23) $b_j^{(1)} = \sigma_j^{(2)}$;
(24) **for** $i = 1, 2, \ldots, n$ **do**
(25) $\Delta w_{ji}^{(1)} = x_i \cdot \sigma_j^{(2)}$;

Algorithm 1: Adaptive dropout backpropagation neural network learning algorithm.

4.2. Feature Fusion Using Vector Outer Product. The vector outer product is one of the widely used operations in mathematics, denoted by ⊗. If A is an m-dimension vector and B is an n-dimension vector, their outer product will produce an $m \times n$ matrix C; $C = A \otimes B$. Each entry in the matrix C is defined as $c_{ij} = a_i \cdot b_j$, where a_i and b_j are one entry in vectors A and B, respectively. One example of the vector outer product is as shown in (6):

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \otimes \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} a_1 b_1 & a_2 b_1 & a_3 b_1 \\ a_1 b_2 & a_2 b_2 & a_3 b_2 \\ a_1 b_3 & a_2 b_3 & a_3 b_3 \\ a_1 b_4 & a_2 b_4 & a_3 b_4 \end{bmatrix}. \tag{12}$$

More generally, the outer product of n vectors $A_1 \in R^{I_1}$, $A_2 \in R^{I_2}, \ldots, A_n \in R^{I_n}$ will produce an n-order tensor $B \in R^{I_1 \times I_2 \times \cdots \times I_n}$, $B = A_1 \otimes A_2 \otimes \cdots \otimes A_n$, in which each entry is defined as $b_{i_1 i_2 \cdots i_n} = a_{1_{i_1}} \cdot a_{2_{i_2}} \cdot \ldots \cdot a_{n_{i_n}}$.

After using the adaptive deep learning model to learn features of heterogeneous data, each type of data can be represented by a feature vector. Particularly, for the heterogeneous dataset in which each object consists of one image, one text, and one piece of video, three vectors, a, b, and c, are used to represent the feature vectors learned from the adaptive

Input $X = \{X_1, X_2, \ldots, X_N\}, d_c$
Output cl[n], center[k]
(1) **for** $i = 1, 2, \ldots, n$ **do**
(2) **for** $j = i + 1, i + 2, \ldots, n$ **do**
(3) $d_{ij} = \sqrt{(X_i - X_j)^T G (X_i - X_j)}$;
(4) **for** $i = 1, 2, \ldots, n$ **do**
(5) $\rho_i = \sum_j \chi(d_{ij} - d_c)$;
(6) **for** $i = 1, 2, \ldots, n$ **do**
(7) $\delta_i = \min_{j:\rho_j > \rho_i} \{d_{ij}\}$;
(8) $\gamma_i = \rho_i \times \delta_i$;
(9) Select clustering centers according to γ_i;
(10) **for** $i = 1, 2, \ldots, n$ **do**
(11) cl[i] = $\min_{j:\text{centers}[k]} \{d_{ij}\}$;

Algorithm 2: High-order CFS clustering algorithm.

dropout deep learning model, respectively. In this subsection, such feature vectors are fused by the vector outer product to form one feature tensor X for joint representation of one object in the heterogeneous dataset according to the following rules:

(1) For the object with only one image and one text, its feature tensor is represented by $X = a \otimes b$.

(2) For the object with only one image and one piece of video, its feature tensor is represented by $X = a \otimes c$.

(3) For the object with only one text and one piece of video, its feature tensor is represented by $X = b \otimes c$.

(4) For the object with only one image, one text, and one piece of video, its feature tensor is represented by $X = a \otimes b \otimes c$.

4.3. The High-Order CFS Clustering. As discussed in Section 2, the conventional CFS algorithm cannot cluster heterogeneous data directly because it works in the vector space while each object in the heterogeneous dataset is represented by a feature tensor. To tackle this problem, we propose a high-order CFS algorithm for clustering heterogeneous data.

To calculate the distance between two points in high-order tensor space, represented by two tensors, $X, Y \in R^{I_1 \times I_2 \times \cdots \times I_N}$, they need to be unfolded to the corresponding vectors. In detail, the item $X_{i_1 i_2 \cdots i_N}$ is unfolded to x_l by $l = i_1 + \sum_{j=2}^{N} \prod_{t=1}^{j-1} I_t$.

The proposed high-order CFS clustering algorithm (HOCFS) based on the feature tensor is outlined in Algorithm 2.

5. Performance Evaluation of Adaptive Dropout Model

In this part, we assess the adaptive dropout deep learning model on the STL-10 and CIFAR-10 datasets by comparison with the conventional dropout model.

5.1. Experiments on the STL-10 Dataset. We initially explored the effectiveness of adaptive dropout using STL-10, a widely used benchmark for machine learning algorithms. It contains 500 training images, 800 testing images that are grouped by 10 classes, and 100000 unlabeled images for unsupervised learning. We combine the adaptive dropout distribution model with stacked autoencoders to train two deep learning models. One has 4 hidden layers while the other has 5 hidden layers. Both of them have one logistic regression layer on the top. For the adaptive dropout deep learning model, we use the proposed algorithm to set the omitted rate of hidden units while setting omitted rate of 0.5 of hidden units for the conventional dropout deep learning model. The classification results are presented in Figures 3 and 4.

From Figures 3 and 4, the classification error decreases with the epoch increasing. The classification error produced by the adaptive dropout model is lower than that produced by the conventional dropout model. Particularly, we achieved

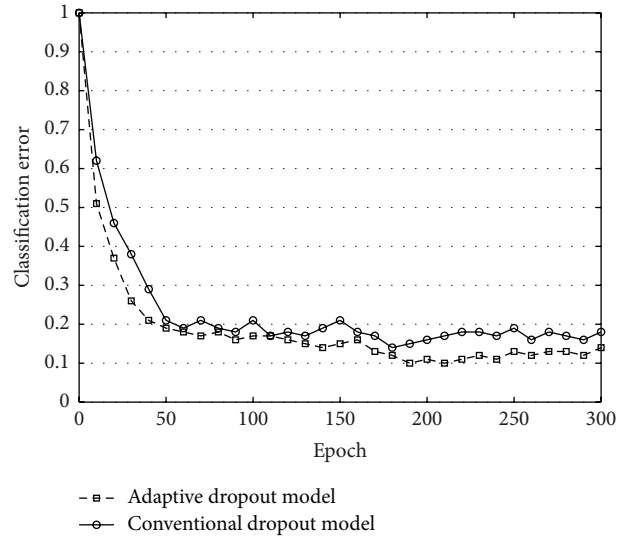

Figure 3: Classification result on STL-10 with 4 hidden layers.

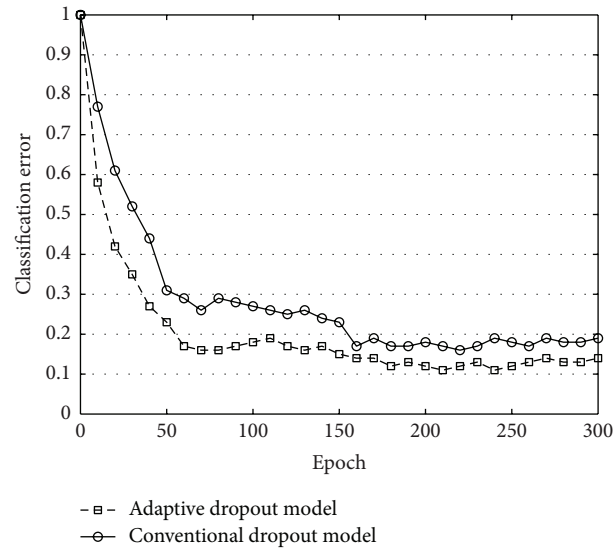

Figure 4: Classification result on STL-10 with 5 hidden layers.

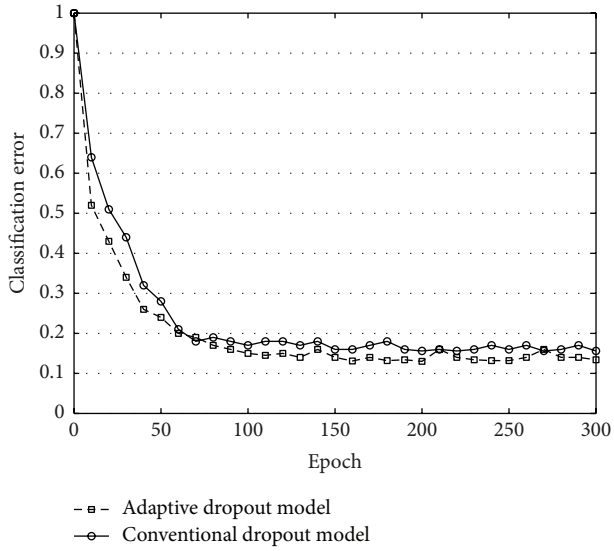

Figure 5: Classification result on CIFAR-10.

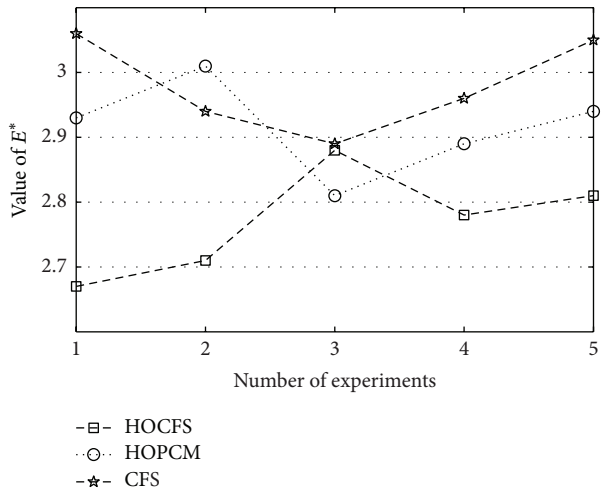

Figure 6: Clustering result on NUS-WIDE in terms of E^*.

the best classification error rate of 0.10 by using adaptive dropout model with 4 hidden layers while the best classification error rate given by the conventional dropout model is 0.12, which indicates that our proposed model performs better than the conventional dropout model in classifying the STL-10 dataset.

5.2. Experiments on the CIFAR-10 Dataset. CIFAR-10 is a benchmark task for object recognition, consisting of 60000 color images in 10 groups, with 6000 images per group. These images were labeled by hand to produce 50000 training images and 10000 test images. We built a classification network with three convolutional layers and three pooling and two fully connected layers to explore the effectiveness of the adaptive dropout model on CIFAR-10 dataset. Each convolutional layer has an exclusive ReLU layer and a dropout layer. Specially for the adaptive dropout deep learning model, we use the proposed algorithm to set the omitted rate of hidden units while setting omitted rate of 0.5 of hidden units for the conventional dropout deep learning model. The classification results are presented in Figure 5.

From Figure 5, the error rate produced by the adaptive dropout model is lower than that produced by the conventional dropout model in most cases. More importantly, using the conventional dropout model gives the best error rate of 0.156. This is reduced to 0.136 by using the adaptive dropout model, which implies that the proposed model works much better than the conventional dropout model for CIFAR-10.

6. Performance Evaluation of the High-Order CFS Algorithm

In this part, we evaluate the high-order CFS clustering algorithm by comparison with the HOPCM algorithm and the conventional CFS algorithm on two representative heterogeneous datasets, namely, NUS-WIDE and CUAVE, in terms of E^* and Rand Index (RI).

HOCPM was developed in 2015 for clustering heterogeneous data by combining the autoencoder model and the possibilistic c-means algorithm [9]. For the conventional CFS algorithm, we perform the same preprocessing step with our proposed algorithm. Particularly, we first use the adaptive dropout deep learning model to learn features of texts, audios, and images of each object and then form a feature vector for the object by concatenating the learned features. Finally, the Euclidean distance is applied for the conventional CFS algorithm to cluster the heterogeneous dataset where each object is represented by a learned feature vector.

The evaluation criteria are described in Section 6.1, followed by the experimental results.

6.1. Experiments on the NUS-WIDE Dataset. The NUS-WIDE dataset is the biggest image set, consisting of 269, 648 annotated images. To compare the proposed algorithm with the HOPCM algorithm and the conventional CFS algorithm fairly, we use the same image dataset collected from the NUS-WIDE with literature [9], which consists of 8 different subsets, each with 10,000 annotated images falling into 14 categories.

First, we carried out the experiments on the overall image set for five times. The clustering results are shown in Figures 6 and 7.

Figure 6 shows the clustering result in terms of E^* on the overall dataset. We observe that the proposed algorithm got the lowest values of E^* in most cases, which implies that the proposed algorithm produced the most accurate clustering centers.

From Figure 7, HOCFS produced the highest values of RI in most cases, which indicates that HOCFS performs best in clustering NUS-WIDE dataset. Moreover, the conventional CFS algorithm performs worst in terms of E^* and RI, demonstrating that the proposed algorithm could effectively capture the complex correlations over the heterogeneous data by applying the vector outer product to feature fusion and using tensor distance to measure the similarity between two objects.

Next, we carried out the experiment on the 8 subsets for 5 times to evaluate the robustness of the clustering algorithms.

Table 1: Clustering result on NUS-WIDE in terms of E^*.

Algorithm/subset	1	2	3	4	5	6	7	8
CFS	2.64	3.01	2.99	3.04	2.73	3.02	3.08	2.82
HOPCM	2.04	2.57	2.91	2.63	2.12	2.91	2.99	2.08
HOCFS	1.96	2.24	2.37	2.28	1.95	2.16	2.39	2.01

Table 2: Clustering result on NUS-WIDE in terms of RI.

Algorithm/subset	1	2	3	4	5	6	7	8
CFS	0.86	0.79	0.87	0.82	0.76	0.79	0.83	0.69
HOPCM	0.91	0.84	0.93	0.91	0.88	0.92	0.82	0.84
HOCFS	0.95	0.84	0.94	0.95	0.93	0.96	0.89	0.91

Tables 1 and 2 present the average clustering results of 5 times on every subset.

From Tables 1 and 2, the average values of E^* obtained by HOCFS are lowest for each subset while the average values of RI obtained by HOCFS are significantly larger than that obtained by HOPCM and CFS. In other words, the proposed algorithm produced the best clustering results in terms of E^* and RI for NUS-WIDE dataset.

6.2. Experiments on the CUAVE Dataset. CUAVE is a typical multimodal dataset consisting of some digits, 0 to 9, reported by 36 individuals. To assess HOCFS for clustering heterogeneous data, we added some annotations to each object as the literature [9].

We first carried out the experiment on the CUAVE dataset for 5 times to judge HOCFS for clustering heterogeneous data in terms of RI. The result is presented in Figure 8.

According to Figure 8, the value of RI obtained by HOCFS is highest for each experiment, implying that the proposed algorithm produced the best clustering result for the CUAVE dataset in terms of RI. On the one hand, the proposed algorithm uses the hybrid stacked autoencoder model to learn features of each object in the CUAVE dataset while HOPCM only uses the basic autoencoder model to learn features, leading to the more accurate clustering result produced by the proposed algorithm compared to HOPCM. On the other hand, HOCFS fuses the learnt features of each modality for capturing the nonlinear correlations over multiple modalities of each object while CFS formed the feature vector for each object by only concatenating the learned features. Thus, the proposed algorithm performed the best for clustering the CUAVE dataset.

Next, we evaluate the robustness of the proposed algorithm by generating three different subsets, each with a distinct combination of two modalities. We carried out the experiment on these subsets for 5 times. The results are shown in Figures 9–11.

According to Figures 9–11, the proposed algorithm outperformed HOPCM and CFS since HOCFS got higher values of RI than the other two algorithms in most cases, especially for clustering the text-audio subset. In other words, the proposed algorithm produced the best clustering results in terms of RI for the CUAVE subsets.

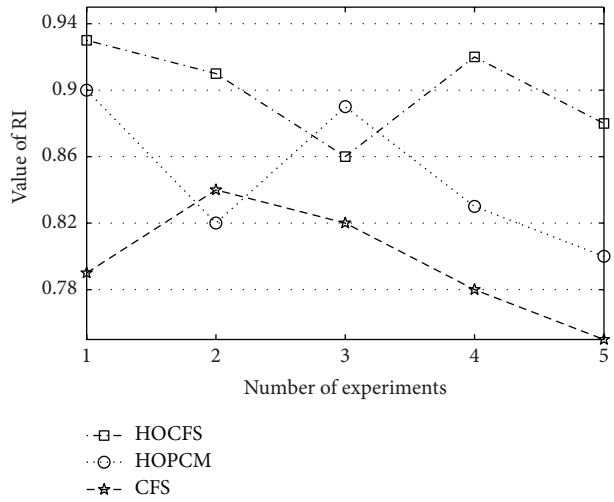

Figure 7: Clustering result on NUS-WIDE in terms of RI.

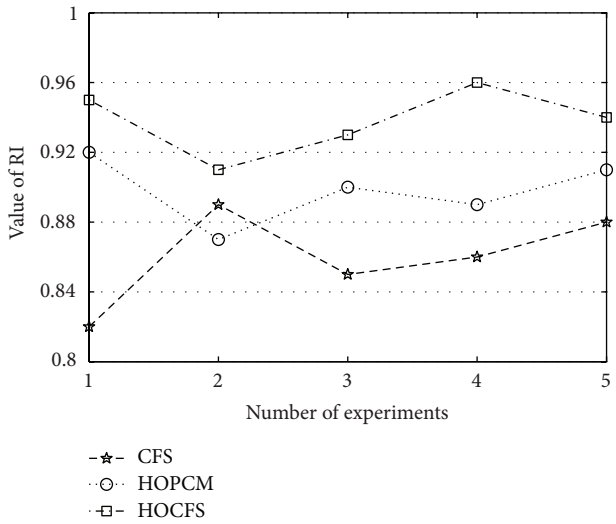

Figure 8: Clustering result on CUAVE in terms of RI.

Finally, we studied the relationship between the clustering result and the different combinations of modalities by analyzing the clustering results, as shown in Table 3.

From Table 3, the best clustering result is always produced on the overall dataset, implying that the clustering result of heterogeneous data relies on the joint features of image-text-audio modalities. Moreover, the proposed algorithm produced the worst clustering result on text-audio subset, which demonstrates that only features learned from the text-audio modalities could not effectively represent the objects in the CUAVE dataset.

7. Conclusion

In this paper, we proposed a high-order CFS algorithm for clustering heterogeneous data. One property of the paper is to devise an adaptive deep learning model and to apply it to learning features of each type of data. Furthermore, the vector

Table 3: Clustering result on different subsets in terms of RI.

Algorithm/subset	1	2	3	4	5
Image-text	0.92	0.88	0.87	0.89	0.93
Text-audio	0.81	0.79	0.78	0.83	0.80
Image-audio	0.89	0.83	0.89	0.85	0.87
Overall	0.96	0.91	0.93	0.96	0.94

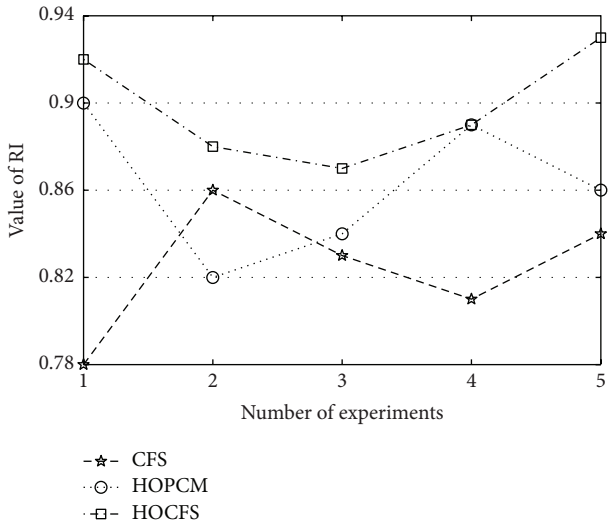

Figure 9: Clustering result on image-text subset in terms of RI.

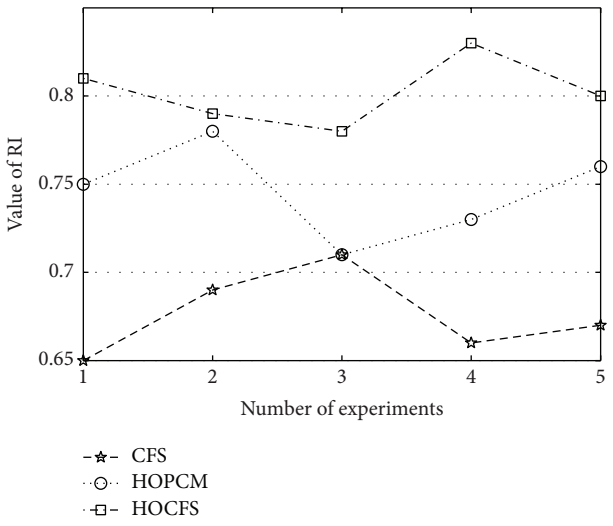

Figure 10: Clustering result on text-audio subset in terms of RI.

outer product was used to model the correlations of each type of data to form a feature tensor for every heterogeneous data object. Another property of the proposed algorithm is to adopt the tensor distance to measure the similarity between every two heterogeneous objects. Experimental results showed that our proposed algorithm produced more accurate results than HOPCM and CFS in terms of E^* and RI.

Recently, more and more complex heterogeneous data have been generated in many applications. For example, there

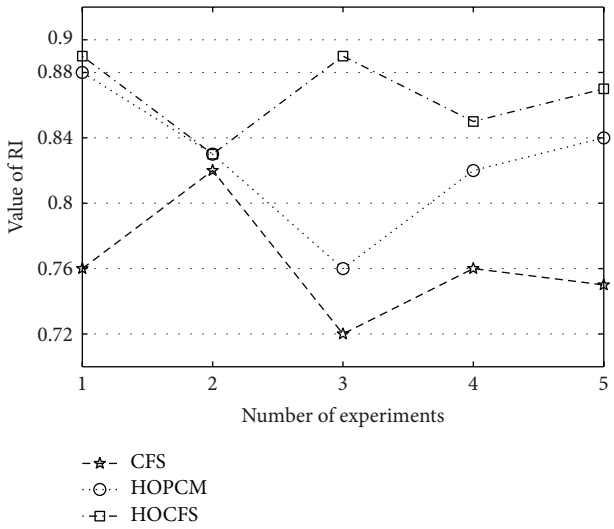

Figure 11: Clustering result on image-audio subset in terms of RI.

are simultaneously many images and audio pieces in one web document. The future work will focus on how to cluster such complex heterogeneous dataset.

Competing Interests

The authors declare that they have no competing interests.

References

[1] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[2] Q. Zhang and Z. Chen, "A weighted kernel possibilistic c-means algorithm based on cloud computing for clustering big data," *International Journal of Communication Systems*, vol. 27, no. 9, pp. 1378–1391, 2014.

[3] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[4] X. Yang, T. Zhang, and C. Xu, "Cross-domain feature learning in multimedia," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 64–78, 2015.

[5] L. Meng, A.-H. Tan, and D. Xu, "Semi-supervised heterogeneous fusion for multimedia data co-clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2293–2306, 2014.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[7] Q. Zhang, L. T. Yang, and Z. Chen, "Deep computation model for unsupervised feature learning on big data," *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 161–171, 2016.

[8] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1351–1362, 2016.

[9] Q. Zhang, L. T. Yang, Z. Chen, and F. Xia, "A high-order possibilistic-means algorithm for clustering incomplete multimedia data," *IEEE Systems Journal*, 2015.