Bu, F, Hu, C, Zhang, Q, Bai, C, T. Yang, L and Baker, T

A Cloud-Edge-aided Incremental High-order Possibilistic c-Means Algorithm for Medical Data Clustering

http://researchonline.ljmu.ac.uk/id/eprint/14065/

Article

A Cloud-Edge-aided Incremental High-order Possibilistic c-Means Algorithm for Medical Data Clustering

Fanyu Bu, Chengsheng Hu, Qingchen Zhang, Changchuan Bai, Laurence T. Yang, *Fellow, IEEE,* and Thar Baker

Abstract—Medical Internet of Things are generating a big volume of data to enable smart medicine that tries to offer computeraided medical and healthcare services with artificial intelligence techniques like deep learning and clustering. However, it is a challenging issue for deep learning and clustering algorithms to analyze large medical data because of their high computational complexity, thus hindering the progress of smart medicine. In this paper, we present an incremental high-order possibilistic c-means algorithm on a cloud-edge computing system to achieve medical data co-clustering of multiple hospitals in different locations. Specifically, each hospital employs the deep computation model to learn a feature tensor of each medical data object on the local edge computing system and then uploads the feature tensors to the cloud computing platform. The high-order possibilistic cmeans algorithm (HoPCM) is performed on the cloud system for medical data clustering on uploaded feature tensors. Once the new medical data feature tensors are arriving at the cloud computing platform, the incremental high-order possibilistic cmeans algorithm (IHoPCM) is performed on the combination of the new feature tensors and the previous clustering centers to obtain clustering results for the feature tensors received to date. In this way, repeated clustering on the previous feature tensors is avoided to improve the clustering efficiency. In the experiments, we compare different algorithms on two medical datasets regarding clustering accuracy and clustering efficiency. Results show that the presented IHoPCM method achieves great improvements over the compared algorithms in clustering accuracy and efficiency.

Index Terms—Medical Data Clustering; Cloud-edge computing; Deep computation model; Possibilistic c-means; Smart Medicine.

I. Introduction

INTERNET of Things (IoT) are creating a large-scale network of many things by deploying lots of sensors [1,2]. Particularly, IoT systems support various services such as intelligent transportation and environmental surveillance by analyzing large amounts of data gathered from the deployed sensors [3,4]. More recently, some medical IoT systems are created to collect medical data via wearable devices and body sensors for enabling smart medicine. Smart medicine is the deep integration of the medicine knowledge and advanced information technologies, especially artificial intelligence techniques like deep learning and clustering, with the intent of offering computer-aided medical services and healthcare services [5,6]. Smart medicine is facing a broad of applications and has made wonderful achievements. For example, smart medicine made a significant contribution to computer-aided diagnosis by applying the convolutional neural networks to medical image classification [7]. A deep convolutional neural network has been created to obtain the considerable level of specialists in skin cancer diagnosis [8]. Besides, smart medicine is playing a key role in providing pervasive healthcare monitoring for the aged.

The crucial component of smart medicine is to use various machine learning algorithms for data analytics. A typical machine learning technology is clustering, also called unsupervised learning, that partitions a dataset into some distinct groups relying on a specific proximity measure. The goal of clustering is to make the objects in the same group as similar or related as possible and the objects in different groups as different or unrelated as possible [9]. Clustering is important to smart medicine. For example, it has used to segment medical images for computer-aided diseases diagnosis [10]. Representative examples of clustering are prototypebased approaches, density-based approaches, and graph-based approaches. A typical prototype-based approach is the fuzzy c-means algorithm (FCM)[11]. Differing from the classical crisp clustering algorithms that assign each object into exact one group, FCM allows gradual memberships of each object to groups measured as degrees in [0,1], which provides the flexibility to express that objects can belong to more than one group. Generally, FCM achieves the comparatively better results for overlapped data sets that widely exist in smart medicine than crisp clustering algorithms. Therefore, FCM has been extensively used in smart medicine in recent years [10,12]. However, FCM forces the sum of the degrees of each object in the groups to be equal to 1, which often leads to meaningless results for the datsets with noise and outliers [13]. To overcome this disadvantage, an improved fuzzy clustering algorithm was proposed based on the possibility theory, called possibilistic c-means algorithm (PCM)that relaxes the constraint of FCM [14]. PCM gives significantly better results for the datasets with noise and outliers than FCM [13]. Therefore, this study focuses on the application of PCM in smart medicine. Recently, Zhang et al. [15] generalises PCM for multi-modal data clustering with the tensor-based representation model. Furthermore, they presented a highorder PCM approach (DCMHoPCM) for big data clustering combined with deep computation models. In particular, the deep computation models are aimed at learning a feature tensor for each instance. Afterwards, the feature tensors are taken as input of HoPCM to obtain the cluster result. Their results demonstrate the strong ability of HoPCM for big data

Fanyu Bu is with Inner Mongolia University of Finance and Economics, Hohhot, 010010, China.

Chengsheng Hu is with Beijing Thunisoft Information Technology Co., LTD, Beijing, 100084, China.

Qingchen Zhang and Laurence T. Yang are with St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada. Email: qzhang@stfx.ca. Changchuan Bai is with Dalian Hospital of Traditional Chinese Medicine, Dalian, 116033, China. Changchuan Bai and Fanyu Bu are the co-first authors.

Thar Baker is with Liverpool John Moores University, Liverpool, L3 3AF, UK.

clustering. However, this algorithm is facing a challenging problem for medical data clustering. It does not support incremental clustering. Specifically, medical data is generated continuously in smart medicine. To integrate the newly arriving data into the previous results, the previous data must be combined with the newly arriving data, leading to the repeated clustering of the previous data. With the continuous generation of medical data, the size becomes bigger and bigger. When the size of medical data exceeds the power of computing platform, HoPCM fails to cluster medical data.

To address this problem, this paper presents an incremental version of HoPCM for medical data clustering to support smart medicine. Particularly, the presented algorithm is based on the idea of single-pass clustering [16,17]. When new instances are arriving, only the previous cluster centers are required to combine with the new instances to produce the whole clustering result. Furthermore, we develop a cloudedge computing system to achieve co-clustering of multiple hospitals that are possible in different locations. To produce better results, a large-scale data pertaining to these hospitals is required for clustering. However, each hospital does not prefer to disclose its original data because the personal medical data is private and shall be protected. In our scheme, the edge computing devices are deployed in each hospital. Each hospital runs the deep computation model on the local edge computers to learn a feature tensor for each instance and then uploads the feature tensors to cloud computing platform. This way does not disclose the original data. Once the feature tensors arrive at the cloud computing platform, the incremental HoPCM approach is performed to obtain the clustering result combined with the previous cluster centers. Experiments are carried out to compare other clustering approaches on two datasets in terms of clustering accuracy and efficiency. Results show that our designed scheme achieves great improvements over other approaches in medical data clustering, proving its promise to support smart medicine.

Three contributions are presented in this article. • We devise an incremental version of HoPCM for medical data clustering to support smart medicine.

- We develop a cloud-edge computing system to efficiently run the incremental HoPCM (IHoPCM) approach, which can achieve the co-clustering of different hospitals without the disclosure of the original medical data.
- We conduct a set of experiments to validate the developed scheme in terms of clustering accuracy and clustering efficiency on two medical datasets.

This article is organized into five sections. The deep computation model and high-order possibilistic c-means are described as the preliminaries of our scheme in Section 2. Section 3 provides the details of our cloud-edge system and the incremental high-order possibilistic c-means approach. Experimental details and results are stated in Section 5. Finally, we summarize this study and discuss the future work in Section 5.

II. Related work and Preliminaries

The designed scheme is based on HoPCM and deep computation in this paper, so this section presents HoPCM and deep computation to make the paper more readable. The main notations in this paper are listed in Table 1.

TABLE I
VARIABLES LIST

	Meaning
	bias and weight tensors
	activation function
	dimension of tensor
	order of tensor
X,Y,H	tensor representation of input, hidden, and output, respectively
U	membership matrix
V	cluster centers
d	distance between two instances

A. FCM and PCM

Since the concept of fuzzy subsets was proposed in 1965, fuzzy systems that are structures based on fuzzy techniques oriented towards information processing have been developed greatly [18]. Typically, a fuzzy system is composed of four functional blocks: a fuzzifier, a fuzzy inference engine, a knowledge base, and a defuzzifier. Starting from the macroscopic view to simulate the fuzzy characteristics of human brain thinking, fuzzy systems have their advantages in describing high-level knowledge, so they have been widely applied in data science, especially big data based systems like decision analysis, economic information system, and medical diagnosis system [19]. Recently, some techniques have been presented to construct effective fuzzy systems for big data mining and analytics, for example fuzzy neural networks and fuzzy cmeans clustering [20,21]. The fuzzy c-means algorithm (FCM) typically gives the result for a dataset $X= \{x_1,x_2,\cdots,x_n\}$ with a center set $V = \{v_1,v_2,\cdots,v_c\}$ and a degree matrix $U = \{u_{ij}|1 \le i \le c, 1 \le j \le n\}$ with u_{ij} denoting the degree of the object x_j belonging to the i-th cluster by minimizing the following function $J_m(U,V)$:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u^m_{ij}||x_k - v_i||^2, \quad s.t. \sum_{i=1}^{c} u_{ij} = 1, \quad u_{ij} > 0, \tag{1}$$

where $m > 1$ is used to control the fuzziness of the partition [11].

To improve the performance of FCM, several weighted FCM algorithms have been presented by assigning a weight that defines the importance of each object in the clustering procedure [22,23,24]. The original FCM algorithm is only effective in clustering spherical' clusters, so kernel FCM algorithms have presented for more general datasets typically by replacing the original Euclidean distance with a kernelinduced distance metric [25,26]. In the era of big data, the volume of data is very big. In order for FCM to cluster large data, incremental FCM algorithms have been proposed like single-pass fuzzy c-means and online fuzzy c-means [21,27]. Moreover, more than 75% objects are multi-modal or heterogenous in big data. However, the original FCM algorithm fails to cluster heterogeneous data. To tackle this problem, Li et al. [28] proposed a high-order FCM algorithm by extending FCM to the tensor space.

From the objective function, FCM has a constraint that forces the sum of degrees of each object to 1. However, the constraint often leads to meaningless results for datasets with noise and outliers [13].

To overcome the disadvantage, Krishnauram and Keller proposed a possibilistic c-means algorithm (PCM) as an improved version of FCM by minimizing the following function [14]:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\|x_k - v_i\|^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m,$$
$$s.t.\ u_{ij} \in [0,1],\ 0 < \sum_{j=1}^{n} u_{ij} \le n,\ \max_i u_{ij} > 0. \tag{2}$$

By relaxing the constraint, PCM gives significantly better results for datasets with noise and outliers than FCM. Noise and outliers extensively exist in medical datasets, so PCM has extensively been used in smart medicine [29,30].

Specifically, PCM uses the following equations to obtain the result and update the parameter η_i:

$$v_i = \sum_{j=1}^{n} u_{ij}^{m} x_j \Big/ \sum_{j=1}^{n} u_{ij}^{m}, \tag{3}$$

$$u_{ij} = 1/(1 + (\frac{d_{ij}}{\eta_i})^{1/(m-1)}), \tag{4}$$

where d_{ij} is the distance between x_j and v_i that can be computed by any specific metric.

$$\eta_i = \sum_{j=1}^{n} u_{ij}^{m} d_{ij} \Big/ \sum_{j=1}^{n} u_{ij}^{m}. \tag{5}$$

.

In order for PCM to cluster multi-modal data, Zhang et al. generalised PCM to present a high-order possibilistic c-means algorithm (HoPCM) as following:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} d_{TD} + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m, \tag{6}$$

where d_{TD} is the square of the tensor distance between x_j and v_i [31].

In particular, HoPCM is outlined in Algorithm 1.

From Algorithms 1, the computational complexity of standard HoPCM is dominated by computing the membership matrix U. In particular, this computation costs the complexity of $O(l \times c \times n)$ with l denoting the number of iterations.

Algorithm 1: The High-order Possibilistic c-Means Approach.

```
Input: $X = \{x_1, x_2, \dots, x_n\}$, $c$, $m$, $maxiter$
Output: $U = \{u_{ij}\}, V = \{v_i\}$
Initialize the membership matrix $U$;
for $iter = 1, 2, \dots, maxiter$ do
    for $i = 1, 2, \dots, c$ do
        Update the cluster center $v_i$ via Eq.(5);
        $\eta_i = \sum_{j=1} u_{ij}^{m} d_{TD(ij)} / \sum_{j=1} u_{ij}^{m}$;
    for $i = 1, 2, \dots, c$ do
        for $j = 1, 2, \dots, n$ do
            $u_{ij} = 1/(1 + (\frac{d_{TD(ij)}}{\eta_i})^{1/(m-1)})$;
```

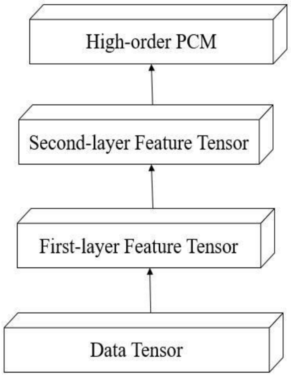


Fig. 1. Example of DCMHoPCM with two hidden layers.

To enhance the accuracy of HoPCM, the deep computation model was integrated to learn a feature tensor for each instance for DCMHoPCM, shown in Fig. 1.

Generally, the feature tensor is a representation of the original input instance and it has strong ability to reveal the distinct features from other instances, so it is beneficial to improve the clustering performance of HoPCM. Therefore, the feature tensors learned by DCM are fed to HoPCM for improving the clustering performance. Although Fig.1 presents an example of DCMHoPCM with two hidden layers, DCM could have an arbitrary number of hidden layers in practice.

A similar algorithm to DCMHoPCM is the multi-modal learning based HoPCM, which uses multiple deep learning to learn features for each modal data, shown in Fig. 2 [32].

There is a remarkable difference between DCM utilized in DCMHoPCM and the standard DCM. Standard DCM is aimed at classification and recognition, so it is a supervised model. Specifically, standard DCM involves a fine-tuning step in which many labeled instances are required to train the parameters with a global back-propagation. Distinguished from the standard version, DCM utilized in DCMHoPCM is an unsupervised model so it does not require labeled instances to fine-tune the parameters. Only greedy layer-wise pre-training is involved in training its parameters. Once the parameters are trained well, DCM is used to learn feature tensors before HoPCM is performed.

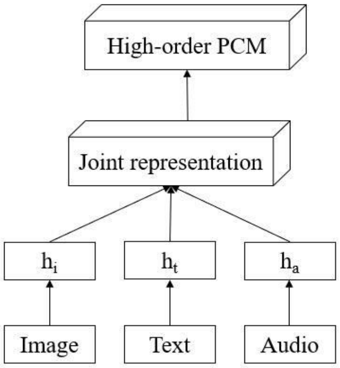


Fig. 2. Example of HoPCM based on deep learning. The joint representation represents the future tensor via the outer product.

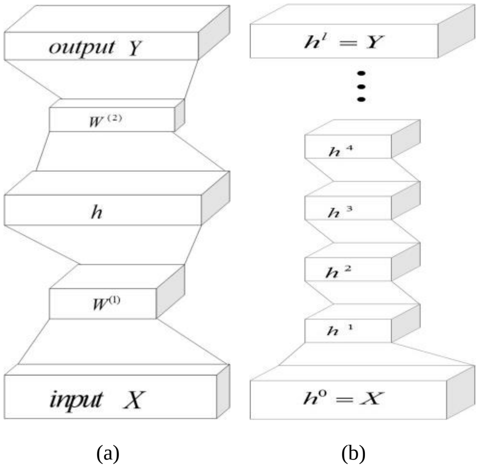
Although DCMHoPCM works well for big data clustering, it has a drawback for medical data clustering in smart medicine. In detail, DCMHoPCM is a static approach, leading to failure for clustering continuously generated medical data. Thus, once new medical instances are coming, they have to be incorporated with the previous instances that have been clustered and then DCMHoPCM is performed on the combined dataset for global results. This way leads to two problems. Firstly, old instances are clustered repeatedly. Second, the size of the combined dataset becomes bigger and bigger with the incorporation of new instances, thus inevitably increasing the clustering cost. What's worse, when the size of the combined dataset exceeds the power of computing platforms, DCMHoPCM cannot be performed. This article addresses this problem by presenting an incremental high-order possibilistic c-means based on the idea of single-pass, illustrated in the next section.

Recently, some incremental approaches have been presented for clustering. The most commonly used approaches for fuzzy clustering are single-pass fuzzy c-means and online fuzzy c-means [16,17]. The former processed a dataset chunk by chunk, where a chunk is a subset of data objects. For each chunk, only the cluster centers are kept as historical data to be employed together with the new arriving chunk for next round in the clustering process. The latter clusters each chunk of objects individually and obtains the final cluster centers of the whole dataset by performing FCM on the centers of all the chunks. Obviously, the single-pass approach is more suitable for dynamic datasets and is able to obtain the clustering results to date, especially suitable to cluster medical datasets in this study.

B. Deep Computation

The Deep Computation Model (DCM) is created with the Tensor Auto-encoder (TAE) as the basic component, presented in Fig. 3 [33].

A distinct property of TAE is to use an$_{1\,2\,N}$ N-order tensor$_{1\,2\,M}$ $X \in R_{I \times I \times \cdots \times I}$ and an M-order tensor $H \in R_{J \times J \times \cdots \times J}$ to represent the input layer and the hidden layer, respectively. Sigmoid is used as the activating function with the following forward pass:



(a) (b)

Fig. 3. Example of TAE and DCM. (a) represents TAE. (b) represents DCM.

$$h_{j_1 \cdots j_M} = f(\sum_{i_1 \cdots i_N}^{I_1 \cdots I_N} w^{(1)}_{i_1 \cdots i_N j_1 \cdots j_M} \cdot x_{i_1 \cdots i_N} + b^{(1)}_{j_1 \cdots j_M}) \quad (7)$$

$$y_{i_1 \cdots i_N} = f(\sum_{j_1 \cdots j_N}^{J_1 \cdots J_M} w^{(2)}_{j_1 \cdots j_M i_1 \cdots i_N} \cdot h_{j_1 \cdots j_M} + b^{(2)}_{i_1 \cdots i_N}), \quad (8)$$

where $\theta = \{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}\}$ is a set of parameters.

θ is trained to minimize the following objective function with m instances using the high-order back-propagation algorithm and the gradient descent approach:

$$\begin{aligned} J_{TAE}(\theta) &= \tfrac{1}{m} \sum_{i=1}^{m} (\tfrac{1}{2} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (Y^{(i)}_{i_1 \cdots i_N} - X^{(i)}_{i_1 \cdots i_N})^2) \\ &+ \tfrac{\lambda}{2} (\sum_{p=1}^{J_1 \times \cdots \times J_M} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (W^{(1)}_{p i_1 \cdots i_N})^2 \\ &+ \sum_{q=1}^{I_1 \times \cdots \times I_N} \sum_{j_1=1}^{J_1} \cdots \sum_{j_M=1}^{J_M} (W^{(2)}_{q j_1 \cdots j_M})^2) \end{aligned} \quad . \quad (9)$$

Fig. 1(b) presents the creation of a DCM model with stacked TAEs for multi-layer feature learning on input.

The deep computation is trained by a double-stage strategy. First, the parameters of each tensor auto-encoder are trained by the high-order back-propagation algorithm from bottom to up, which is commonly called the pre-training stage in the area of deep learning. Afterwards, the deep computation model is trained by a global back-propagation algorithm with the pre-trained parameters as the initialization, which is commonly called fine-tuning. Typically, the pre-training is an unsupervised learning process while the fine-tuning is a supervised one.

III. Incremental High-order Possibilistic c-Means on Cloud-edge System

The presented incremental HoPCM (IHoPCM) is based on the idea of single-pass clustering. In order for HoPCM to work in the single-pass way, we modify standard HoCPM by introducing weights, resulting in the weighted HoPCM with the following objective:

$$J_m(U,V;w) = \sum_{i=1}^{c}\sum_{j=1}^{n} u^m_{ij} d_{TD} + \sum_{i=1}^{c}\eta_i \sum_{j=1}^{n}(w_j - u_{ij})^m, \tag{10}$$

where w_j is the weight assigned to x_j.

The goal of updating $U = \{u_{ij}\}$ and centers $V = \{v_i\}$ is to minimize $J_m(U,V;w)$. Since u_{ij} are independent of each other, minimizing $J_m(U,V;w)$ regarding U can be achieved by minimizing the following function regarding u_{ij}:

$$J_{mij}(u_{ij},v_i;w_j) = u_{mij}d_{TD} + \eta_i(w_j - u_{ij})_m. \tag{11}$$

Furthermore, we obtained the following equation by differentiating Eq.(10) regarding u_{ij} and letting it equal to 0:

$$mu_{mij-1}d_{TD} - m\eta_i(w_i - u_{ij})_{m-1} = 0. \tag{12}$$

Solving Eq.(11) to obtain the equation for updating u_{ij} in Eq.(12):

$$u_{ij} = w_j/(1+(\frac{d_{TD}}{\eta_i})^{1/(m-1)}) \tag{13}$$

The weighted HoPCM is presented in Algorithm 2.

Algorithm 2: The Weighted HoPCM Approach.

```
Input: $X = \{x_1,x_2,...,x_n\}$, $c$, $m$, $maxiter$ Output: $U = \{u_{ij}\}, V = \{v_i\}$
Initialize the membership matrix $U$;
for $iter = 1,2,...,maxiter$ do
    
        Update the cluster center $v_i$ via Eq.(5);
        $\eta_i = \sum_{j=1}^{n} u_{ij}^m d_{TD\,(ij)} / \sum_{j=1}^{n} u_{ij}^m$;
    for $i = 1, 2,...,c$ do
        for $j = 1, 2,...,n$ do
            $u_{ij} = w_j/(1+(\frac{d_{TD}}{\eta_i})^{1/(m-1)})$;
```

for $i = 1,2,...,c$ do

From Algorithms 2, the computational complexity of the weighted HoPCM is also dominated by computing the membership matrix U. So, it has the same computational complexity of $O(l \times c \times n)$ as Algorithm 1.

Denoting the first chunk of medical data arriving as X^1, the standard HoPCM is performed on this chunk to produce the result, denoting as the membership matrix U^1 and centers V^1. When a chunk of new medical data, denoted as X^2, is arriving, X^2 is combined with the previous centers V^1 to form the new dataset $Y = [V^1,X^2]$. Each instance in Y is assigned to a weight, all weights forming a set w^2 by the following rules:

(1) the weights of instances in X^2 are set to 1, denoting as $w_{ins^n\,tance} = [1,1,...,1]^T$ with n representing the number of instances in X^2,

(2) the weight of each center in V^1 is set to the weighted sum of memberships in U^1, all weights of centers forming $w_{center1}$, i.e., $w_{center1} = U_1 w_{insm\,tance}$ with m representing the number of instances in X^1.

Based on the above rules, we have $w^2 = [w_{center^1}, w_{ins^n\,tance}]$. The weighted HoPCM is performed on the dataset Y to produce the global result, U^2 and V^2, to date with the weight set w^2.

Similarly, the weight of each center in V^2 can be obtained by the following equation:

$$w_{center2} = U_2 w_2 = U_2[w_{center1}, w_{insn\,tance}]. \tag{14}$$

More generally, denoting the results of the dataset of the first s chunks of medical data as U^s and V^s, the weight of each center in V^s is computed as the weighted sum of memberships of the centers of $s-1$th chunk and the total membership of instances in X_s, namely, $w_{centers} = U_s[w_{centers-1}, w_{insk\,tance}]$ where k denotes the number of instances in X^s. When the s+1 chunk of new t instances is arriving, the dataset Y is formed by the combination of the new instances and the centers V^s, i.e., $Y = [V^s,X^{s+1}]$. The weight of instance in Y is obtained by the combination of w_{center^s} and $w_{ins^t\,tance} = [1,1,...,1]^T$, i.e., $w_{s+1} = [w_{centers}, w_{inst\,tance}]$. The clustering results to date is produced by performing weighted HoPCM on Y with the weights w^{s+1}.

The incremental HoPCM is outlined in Algorithm 3.

Algorithm 3: The Incremental HoPCM Approach.

```
Input: $X^{s+1}(s \geq 1)$, $V^{s}$, $w_{center}^{s}$, $c$, $m$, $maxiter$
Output: $U^{s+1}$, $V^{s+1}$, $w_{center}^{s+1}$
Form the dataset $Y$ with $Y = [V^{s}, X^{s+1}]$;
Set the weight vector $w_{ins\tan ce}^{t}$ with $w_{ins\tan ce} = [1,1,\ldots,1]^{T}$;
Form the weight vector for $Y$ with
    $w^{s+1} = [w_{center}^{s}, w_{ins\tan ce}^{t}]$;
Initialize the membership matrix $U^{s+1}$;
Perform weighted HoPCM on $Y$ with $w^{s+1}$ to obtain $U^{s+1}$ and $V^{s+1}$;
Compute $w_{center}^{s+1}$ with
    $w_{center}^{s+1} = U^{s+1}[w_{center}^{s}, w_{ins\tan ce}^{t}]$;
```

From Algorithms 3, the computational complexity of incremental HoPCM for clustering the total s+1 chunks of medical instances, with t instances in the s+1th chunk, is dominated by computing the membership matrix U^{s+1}. In particular, this computation costs the complexity of $O(l \times c \times (t + c))$ where l and c are the number of iterations and cluster centers. Since c is typically significantly less than t, the computational complexity can be approximated as $O(l \times c \times t)$.

Denoting the number of instances in the ith chunk X^{i} as t_i, the number of instances in the total $s + 1$ chunks is × × $q = \sum_{i=1}^{s+1} t_i$. If standard HoPCM is used to cluster the $s + 1$ chunks of medical instances, the time cost will be $O(l\ c\ q)$. For simplicity, supposing the number of instances in each chunk is t, the complexity of standard HoPCM for clustering the $s + 1$ chunks of instances is $O(l \times c \times s \times t)$. With the continuous generation of medical data, s is continuously increasing. In the case with a large s, the computational complexity of standard HoPCM is significantly higher than that of incremental HoPCM. So, incremental HoPCM achieves great improvement of clustering efficiency for medical data clustering.

Furthermore, we develop a cloud-edge computing system to perform the incremental HoPCM based on the deep computation model for achieving co-clustering of medical data in different hospitals. In edge computing paradigm, the computing devices are deployed at the site close to data [34]. Examples of such the computing devices, also called edge devices, are personal computers and mobile phones. Typically, edge devices have a limited power of computation, storage and energy, so they perform some simple and preliminary computation for data preprocessing.

In the proposed system, edge devices are the local computing centers deployed in hospitals. When new medical data instances are generated, they are stored in these local computing centers. When they need to be clustered, the deep computation model is performed to learn feature tensors for them on the edge devices. Afterwards, the learned feature tensors are uploaded to the cloud computing platform and then the incremental HoPCM is performed on the feature tensors with the previous centers to obtain the clustering results on cloud.

The process of using the deep computation model to learn feature tensors for medical instances only involves the forward propagation, so this process does not require powerful computing resources and it is of high efficiency. Edge devices are fully qualified for this task. Moreover, the feature tensors of medical instances can be learned in parallel since the deep computation model learns the feature tensor for each instance independently. The task of training the deep computation model can be completed on cloud computing platform or by a GPU Cluster. Training the deep computation model is a onetime task. Once the deep computation model is trained well, it can be used by all the hospitals all the time. So training the deep computation model is not an obstacle to the smart medicine though it is about high time complexity.

Typically, the size of a feature tensor is much less than the size of the corresponding original instance. So uploading the feature tensors instead of the original instances to the cloud computing platform can reduce the communication cost significantly, which is crucial in the case of limited network bandwidth. More notably, only uploading feature tensors can protect the privacy of the original instances to some extent since the forward propagation is a nonlinear computation process. Without the parameters of the deep computation model, it is difficult to infer the original instances relying on the feature tensors, especially when the deep computation model has many hidden layers.

Although the computational complexity of incremental HoPCM is not high, it still requires powerful computing devices for clustering medical data when the size of a chunk of medical instances is large. Fortunately, cloud computing platform offers enough computing devices of strong power to perform incremental HoPCM for clustering results.

IV. Experiments

In the experiments, the server with 6-cores, 2.4GHz CPU and 64GB memory is employed as the local edge devices deployed in hospitals and the server with 20-cores, 2.9GHz CPU and 745GB memory is used as the cloud platform. To evaluate the performance of the proposed algorithm, we developed an incremental high-order fuzzy c-means algorithm (IHoFCM) by the combination of the high-order FCM [28] algorithm and deep computation in the single-pass way. So, we compare the presented incremental HoPCM (IHoPCM) with IHoFCM and standard HoPCM (DCMHoPCM) on two synthesized medical datasets. To make the fair comparison, we use the same deep computation model to learn features of each object in the three schemes. The first dataset contains 8000 images fallen into 4 clusters sampled. The second dataset has 6000 medical images fallen into 3 clusters. To report the experimental results clearly, we denote the two datasets as Medical-4 and Medical-3, respectively.

We use Silhouette Coefficient (SC) to verify the clustering accuracy in the experiments. To compute silhouette coefficient, we convert the soft results of the two algorithms into the hard results by assigning every instance to the cluster where it has the biggest membership.

Silhouette coefficient verifies a specific clustering algorithm with the combination of cohesion and separation. Specifically, the silhouette coefficient of a clustering result is computed in the following four steps.

Step 1. For the *i*th instance in a cluster, compute its average distance to other instances in this cluster, called a_i.

Step 2. For the same instance and any other cluster, compute the average distance between the instance and the instances in the given cluster. Denote the minimum distance value as b_i.

Step 3. Compute the silhouette coefficient s_i for the instance as:

$$s_i = (b_i - a_i)/\max(a_i,b_i). \tag{15}$$

Step 4. Compute the silhouette coefficient of the clustering result as the mean of all the instances.

The SC value ranges from −1 to 1. The closer this value is to 1, the better the clustering result.

A. Results on Medical-4

We divide Medical-4 into 8 chunks, each chunk with 1000 instances. The results regarding silhouette coefficient are summarized in Table II.

The results in Table II make three interesting observations. First, both DCMHoPCM and IHoPCM produce the same clustering result on the first chunk because both the two algorithms perform the standard HoPCM on the same feature tensors. In fact, the two algorithms are exactly the same in this case. Second, from the second chunk, DCMHoPCM produces more accurate results than IHoPCM and IHoFCM in most cases. However, compared with DCMHoPCM, the accuracy drop of IHoPCM is very low with around 2-3%, that is, the clustering accuracy of the incremental HoPCM can be acceptable. More significantly, IHoCPM outperforms IHoFCM, which validates the performance of our proposed algorithm on this dataset. Lastly, when the fourth and the seventh chunks are arriving, IHoPCM outperforms DCMHoPCM with the 2% silhouette coefficient increasing. The possible reason is that the instances in the two chunks are separated well. Generally speaking, IHoPCM is affected by the new arriving chunk greater than DCMHoPCM.

TABLE II

RESULTS ON MEDICAL-4 REGARDING SILHOUETTE COEFFICIENT.

Algorithm/chunk	1	2	3	4	5	6	7	8
DCMHoPCM	0.79	0.78	0.81	0.80	0.81	0.79	0.83	0.81
IHoFCM	0.76	0.73	0.69	0.77	0.75	0.71	0.79	0.74
IHoPCM	0.79	0.76	0.78	0.82	0.78	0.77	0.85	0.79

When we compare the efficiency of the three algorithms, we do not take the running time of feature tensors learning into account because they use the same deep computation model to learn feature tensors on the same computing platform, i.e., the time of feature tensors learning of the three algorithms are the same. There, we compare the running time of the three algorithms on cloud for clustering on feature tensors. In particular, the running time is normalized regards the longest running time to make a clear comparison. The results regarding normalized running time are listed in Table III.

First, Table III shows that IHoPCM has almost the same clustering efficiency as IHoFCM since their execution time is almost the same on each chunk. When the last chunk of feature tensors are arriving at cloud, DCMHoPCM will cluster the whole dataset. So the running time of DCMHoPCM on the whole dataset is the longest, which is also implied by the results in Table III. For DCMHoPCM, the running time continuously increases with the arrival of new chunks. Specifically, the running time increases approximately linearly in the number of chunks since the computational complexity of standard HoPCM is linear in the number of instances. However, we can see that the running time of IHoPCM and IHoFCM remains almost the same from the second chunk arrival. The reason is that the two algorithms cluster the same size of data, i.e., the instances in each chunk plus the centers of the previous chunks. Overall, our proposed algorithm makes great improvements for the clustering efficiency of the standard HoPCM.

B. Experimental results on Medical-3

We divide the Medical-3 dataset into 4 chunks, each chunk with 1500 medical images. The results of the three algorithms regarding silhouette coefficient and normalized running time are presented in Table IV and Table V.

TABLE IV

RESULTS ON MEDICAL-3 REGARDING SILHOUETTE COEFFICIENT.

Algorithm/chunk	1	2	3	4
DCMHoPCM	0.86	0.89	0.82	0.85
IHoFCM	0.83	0.86	0.72	0.81
IHoPCM	0.86	0.91	0.79	0.84

TABLE V

RESULTS ON MEDICAL-3 REGARDING NORMALIZED RUNNING TIME.

Algorithm/chunk	1	2	3	4

DCMHoPCM	0.221	0.479	0.716	1
IHoFCM	0.239	0.228	0.235	0.251
IHoPCM	0.221	0.237	0.258	0.245

Not surprisingly, both DCMHoPCM and IHoPCM obtain the same clustering result in terms of silhouette coefficient and normalized running time for the first chunk of feature tensors from Table IV and Table V because they cluster the same dataset in this case. From Table IV, IHoPCM produces considerable clustering results with DCMHoPCM after the second chunk is arriving at cloud. In particular, when the second chunk arrives, IHoPCM produces 2% higher silhouette coefficient than DCMHoPCM. However, IHoPCM produces significantly more accurate results than IHoPCM in terms of silhouette coefficient. From Table V, IHoPCM improves the clustering efficiency significantly with the arrival of new chunks. Compared with DCMHoPCM, the results in Table IV and Table V clearly argue that IHoPCM makes significant improvements of clustering efficiency, especially on the whole dataset with the efficiency improvement of around 4 times, without clustering accuracy drop. Such the results fully validate the performance of the presented algorithm for medical data clustering in smart medicine.

C. Results on Clustering Efficiency

Furthermore, we evaluate the clustering efficiency of the developed cloud-edge system for IHoPCM based on deep computation. For simplicity, we compare our scheme with the following two schemes on the first two chunks regarding the normalized running time in the experiments.

(1) We run DCM and IHoPCM on edge devices. In this scheme, the total running time is equal to the running time of feature tensors learning plus the running time of clustering on edge devices.

(2) We upload all original data to cloud and run DCM and IHoPCM on cloud. In this scheme, the total running time is equal to the running time of uploading original data plus the running time of feature tensors learning and clustering on cloud.

In our scheme, the total running time is equal to the running time of feature tensors learning on edge devices plus the running time of uploading original data plus the running time of clustering on cloud.

Table VI and Table VII report the comparisons.

TABLE III
RESULTS ON MEDICAL-4 REGARDING NORMALIZED RUNNING TIME.

Algorithm/chunk	1	2	3	4	5	6	7	8
DCMHoPCM	0.069	0.148	0.249	0.417	0.541	0.702	0.841	1
IHoFCM	0.067	0.073	0.072	0.076	0.081	0.079	0.074	0.079
IHoPCM	0.069	0.072	0.076	0.078	0.075	0.079	0.081	0.078

TABLE VI
RESULTS OF THREE SCHEMES ON MEDICAL-4.

System/chunk	1	2
Edge	0.921	1
Cloud	0.694	0.783
Cloud-edge	0.722	0.811

TABLE VII
RESULTS OF THREE SCHEMES ON MEDICAL-3.

System/chunk	1	2
Edge	0.963	1
Cloud	0.782	0.824
Cloud-edge	0.816	0.875

The results show that running both DCM and IHoPCM on the edge computing platform takes longest because of limited computing power on edge devices. Although running DCM on edge devices takes longer than running DCM on cloud, uploading feature tensors can save much time since the size of feature tensors is significantly smaller than the size of original data. So, the developed scheme does not take much longer to run the IHoPCM based on deep computation than the scheme that runs both DCM and IHoPCM on cloud. It should be noted that uploading original data will disclose the privacy of personal medical data.

V. CONCLUSION

In this article, we presented an incremental high-order possibilistic c-means algorithm based on deep computation for medical data clustering to support smart medicine. The presented algorithm makes great improvement with respect to clustering efficiency of the standard possibilistic c-means algorithm especially when a large volume of data is accumulated. Furthermore, we developed a cloud-edge system to achieve coclustering of data in different hospitals. One major advantage of the developed system is to use cloud computing to improve clustering efficiency without disclosure of the original data. The future work will focus on verifying the presented scheme on real applications. Specifically, we will deploy the developed system in a real hospital environment to validate the presented algorithm.

VI. ACKNOWLEDGE

This paper is supported by the National Key Research and Development Program of China (Grants No. 2018YFC0830203).

REFERENCES

[1] M. Lin and L. T. Yang, ”Hybrid genetic algorithms for scheduling partially ordered tasks in a multi-processor environment,” in *Proceedings of IEEE International Conference on Real-Time Computing Systems and Applications*, 1999, pp. 382-387.

[2] S. Li, L. D. Xu, and S. Zhao, ”5G Internet of Things: a survey,” *Journal of Industrial Information Integration*, vol. 10, pp. 1-9, 2018.

[3] V. K. Naik, C. Liu, L. T. Yang, and J. Wagner, ”Online resource matching for heterogeneous grid environments,” in *Proceedings of IEEE International Symposium on Cluster Computing and the Grid*, 2005, pp. 607-614.

[4] M. Dong, K. Ota, L. T. Yang, S. Chang, H. Zhu, and Z. Zhou, ”Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks,” *Computer Networks*, vol. 74, pp. 58-70, 2014.

[5] Q. Zhang, C. Bai, L. T. Yang, Z. Chen, P. Li, and H. Yu, ”A unified smart Chinese medicine framework for healthcare and medical services,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019, DOI: 10.1109/TCBB.2019.2914447.

[6] Q. Zhang, C. Bai, Z. Chen, P. Li, H. Yu, S. Wang, and H. Gao, ”Deep learning models for diagnosing spleen and stomach diseases in smart Chinese medicine with cloud computing,” *Concurrency and Computation: Practice and Experience*, 2019, DOI: 10.1002/cpe.5252.

[7] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A.W.M. Laak, B. Ginneken, and C. I. Snchez, ”A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60-68, 2017.

[8] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, ”Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, pp. 115-118, 2017.

[9] Z. Deng, K. Choi, Y. Jiang, J. Wang, and S. Wang, ”A survey on soft subspace clustering,” *Information Sciences*, vol. 348, pp. 84-106, 2016.

[10] D. Kumar, H. Verma, A. Mehra, and R. K. Agrawal, ”A modified intuitionistic fuzzy c-means clustering approach to segment human brain MRI image,” *Multimedia Tools and Applications*, vol. 78, pp. 1266312687, 2019.

[11] E. H. Ruspini, J. C. Bezdek, and J. M. Keller, ”Fuzzy clustering: a historical perspective,” *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 45-55, 2019.

[12] S. R. Kannan, M. Siva, S. Ramathilagam, and R. Devi, ”Effective kernelbased fuzzy clustering systems in analyzing cancer database,” *DataEnabled Discovery and Applications*, vol. 2, no. 1, pp. 5:1-5:12, 2018.

[13] F. Bu, Q. Zhang, L. T. Yang, and H. Yu, ”An edge-cloud-aided highorder possibilistic c-means algorithm for big data clustering,” *IEEE Transactions on Fuzzy Systems*, 2020, DOI: 10.1109/TFUZZ.2020.2992634.

[14] R. Krishnauram and J. M. Keller, ”A possibilistic approach to clustering” *IEEE Transactions on Fuzzy Systems*, vol. 1 no. 2, pp. 98-110, 1993.

[15] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, ”PPHOPCM: Privacypreserving high-order possibilistic c-means algorithm for big data clustering with cloud computing,” *IEEE Transactions on Big Data*, 2017, DOI: 10.1109/TBDATA.2017.2701816.

[16] P. Hore, L. O. Hall, and D. B. Goldgof, ”Single pass fuzzy c means,” in *Proceedings of IEEE International Fuzzy Systems Conference*, 2007, pp. 1-7.

[17] J. Mei, Y. Wang, L. Chen, and C. Miao, ”Incremental fuzzy clustering for document categorization,” in *Proceedings of IEEE International Conference on Fuzzy Systems*, 2014, pp. 1518-1525.

[18] R. Czabanski, M. Jezewski, and J. Leski, ”Introduction to fuzzy systems,” *Theory and Applications of Ordered Fuzzy Numbers*, Springer, Cham pp. 23-43, 2017.

[19] A. Fernandez, C. J. Carmona, M. J. del Jesus, and F. Herrera, ”A view´ on fuzzy systems for big data: progress and opportunities,” *International Journal of Computational Intelligence Systems*, vol. 9, sup. 1, pp. 69-80, 2016.

[20] K. V. Shihabudheen and G. N. Pillai, ”Recent advances in neuro-fuzzy system: a survey,” *Knowledge-Based Systems*, vol. 152, pp. 136-162, 2018.

[21] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, ”Fuzzy c-means algorithms for very large data,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1130-1146, 2012.

[22] G. E. Tsekouras, ”On the use of the weighted fuzzy c-means in fuzzy modeling,” *Advances in Engineering Software*, vol. 36, no. 5, pp. 287-300, 2005.

[23] C. Hung, S. Kulkarni, and B. Kuo, ”A new weighted fuzzy c-means clustering algorithm for remotely sensed image classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 543553, 2010.

[24] M. Hashemzadeh, A. G. Oskouei, and N.Farajzadeh, ”New fuzzy cmeans clustering method based on feature-weight and cluster-weight learning,” *Applied Soft Computing*, vol. 78, pp. 324-345, 2019.

[25] H. Huang, Y. Chuang, and C. Chen, ”Multiple kernel fuzzy clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120-134, 2011.

[26] F. Zhao, Z. Zeng, H. Liu, R. Lan, and J. Fan, ”Semisupervised approach to surrogate-assisted multiobjective kernel intuitionistic fuzzy clustering algorithm for color image segmentation,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 6, pp. 1023-1034, 2020.

[27] Y. Wang, L. Chen, and J. Mei, ”Incremental fuzzy clustering with multiple medoids for large data,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no.6, pp. 1557-1568, 2014.

[28] P. Li, Z. Chen, L. T. Yang, L. Zhao, and Q. Zhang, ”A privacy-preserving high-order neuro-fuzzy c-means algorithm with cloud computing,” *Neurocomputing*, vol. 256, pp. 82-89, 2017.

[29] J. Yang and D. Tan, ”A modified improved possibilistic c-means method for computed tomography image segmentation,” *Journal of Medical Imaging and Health Informatics*, vol. 8, no. 3, pp. 555-560, 2018.

[30] F. Li, S. Wang, and G. Liu, ”A Bayesian possibilistic c-means clustering approach for cervical cancer screening,” *Information Sciences*, vol. 501, pp. 495-510, 2019.

[31] Y. Liu, Y. Liu, and K. C. C. Chan, ”Tensor distance based multilinear locality-preserved maximum information embedding,” *IEEE Transactions on Neural Networks*, vol. 21. no. 11, pp. 1848-1854, 2010.

[32] Q. Zhang, L. T. Yang, Z. Chen, and F. Xia, ”A high-order possibilistic c-means algorithm for clustering incomplete multimedia data,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 2160-2169, 2015.

[33] 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.

[34] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, ”Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.