Information Fusion 39 (2018) 72–80

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT

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

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

ARTICLE INFO

Article history:
Received 17 September 2016
Revised 30 March 2017
Accepted 2 April 2017
Available online 4 April 2017

Keywords:
Big data
IoT
Possibilistic c-means clustering
Canonical polyadic decomposition
Tensor-train network

ABSTRACT

Internet of Things (IoT) connects the physical world and the cyber world to offer intelligent services by data mining for big data. Each big data sample typically involves a large number of attributes, posing a remarkable challenge on the high-order possibilistic c-means algorithm (HOPCM). Specially, HOPCM requires high-performance servers with a large-scale memory and a powerful computing unit, to cluster big samples, limiting its applicability in IoT systems with low-end devices such as portable computing units and embedded devises which have only limited memory space and computing power. In this paper, we propose two high-order possibilistic c-means algorithms based on the canonical polyadic decomposition (CP-HOPCM) and the tensor-train network (TT-HOPCM) for clustering big data. In detail, we use the canonical polyadic decomposition and the tensor-train network to compress the attributes of each big data sample. To evaluate the performance of our algorithms, we conduct the experiments on two representative big data datasets, i.e., NUS-WIDE-14 and SNAE2, by comparison with the conventional high-order possibilistic c-means algorithm in terms of attributes reduction, execution time, memory usage and clustering accuracy. Results imply that CP-HOPCM and TT-HOPCM are potential for big data clustering in IoT systems with low-end devices since they can achieve a high compression rate for heterogeneous samples to save the memory space significantly without a significant clustering accuracy drop.

1. Introduction

Internet of Things (IoT) aims to improve our lives by connecting the physical objects to the cyber space. IoT has been successfully used in many domains, such as the industrial automation, smart home and intelligent transportation. A typical IoT system is constituted by five layers, i.e., object layer, object abstraction layer, service management layer, application layer and business layer [1,2].

Currently, big data sets are generated and collected from many areas such as social networks, scientific computing and IoT. For an IoT system, the data sets are collected from the object layer and transferred to the service management layer via the object abstraction layer. The service management layer offers decision support and prediction services to upper layers by analyzing the received data. The top two layers provide interfaces to the customers, making it possible to offer intelligent services based on data mining for big data.

From the architecture of Internet of Things, data mining for big data is crucial for the IoT to offer the intelligent services in various applications. As one typical technique of data mining, clustering usually uses the unsupervised/semisupervised strategy to divide the collected samples into several groups such that the similar samples are divided in the same group [3,4]. Over past decades, a lot of clustering techniques have been developed, which can be classified into two categories, i.e., hard clustering and soft clustering [5]. In the hard clustering, each sample can only be partitioned into one group while the assignment of each sample is a distribution over all the groups in the soft clustering. One of the most well-known soft clustering technique is the possibilistic c-means algorithm (PCM) that is defined as a set of clustering centers and a membership matrix [6]. Specially, each element in the membership matrix denotes the degree of one sample belonging to a specific group. PCM can reveal the data distribution more effectively than other soft clustering techniques(e.g., fuzzy c-means clustering) and the hard clustering algorithms(e.g., k-means and affinity propagation). Furthermore, PCM has some well understood properties and is well implemented due to its simplicity and efficiency

* Corresponding author.
E-mail address: ltyang@gmail.com (L.T. Yang).

http://dx.doi.org/10.1016/j.inffus.2017.04.002
1566-2535/

[7]. Therefore, PCM has been widely used in image segmentation, fault detection and data mining [8–10].

With the development of the communication and sensor technologies, an increasing number of heterogeneous data are collected in the big data sets [11]. To cluster the heterogeneous data, Zhang et al. [7] presented a high-order possibilistic c-means algorithm by extending the conventional possibilistic c-means algorithm (HOPCM) to the tensor space. HOPCM achieves super performance for heterogeneous data clustering. However, the heterogeneous samples in big data sets usually involve a large number of attributes, called large-scale samples, posing a remarkable challenge on the high-order possibilistic c-means algorithm. Specially, HOPCM needs to be performed on the high-performance servers with big memory space and powerful computing units since it requires to load all the large-scale samples into memory to achieve a desired clustering result. However, the large-scale samples with a large number of attributes typically occupy 80%, even more, of memory space. Therefore, it is almost impossible to use the high-order possibilistic c-means algorithm to cluster large-scale heterogeneous samples for IoT systems with on low-end devices such as portable computing units and embedded devices because of their limited memory space and computing power.

Some strategies can be applied to the high-order possibilistic c-means algorithm for clustering large-scale samples, such as the sampling strategy and the online strategy [12]. The sampling strategy chooses a small set of objects from the original big data set randomly and then calculates the clustering centers from the chosen set [13]. The online strategy clusters small sets, each of which can be loaded into the memory, sequentially and then combines the patterns of each set for the final clustering result [14]. The high-order possibilistic c-means algorithms based on these strategies often result in a large clustering accuracy drop. The high-order possibilistic c-means algorithms based on cloud computing and other distributed computing strategies have been developed for big data clustering without a large accuracy drop [15]. However, these distributed schemes need to be performed in some large-scale data computing centers, limiting the use of the high-order possibilistic c-means algorithm for clustering large-scale samples in IoT systems.

In this paper, we propose two improved high-order possibilistic c-means algorithms based on tensor decompositions, namely canonical polyadic decomposition (CPD) and tensor-train network (TT). The goal of the proposed algorithms is to make the high-order possibilistic c-means clustering potential to work for IoT systems with low-end devices which typically have limited memory space. To achieve the goal, we used tensor decomposition schemes to compress the heterogeneous samples so that the clustering algorithms requires less memory space to cluster big data heterogeneous dataset. Specially, we design a CP high-order possibilistic c-means algorithm (CP-HOPCM) and a TT high-order possibilistic c-means algorithm (TT-HOPCM) by using the canonical polyadic decomposition and tensor-train network to compress each sample, respectively. There are two reasons for applying the canonical polyadic decomposition and tensor-train network in this paper [16,17]. First, they achieve the highest compression rate for big tensors compared to other tensor decomposition schemes such as the Tucker decomposition and the CUR decomposition. Given an N-order data tensor, it will grow exponentially with the number of the order N in the original format. However, the number of the parameters in the CPD format and in the TT format will increase linearly with regard to N. Second, they have some well understood properties and simple practical implementation by current decomposition algorithms such as alternating least squares and alternative low-rank matrix approximations. Finally, we derive the functions for updating the membership matrix and the clustering centers for the CP high-order possibilistic c-means algorithm (CP-HOPCM) and the TT high-order possibilistic c-means algorithm (TT-HOPCM), respectively. In the experiments, we evaluate the performance of our algorithms on two representative heterogeneous datasets, i.e., NUS-WIDE-14 and SNAE2, by comparison with the conventional high-order possibilistic c-means algorithm in terms of attributes compression, execution time, memory usage and clustering accuracy. Results demonstrate that our schemes achieve a high compression rate for the attributes without a significant clustering accuracy drop. Moreover, CP-HOPCM yields a higher compression rate than TT-HOPCM while TT-HOPCM produces a higher clustering accuracy than CP-HOPCM. From such results, we can adopt the CP high-order possibilistic c-means algorithm for clustering big data when performing the clustering task in IoT systems with low-end devices with limited memory space and computing power. Otherwise, we can use the TT-HOPCM when a higher accurate clustering result is required.

Therefore, there are three major contributions in the paper:

- We present a CP high-order possibilistic c-means algorithm (CP-HOPCM) to compress the attributes by applying the canonical polyadic decomposition to the high-order possibilistic c-means scheme. CP-HOPCM can be used in IoT systems with low-end devices which have limited memory space and computing power since it can achieve a significant compression rate for samples with a small accuracy drop.
- We present a TT high-order possibilistic c-means algorithm (TT-HOPCM) by using the tensor-train network to compress the attributes in the high-order possibilistic c-means algorithm. We can use the TT-HOPCM to cluster big data with a large number of attributes when a higher accurate clustering result is required since it can yield almost the same accurate clustering result as the conventional high-order possibilistic c-means algorithm with a very high compression rate for heterogeneous data.
- We conduct the experiments on two representative heterogeneous datasets, i.e., NUS-WIDE-14 and SNAE2, to compare CP-HOPCM and TT-HOPCM in terms of attributes reduction, execution time, memory usage and clustering accuracy. Furthermore, we discuss the application conditions of the two proposed high-order possibilistic c-means algorithms based on their properties drawn from the experimental results.

In the rest of this paper, we present the preliminaries, including the conventional possibilistic c-means algorithm, the high-order possibilistic c-means algorithm, the canonical polyadic composition and the tensor-train network, in Section 2. We describe the CP high-order possibilistic c-means algorithm and the TT high-order possibilistic c-means algorithm in Section 3 and Section 4, representatively. Section 5 reports the experimental results and Section 6 reviews the related work. Finally we conclude the paper in Section 7.

2. Preliminaries

In this section, we describe the conventional possibilistic c-means algorithm (PCM), the high-order possibilistic c-means algorithm (HOPCM), the canonical polyadic composition (CPD) and the tensor-train network (TT), which are the preliminaries required in the proposed methods. PCM and HOPCM are described first, followed by CPD and TT.

2.1. Possibilistic c-means algorithm (PCM)

The possibilistic c-means algorithm was developed by Krishnapuram et al. [6] by defining a $c \times n$ matrix $U = \{u_{ij}\}$, where c and n denote the number of the clustering centers and the samples in