Secure weighted possibilistic c-means algorithm on cloud for clustering big data

Qingchen Zhang[a,b], Laurence T. Yang[a,b,*], Arcangelo Castiglione[c], Zhikui Chen[d], Peng Li[d]

[a] School of Information and Communication Engineering, University of Electronic Science and Technology of China, China
[b] Department of Computer Science, St. Francis Xavier University, Antigonish, Canada
[c] Department of Computer Science, University of Salerno, Fisciano, Italy
[d] School of Software Technology, Dalian University of Technology, Dalian, China

Keywords:
Big data
Possibilistic c-means algorithm
Cloud computing
BGV

A B S T R A C T

The weighted possibilistic c-means algorithm is an important soft clustering technique for big data analytics with cloud computing. However, the private data will be disclosed when the raw data is directly uploaded to cloud for efficient clustering. In this paper, a secure weighted possibilistic c-means algorithm based on the BGV encryption scheme is proposed for big data clustering on cloud. Specially, BGV is used to encrypt the raw data for the privacy preservation on cloud. Furthermore, the Taylor theorem is used to approximate the functions for calculating the weight value of each object and updating the membership matrix and the cluster centers as the polynomial functions which only include addition and multiplication operations such that the weighed possibilistic c-means algorithm can be securely and correctly performed on the encrypted data in cloud. Finally, the presented scheme is estimated on two big datasets, i.e., eGSAD and sWSN, by comparing with the traditional weighted possibilistic c-means method in terms of effectiveness, efficiency and scalability. The results show that the presented scheme performs more efficiently than the traditional weighted possiblistic c-means algorithm and it achieves a good scalability on cloud for big data clustering.

1. Introduction

Recent years have witnessed a considerable development in Internet of Things with the rapid proliferation of mobile devices and sensing techniques [1,6,26]. Specially, Internet of Things are being widely used in smart cities, intelligent transportation and industrial manufacture [7]. A typical Internet of Things system usually consists of three layers from bottom to top, i.e., physical layer, network layer and application layer, as presented in Fig. 1.

In the Internet of Things systems, the physical layer uses sensing devices such as sensors, RFID and two-dimensional codes to collect data and then the collected data is transmitted to the application layer through the network layer. In Internet of Things, a dedicated network is usually combined with the Internet to achieve the real-time and dependable transmission for the collected data. In the application layer, the collected data is analyzed typically using the cloud computing techniques

* Corresponding author at: Department of Computer Science, St. Francis Xavier University, Antigonish, Canada.
E-mail address: ltyang@stfx.ca (L.T. Yang).

https://doi.org/10.1016/j.ins.2018.02.013

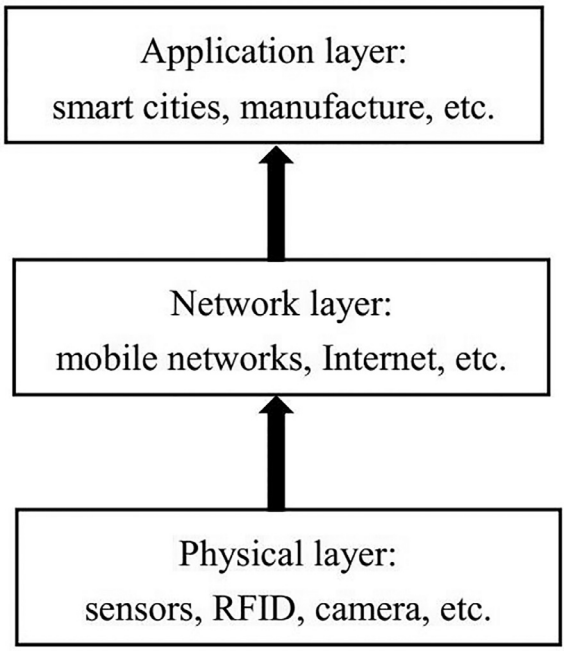

Fig. 1. Architecture of Internet of Things.

to provide predictive services and intelligent decisions [4]. Therefore, data analytic is playing an important role for Internet of Things to offer various services [10,20,21].

Clustering, as a crucial and challenging technique for data analytics, partitions objects into different groups based on some similarity metrics so that the objects in the same group share more similarity than others in different groups [25]. In the past few decades, many clustering algorithms have been developed, which can be roughly grouped by two categories, i.e., hard clustering and soft clustering [9,12]. Typical hard clustering algorithms include *k*-means and affinity propagation while representative soft clustering algorithms include the fuzzy c-means algorithm and the possibilistic c-means algorithm. In hard clustering, each object is assigned to only one cluster while each object is assigned to multiple clusters with different memberships in soft clustering. As a representative soft clustering algorithm, possibilistic c-means (PCM) has been successfully utilized in fault diagnosis, nonlinear system identification and incomplete data clustering [11]. Specially, PCM is viewed as a potential technique for big data anaytics. However, the traditional PCM algorithm cannot obtain the desirable clustering results for the datasets including some noisy objects. To tackle this problem, Schneider proposed a weighted possibilistic c-means algorithm (WPCM) to minimize the negative effect of noisy objects by assigning a small weight to each noisy object [16]. Generally, WPCM could yield significantly more accurate clustering results than PCM for the datasets including noisy objects.

Currently, big data collected from Internet of Things is posing a novel challenge on WPCM [17]. Big data is typically defined by four characteristics, i.e., large volume, large variety, large value and large velocity. Large volume is the dominated characteristic of big data, implying that there are a large number of objects in a big data set. Large variety indicates the different types of data including structured, semi-structured and unstructured data which the third characteristic refers to the hidden valuable information in big data. Moreover, big data is continually generated quickly and it requires to be processed in real time. It is difficult for WPCM to cluster big data with a large number of objects efficiently since WPCM has a high computational complexity [22]. Furthermore, WPCM needs to load all the objects into the memory for big data clustering. In some cases, the memory space of the computing devices is limited, leading to the failure of WPCM for big data clustering. Although some improved WPCM algorithms such as online WPCM and incremental WPCM have been proposed for big data clustering [9], they always produce lower clustering accuracy than the conventional WPCM algorithm. Cloud computing, as an emerging computing paradigm, is offering a scalable and cost-efficient solution for big data analytics by providing tremendous memory space and strong computing power [2]. Cloud computing has enjoyed its success in mobile crowdsourcing, scientific computing and machine learning [13,14,23]. Based on cloud computing, a distributed weighted possibilisic c-means algorithm was developed for big data clustering efficiently by uploading objects on cloud [22]. However, the private data will be disclosed when uploading the raw data to cloud directly, posing serious threat to human security. Specially, big data usually includes some private information such as personal medical records and bank counts. Once they are leaked, personal life and property will be threaten.

In this paper, a secure weighted possibilistic c-means algorithm (SWPCM) on cloud is presented for efficient big data clustering. To prevent the disclosure of the private data, BGV is utilized to encrypt the raw objects before uploading them

on cloud. BGV is one of the most efficient fully homomorphic encryption schemes and it has obtained the successful application in cloud computing and deep computation models [5]. However, BGV does not support the exponential and division operations that are included in the functions of the WPCM algorithm for calculating the weighted values and updating the membership matrix and the clustering centers. Therefore, the Taylor theorem is employed to approximate the functions as the polynomial functions which only include addition and multiplication operations such that the proposed SWPCM algorithm can yield the correct clustering result on the encrypted data based on BGV. Finally, the proposed algorithm is evaluated on two representative datasets, i.e., eGSAD and sWSN [22], by comparing with the traditional weighted possibilistic c-means algorithm in terms of effectiveness, efficiency and scalability.

So, the presented scheme includes there contributions, as listed below.

- A secure weighted possibilistic c-means algorithm based on BGV is presented for big data clustering on cloud. To protected the private data, the BGV encryption technique is employed to encrypt the data and thus the weighted possibilistic c-means algorithm is run on the encrypted data to prevent the disclosure of the raw data on cloud.
- BGV cannot support division and exponential operations on the encrypted data directly. To obtain correct clustering results on the encrypted data, the Taylor theorem is employed to approximate the functions for computing the weighted values and updating the membership matrix and the clustering centers as the polynomial functions to remove the division and exponential operations.
- Extensive experiments are conducted to evaluate the presented SWPMC algorithm by comparing the traditional WPCM algorithm in terms of efficiency, effectiveness and scalibility. The results show that the presented scheme performs more efficiently than the traditional weighted possiblistic c-means algorithm and it achieves a good scalability on cloud for big data clustering.

The paper is organized as follows. The possibilistic c-means algorithm and the related work are reviewed in Section 2 and the proposed algorithm is illustrated in Section 3. The experimental results are shown in Section 4 and the paper is concluded in the last section.

2. Related works

2.1. Possibilistic c-means clustering algorithm

The possibilistic c-means algorithm (PCM) is a typical soft clustering technique that was developed by Krishnapuram and Keller [11]. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with n objects, each object with m attributes, PCM is defined by a $c \times n$ membership matrix $U = \{u_{ij} | 1 \le i \le c; 1 \le j \le n\}$ in which u_{ij} denotes the membership of x_j towards the ith clustering center and c denotes the number of clustering centers denoted by $V = \{v_1, v_2, \ldots, v_c\}$. Therefore, PCM aims to calculate the membership matrix U and the clustering centers V by minimizing the following objective function:

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

where m and η_i denote are two constants pre-defined in the initialization step [3].

The functions for updating the membership matrix and the clustering centers can be obtained by minimizing Eq. (1):

$$u_{ij} = \frac{1}{1 + (d_{ij}^2/\eta_i)^{1/(m-1)}}, \forall i, j, \tag{2}$$

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

where d_{ij} denotes the distance between x_j and v_i. Specially, PCM is outlined in Algorithm 1.

PCM assigns the same weight to each object and thus is cannot distinguish the importance of different objects. Furthermore, PCM cannot yield the desirable clustering results for the datasets including noisy objects or outliers. To tackle this problem, Schneider presented a weighted possibilistic c-means algorithm (WPCM) by assigning a weight value to each object, which leads to an objective function of WPCM [16]:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m ||x_j - v_i||^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (w_j-u_{ij})^m, \tag{4}$$

where w_j is the weight value of x_j and it can typically be calculated via:

$$w_j = \sum_{i=1}^{c} \exp\{-\alpha ||x_j - v_i||^2\}. \tag{5}$$

Algorithm 1: The standard posssibilistic c-means algorithm.

Input: $X = \{x_1, x_2, \ldots, x_n\}$, *maxiter*
Output: $U = \{u_{ij}\}, V = \{v_i\}$
Initialize c, m, v_i, u_{ij} and η_j;
for *iteration* $= 1, 2, \ldots,$ *maxiter* **do**
  for $i = 1, 2, \ldots, c$ **do**
    $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$;
  for $i = 1, 2, \ldots, c$ **do**
    for $j = 1, 2, \ldots, n$ **do**
      $u_{ij} = \frac{1}{1+(d_{ij}^2/\eta_i)^{1/(m-1)}}$;

By minimizing the objective function of WPCM, the membership matrix can be updated via:

$$u_{ij} = \frac{w_j}{1 + (d_{ij}^2/\eta_i)^{1/(m-1)}}. \tag{6}$$

WPCM has the same function for updating the clustering centers as PCM.

Obviously, the weight values are used to distinguish the importance of different objects, so WPCM reduces the negative effect of noisy data by assigning a lower weight to each noisy object. Specially, WPCM is outlined by Algorithm 2.

Algorithm 2: The weighted posssibilistic c-means algorithm.

Input: $X = \{x_1, x_2, \ldots, x_n\}$, *maxiter*
Output: $U = \{u_{ij}\}, V = \{v_i\}$
Initialize c, m, v_i, u_{ij} and η_j;
for *iteration* $= 1, 2, \ldots,$ *maxiter* **do**
  for $i = 1, 2, \ldots, c$ **do**
    $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$;
  for $j = 1, 2, \ldots, n$ **do**
    $w_j = \sum_{i=1}^{c} \exp\{-\alpha||x_j - v_i||^2\}$;
  for $i = 1, 2, \ldots, c$ **do**
    for $j = 1, 2, \ldots, n$ **do**
      $u_{ij} = \frac{w_j}{1+(d_{ij}^2/\eta_i)^{1/(m-1)}}$;

From Algorithm 2, the computational complexity of WPCM is dominated by the step for updating the membership matrix. Specially, this step has a computational complexity of $O(cn)$, resulting in a total computational complexity of $O(lcn)$ where l denotes the number of the iterations.

2.2. Improved possibilistic c-means algorithms

Some other improved possibilistic c-means algorithms have been presented since PCM was proposed. PCM is effective and efficient to cluster small datasets in most cases. However, PCM is very sensitive to the initialization. Speically, PCM will yield a coincident result if it is not initialized appropriately. To address this problem, two variants, i.e., fuzzy possibilistic c-means algorithm and possibilistic fuzzy c-means algorithm, were developed by combining fuzzy clustering and possibilistic clustering [15]. Besides, an enhanced possibilistic c-means method was presented to avoid the coincident clustering by partitioning the objects into one major subset and one assistant subset [19].

Kernel possibilistic c-means methods (KPCM) were presented for non-spherical data clustering [8]. KPCM maps the data objects to the high-dimensional space in which the similarity of each object and the clustering centers can be measured

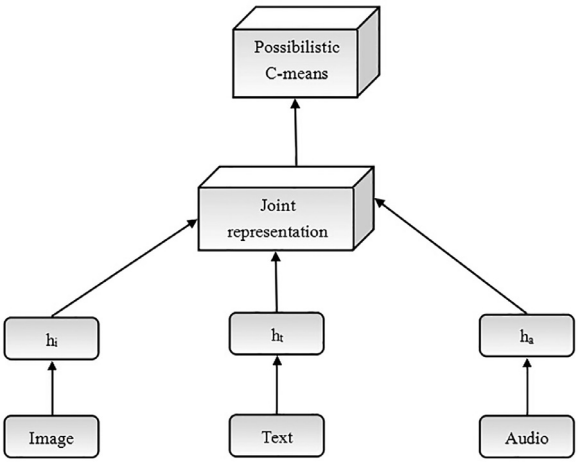

Fig. 2. Architecture of the high-oder PCM algorithm.

more effectively. Specially, given a kernel function such as the Gaussian kernel, KPCM defines the objective function as:

$$J_m(U,V;k) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \Phi_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (w_j - u_{ij})^m, \tag{7}$$

where Φ_{ij} denotes the kernel-based distance between x_j and v_i in the kernel space. KPCM is outlined in Algorithm 3.

Algorithm 3: The Kernel posssibilistic c-means algorithm.

Input: $X = \{x_1, x_2, \ldots, x_n\}$, *maxiter*
Output: $U = \{u_{ij}\}, V = \{v_i\}$
Initialize c, m, v_i, u_{ij} and η_j;
for $iteration = 1, 2, \ldots, maxiter$ **do**
  for $i = 1, 2, \ldots, c$ **do**
    $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m K(x_j, v_i) x_j}{\sum_{j=1}^{n} u_{ij}^m K(x_j, v_i)}$;
  for $i = 1, 2, \ldots, c$ **do**
    for $j = 1, 2, \ldots, n$ **do**
      $u_{ij} = \frac{1}{1+(2(1-K(x_j,v_i))/\eta_i)^{1/(m-1)}}$;

In addition, two sparse possibilistic c-means algorithms were proposed to address the closely located clusters by imposing a regularization item on the objective function of PCM [18].

More recently, a high-order possibilistic c-means algorithm was presented for heterogenous data clustering based on deep learning [24]. This algorithm first uses stacked auto-encoders to learn the features of each modality and then concatenates the learned features as a joint representation of each heterogeneous object. Finally, PCM is performed on the joint representations to obtain the clustering result. Fig. 2 presents the architecture of the high-order possibilistic c-means algorithm [24].

Furthermore, some variants have been presented to improve the clustering efficiency for the possibilistic c-means algorithm on big data. For example, an online possibilistic c-means algorithm was presented by clustering large datasets in an incremental way. This algorithm performs particularly well for streaming data and large datasets that cannot be loaded into the memory [9]. To improve the clustering efficiency of the high-order possibilistic c-means algorithm (HOPCM), two tensor decomposition schemes including the canonical polyadic decomposition and the tensor-train network were applied to HOPCM by compressing the raw data [20]. In addition, Zhang et al. [22] presented a distributed possibilistic c-means algorithm based on MapReduce to cluster large datasets, which can be performed efficiently on cloud.

Generally, cloud computing is the most efficient for the possibilistic c-means algorithm and its variants to cluster big data. However, the private data is disclosed when uploading the raw data to the cloud directly, posing serious threat to human security. Specially, security is a crucial and challenging issue to Internet of Things since there are a large number of

private objects in Internet of Things such as personal diagnosing records in smart medicine and government core data in smart cities. The disclosure of the private data may result in a terrible catastrophe. Therefore, a secure weighted possibilistic c-means algorithm based on BGV is proposed in this paper for big data clustering on cloud.

3. Secure weighted possibilistic c-means algorithm

This section illustrates the proposed secure weighted possibilistic c-means algorithm based on BGV for big data clustering on cloud. BGV is a fully homomorphic encryption scheme which can perform addition and multiplication operations on the encrypted data securely and correctly. As one of the most efficient fully homomorphic encryption schemes, BGV has successfully employed in cloud computing and deep computation models. However, BGV does not directly support division and exponential operations that are included in the functions for calculating weight values and updating the membership matrix and the clustering centers on encrypted data. Thus, directly applying BGV to the weighted possibilistic c-means algorithm on encrypted data on cloud cannot obtain the correct clustering result. Therefore, the Taylor theorem is used to approximate the functions as the polynomial functions to remove the division and exponential operations and then the overall scheme of the proposed algorithm is described in this section. As the important part in the proposed scheme, the BGV encryption scheme is presented first in this section.

3.1. BGV encryption scheme

BGV is a LWE/RLWE-based leveled full homomorphic encryption scheme introduced by Brakerski, Gentry and Vaikuntanathan [5]. The BGV scheme begins with a Setup procedure that is used to choose a μ-bit modulus q and the following parameters: the dimension $n = n(\lambda, \mu)$, the degree $d = d(\lambda, \mu)$, the distribution $\chi = \chi(\lambda, \mu)$, and $N = \lceil (2n+1)\log q \rceil$. Furthermore, two additional procedures, i.e., key Switching and modulus Switching, are designed to ensure correct result on the encrypted data. The goal of the key Switching step is decreasing the ciphertext's dimension while the task of the modulus Switching step is minimizing the effect of the noise.

Except for the encryption and decryption operations, there are two major components for secure computation on the encrypted data, i.e, the BGV addition and the BGV multiplication. Let m_1 and m_2 denote two plaintexts and their corresponding ciphertexts are c_1 and c_2, respectively. The BGV addition computes the sum c_4 of c_1 and c_2 in two steps: (1) $c_3 \leftarrow (c_1 + c_2)\%q_j$ and (2) $c_4 \leftarrow \mathrm{Re}fresh(c_3, \tau(s_j' \rightarrow s_{j-1}), q_j, q_{j-1})$. Also, the BGV multiplication computes the product c_6 of c_1 and c_2 using the following two steps: (1) $c_5 \leftarrow c_1 \otimes c_2 \% q_j$ and (2) $c_6 \leftarrow \mathrm{Re}fresh(c_5, \tau(s_j' \rightarrow s_{j-1}), q_j, q_{j-1})$. Furthermore, the sum and product of m_1 and m_2 can be obtained by decrypting c_4 and c_6, respectively. More details can be found in [5].

3.2. Function approximation

From Algorithm 1, the weighted possiblistic c-means algorithm has three key functions for calculating weight values and updating the membership matrix and the clustering centers, respectively. Specially, the first key function, Eq. (3), is used to update the clustering centers. However, Eq. (3) involves a division operation that cannot be supported by BGV. So, we approximate Eq. (3) as a polynomial function according to multivariate Taylor formula by viewing Eq. (3) as a function with regard to $(u_{i1}, u_{i2}, \ldots, u_{in})$ [22]:

$$\begin{aligned} v_i &= f(u_{i1}, u_{i2}, \ldots, u_{in}) \\ &\approx \varphi + \sum_{j=1}^{n} \delta_j (u_{ij} - \gamma_j), \end{aligned} \tag{8}$$

where $\gamma_1, \gamma_2, \ldots, \gamma_n \in (0,1)$ denote the expansion points, $\varphi = \sum_{j=1}^{n} \gamma_j^m x_j / \sum_{j=1}^{n} \gamma_j^m$ and $\delta_k = m\gamma_k^{m-1}(x_k \sum_{j=1}^{n} \gamma_j^m - \sum_{j=1}^{n} \gamma_j^m x_j)/(\sum_{j=1}^{n} \gamma_j^m)^2 (1 \le k \le n)$. Once the expansion points are fixed, φ and $\delta_k (1 \le k \le n)$ are the determined constants during the clustering process. Thus, Eq. (8) includes only addition and multiplication operations for updating the clustering centers, which can be securely computed by Algorithm 4.

Algorithm 4: Secure calculation for v_i on cloud.

Input: $C(\varphi)$, $C(\delta_j (1 \le j \le n))$ and $C(u_{ij})$
Output: $C(v_i)$

```
for j = 1, 2, ..., n do
    Calculate C(φ_j) according to the BGV multiplication: ;
    C(φ_j) = C(δ_j) × C(u_ij − γ_i) ;
for i = 1, 2, ..., c do
    Calculate C(v_i) according to the BGV addition: ;
    C(v_i) = C(φ) + Σ_{j=1}^{n} C(φ_j);
return C(v_i);
```

The second key function, Eq. (5), aims to calculate the weight value w_j for x_j, which includes an exponential operation. Therefore, we approximate Eq. (5) as a polynomial function to remove the exponential operation. It is apparent from Eq. (5) that w_j is an exponential function with regard to d_{ij}^2. Let $q_i = d_{ij}^2$ with $q_i \in [0, +\infty)$, the function for computing w_j can be rewritten as:

$$w_j = \sum_{i=1}^{c} \exp\{-\alpha q_i\}. \tag{9}$$

Depending on the Maclaurin formula, Eq. (9) can be approximated as:

$$\begin{aligned} w_j &= \sum_{l=0}^{+\infty} (-\alpha)^l \frac{1}{l!} {p_i}^l \\ &\approx 1 - \alpha p_i + 0.5\alpha^2 {p_i}^2, \end{aligned} \tag{10}$$

where α is an appropriate constant.

Eq. (10) shows that the approximated function for computing w_j includes only addition and multiplication operations. Therefore, $w_j (1 \le j \le n)$ can also be securely computed by BGV according to Algorithm 5.

Algorithm 5: Secure calculation for $C(w_j)$ on Cloud.

Input: $C(\alpha), C(d_{ij}^2)$
Output: $C(w_j)$
1 Calculate C_1 according to the BGV multiplication: ;
2 $C_1 = C(\alpha) \times C(d_{ij}^2)$;
3 Calculate C_2 according to the BGV addition: ;
4 $C_2 = 0.5 \times C_1 \times C_1$;
5 Calculate C_2 according to the BGV addition: $C(w_j) = 1 - C_1 + C_2$;
6 **return** $C(w_j)$;

The task of the last key function, Eq. (6), is to update the membership matrix u_{ij}, which also includes a division operation. So, this paper approximates it as a polynomial function based on the Taylor theorem to remove the division operation. From Eq. (6), u_{ij} can be rewritten as a function of $p = d_{ij}$ [22]:

$$u_{ij} = w_j f(p) = \frac{w_j}{1 + (p/\eta_i)^b}. \tag{11}$$

Let $r = 1/(1 + (a/\eta_i)^b)$, $s = b\eta_i^{2b} a^{b-1}/(\eta_i^b + a^b)^2$ and $t = (b(b-1)\eta_i^{3b} a^{b-2} - b(b+1)\eta_i^{2b} a^{3b-2} - 2b\eta_i^{3b} a^{2b-2})/2(\eta_i^b + a^b)^4$, u_{ij} can be approximated by:

$$\begin{aligned} u_{ij} &\approx w_j f(p) \\ &\approx w_j (r + s(p-a) + t(p-a)^2). \end{aligned} \tag{12}$$

Apparently, Eq. (12) includes only addition and multiplication operations so u_{ij} can be securely calculated by Algorithm 6.

Algorithm 6: Secure calculation for $C(u_{ij})$ on Cloud.

Input: $C(\alpha), C(d_{ij}^2)$, $C(r)$ and $C(w_j)$
Output: $C(u_{ij})$
1 Calculate C_1 according to the BGV multiplication: ;
2 $C_1 = C(s) \times C(d_{ij}^2 - \alpha)$;
3 Calculate C_2 according to the BGV multiplication: ;
4 $C_2 = C(t) \times C(d_{ij}^2 - \alpha) \times C(d_{ij}^2 - \alpha)$;
5 Calculate C_3 according to the BGV addition: $C_3 = C(r) + C_1 + C_2$;
6 Calculate u_{ij} according to the BGV multiplication: $C(u_{ij}) = C(w_j) C_3$;
7 **return** $C(u_{ij})$;

3.3. Secure weighted possibilistic c-means algorithm on cloud

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, the goal of the proposed scheme is to yield a correct weighted possiblistic c-means clustering result including a membership matrix $U = \{u_{ij}\}$ and a center set $V = \{v_1, v_2, \ldots, v_c\}$ on cloud for X without the

disclosure of the private data efficiently. To achieve this goal, the overall scheme includes two major components: (1) the client encrypts and decrypts the raw data and (2) the cloud performs the secure weighted possibilistic c-means algorithm on the encrypted data to calculate the membership matrix and the clustering centers. Specially, the secure weighed possibilistic c-means algorithm based on the BGV encryption scheme on cloud is presented in Algorithm 7.

Algorithm 7: Secure weighted posssibilistic c-means scheme.

Input: $X = \{x_1, x_2, \ldots, x_n\}$
Output: U and V
1 On client: ;
2 Parameters initialization;
3 Use BGV to encrypt all the parameters and the data objects;
4 Upload encrypted parameters and objects on cloud;
5 **for** $iteration = 1, 2, \ldots, maxiter$ **do**
6 Use BGV to encrypt the membership matrix and the clustering centers;
7 Upload encrypted membership matrix and centers on cloud;
8 On cloud: ;
9 **for** $i = 1, 2, \ldots, c$ **do**
10 Use Secure Algorithm 2 to compute $C(v_i)$;
11 **for** $j = 1, 2, \ldots, n$ **do**
12 Use Secure Algorithm 3 to compute $C(w_j)$;
13 **for** $i = 1, 2, \ldots, c$ **do**
14 **for** $j = 1, 2, \ldots, n$ **do**
15 Use Secure Algorithm 4 to compute $C(u_{ij})$;
16 Send the encrypted membership matrix and centers to client;
17 On client: ;
18 Decrypt the immediate results to update U and V;

In the weighted possibilistic c-means algorithm, the major parameters include m, c, η_i and α. Therefore, before performing the secure weighted possibilistic c-means algorithm, the cloud initializes the parameters, as well as U and V. Afterwards, the client runs the BGV encryption operation to encrypt all the parameters and the data objects on line 3 in Algorithm 7 and then uploads the encrypted data on cloud. During each iteration, the client encrypts U and V on line 6 in Algorithm 7 while the cloud runs secure Algorithm 4, secure Algorithm 5 and secure Algorithm 6 to compute the weight values, the membership matrix and the clustering centers, respectively, on lines 9–15 in Algorithm 5. Afterwards, the cloud sends the encrypted membership matrix and the encrypted clustering centers that denote the ciphertexts of the immediate results to the client. The client updates the membership matrix and the clustering centers by decrypting the immediate results.

From Algorithm 5, the client only needs to perform encryption and decryption and all clustering operations are performed on cloud, so the proposed scheme is highly efficient. More importantly, since the cloud performs all clustering operations on the encrypted data, the raw data will not be disclosed, ensuring the security for the private data.

3.4. Complexity analysis

We analyze the complexity of the presented secure weighted possibilistic c-means algorithm in terms of computation cost and communication cost in this paper. Specially, we use ADD, MUL and MOD to denote the computation cost of one addition operation, one multiplication operation and one modulus operation on Ring R, respectively.

Computation Cost. In the presented algorithm, the client encrypts the raw objects only once before uploading the encrypted data on cloud. During each iteration, the client encrypts the membership matrix and the clustering centers once while the cloud performs Algorithms 4–6 once. For each object with m attributes in the dataset $X = \{x_1, x_2, \ldots, x_k\}$ with c clustering centers, the client encrypts the dataset with $km(n+1)N$ (ADD+MUL) using the BGV encryption operation with the parameter set $\{\mu, q, d, n, N, \chi\}$. Furthermore, the client encrypts the membership matrix and the clustering centers with $kc(n+1)N$ (ADD+MUL) and $cm(n+1)N$ (ADD+MUL), respectively, and decrypts the immediate results with $c(m+k)(n+1)$ MUL, $c(m+k)n$ ADD and $2c(m+k)$ MOD in each iteration. Moreover, the cloud performs $(n+1)(4k(c+1)(2N+n)+8N+1)$ MUL, $(n+2)(2k(c+1)+4k+1)N$ ADD and $(n+1)(((6k(c+1)+c)(n+2))+4n)$ MOD for calculating the weight values and updating the membership matrix and the clustering centers on the encrypted data during each iteration.

Communication cost. At the beginning, the client uploads km messages with $km(n+1)\mu$ bits to cloud. During each iteration, the client exchanges $2c(k+m)$ messages with $2c(k+m)(n+1)\mu$ bits with the cloud.

Table 1
Result in terms of C^* on eGSAD.

Algorithm/dataset	1	2	3	4	5	6	7	8	Average	Whole
WPCM	14.06	12.66	13.83	10.98	11.79	12.96	11.81	13.82	12.68	12.16
SWPCM	14.06	14.79	13.71	10.98	12.01	12.96	11.81	13.91	13.02	12.16

Table 2
Result in terms of C^* on sWSN.

Algorithm/dataset	1	2	3	4	5	6	7	8	Average	Whole
WPCM	0.53	0.39	0.47	0.66	0.51	0.48	0.71	0.64	0.55	0.48
SWPCM	0.56	0.39	0.47	0.71	0.52	0.52	0.69	0.64	0.56	0.51

Table 3
Result in terms of $ARI(U,U^*)$ on eGSAD.

Algorithm/dataset	1	2	3	4	5	6	7	8	Average	Whole
WPCM	0.90	0.87	0.91	0.93	0.88	0.92	0.89	0.84	0.89	0.92
SWPCM	0.89	0.84	0.91	0.91	0.84	0.92	0.87	0.85	0.88	0.91

Table 4
Result in terms of $ARI(U,U^*)$ on sWSN.

Algorithm/dataset	1	2	3	4	5	6	7	8	Average	Whole
WPCM	0.88	0.91	0.82	0.88	0.92	0.89	0.84	0.91	0.88	0.89
SWPCM	0.87	0.88	0.83	0.83	0.90	0.84	0.86	0.91	0.86	0.87

4. Experiments

In the experiments, the performance of the presented secure weighted possibilistic c-means scheme (SWPCM) based on the BGV encryption is evaluated in a cloud platform which includes 20 personal computers, each with 3.2 GHz Core i7 CPU and 4 GB memory. Specially, the SWPCM scheme is compared with the traditional weighted possibilistic c-means method (WPCM) from three aspects, i.e., efficiency, effectiveness and scalability. In this paper, C^* and $ARI(U,U^*)$ are used to compare the clustering accuracy between SWPCM and WPCM. C^* denotes the error between the actual clustering centers and the clustering ones yielded by the specific clustering method, which can be computed via [24]:

$$C* = \sqrt{\sum_{i=1}^{c} ||v_i^* - v_i||^2}, \tag{13}$$

where v_i^* and v_i denote the ith clustering center yielded by the clustering method and the actual clustering center, respectively.

$ARI(U,U^*)$ is a most widely used metric to evaluate the performance of soft clustering techniques. U denotes the actual membership matrix while U^* denotes the produced membership matrix.

A smaller C^* and a bigger $ARI(U,U^*)$ imply that the clustering method produces a more accuracy result.

Furthermore, two representative big datasets, namely eGSAD and sWSN [24], are employed to estimate the effectiveness and efficiency of two clustering algorithms.

4.1. Results on clustering effectiveness

To estimate the robustness of SWPCM and WPCM, each dataset is averaged to 8 subsets and each algorithm is performed for clustering each subset and the whole dataset. The clustering results in terms of C^* are listed in Tables 1 and 2.

We can make two important observations from the results shown in Tables 1 and 2. First, in most cases, SWPCM yields the slightly bigger C^* values than WPCM since the approximation of the functions for calculating the weight values and updating the membership matrix and the clustering centers results in the drop of the clustering accuracy. For example, SWPCM yields the C^* value of 0.51 on the whole sWSN dataset while WPCM obtains the C^* value of 0.48. Second, in some cases, SWPCM obtains the same C^* values as WPCM. For instance, the C^* value produced by both SWPCM and WPCM on the whole eGSAD dataset is the same, namely 12.16. Such observations imply that SWPCM yields the considerably accurate clustering centers with WPCM.

Tables 3 and 4 show the clustering results in terms of $ARI(U,U^*)$.

According to the results presented in Tables 3 and 4, SWPCM produces smaller $ARI(U,U^*)$ than WPCM in most cases. For example, SWPCM and WPCM yield the $ARI(U,U^*)$ values of 0.92 and 0.91, respectively, on the whole eGSAD dataset. However,

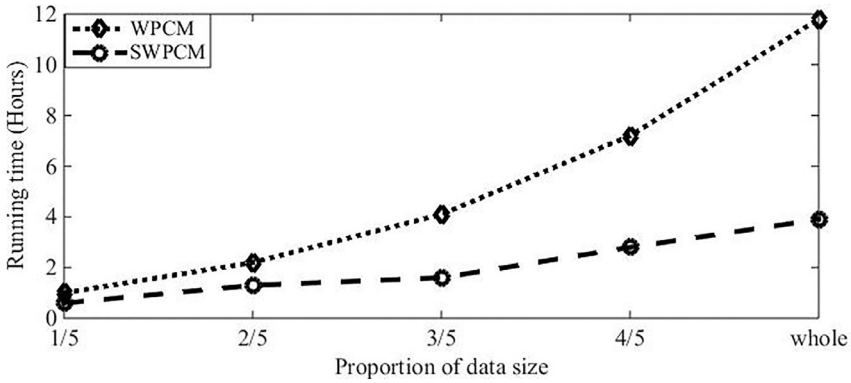

Fig. 3. Running time on eGSAD.

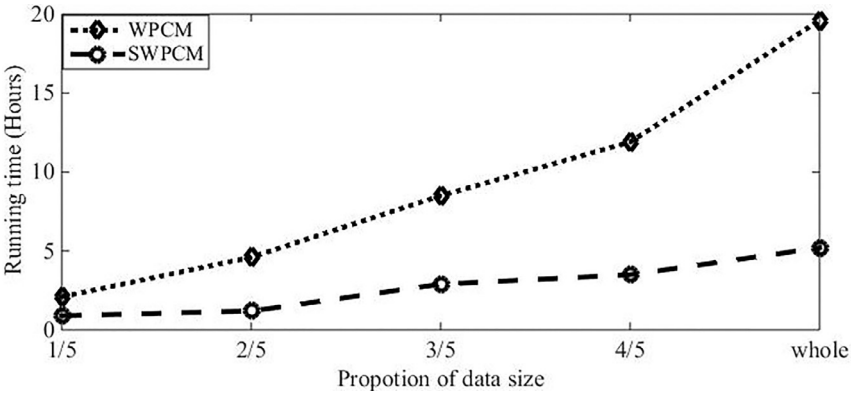

Fig. 4. Running time on sWSN.

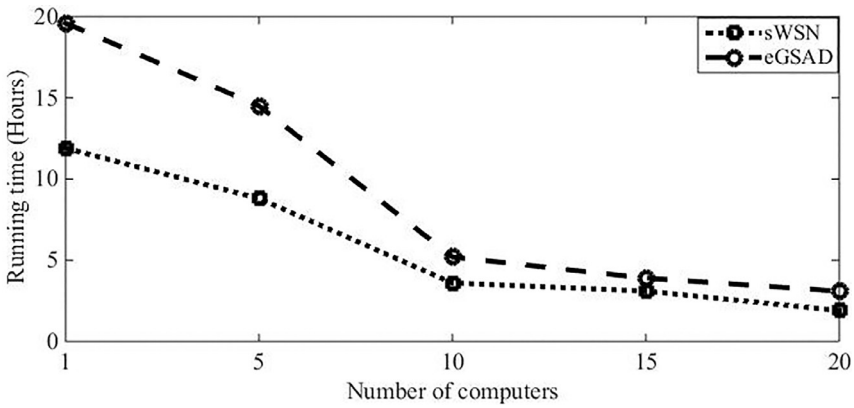

Fig. 5. Result on speedup.

the difference of *ARI(U,U*)* produced by SWPCM and WPCM is very low, demonstrating a low drop of clustering accuracy by SWPCM.

4.2. Results on clustering efficiency

To estimate the efficiency of two algorithms, SWPCM is run on cloud with 10 personal computers and WPCM is run on a single computer. The running time is shown in Figs. 3 and 4.

From Figs. 3 and 4, with the increasing of the data size, the running time of WPCM and SWPCM grows. For example, when the proportion of eGSAD increases from 1/5 to 4/5, the running time of WPCM and SWPCM grows from 1 to 7.2 h and from 0.6 to 2.8 h, respectively. Such observation demonstrates that the data size has an important influence on the running time for the clustering algorithms. However, SWPCM took significantly shorter than WPCM to cluster two datasets.

For instance, the running time of SWPCM is 5.2 h while that of WPCM is 19.6 h on sWSN. This result indicates that SWPCM achieves a great improvement for clustering efficiency.

4.3. Results on scalability

Speedup is employed to estimate the scalability of the presented SWPCM scheme by running SWPCM in the cloud platform with different number of computers in this section. The result is presented in Fig. 5.

Fig. 5 shows that the running time of SWPCM decreases with the growth of the number of the computers on cloud for clustering two datasets. For example, when the number of the computers grows from 5 to 15, the running time decreases from 8.8 to 3.1 h on the eGSAD dataset. Such results demonstrate that SWPCM achieves a good scalability.

5. Conclusion

In this paper, a secure weighted possibilistic c-means algorithm based on the BGV encryption scheme is presented for big data clustering on cloud. One property of the presented scheme is the combination of the fully homomorphic encryption scheme and the cloud computing to improve the clustering efficiency for big data without the disclosure of the private data. The key idea is to approximate the functions for calculating the weight values and updating the membership matrix and the clustering centers as three polynomial functions to remove the division and exponential operations such that the presented scheme can obtain the correct clustering result on the encrypted data. Experimental results clearly demonstrate three points. First, the presented algorithm produces a considerable clustering accuracy with the traditional weighted possibilistic c-means algorithm in terms of C^* and $ARI(U,U^*)$ on the experimental datasets. Second, the presented algorithm performs significantly more efficiently than the traditional weighted possibilistic c-means algorithm. Last but not least, the presented algorithm achieves a good scalability, implying that the performance of the presented algorithm could be further improved by adding more computers on cloud. In the future work, other encryption schemes such as somewhat homomorphic encryption schemes and garbled circuit will be investigated to implement the secure weighted possibilistic c-means algorithm, which is expected to further improve the clustering efficiency without the disclosure the private data on cloud.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutorials 17 (4) (2015) 2347–2376.
[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.
[3] M. Barni, V. Cappellini, A. Mecocci, Comments on a possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 4 (3) (1996) 393–396.
[4] M.Z.A. Bhuiyan, J. Wu, G. Wang, M. Hassan, E-sampling: event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems, ACM Trans. Auton. Adapt. Syst. 12 (1) (2017) 1–29.
[5] Z. Brakerski, C. Gentry, V. Vaikuntanathan, (Leveled) fully homomorphic encryption without bootstrapping, in: Proceedings of ACM Innovations in Theoretical Computer Science Conference, ACM, 2012, pp. 309–325.
[6] X. Deng, Z. Tang, L.T. Yang, M. Lin, B. Wang, Confident information coverage hole healing in hybrid industrial wireless sensor networks, IEEE Trans. Ind. Inf. (2017), doi:10.1109/TII.2017.2764038.
[7] X. Deng, Z. Tang, L. Yi, L.T. Yang, Healing multi-modal confident information coverage holes in NB-iot-enabled networks, IEEE IoT J. (2017), doi:10.1109/JIOT.2017.2783258.
[8] M. Filippone, F. Masulli, S. Rovette, Applying the possibilistic c-means algorithm in kernel-induced spaces, IEEE Trans. Fuzzy Syst. 18 (3) (2010) 572–584.
[9] T. Havens, J. Bezdek, C. Leckie, L. Hall, M. Palaniswami, Fuzzy c-means algorithms for very large data, IEEE Trans. Fuzzy Syst. 20 (6) (2012) 1130–1146.
[10] H. Hui, M. Lin, Q. Zhang, Hybrid DVFS scheduling for real-time systems based on reinforcement learning, in: Proceedings of IEEE International Conference on Green Computing and Communications, 2017. http://cse.stfx.ca/GreenCom2017/acceptedlist.htm.
[11] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (2) (1993) 98–110.
[12] P. Li, Z. Chen, L.T. Yang, L. Zhao, Q. Zhang, A privacy-preserving high-order neuro-fuzzy c-means algorithm with cloud computing, Neurocomputing 256 (2017) 82–89.
[13] H. Li, K. Ota, M. Dong, M. Guo, Mobile crowdsensing in software defined opportunistic networks, IEEE Commun. Mag. 55 (6) (2017) 140–145.
[14] H. Li, K. Ota, M. Dong, A. Vasilakos, K. Nagano, Multimedia processing pricing strategy in GPU-accelerated cloud computing, IEEE Trans. Cloud Comput. (2017), doi:10.1109/TCC.2017.2672554.
[15] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, IEEE Trans. Fuzzy Syst. 13 (4) (2005) 517–530.
[16] A. Schneider, Weighted possibilistic c-means clustering algorithms, in: Proceedings of IEEE International Conference on Fuzzy Systems, IEEE, 2000, pp. 176–180.
[17] X. Wu, X. Zhu, G.Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data. Eng. 26 (1) (2014) 97–107.
[18] S.D. Xenaki, K.D. Koutroumbas, A.A. Rontogiannis, Sparsity-aware possibilistic clustering algorithms, IEEE Trans. Fuzzy Syst. 24 (6) (2016) 1611–1626.
[19] Z. Xie, S. Wang, F.L. Chung, An enhanced possibilistic c-means clustering algorithm EPCM, Soft Comput. 12 (6) (2008) 593–611.
[20] Q. Zhang, L.T. Yang, Z. Chen, P. Li, A survey on deep learning for big data, Inf. Fusion 42 (2018) 146–157.
[21] Q. Zhang, L.T. Yang, Z. Chen, P. Li, An improved deep computation model based on canonical polyadic decomposition, IEEE Trans. Syst. Man Cybern. (2017), doi:10.1109/TSMC.2017.2701797.
[22] Q. Zhang, L.T. Yang, Z. Chen, P. Li, PPHOPCM: Privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing, IEEE Trans. Big Data (2017), doi:10.1109/TBDATA.2017.2701816.
[23] Q. Zhang, L.T. Yang, Z. Chen, P. Li, M.J. Deen, Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning, IEEE IoT J. (2017), doi:10.1109/JIOT.2017.2732735.
[24] Q. Zhang, L.T. Yang, Z. Chen, F. Xia, A high-order possibilistic-means algorithm for clustering incomplete multimedia data, IEEE Syst. J. 11 (4) (2017) 2160–2169.
[25] L. Zhao, Z. Chen, Y. Yang, Z. Wang, V. Leung, Incomplete multi-view clustering via deep semantic mapping, Neurocomputing (2017). https://doi.org/10.1016/j.neucom.2017.07.016
[26] Z. Zhao, W. Dong, J. Bu, Y. Gu, C. Chen, Link-correlation-aware data dissemination in wireless sensor networks, IEEE Trans. Ind. Electron. 62 (9) (2015) 5745–5757.