PPHOPCM: Privacy-Preserving High-Order Possibilistic C-Means Algorithm for Big Data Clustering With Cloud Computing

SOUNTHARARAJ.A.C

ASSISTANT PROFESSOR

SRI KRISHNA COLLEGE OF ARTS AND SCIENCE

AFFILATED TO BARATHIYAR UNIVERSITY

COIMBATORE-07

[sunithac@skasc.ac.in](mailto:sunithac@skasc.ac.in)

DHANYA.V.A

DEPT. OF MSC.SS

SRI KRISHNA COLLEGE OF ARTS AND SCIENCE

AFFILATED TO BARATHIYAR UNIVERSITY

COIMBATORE-07

[dhanyava15mss013@skasc.ac.in](mailto:dhanyava15mss013@skasc.ac.in)

DEEPIKA.G.KRISHNAN

DEPT. OF MSC.SS

SRI KRISHNA COLLEGE OF ARTS AND SCIENCE

AFFILATED TO BARATHIYAR UNIVERSITY

COIMBATORE-07

[deepikagkrishnan15mss010@skasc.ac.in](mailto:dhanyava15mss013@skasc.ac.in)

*ABSTRACT*

The world is moving with more advancement with a huge amount of memory requirement. To do this we are in need of big data a technique which resolves the conflict of memory space requirement which is much larger that few petabytes. One of the important knacks of fuzzy clustering in data mining and pattern recognition is being solved with the c-means algorithm (PCM) has been widely used in image analysis and knowledge discovery. It is always difficult for the PCM to produce a good result for clustering big data. Especially for heterogeneous data, the PCM can never be implemented as it was designed only small structured datasets. To overcome this problem, the PCM algorithm (HOPCM) can be implemented for big data clustering by optimizing the objective function in the tensor space .this topic is being discussed in this paper. To enhance the processing of a large amount of heterogeneous data set, a distributed HOPCM can be implemented. Finally, to protect the private data on the cloud by applying the BGV encryption scheme to HOPCM, a privacy-preserving HOPCM algorithm (PPHOPCM) can b used. In this technique of PPHOPCM, the functionalities such as updating the membership matrix and clustering centers are approximated as polynomial functions to support the secure computing of the BGV scheme. It is found that the algorithm of PPHOPCM works effectively in clustering large number of heterogeneous data which uses cloud computing without disclosure of private data.

*KEYWORDS: -* big data clustering; cloud computing; privacy preserving; possibility c-means; tensor space.

1. INTRODUCTION

AS personal computing technology and social websites, such as Facebook and Twitter, become increasingly popular, big data is in the explosive growth [1]. Bigdata are typically heterogeneous i.e., each object in bigdata set is multi-modal [2]. Specially, big data sets include various interrelated kinds of objects, such as texts, images and audios, resulting in high heterogeneity in terms of structure form, involving structured data and unstructured data. Moreover, different types of objects carry different information while they are interrelated with each other [3]. For example, a piece of sport video with meta-information uses a large number of subsequent images to display the exercise process and uses some meta-information, such as annotation and surrounding texts, to show additional information which are not displayed in the video, for instance the names of athletes .Although the subsequent images pass on different information from the surrounding texts, they describe the same objects from different perspectives. Furthermore, big data are usually of huge amounts. For example, Facebook, the famous social websites, collects about 500 terabytes (TB) data every day [4]. These features of big data bring a challenging issue to clustering technologies. Clustering is designed to separate objects into several different groups according to special metrics, making the objects with similar features in the same group [5, 6]. Clustering techniques have been successfully applied to knowledge discovery and data engineering [7].With the increasing popularity of bigdata, bigdata clustering is attracting much attention from data engineers and researchers. For example, Gao et al. [8] designed a graph-based co-clustering algorithm for big data by generalizing their previous image-text clustering method. Chen et al. [9, 10] designed a non-negative matrix tri-factorization algorithm to cluster big data sets by capturing the correlation over the multiple modalities. Zhang et al. [11] proposed a high-order clustering algorithm for big data by using the tensor vector space to model the correlations over the multiple modalities. However, it is difficult for them to cluster big data effectively, especially heterogeneous data, due to the following two reasons. First, they concatenate the features from different modalities linearly and ignore the complex correlations hidden in the heterogeneous data sets, so they are not able to produce desired results. Second, they often have a high time complexity, making them only applicable to small data sets. Thus, they cannot cluster large amounts of heterogeneous data efficiently. To tackle the above problems, this paper proposes a privacy preserving high-order PCM scheme (PPHOPCM) for big data clustering. PCM is one important scheme of fuzzy clustering [12, 13]. PCM can reflect the typicality of each object to different clusters effectively and it is able to avoid the corruption of noise in the clustering process [14]. However, PCM cannot be applied to big data clustering directly since it is initially designed for the small structured dataset. Specially, it cannot capture the complex correlation over multiple modalities of the heterogeneous data object. The paper proposes a high-order PCM algorithm by extending the conventional PCM algorithm in the tensor space. Tensor is called a multidimensional array in mathematics and it is widely used to represent heterogeneous data in bigdata analysis and mining. In this paper, the proposed HOPCM algorithm represents each object by using a tensor to reveal the correlation over multiple modalities of the heterogeneous data object. To increase the efficiency for clustering big data, we design a distributed HOPCM algorithm based on MapReduce to employ cloud servers to perform the HOPCM algorithm. However, the private data tends to be in disclosure when performing HOPCM on cloud. Take the medical data which is a typical type of big data for example. A large amount of private information such as personal email address and diagnostic data is included in the medical records. The disclosure of the private information will threaten people’s lives and property greatly. Therefore, to protect the private data on cloud, we propose a privacy preserving HOPCM scheme by using the BGV technique that is of high efficiency [15]. Unfortunately, BGV does not support the division operations and square root operations that are the necessary computation in the functions for updating the membership matrix and clustering centers in the HOPCM algorithm although it is a fully homomorphism encryption scheme. To tackle this issue, we use the Taylor’s theorem to transform these functions to polynomial functions to remove these operations. We conduct the experiments on the two representative big data sets, i.e., NUS-WIDE and SNAE2, to assess the clustering accuracy and efficiency of our algorithms by comparison with three representative possibility c-means clustering algorithms, namely HOPCM-15 [11], wPCM [14] and PCM [12]. Results demonstrate that HOPCM outperforms other algorithms in clustering accuracy for big data, especially for heterogeneous data. Furthermore, PPHOPCM can use cloud servers cluster big data efficiently without disclosure of the private data. Therefore, our contributions are summarized as the following three aspects:

• The conventional PCM algorithm cannot cluster heterogeneous data. Aiming at this problem, the paper proposes a high-order PCM algorithm by optimizing the objective function in the high-order tensor space for heterogeneous data clustering.

• To employ cloud servers to improve the clustering efficiency, we design a distributed high-order possibility algorithm based on MapReduce.

• To protect the sensitive data when performing HOPCM on the cloud platform, we develop a privacy-preserving high-order possibilistic c-means scheme by using the BGV encryption method. The rest of the paper is organized as follows. Section II reviews the related work on the possibilistic c-means algorithm and big data clustering methods. HOPCM is illustrated in Section III and the distributed HOPCM scheme based on MapReduce is presented in Section IV. Section V illustrates the privacy-preserving HOPCM method and Section VI reports experimental results. The whole paper is concluded in Section VII.

2. RELATED WORKS

This section reviews the related work on the possibilistic c-means algorithm and heterogeneous data clustering methods. As the preliminary, the PCM algorithm is described ﬁrst, followed by the heterogeneous data clustering methods.

2.1 Possibilistic c-Means Algorithm

The possibilistic c-means algorithm is one of fuzzy clustering schemes. Different from the traditional clustering schemes which assign each object into only one group; fuzzy clustering schemes assign each object into multiple groups. Specially, the assignment of each object is typically a distribution over all the groups in the fuzzy clustering. Given a data set X = {x1,··· ,xn}, PCM is defined as ac ×n membership matrix U = {uij},with the following objective function:

Equation

Where V = {v1,··· ,vc}represents the set of clustering centers, uij denotes the membership of xj belonging to vi. By minimizing Eq. (1), the membership matrix and the clustering centers can be updated by Eq.(2) and Eq.(3).

Equation

where dij denotes the distance between the j-th object xj and the i-th clustering center vi, and ηi is a scale parameter which can be estimated by using Eq.(4): ηi = ∑n j=1 um ij×d2 ij∑ n j=1 um ij . (4) Typically, the computational complexity of the traditional possibilistic c-means algorithm is dominated by calculating the distance between each object xj and every clustering center vi, which requires o(n × c) for each iteration. So, PCM has a computational complexity of o(tn × c) with t indicating the number of iterations. PCM is able to avoid the corruption of noise in the clustering big data sets. However, PCM is sensitive to initial parameters and usually produces a coincident clustering result [13]. Aiming at this problem, FPCM and PFCM were proposed by combining PCM and FCM [16]. Xie et al. [5] developed an enhanced PCM algorithm by grouping the data set into one main subset and one assistant subset to avoid the coincident result. In addition, PCM is not robust to the additional parameters. To tackle this problem, Yang et al. [17] proposed an unsupervised PCM scheme to improve the robustness of the conventional PCM algorithm. To cluster non-spherical data sets, some kernel-based possibilistic clustering algorithms have been proposed by mapping the objects of the data set into high order data space [18]. Other PCM variants include weighted PCM algorithm and sample-weighted PFCM algorithm [19, 20]. Although these algorithms can improve the performance of the conventional PCM clustering, they are all limited in the structured data clustering. Therefore, the paper proposes a high-order PCM algorithm to cluster heterogeneous data.

2.2 Big Data clustering

Over the past few years, some algorithms have been proposed for big data clustering, especially for heterogeneous data sets. Early works focused on image-text co-clustering by information fusion [10]. Specially, many algorithms ﬁrst extracted the image features and the text features separately, and then concatenated them into a single vector [21].However, the semethodsare difficult to produce desired clustering results since they cannot capture the complex correlations over the bi-modalities of the objects by concatenating the features in linear way. To tackle this problem, Jiang and Tan [22] proposed two methods based on the vague information and the Fusion ART to learn the visual-textual correlations by measuring the image-text similarities. Most of heterogeneous data clustering schemes are developed depending on graph theory. They usually transform the heterogeneous data clustering task into a graph partitioning problem. The most representative scheme of this type is the bipartite graph partition scheme proposed by Gao [8] for image-text clustering by interpreting the clustering task as a tripartite graph. Afterward, they extended this method for heterogeneous data clustering. The similar work is the isoperimetric co-clustering algorithm proposed by Rege et al. [23]. This algorithm clusters heterogeneous data by solving a set of linear equations. In addition, Cai et al. [24] developed as pectral clustering algorithm, which is are presentative method based on graph theory, for heterogeneous data clustering by designing an iterative process to optimize a uniﬁed objective function. Another graph theory based method is spectral relational clustering (SRC) presented by Long et al. [25]. SRC ﬁrst produces a collective clustering and then achieves the ﬁnal result by deriving an iterative spectral clustering. The major weakness of the heterogeneous data clustering algorithms based on the graph theory is that they have a high time complexity. Therefore, they are often inefﬁcient for large amounts of data. Another kind of heterogeneous data clustering is based on the matrix factorization theory. For instance, Chen et al. [9] presented a clustering algorithm based on non-negative matrix factorization scheme for heterogeneous data clustering by minimizing the global reconstruction function of the relational matrix over multiple modalities [26].Other heterogeneous data clustering techniques, such as the combinatorial Markov random fields (Comrafs), are depending on information theory [27]. Similar to the methods based on graph theory, these algorithms still have a high time complexity. For example, the computational complexity of Comrafs will increase signiﬁcantly with the growing amount of heterogeneous data.

3 HIGH-ORDER POSSIBILISTIC C-MEANS ALGORITHM BASED ON TENSOR REPRESENTATION MODEL

In this part, we present the HOPCM scheme for heterogeneous data clustering based on the tensor data representation model. The tensor data model represents each object by using a tensor [28]. For example, a colorful image can be represented as a 3-order tensor RIw×Ih×Ic, where Iw,Ih, and Ic denote width, height and color space, representatively. Specially, an image with 560×480 in the RGB color space can be represented by R560×480×3. Furthermore, a piece of video with MPEG-4 format can be represented as a 4-order tensor RIw×Ih×Ic×If with If denoting the frames. The tensor model can represent any heterogeneous data object. More importantly, it can capture the complex correlations over the multiple modalities of each heterogeneous data object. The tensor-based representation models have been successfully used in big data analysis and mining in past few years [3, 11, 26]. Therefore, HOPCM extends the conventional possibilistic c-means algorithm using the tensor data representation model. Assume that each object in the heterogeneous data set is represented by a T-order tensor RI1×I2×···×IT . By introducing the tensor model for representing heterogeneous data object, HOPCM is deﬁned as:

Equation

Where, m > 1 denotes a fuzziﬁcation constant. Generally, m → 1 results in approaching a hard cluster result while m →∞causes a high level of fuzziness [10, 11]. Eq. (5) shows that HOPCM has the similar objective function with PCM. However, different from PCM, d2 (T)ij indicates the distance between two tensors, namely the jth object xj and the ith clustering center vi. To calculate the distance d2 (T)ij between xj and vi, we unfold each tensor O ∈ RI1×I2×···×IT used to represent the object xj or the clustering center vi to its corresponding vector o. Specially, ol, the l-th item of the vector o, denotes xji1i2...iT by l = i1 + ∑T k=2∏k−1 t=1 It [29]. Thus, we can calculate the distance d2 (T)ij between xj and vi by an inner product as Eq. (6). d2 (T)ij =∑I1×I2×···IT l=1 (xjl −vil)2. (6) The goal of the high-order possibilistic c-means algorithm is to minimize the objective function (5). To update the membership value uij, we differentiate (5) with respect to uij and we can get:

∂Jm(U,V ) ∂uij = ∂ ∂uij (um ijd2 (T)ij + ηi(1−uij)m) = m·d2 (T)ij ·um−1 ij + m·ηi(1−uij)m−1

. (7) We can get the equation for updating uij by setting Eq. (7) to 0 as Eq. (8).

uij = 1 (1+(d2 (T)ij/ηi)1/(m−1))

,∀i,j. (8) Similarly, we can get the equation for updating vi with the same format as Eq. (3). Therefore, HOPCM is outlined in Algorithm 1.

Algorithm 1: The High-order Posssibilistic c-Means Algorithm. Input: X = {X1,X2,...,XN}, c, m, maxiter Output: U = {uij},V = {vi}1 for iteration = 1,2,...,maxiter do 2 for i = 1,2,...,c do 3 vi = ∑n j=1 um ijxj ∑n j=1 um ij ; 4 ηi = ∑n j=1 um ij×d2 (T)ij∑ n j=1 um ij ; 5 for i = 1,2,...,c do 6 for j = 1,2,...,n do 7 uij = 1 (1+(d2 (T)ij/ηi)1/(m−1)) ;

From Algorithm 1, the time complexity of HOPCM is controlled by calculating the tensor distance between nxk and vi, which needs O(n × c) for each iteration. Therefore, HOPCM has a computational complexity of O(tn×c) with t iterations.

4 DISTRIBUTED HIGH-ORDER POSSIBILISTIC CMEANS ALGORITHM BASED ON MAPREDUCE

In this part, to increase the efficiency of HOPCM for big data, we design a distributed high-order prossibilistic c-means algorithm (DHOPCM) based on MapReduce which is an efficient cloud computing programming model for massive data computation [30].

Algorithm 1 shows that the most important steps of HOPCM is to calculate the membership matrix and the clustering centers. Therefore, we use the Map function to calculate the membership matrix and use the Reduce function to calculate the clustering centers. The whole scheme is outlined in Fig. 1.

From Eq. (8) for updating the membership matrix, only the object xi and clustering centers V = {v1,v2,...,vc} are required for calculating the membership values of the object xi towards each clustering center. Therefore, to reduce the storage of each computing node and the communication, we partition the membership matrix into p sub-matrices U = {U1,U2,...,Up} by columns. The dataset X is also partitioned into p subsets X = {X1,X2,...,Xp} accordingly. As shown in Fig. 1, in the Map phrase, we dispatch each sub-matrix, the corresponding subset and all the clustering centers to one computing node for updating the membership matrix.

Eq. (3) shows that it requires all the data objects to calculate the clustering center vi. Therefore, the communication will increase signiﬁcantly if calculating the clustering centers directly in the Reduce function. So, two parameters, i.e., αi = {α(1) i ,α(2) i ,...,α(p) i } and βi = {β(1) i ,β(2) i ,...,β(p) i }, are deﬁned to help calculate the clustering center vi. α(t) i and β(t) i can be calculated in the t-th computing node by the following equations.

Equation

In the Reduce phrase, α(t) i (t = 1,2,...,p) and the corresponding β(t) i (t = 1,2,...,p) are dispatched to the same computing node for calculating the clustering center vi by Eq. (11). vi = ∑p t=1 α(t) i∑ p t=1 β(t) i . (11)

Equation

After all the clustering centers are calculated by the Reduce function using Eq.(11),they will be dispatched to each computing node for restarting another MapReduce job until convergence.

DHOPCM has a time complexity of O (tnc/p)+O(commu), p indicating the number of computers employed to perform HOPCM and O(commu) denoting the communication overhead. However, the communication takes signiﬁcantly less time than the calculation on the cluster process, especially in a centralized the cloud computing platform. So, it can be ignored. Therefore, DHOPCM has an approximate computational complexity of O (tnc/p).

5 PRIVACY-PRESERVING HIGH-ORDER POSSIBILISTIC C-MEANS ALGORITHM BASED ON BGV

In the last part, we present a DHOPCM scheme based on MapReduce, which can signiﬁcantly increase the efficiency for clustering big data by employing many cloud servers. However, private data usually suffers from disclosure when performing DHOPCM on cloud. Specially, there is huge scale of sensitive heterogeneous data, such as medical information and clinical charts, in medical area, which are vital to patients. Once they are leaked, patient’s lives and property will suffer a great threat [31].

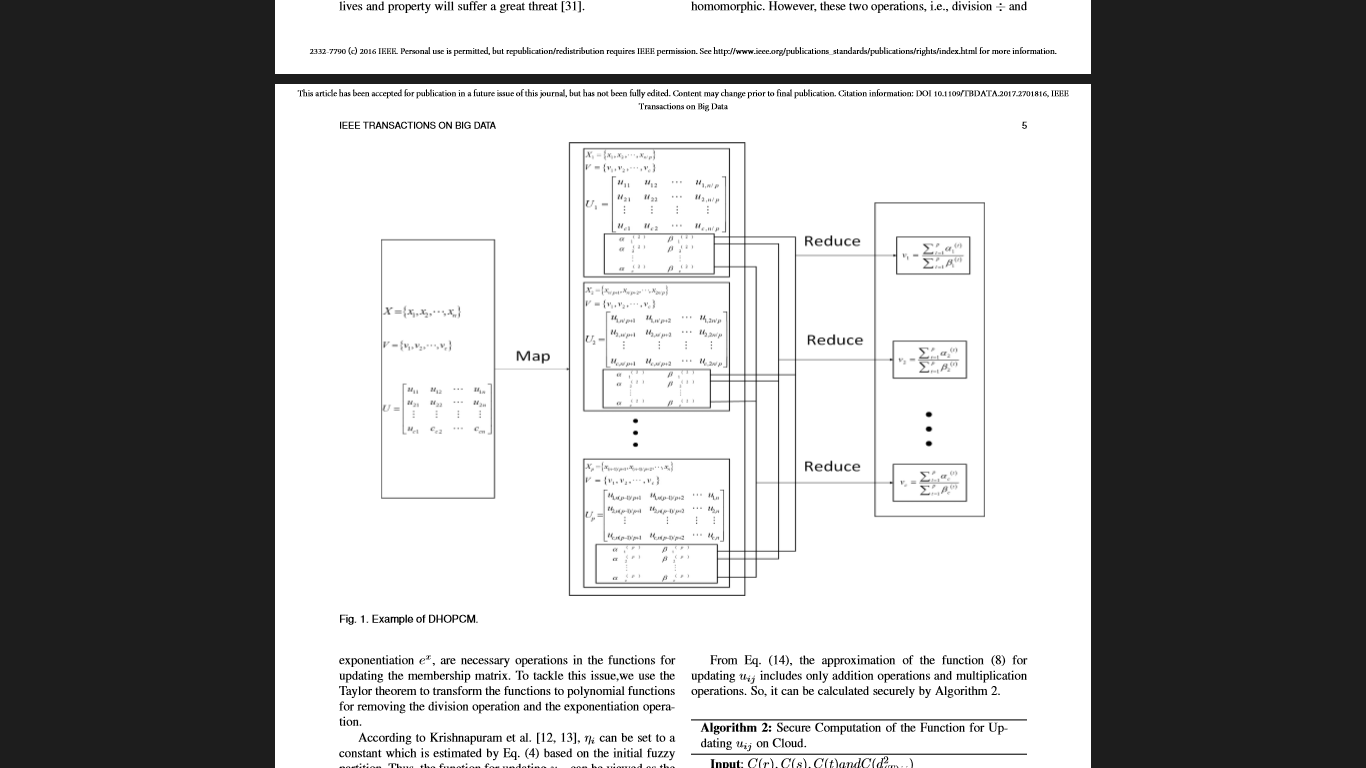
To protect the private data, we devise a privacy-preserving HOPCM scheme (PPHOPCM) based on BGV in this section. The proposed scheme cannot only employ cloud servers to increase the clustering efficiency for large amounts of heterogeneous data, but also avoid the disclosure of the private data. BVG secure operations required for PPHOPCM are described ﬁrst, followed by the details of the proposed scheme.

5.1 BGV Secure Operations

BGV is a leveled fully homomorphism encryption technique. It uses a Setup procedure to select a µ-bit modulus q and the following parameters: the dimension n = n(λ,µ), the degree d = d(λ,µ), the distribution χ = χ(λ,µ), and N = ⌈(2n + 1)logq⌉. Furthermore, a key Switching procedure and a modulus Switching procedure are implemented in the BGV scheme. The former is used to reduce the dimension of the cipher text while the latter is aims to reduce the noise. The BGV technique has four major secure operations, i.e., encryption, decryption, secure addition and secure multiplication, required for implementing our proposed PPHOPCM scheme, listed as follows [15]. (1) Encryption: Encrypt a plaintext m ∈ R2 as a cipher text c ← m + ATr ∈ Rn+1 q . (2) Decryption: Decrpt a cipher text c to its plaintext m ←(( < c,sj > modq) mod 2) using the corresponding secret key sj. (3) Secure Addition: Add two cipher texts, i.e., c1 and c2, to their sum c4 on cloud by c3 ← c1 + c2 mod qj, and c4 ← Refresh(c3,τ(sj′ → sj−1),qj,qj−1). (4) Secure Product: Multiply two cipher texts, i.e., c1 and c2, to their product c4 on cloud by c3 ← c1 ⊗c2 mod qj, and c4 ← Refresh(c3,τ(sj′ → sj−1),qj,qj−1). Compared with other encryption schemes, there are two major advantages of the BGV technique. First, it is a fully homomorphic encryption scheme which supports an arbitrary number of addition operations and multiplication operations simultaneously. Second, BGV is more efficient than most of other encryption schemes. So, BGV is used to encrypt the private data in this paper.

5.2 Approximation of the Functions for Updating Membership Matrix and Clustering Centers

Table 1 lists ﬁve types of operations required for updating the membership matrix and the clustering centers in the high-order possibilistic c-means algorithm, i.e., addition +, subtraction −, multiplication×, division÷, and exponentiation ex. TABLE 1 Operations required in HOPCM.



Operation Homomorphic Example + Y ∑n j=1 um ij − Y m−1 × Y um ijxj ex N um ij ÷ N d2 ij/ηi

ForﬁvetypesofoperationslistedinTable1,only addition +, subtraction −, and multiplication × are homorphic. The last two operations are not supported by BGV since they are not homomorphic. However, these two operations, i.e., division ÷ and exponentiation ex, are necessary operations in the functions for updating the membership matrix. To tackle this issue, we use the Taylor theorem to transform the functions to polynomial functions for removing the division operation and the exponentiation operation. According to Krishnapuram et al. [12, 13], ηi can be set to a constant which is estimated by Eq. (4) based on the initial fuzzy partition. Thus, the function for updating uij can be viewed as the functionwithregardtod2 (T)ij with the deﬁnition domain[0,+∞). So, we may rewrite the function for updating uij as:

uij = f(x) = 1 1+(x/ηi)b ,x ∈ [0,+∞). (12) Obviously, f(x) is an elementary function so we can write it as a polynomial function by the Taylor theorem with the expansion point a ∈ (0,+∞) as follows:

Equation (13)

Let r = 1/(1 + (a/ηi)b), s = bη2b i ab−1/(ηb i + ab)2 and t = (b(b − 1)η3b i ab−2 − b(b + 1)η2b i a3b−2 −2 bη3b i a2b−2)/2(ηb i + ab)4, we can get the approximation of the function (12) for uij as:

uij = f(x) ≈ r + s(x−a) + t(x−a)2. (14)

From Eq. (14), the approximation of the function (8) for updating uij includes only addition operations and multiplication operations. So, it can be calculated securely by Algorithm 2.

Algorithm 2: Secure Computation of the Function for Updating uij on Cloud. Input: C(r),C(s),C(t)andC(d2 (T)ij) Output: C(uij) 1 Using secure multiplication to compute: ; 2 C1 = C(s)×C(d2 (T)ij −α); 3 Using secure multiplication to compute: ; 4 C2 = C(t)×C(d2 (T)ij −α)×C(d2 (T)ij −α); 5 Using secure addition to compute: C(uij) = C(r) + C1 + C2; 6 return C(uij);

Similarly, the function (3) for updating vi can be viewed as the multivariable function with regards to (ui1,ui2,...,uin) and can be rewritten as: vi = f(ui1,ui2,...,uin) = ∑n j=1 um ijxj ∑n j=1 um ij . (15) So, we can transform Eq. (3) to a polynomial function by the multi-variable Taylor theorem with the expansion points α1,α2,...,α ∈ (0,1) as follows:

Equation

The secure computation of the function (16) is outlined in Algorithm 3.

Algorithm 3: Secure Computation of the Function for Updating vi on Cloud.

Input: C (λ),C (βj),C (uij),∀I,j

Output: C(vi)

1 for j = 1,2,...,n do

2 Using secure multiplication to compute: ;

3 C (λi) = C (βj)×C (uij –αi) ;

4 Using secure addition to compute: ;

5 C (vi) = C (λ) +∑n j=1 C (λi);

6 return C (vi);

5.3 Privacy-preserving High-order Possibilistic cMeans Algorithm on Cloud Give the heterogeneous dataset X = {x1,x2,...,xn}, the privacy-preserving high-order possibilistic c-means algorithm (PPHOPCM) aims to group X into c clusters by performing the secure HOPCM algorithm on cloud without disclosure of the private data. PPHOPCM is outlined in Algorithm 4. To cluster the dataset X securely, PPHOPCM initializes the membership matrix U = {uij} and clustering centers V = {vi} in the client, and estimates the value of ηi on line 2. At the beginning of clustering, the client encrypts the objects X = {x1,x2,...,xn}, the membership matrix U = {uij},the clustering V = {vi}and other parameters, e.g., m and ηi on line 3. The cloud servers performs one iteration of the PPHOPCM algorithm on the cipher texts for updating the membership matrix and the clustering centers, and then transmits the intermediate results, i.e., the membership matrix U = {uij} and the clustering centers V = {vi} to the client on lines 914. After downloading the intermediate results, the client decrypts U = {uij} and V = {vi} on line 16, and then re-encrypts them to upload them on the cloud for another iteration on lines 6-7. The process will be repeated until the convergence. FromAlgorithm4, PPHOPCM cannot only protect the dataset but also the clustering results since the heterogeneous data sets, the membership matrix and the clustering centers are encrypted simultaneously. More importantly, the whole clustering process is executed on the cipher texts. Therefore, PPHOPCM is able to preserve the privacy effectively.

5.4 Complexity Analysis

Now, we estimate the computation complexity and the communication complexity of the PPHOPCM scheme. We use ADD and MUL to represent the time cost of one addition operation and one multiplication, representatively. Computation Cost. Assume that the dataset X = {x1,x2,...,xk}, each represented by a T-order tensor RI1×I2×···×IT , is grouped into c clusters. PPHOPCM encrypts all the objects, the fuzzy constant m and η with N × (n + 1)×(k∏T t=1 It +c+1) (ADD+MUL) before uploading them to the cloud. When performing PPHOPCM on the cloud, the client encrypts the membership U = {uij} and the cluster centers V = {vi}with c×(k2 + k + 1)×(n + 1)×N (ADD+MUL), and decrypts the intermediate results with (c×(k2 + k + 1)× n×N)MUL + (c×k(k + 1)×(n + 1)×N)ADD once in each iteration. In each iteration, PPHOPCM performs (c2 ×k×∏T t=1 It + 3)×(2n+N+1)ADDand(c2×k×∏T t=1 It+2)×(4n+2N+1) MUL to calculate the membership matrix by using Algorithm 2. Simultaneously, it performs(2c×k2+1)×(2n+N+1)ADD and (c×k2+2)×(4n+2N+1)MUL to calculate the cluster centers in each iteration. Obviously, the computation cost will increase with the growing number of the objects, however, a large number of cloud servers can be employed to perform the clustering algorithm efficiently. Communication Cost. Before performing PPHOPCM, the client uploads k∏T t=1 It+c+1 messages, with (k∏T t=1 It+c+ 1)×(n+1)×µ bits, to the cloud. And then, the client exchanges c×(k +1) messages, with c×(k +1)×(n+1)×µ bits, with the cloud in each iteration.

Algorithm 4: The Privacy-preserving High-order Posssibilistic c-Means Algorithm.

Input: X = {X1,X2,...,XN}, c, m, maxiter

Output: U = {uij},V = {vi}

1 Client: ;

2 Randomly initialize the parameters ;

3 Encrypt X, m, and η ;

4 Upload the ciphertexts to the cloud ;

5 for iteration = 1,2,...,maxiter do

6 Encrypt U and V ;

7 Upload ciphertexts to the cloud ;

8 Cloud: ;

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

10 Perform Algorithm 3 to calculate vi;

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

12 for j = 1,2,...,n do

13 Perform Algorithm 2 to calculate uij;

14 Send the immediate results to the client ;

15 Client: ;

16 Decrypt immediate results for updating U and V ;

6 EXPERIMENTS

To estimate the clustering accuracy and efficiency of our schemes, we perform the proposed algorithms on the cloud platform including 20 nodes, each with 3.2 GHz Core i7 CPU and 8GB memory. We ﬁrst compare our HOPCM algorithm with HOPCM-15, wPCM and PCM in clustering accuracy on two representative big data sets, i.e., NUS-WIDE and SNAE2. And then, we evaluate the clustering efficiency of the PPHOPCM algorithm by comparison with HOPCM and DHOPCM. At last, we estimate the scalability of PPHOPCM and DHOPCM based on speedup.

6.1 Data Sets and Evaluation Criteria

Two representative big data sets, i.e., NUS-WIDE and SNAE2, are used to estimate the clustering accuracy and efficiency of our schemes. NUS-WIDE is downloaded from Flikr.com [32]. It consists more than 260,000 images which are grouped by 81 classes. All the images are annotated by some texts, constituting a heterogeneous dataset. To evaluate the robustness of our proposed schemes, we sampled 80, 000 representative images which can be averaged to 8 subsets, each grouped by 14 categories, from NUSWIDE. SNAE2, downloaded from YouTube, includes 1800 pieces of videos, grouped by four classes, i.e., sport, news, advertisement and entertainment. Each video consists 100 frames, represented by a 4-order tensor in our schemes. In this paper, we use two criteria, i.e., E∗ and ARI, to evaluate the clustering accuracy of the proposed schemes [11]. E∗ is aimed to estimate the clustering accuracy of the centers gotten by the clustering technique ∗ by calculating the distance between the produced centers v∗ and the actual ones vi deal as Eq. (17). E∗ =√ c ∑ i=1||vii deal −vi ∗||2, (17) ARI(U,U′) is aimed to evaluate the clustering accuracy by measuring the agreement between U and U′, U indicating the actual labels and U′ denoting the labels gotten by the clustering technique∗. Obviously, a smaller E∗ and a bigger ARI demonstrate a higher clustering quality produced by the proposed method.

6.2 Performance Evaluation of HOPCM

The task of this experiment is to evaluate the clustering accuracy of HOPCM in terms of E∗ and ARI by comparison with three representativepossibilisticclusteringalgorithms, i.e., HOPCM-15, wPCM, and PCM. HOPCM-15 is proposed by Zhang et al. [11] for heterogeneous data clustering while wPCM is a weighted PCM scheme. Different from our HOPCM scheme, HOPCM-15 learns features from heterogeneous data using improved auto-encoder model before clustering. For wPCM and PCM, we concatenate the attributes of each modality to form a single vector for clustering heterogeneous data. Table 2 shows the clustering result on the NUS-WIDE based on E∗ while Table 3 presents the result based on ARI. From the results, we can observe that HOPCM produces the lowest values of E∗ and the highest values of ARI in most cases. Specially, HOPCM yields the E∗ value of 2.72 and the ARI value of 0.91, representatively, on the whole dataset. Such results imply that HOPCM performs best in clustering accuracy. PCM and wPCM perform worst in terms of E∗ and ARI since they cannot capture the complex correlations over the multiple modalities of heterogeneous objects only by concatenating their attributes. HOPCM-15 usually performs similarly with HOPCM in clustering the NUS-WIDE dataset, sometimes even better than HOPCM. This is because HOPCM-15 calculates the similarity between each object and the clustering centers by adopting tensor distance that can reveal the distributions for some heterogeneous objects. However, HOPCM-15 performs signiﬁcantly less efficiently than HOPCM, which will be shown subsequently. We perform the four algorithms on the different proportions of the NUS-WIDE dataset to evaluate the clustering efficiency. The result is shown in Fig. 2.

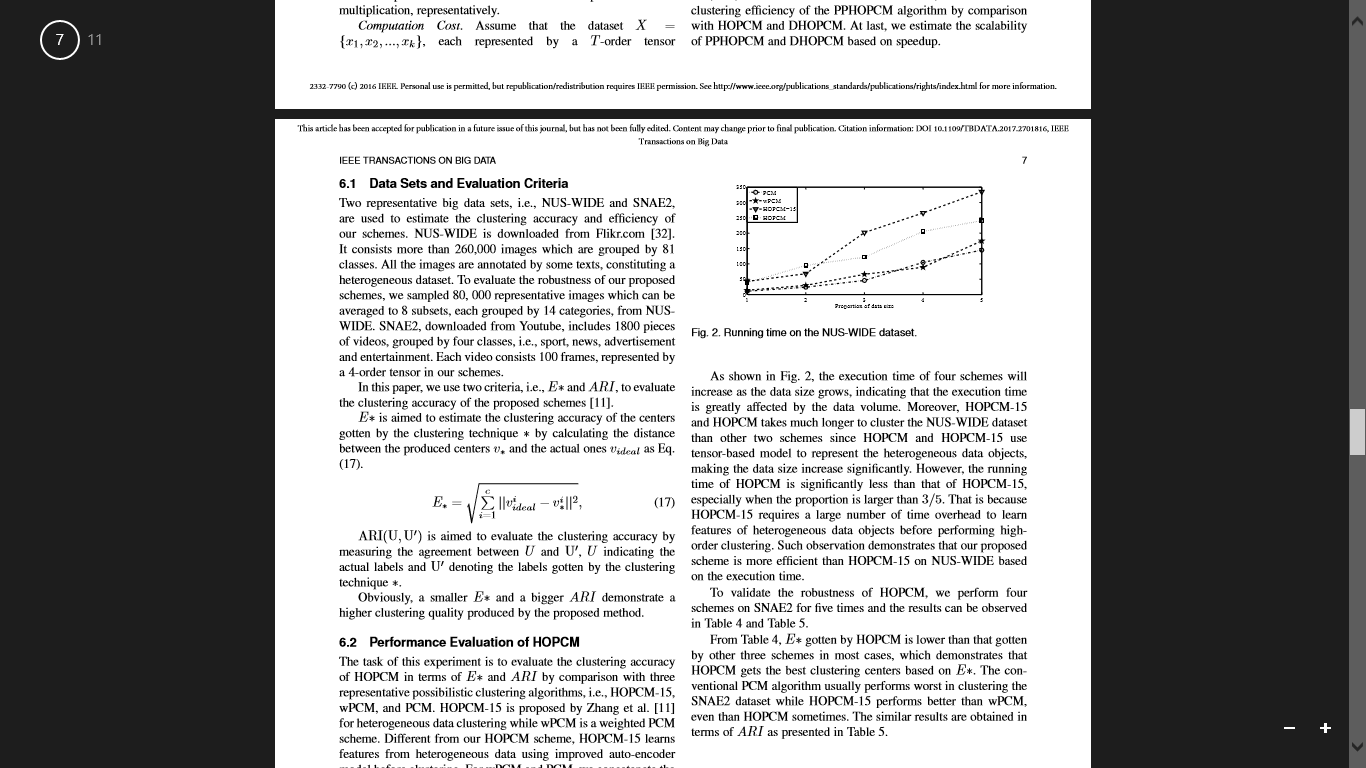


Fig. 2. Running time on the NUS-WIDE dataset.

As shown in Fig. 2, the execution time of four schemes will increase as the data size grows, indicating that the execution time is greatly affected by the data volume. Moreover, HOPCM-15 and HOPCM takes much longer to cluster the NUS-WIDE dataset than other two schemes since HOPCM and HOPCM-15 use tensor-based model to represent the heterogeneous data objects, making the data size increase signiﬁcantly. However, the running time of HOPCM is signiﬁcantly less than that of HOPCM-15, especially when the proportion is larger than 3/5. That is because HOPCM-15 requires a large number of time overhead to learn features of heterogeneous data objects before performing high order clustering. Such observation demonstrates that our proposed scheme is more efficient than HOPCM-15 on NUS-WIDE based on the execution time. To validate the robustness of HOPCM, we perform four schemes on SNAE2 for ﬁve times and the results can be observed in Table 4 and Table 5. From Table 4, E∗ gotten by HOPCM is lower than that gotten by other three schemes in most cases, which demonstrates that HOPCM gets the best clustering centers based on E∗. The conventional PCM algorithm usually performs worst in clustering the SNAE2 dataset while HOPCM-15 performs better than wPCM, even than HOPCM sometimes. The similar results are obtained in terms of ARI as presented in Table 5.

6.3 Performance Evaluation of PPHOPCM

We evaluate the clustering accuracy and efficiency of PPHOPCM by comparison with DHOPCM and HOPCM on NUS-WIDE and SNAE2. First, we present the results of three schemes in clustering accuracy in Table 6 - Table 9. AspresentedinTable6-Table9, PPHOPCM produces lower accuracy than HOPCM and DHOPCM, which is demonstrated by that PPHOPCM, achieves a higher value of E∗ and a lower value of ARI. This is because the approximation of the functions for updating membership matrix and clustering centers will lead to some errors. Specially, PPHOPCM produces about 3% more error than HOPCM for clustering NUS-WIDE and SNAE2 in terms of ARI. However, such an error is permitted for massive heterogeneous data. Therefore, PPHOPCM can satisfy the accuracy requirement for massive heterogeneous data clustering. Moreover, DHOPCM performs the same as HOPCM in terms of E∗ and ARI for clustering NUS-WIDE and SNAE2 since they uses the same tensor-based data representation model and distance metrics. The difference is that DHOPCM is performed on cloud servers while HOPCM on a single server, resulting a different clustering efficiency.

Table 2

Clustering in terms of E\* on NUS-WIDE to evaluate HOPCM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Overall |
| PCM | 4.52 | 6.15 | 5.22 | 5.47 | 5.51 | 4.12 | 4.29 | 5.69 | 5.78 |
| wPCM | 4.21 | 3.96 | 4.11 | 5.23 | 3.82 | 4.18 | 4.64 | 5.25 | 4.83 |
| HOPCM-15 | 1.98 | 2.57 | 2.91 | 2.63 | 2.12 | 2.91 | 3.29 | 2.08 | 2.93 |
| HOPCM | 2.06 | 2.13 | 2.26 | 3.08 | 2.03 | 2.67 | 2.25 | 2.18 | 2.72 |

Table 3

Clustering in terms of ARI on NUS-WIDE to evaluate HOPCM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Overall |
| PCM | 0.63 | 0.59 | 0.72 | 0.75 | 0.71 | 0.64 | 0.73 | 0.59 | 0.71 |
| wPCM | 0.69 | 0.71 | 0.78 | 0.81 | 0.75 | 0.79 | 0.77 | 0.69 | 0.77 |
| HOPCM-15 | 0.91 | 0.84 | 0.94 | 0.91 | 0.88 | 0.92 | 0.82 | 0.84 | 0.90 |
| HOPCM | 0.90 | 0.87 | 0.92 | 0.94 | 0.89 | 0.93 | 0.85 | 0.82 | 0.91 |

Table 4

Clustering in terms of E\* on SNAE2 to evaluate HOPCM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | Average |
| PCM | 9.06 | 8.54 | 10.15 | 8.75 | 9.63 | 9.226 |
| wPCM | 8.46 | 9.21 | 8.13 | 7,92 | 9.01 | 8.546 |
| HOPCM-15 | 7.96 | 8.15 | 7.37 | 8.24 | 8.55 | 8 |
| HOPCM | 7.89 | 8.62 | 7.21 | 7.96 | 8.04 | 7.944 |

Table 5

Clustering in terms of ARI on SNAE2 to evaluate HOPCM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | Average |
| PCM | 0.69 | 0.78 | 0.64 | 0.73 | 0.65 | 0.698 |
| wPCM | 0.75 | 0.71 | 0.82 | 0.76 | 0.70 | 0.748 |
| HOPCM-15 | 0.87 | 0.84 | 0.89 | 0.81 | 0.76 | 0.834 |
| HOPCM | 0.90 | 0.81 | 0.89 | 0.85 | 0.83 | 0.856 |

Table 6

Clustering in terms of E\* on NUS-WIDE to evaluate PPHOPCM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Overall |
| HOPCM | 2.06 | 2.13 | 2.26 | 3.08 | 2.03 | 2.67 | 2.25 | 2.18 | 2.72 |
| DHOPCM | 2.06 | 2.13 | 2.26 | 3.08 | 2.03 | 2.67 | 2.25 | 2.18 | 2.72 |
| PPHOPCM | 2.19 | 2.42 | 2.71 | 3.66 | 2.11 | 2.85 | 3.05 | 2.17 | 2.91 |

Table 7

Clustering in terms of ARI on NUS-WIDE to evaluate PPHOPCM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Overall |
| HOPCM | 0.90 | 0.87 | 0.92 | 0.94 | 0.89 | 0.93 | 0.85 | 0.82 | 0.91 |
| DHOPCM | 0.90 | 0.87 | 0.92 | 0.94 | 0.89 | 0.93 | 0.85 | 0.82 | 0.91 |
| PPHOPCM | 0.86 | 0.85 | 0.89 | 0.91 | 0.88 | 0.91 | 0.84 | 0.79 | 0.88 |

Table 8

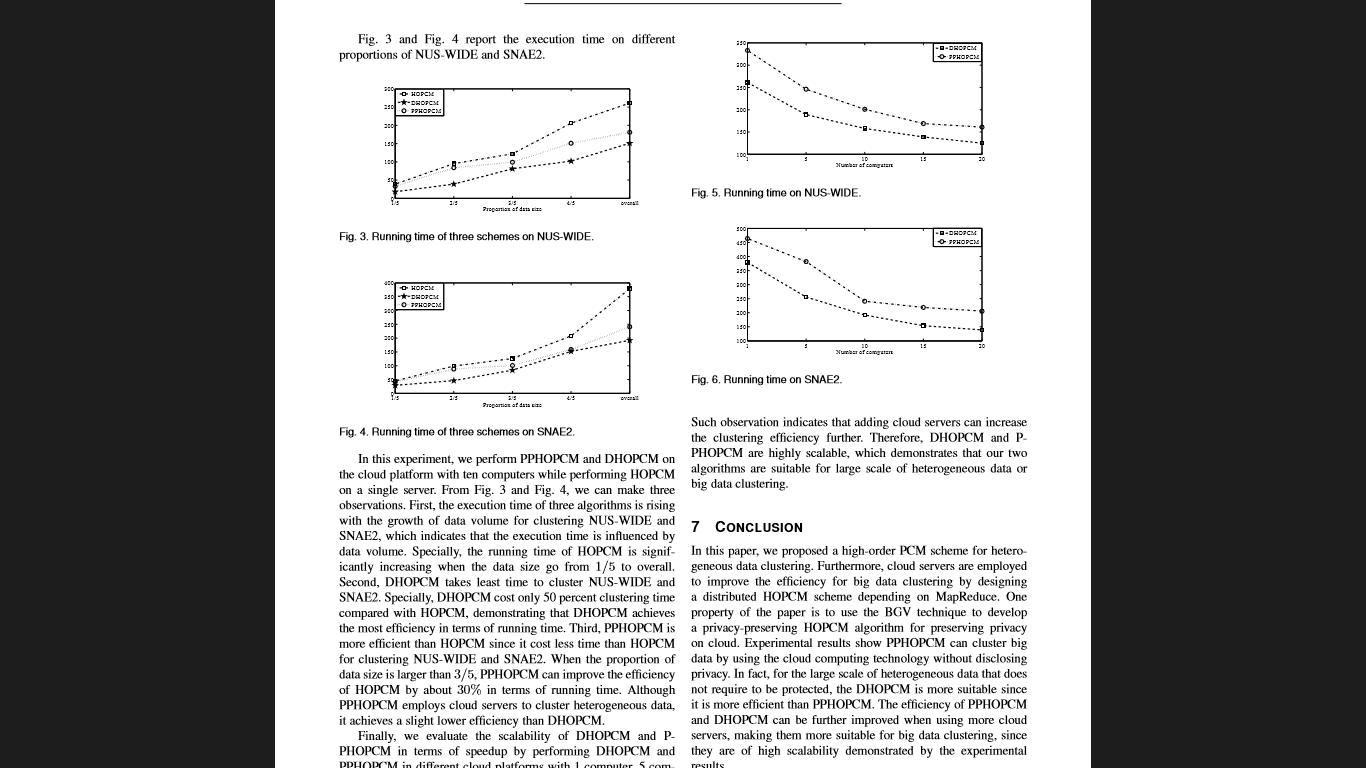
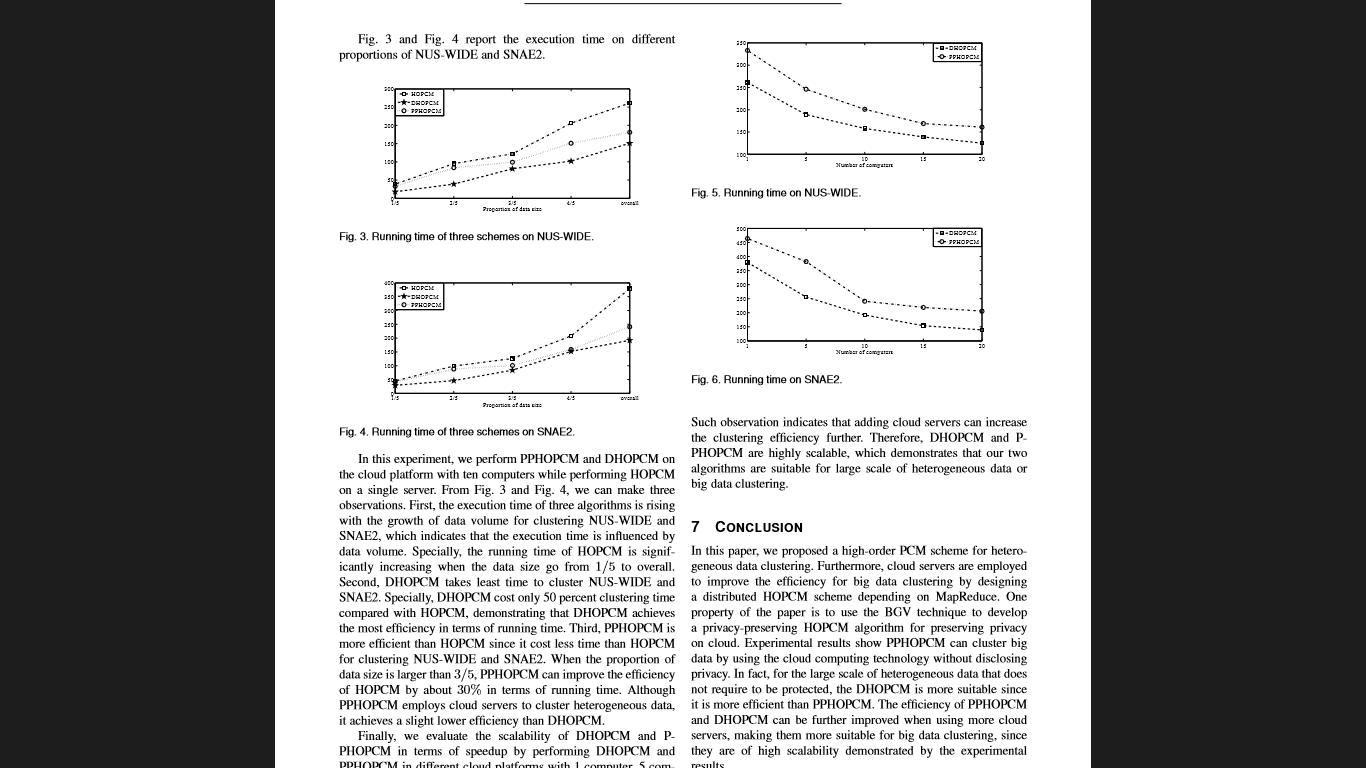
Clustering in terms of E\* on SNAE2 to evaluate HOPCM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | Average |
| HOPCM | 7.89 | 8.62 | 7.21 | 7.96 | 8.04 | 7.944 |
| DHOPCM | 7.89 | 8.62 | 7.21 | 7.96 | 8.04 | 7.944 |
| PPHOPCM | 8.01 | 8.89 | 7.64 | 8.12 | 8.58 | 8.408 |

Table 9

Clustering in terms of ARI on SNAE2 to evaluate HOPCM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithmic dataset | 1 | 2 | 3 | 4 | 5 | Average |
| HOPCM | 0.90 | 0.81 | 0.89 | 0.85 | 0.83 | 0.856 |
| DHOPCM | 0.90 | 0.81 | 0.89 | 0.85 | 0.83 | 0.856 |
| PPHOPCM | 0.88 | 0.76 | 0.76 | 0.81 | 0.82 | 0.828 |



In this experiment, we perform PPHOPCM and DHOPCM on the cloud platform with ten computers while performing HOPCM on a single server. From Fig. 3 and Fig. 4, we can make three observations. First, the execution time of three algorithms is rising with the growth of data volume for clustering NUS-WIDE and SNAE2, which indicates that the execution time is inﬂuenced by data volume. Specially, the running time of HOPCM is significantly increasing when the data size go from 1/5 to overall. Second, DHOPCM takes least time to cluster NUS-WIDE and SNAE2. Specially, DHOPCM cost only 50 percent clustering time compared with HOPCM, demonstrating that DHOPCM achieves the most efficiency in terms of running time. Third, PPHOPCM is more efficient than HOPCM since it cost less time than HOPCM for clustering NUS-WIDE and SNAE2. When the proportion of data size is larger than 3/5, PPHOPCM can improve the efficiency of HOPCM by about 30% in terms of running time. Although PPHOPCM employs cloud servers to cluster heterogeneous data, it achieves a slight lower efficiency than DHOPCM. Finally, we evaluate the scalability of DHOPCM and PPHOPCM in terms of speedup by performing DHOPCM and PPHOPCM in different cloud platforms with 1 computer, 5 computers, 10 computers and 20 computers, respectively, on NUSWIDE and SNAE2. From Fig. 5 and Fig. 6, the execution time of two schemes reduces as the number of the employed cloud server’s increase.

Such observation indicates that adding cloud servers can increase the clustering efficiency further. Therefore, DHOPCM and PPHOPCM are highly scalable, which demonstrates that our two algorithms are suitable for large scale of heterogeneous data or big data clustering.

7 CONCLUSIONS

In this paper, we proposed a high-order PCM scheme for heterogeneous data clustering. Furthermore, cloud servers are employed to improve the efficiency for big data clustering by designing a distributed HOPCM scheme depending on MapReduce. One property of the paper is to use the BGV technique to develop a privacy-preserving HOPCM algorithm for preserving privacy on cloud. Experimental results show PPHOPCM can cluster big data by using the cloud computing technology without disclosing privacy. In fact, for the large scale of heterogeneous data that does not require to be protected, the DHOPCM is more suitable since it is more efficient than PPHOPCM. The efficiency of PPHOPCM and DHOPCM can be further improved when using more cloud servers, making them more suitable for big data clustering, since they are of high scalability demonstrated by the experimental results. In this work, the proposed schemes are preliminarily evaluated on two representative heterogeneous datasets. In the future work, the proposed algorithms will be further validated on larger actual datasets.

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, Jan. 2014.

[2] B. Ermis, E. Acar, and A. T. Cemgil, ”Link Prediction in Heterogeneous Data via Generalized Coupled Tensor Factorization,” Data Mining and Knowledge Discovery, vol. 29, no. 1, pp. 203-236, 2015.

[3] 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, Jan. 2016.

[4] N.SoniandA.Ganatra,”MOiD(MultipleObjectsIncrementalDBSCAN) - A Paradigm Shift in Incremental DBSCAN,” International Journal of Computer Science and Information Security, vol. 14, no. 4, pp. 316-346, 2016.

[5] Z. Xie, S. Wang, and F. L. Chung, ”An Enhanced Possibilistic c-Means ClusteringAlgorithmEPCM,”SoftComputing,vol.12,no.6,pp.593-611, 2008.

[6] Q. Zhang, C. Zhu, L. T. Yang, Z. Chen, L. Zhao, and P. Li, ”An Incremental CFS Algorithm for Clustering Large Data in Industrial Internet of Things,” IEEE Transactions on Industrial Informatics, 2015. DOI: 10.1109/TII.2017.2684807.

[7] X. Zhang, ”Convex Discriminative Multitask Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 1, pp. 28-40, Jan. 2015.

[8] B. Gao, T. Liu, T. Qin, X. Zheng, Q. Cheng, and W. Ma, ”Web Image Clustering by Consistent Utilization of Visual Features and Surrounding Texts,” in Proceedings of the 13th Annual ACM International Conference on Multimedia, 2005, 112-121.

[9] Y. Chen, L. Wang, and M. Dong, ”Non-Negative Matrix Factorization for Semisupervised Heterogeneous Data Coclustering,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1459-1474, Oct. 2010.

[10] L. Meng, A. 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, Aug. 2014.

[11] Q. Zhang, L. T. Yang, Z. Chen, and Feng Xia, ”A High-Order Possibilistic-Means Algorithm for Clustering Incomplete Multimedia Data,” IEEE Systems Journal, 2015, DOI: 10.1109/JSYST.2015.2423499.

[12] R. Krishnapuram and J. M. Keller, ”A Possibilistic Approach to Clustering,” IEEE Transactions on Fuzzy Systems, vol. 1, no. 2, pp. 98-110, May 1993.

[13] R.KrishnapuramandJ.M.Keller,”ThePossibilisticc-MeansAlgorithm: Insights and Recommendations,” IEEE Transactions on Fuzzy Systems, vol. 4, no. 3, pp. 385-393, Aug. 1996.

[14] 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.

[15] 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, May 2016.

[16] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, ”A Possibilistic Fuzzy c-MeansClusteringAlgorithm,”IEEETransactionsonFuzzySystems,vol. 13, no. 4, pp. 517-530, Aug. 2005.

[17] M. Yang and C. Lai, ”A Robust Automatic Merging Possibilistic Clustering Method,” IEEE Transactions on Fuzzy Systems, vol. 19, no. 1, pp. 26-41, Feb. 2011.

[18] M. Filippone, F. Masulli, and S. Rovette, ”Applying the Possibilistic cMeansAlgorithminKernel-InducedSpaces,”IEEETransactionsonFuzzy Systems, vol. 18, no. 3, pp. 572-584, Jun. 2010.

[19] A. Schneider, ”Weighted Possibilistic c-Means Clustering Algorithms,” in Proceedings of the 9th IEEE International Conference on Fuzzy Systems, 2000, pp. 176-180.

[20] B. Liu, S. Xia, Y. Zhou, and X. Han, ”A Sample-Weighted Possibilistic Fuzzy Clustering Algorithm,” Acta Electronica Sinica, vol. 30, no. 2, pp. 371-375, 2012.

[21] R. Zhao and W. Grosky, ”Narrowing the Semantic Gap Improved TextBased Web Document Retrieval Using Visual Features,” IEEE Transactions on Multimedia, vol. 4, no. 2, pp. 189-200, Jun. 2002.

[22] T. Jiang and A.-H. Tan, ”Learning Image-Text Associations,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 2, pp. 161-177, Feb. 2009.