Journal of Computers Vol. 32 No. 1, 2021, pp. 83-94
doi:10.3966/199115992021023201007

A Fast Clustering Method for Real-Time IoT Data Streams

Jing Sun*, Xin Yao

Department of Information Engineering, Zhongshan Polytechnic College, Boai 7th Road 25, Zhongshan, China
sunjing_zspt@sina.com, gotoyaoxin@163.com

Received 12 November 2019; Revised 6 April 2020; Accepted 29 May 2020

Abstract. As an effective way of data analysis, clustering is widely applied in the IoT based applications. By studying the related existing proposals of data clustering, a new clustering method for IoT Data streams is proposed in the present work. Firstly, the characteristics of PML documents in the process of data acquisition and identification are introduced and a hybrid PML document similarity calculation method based on the Bayesian network is developed and expected to assist in data streams clustering. Secondly, a PML data streams clustering method based on a dynamic sliding window is proposed. Finally, we evaluate the performance of our clustering method and the related methods with respect to Running time, Similarity, Purity, Entropy, and F-measure. Experimental results exhibit that the innovative clustering approach can adaptively learn from data streams that change over time, while still maintains comparable accuracy and speed.

Keywords: PML, Bayesian network model, data streams clustering, dynamic sliding window

1 Introduction

Progress in IoT technology is helping people improve lives and works around the world. Nowadays, new businesses and serves are created continuously. More and more cities use IoT devices such as connected sensors, lights, and meters to analyze environments and develop infrastructure, public utilities, and more. Various types of smart devices are connected to the Internet, share and transfer information. Thus, all kinds of data are generated, and the IoT network provides a large platform for information interaction [1].

However, while industrial IoT is improving the effectiveness of modern production and applications, it also brings many new challenges regarding smart control and large data management [2]. IoT is considered as part of the emerging technology ecosystem with big data analytics and cloud computing. Interactions occur among and between people, objects, and events in environments that can leverage new services supported by a strong set of analysis tools [3]. In this sense, sophisticated data analysis techniques are needed to enable applications to aggregate and process a significant volume of data streams generated by homing devices, public spaces, etc.

As an effective way of data analysis, clustering is widely applied in the area of education, business, transportation, and other IoT based applications. Deep researches on data clustering have been carried out these years [4-9]. In general, the target of clustering is to group the collected data into different clusters according to their similarity so that data objects within a cluster have high similarity, but are distinctly different from data objects in other clusters.

1.1 Related Works

At present, clustering has become an active research area and various proposals exist in the literature.

Hierarchical Clustering algorithm [10] provides a fast way of clustering as high intra-class similarity and low inter-class dissimilarity. In this algorithm, selecting different dimensional space and frequency levels leads to different similarity calculation accuracy in the clustering results. Therefore, one important

* Corresponding Author

improvement would be to employ different similarity calculation strategies for similarity calculation between documents.

Clustream [11] is one of the most widely used stream clustering algorithms. It normally uses a two-phase scheme which consists of an online micro-clustering phase and an offline macro-clustering phase. The online micro-cluster processes data streams in real-time and produces summary statistics; the offline macro-clustering part uses the summary statistics to generate new clusters. Because the algorithm requires expert-level parametrization, the real cluster distribution can be affected critically and it suffers from the common problems of noise, high-dimensional streams which can lead to poor clustering quality [12].

The k-means [13] method is an algorithm that finds K clusters of given datasets. It first randomly assigns a cluster to each observation, thus computing the initial mean to be the centroid of the cluster's randomly assigned points. However, K-means are not suitable for all data types in the IoT network because it cannot generate non-spherical clusters and clusters of different sizes and densities.

DenStream [14] is a density-based data streams clustering algorithm. It distinguishes the outliers and the potential clusters and greatly improves the clustering results of the evolution of the online underlying data stream. However, there is no limit to the number of micro clusters in this method and no corresponding method to delete or reduce the redundant clusters, which will lead to a huge memory overhead.

XCLS [15] groups the XML documents according to the structural similarity. In this algorithm, a new level structure is built to represent the XML documents. It defines a document clustering calculation method based on level similarity and does not need to take much time to compute the similarity between XML documents. The experimental analysis showed that the method to be fast and accurate.

Besides, in order to dominate the cluster process, these cluster methods need some prior parameters such as initial cluster number and radius of cluster.

1.2 The Proposed IoT Data Clustering

Considering the abovementioned advantages and limitations of the previous works, we propose an alternative data clustering method to improve the clustering performance on evolutionary underlying data streams and meet the high computational and space efficiency requirements implicated for IoT applications.

Our work differs from the previous works in that it uses the PML documents to describe the sensor-based data streams. Also, it applies the Bayesian network to define a hybrid similarity calculation method of PML documents, and adopt a hybrid similarity calculation method that considers both the element comparison and edit distance results between PML documents. Moreover, a PML document clustering based on a dynamic sliding window is presented in this work. By analyzing the speed of the data streams, the algorithm can adjust the sliding window size dynamically.

To summarize, the contribution of this work is as follows:

1. The hybrid similarity calculation algorithm based on the Bayesian network produces high similarity results to assist in the clustering task.
2. Using a dynamic sliding observation window, the proposed clustering method can adaptively deal with an infinite and constantly changing data stream and achieve high clustering quality.
3. The proposed clustering method provides computational advantages for distributions that can be represented with a PML document structure.

The rest of this paper is organized as follows. Section 2 demonstrates the characteristics of PML documents in the process of data acquisition and identification. In Section 3, the basic principle of the Bayesian network is described. In Section 4, a hybrid similarity calculation model of PML documents based on the Bayesian network is designed. In Section 5, the clustering method of PML data streams based on a dynamic sliding window is presented. Section 6 discusses the experimental results. Finally, Section 7 is a conclusion of this work.

2 PML Language

Physical Markup Language (PML) is an XML based markup language, which is used to describe physical environments and its objects within them. Each device contains a component that interprets the

PML document associated with the function of the device. In this work, PML is proposed as a general standard method to describe the physical objects and environments of industrial, commercial and consumer applications.

2.1 The Role of PML Language

The main benefit of a PML based IoT system is to automatically track the flow of objects, which is very helpful for asset management in enterprises. It can be seen that the most important role of PML is as a general interface of different components of the IoT system, i.e. SAVANT, third-party applications (such as ERP, MES), and PML servers processing related Auto-ID data, as shown in Fig. 1.

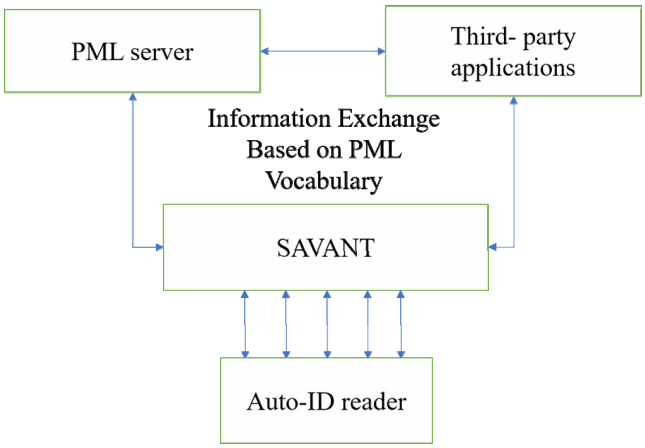

Fig. 1. The role of PML

2.2 PML Document Instance

A PML document represents the whole process that an auto-ID reader acquiring data into hierarchically structured information. For example, Fig. 2 and Fig. 3 demonstrate the tree structures of two PML document examples, and how tag data that acquired by the RFID readers described and transformed by PML language. The PML documents contain information about the unique ID, task, time, or simply more detailed information about the object itself. It also can be seen that by including the XML data in the user storage area of the RFID tags, PML document 2 represents the object information more detailly than PML document 1.

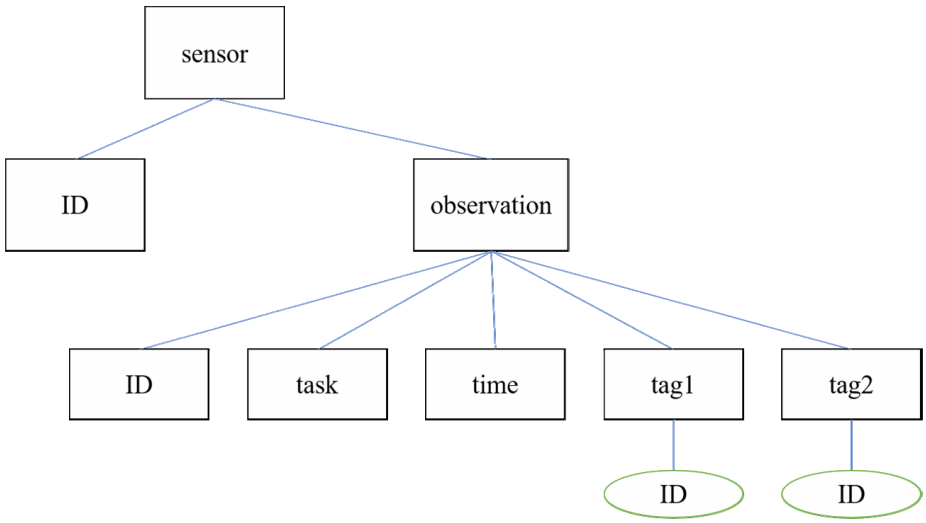

Fig. 2. Tree structure of PML document 1

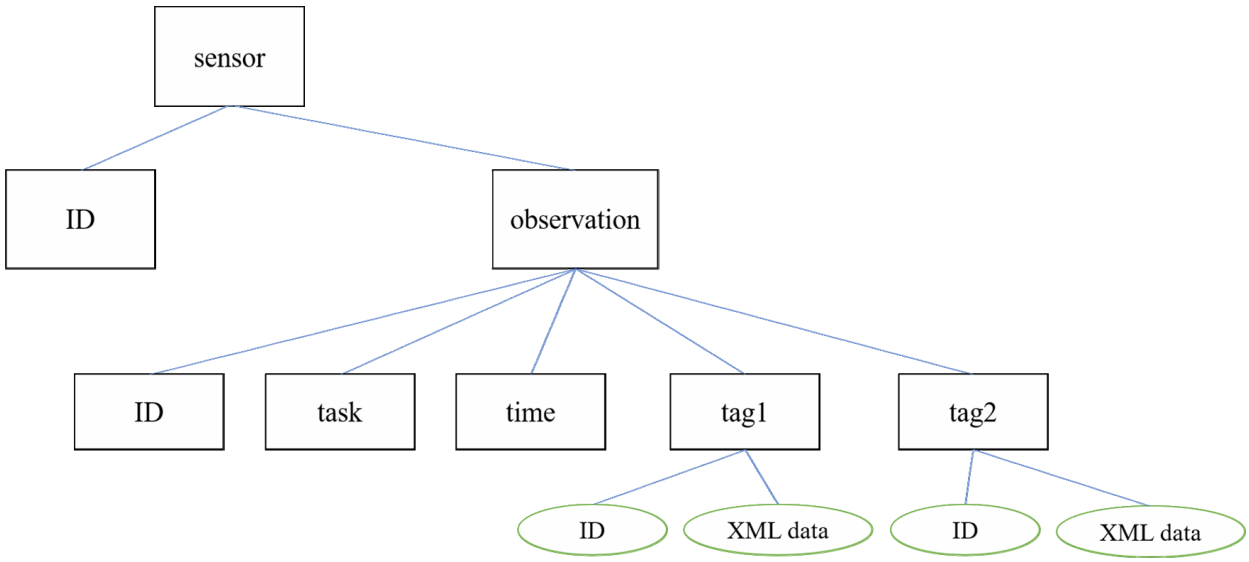

Fig. 3. Tree structure of PML document 2

Additionally, PML documents could change very frequently in highly dynamic environments, both in content and structure. Thus, the target of the PML documents in the present work is to provide a flexible format for the exchange of the data acquired by RFID-based sensors.

3 Bayesian Network

Clustering results always depend on the similarity measure methods and impact the performance of the related applications directly. For Measuring the similarity of PML documents, we establish a Bayesian network model of several RFID-based sensors. The Bayesian network only associates nodes that are probability dependent by arbitrary dependency. Instead of storing all possible states, Bayesian networks only store and process possible state combinations between the related parent node set and the child node set. This greatly saves storage space and computation.

According to the Bayesian network structure, each node represents different attributes and data variables, but there is a certain probability dependence between them. In this work, a Bayesian network is built by defining a conditional probability table for each PML document while highlighting the cause-effect conditional dependence. A naive Bayesian network model can be described as follows in Equation (1).

$$B = (V, E, P) \tag{1}$$

Here, $V = \{V_1, V_2, ..., V_n\}$ represents a set of n random variables, $E = \{V_i V_j \,|\, V_i V_j \in V\}$ represents a set of directed edges, and $P = \{P(V_i \,|\, V_1, V_2, ..., V_{i-1}), V_i \in V\}$ represents the set of conditional probability distribution.

Bayesian networks are based on Bayes theorem which expresses the posterior probability rather than the prior probability [16]. Posterior probability represents the likelihood that an event will occur if a related event has already occurred. Given a set of variables $\{V_1, V_2, ..., V_n\}$, the joint distribution $P(V_1, V_2, ..., V_n)$ is expressed as follows:

$$P(V_1, V_2, ..., V_n) = \prod_{i=1}^{n} P(V_i \mid parents(V_i)), \tag{2}$$

where $parents(V_i) = \{P_1, P_2, ..., P_m\}$ represents the parents set of the variable V_i.

Classification with a Bayesian multinet is carried out by identifying a class c. The probability on c is calculated according to the Bayes rule as Equation (3).

$$P(c|V_1, V_2, ..., V_n)=\frac{P(V_1, ..., V_x \mid c)P(c)}{P(V_1, ..., V_x)} \quad \textbf{(3)}$$

As $P(c|V_1, V_2, ..., V_n)$ is not class-label dependent, it can simplified as:

$$P(c|V_1, V_2, ..., V_n)=\partial P(V_1, ..., V_x \mid c), \quad \textbf{(4)}$$

where ∂ is a prior positive constant, and the Bayesian network representation appears in Fig. 4.

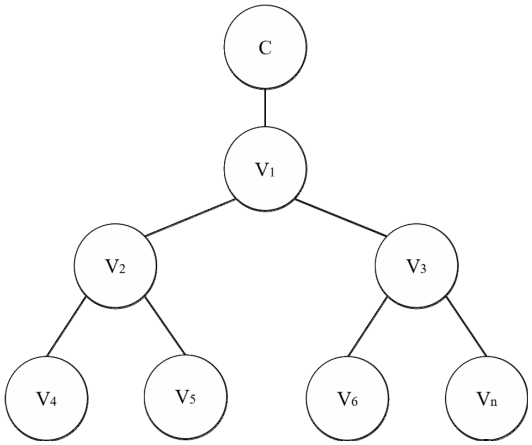

Fig. 4. Bayesian network representation

4 Similarity Calculation

The results of similarity calculation have a great impact on the document clustering quality. However, there is a lack of deep analysis of the document structures in many similarity calculation methods; this can lead to different clustering accuracy results when selecting different types of datasets. With respect to the features of PML documents, we present a hybrid XML document similarity calculation method (see Section 4.3) in this work, which combines the element comparison method (see Section 4.1) and edit distance method (see Section 4.2), thus reducing the deviation caused by using either method alone.

4.1 Element Comparison Method

Element comparison method calculates the similarity between two XML documents by calculating the proportion of the same elements in two XML documents to all elements in the documents. $E(T) = \{e_1, e_2, ..., e_n\}$ represents all the elements contained in the document tree T, and n represents the number of elements. Then, the similarity between the document tree T_1 and T_2 can be formulated as follows in Equation (5).

$$\mathrm{S}im(T_1, T_2) = \frac{\sum_{i=1}^{m} Level(C_i)}{\sum_{i=1}^{n} Level(C'_i) + \sum_{i=1}^{m} Level(C_i)} \quad \textbf{(5)}$$

Here, m is the sum of the same elements in the two document trees, $i = 1, 2, ..., m$, e_{1j} represents an element in T_1, and e_{2k} represents an element in T_2.

4.2 Edit Distance Method

Edit distance method is applied to calculate and display the similarity between two XML document trees by counting the minimum number of steps required to transform one tree into the other. There are three operations permitted on the tree: deletion, insertion and substitution. λ is defined as a cost function, T_1

and T_2 are two document trees and $S = (S_1, S_2, ..., S_n)$ is the sequence of editing operations from T_1 to T_2. Therefore, the number of steps required to transform one tree into the other is expressed as Equation (6).

$$\lambda(S) = \sum_{i=1}^{n} \lambda(S_i) \tag{6}$$

The edit distance which is defined as the minimum cost of operations to transfer T_1 and T_2 can be expressed as follows in Equation (7).

$$ED(T_1, T_2) = Min(\lambda(S)) = Min\left[\sum_{i=1}^{n} \lambda(S_i)\right] \tag{7}$$

4.3 Hybrid Similarity Method

The two methods above represent document similarity from two different aspects. Experimental results [17] showed that the use of either method alone was inadequate. Instead, we adopt a hybrid approach that explores the use of both element comparison and edit distance method. As a linear combination of these two methods, the method is expressed as follows in Equation (8):

$$Sim\_hybrid(T_1, T_2) = w_1 Sim(T_1, T_2) + w_2 ED(T_1, T_2), \tag{8}$$

where w_1 is the weight assigned to $Sim(T_1, T_2)$, w_2 is the weight assigned to $ED(T_1, T_2)$, and w_1 and w_2 are non-negative numbers such that $w_1 + w_2 = 1$. By employing the hybrid similarity calculation strategy, the method is expected to overcome the disadvantages of using either method alone and produce more harmonious clustering accuracy of different types of documents.

5 PML Data Streams Clustering Based on Dynamic Sliding Window

PML is an XML based markup language, the simple, self-describing nature of the XML standard promises to enable a broad suite of IoT network applications, ranging from storage to query. However, existing XML clustering algorithms mainly concern about static data collections, whereas modern information systems frequently deal with streaming XML data that needs to be processed online [18]. This made it difficult for existing data analysis tools, methods, and techniques to be applied directly on the IoT data streams. Additionally, clustering over evolutionary data streams in IoT networks involves more challenges such as dealing with infinite and fast changing data streams. In this sense, we present a clustering method of PML data streams based on a dynamic sliding window to solve this problem.

Fig. 5 shows the clustering model of PML data streams based on the dynamic sliding window. PML documents are randomly selected from different clusters and the similarity calculation between the document and the existing micro cluster feature class is carried out first.

Given the size of sliding window W and the time of PML document passing through the sliding window at a constant speed, ΔT, only unexpired data are summarized in memory and considered for clustering, and the obsolete data beyond that interval are discarded. Our work focuses on continuously producing new clusters on streams over the sliding window, including online handling the newly arrived data and discarding the expired data.

The clustering program determines when to assign data to the sliding window. Before the incoming data is enabled to enter the sliding window, the time of existing data passing through the sliding window ΔT' is computed, and compared with ΔT. while $\Delta T - \Delta T' > \sigma$ the real-time window size W is adjusted to $W' = W + \Delta W$; while $-\sigma \le \Delta T' - \Delta T \le \sigma$ W maintains the same; and while $\Delta T' - \Delta T > \sigma$ W is adjusted to $W' = W + \Delta W$. In this way, the algorithm can adjust the sliding window size dynamically according to the speed of the data streams, and effectively save the time of data buffering and storage, thus adapt to a changing underlying data stream. Particularly, the algorithm does not need extra prior parameters before clustering, such as the initial cluster number and radius of cluster.

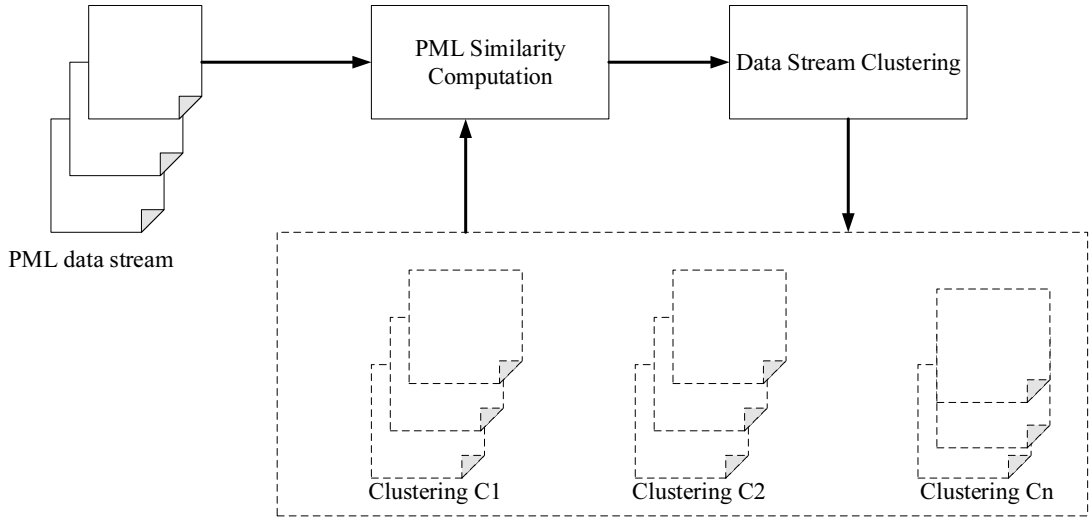

Fig. 5. Data streams clustering model

6 Experimental Simulation and Comparison

The performance of the new clustering method developed in this work is validated by data streams represented by PML documents. The ThinkPad E495 (AMD Ryzen 7 2nd Gen 3700U 2.30 GHz, 8 GB Memory, Windows 10 Pro 64-bit) is employed to implement the proposed method and the related methods and PyCharm 2019 is the development tool.

Three original methods are compared with our proposed method. The first one is the Hierarchical Clustering method presented by I. Ceema et al. [16]. The second one is the Clustream method proposed by Aggarwal et al. [17]. The third one is the XCLS method proposed by Richi Nayak [21]. We evaluate the value range of Running time, Similarity, Purity, Entropy, and F-measure of the clustering results which are discussed and analyzed in Section 6.1, 6.2, 6.3, where the test algorithms are referred to as DSW (short for Dynamic Sliding Window), Hierarchical Clustering, XCLS, and Clustream.

The experiments consist of time performance comparison, similarity calculation comparison, and clustering quality comparison of these methods. First of all, we use an XML generator to simulate a dynamic and variable-speed data streams environment. Then, we select the clustering data set in the simulation platform and select the PML data streams clustering algorithms to Run. Once the program retrieves PML documents in folders, it starts to cluster the PML data streams.

6.1 Time Performance Comparison

To validate the time performance of the similarity calculation algorithms, we first compare DSW with Hierarchical Clustering. Recall from Equation (8) that the hybrid similarity of DSW is a linear combination of the element comparison result $Sim(T_1, T_2)$ and the edit distance result $ED(T_1, T_2)$. Note that we used five pairs of values to represent the varying levels of contribution of each of these similarity methods in our previous works:

1. Element comparison based similarity- $(w_1 = 0.0, w_2 = 1.0)$
2. Edit distance based similarity - $(w_1 = 1.0, w_2 = 0.0)$
3. Hybrid similarity- $(w_1 = 0.25, w_2 = 0.75)$
4. Hybrid similarity- $(w_1 = 0.5, w_2 = 0.5)$
5. Hybrid similarity- $(w_1 = 0.75, w_2 = 0.25)$

To manage the number of experiments to be carried out to make easier comparison, we set $w_1 = 0.5, w_2 = 0.5$ and observe the hybrid similarity measure method in this section.

Fig. 6 shows that the running time of the similarity calculation algorithm exhibits a decreasing trend along with the increase in the document repeatability rate; this attributes to the fact that the number of clusters is decreased with the increase in the document repeatability rate.

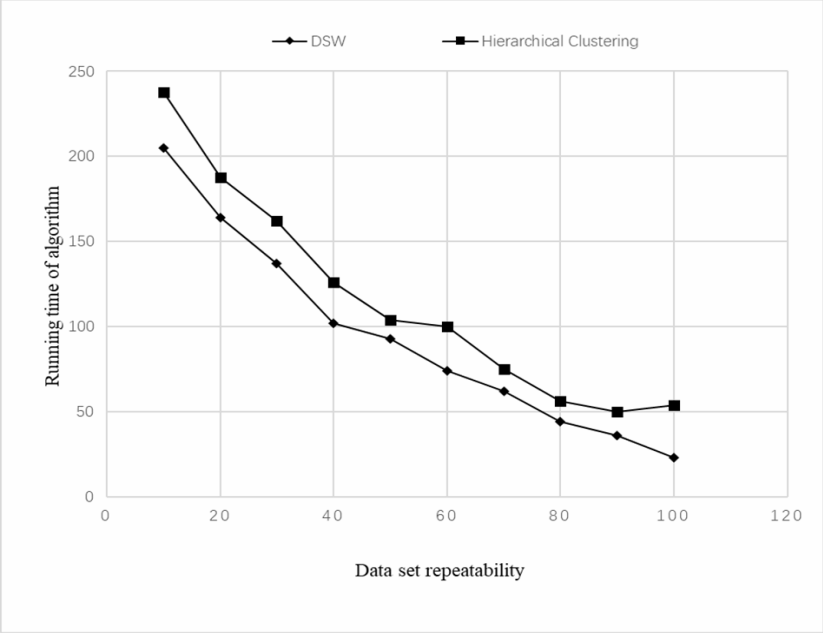

Fig. 6. Time performance of similarity calculation

It is also interesting to observe that the running time of the hybrid similarity calculation algorithm based on the Bayesian network is consistently faster than that of Hierarchical Clustering. Thus, while the Bayesian network only associates nodes that are probability dependent by causal dependency, it provides computational advantages for distributions that can be represented with a PML document structure.

To validate the time performance of the clustering algorithms, we compare DSW with Clustream and XCLS. Fig. 7 shows that the running time of the algorithms exhibits an increasing trend along with the increase in the size of the data set. Given the same size of datasets, DSW performs best (35ms), followed by Clustream (51ms) and XCLS (52ms), which implicates that DSW provides better clustering efficiency.

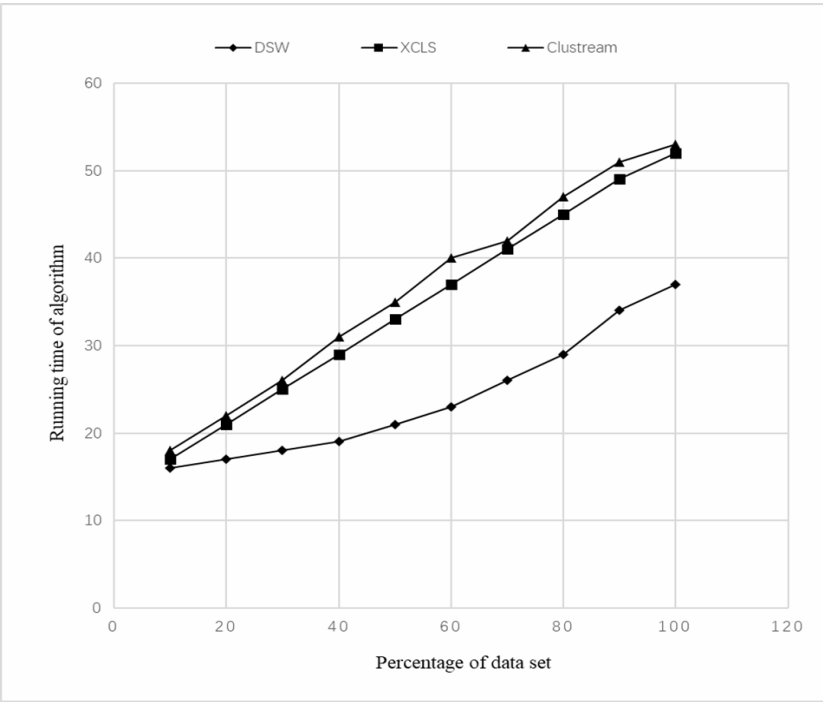

Fig. 7. Time performance of data streams clustering

6.2 Similarity Calculation Comparison

Fig. 8 and Fig. 9 show the testing results of the intra-class and inter-class clustering similarity of the algorithms. Compared with Clustream and XCLS, DSW provides highest intra-class similarity (0.89) and lowest inter-class similarity (0.087) respectively on average. The results also suggest that the hybrid similarity calculation method of DSW, considering the results of both element comparison and edit distance, is better able to assist in the clustering task. By employing the hybrid similarity calculation strategy for different types of datasets, our method produces more harmonious similarity accuracy.

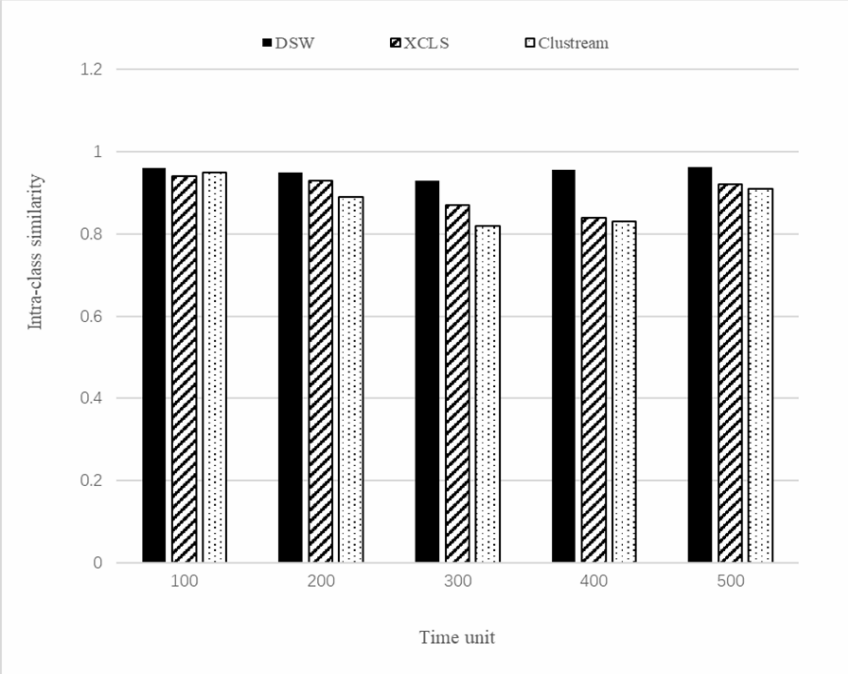

Fig. 8. Similarity within data stream clusters

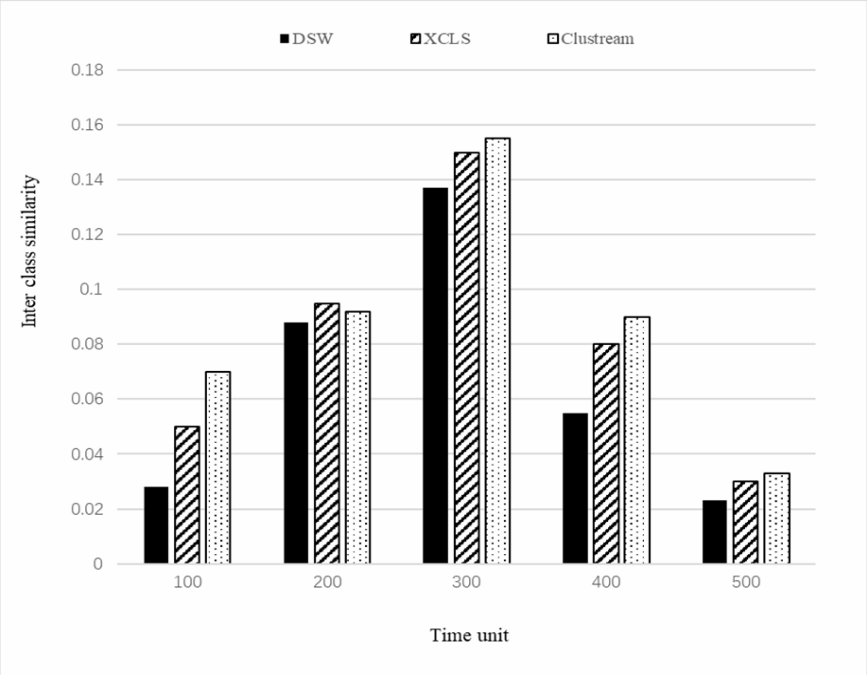

Fig. 9. Similarity between data stream clusters

6.3 Clustering Quality Comparison

To evaluate the global performance of the algorithms, we use validation measures like Purity, Entropy and F-measure. Here, we set the initial window size W = 1024 and Δ W = 0.3W in DSW. Fig. 10 shows that DSW provides the highest purity at 0.89 while XCLS ranks second highest at 0.6. In terms of entropy, DSW ranks lowest at 0.08 while Clustream ranks second highest at 0.23.

The comprehensive evaluation index F-measure is viewed as a measure of an algorithm's accuracy and it is defined as the weighted harmonic mean of the algorithms' precision and recall, as given in Equation (9),

$$F\text{-}measure_{\beta}=(1+\beta^2)\frac{Precision \cdot Recall}{\beta^2 Precision + Recall} \quad \textbf{(9)}$$

where coefficient β is the relative importance of recall and precision. Considering that recall and precision are equally important to the results, we set β =1. It is noticed that DSW yields the best F-measure (0.87), followed by Clustream (0.62) and XCLS (0.61).

To sum up, our method outperforms Clustream and XCLS in terms of clustering accuracy and this might attribute to the fact that our method uses unfixed size observation windows which can adaptively adjust to capture local trends in the most recent data. Besides, by discarding expired PML documents beyond the time interval, the number of documents in DSW is made smaller, which reduces the interference between PML documents thus increase the clustering accuracy.

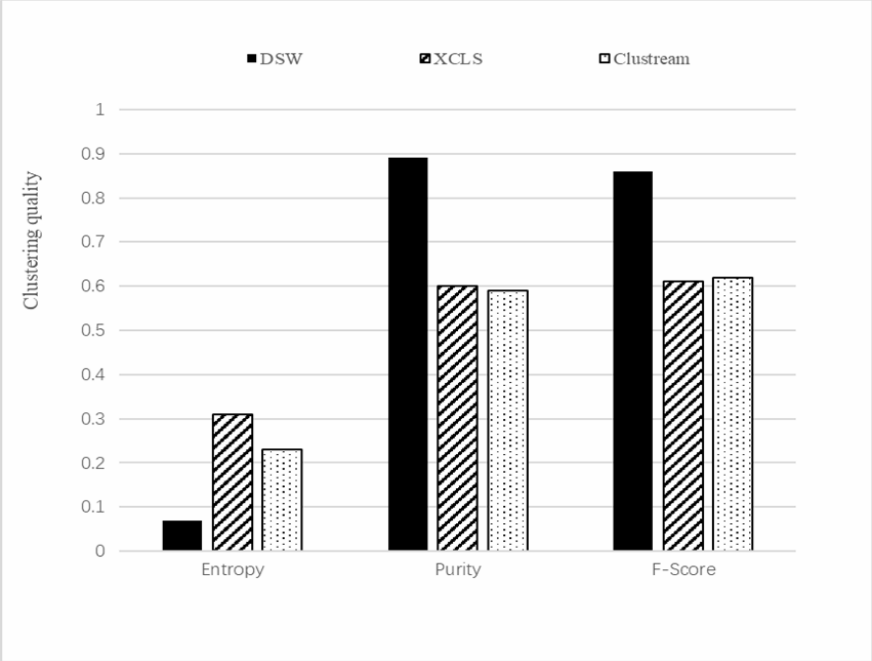

Fig. 10. Entropy, purity and F-measure of clustering algorithms

7 Conclusion

Clustering is a challenging task while clustering over evolutionary data streams in IoT networks involves additional challenges such as dealing with infinite and consistently changing data streams. In this work, we proposed a fast data clustering method to solve this problem.

Firstly, PML documents are presented to describe the objects and environments of industrial, commercial and consumer applications. With respect to the features of PML documents, a hybrid similarity calculation method of PML documents based on the Bayesian network is built to assist in document clustering. In the similarity calculation, we optimized the similarity measure as a linear combination of the element comparison method and the edit distance method instead of using either method alone. Secondly, a clustering model of PML data streams based on a dynamic sliding window is

presented. As the algorithm progresses, it continuously online handles the new arrival data and discarding the expired data. By analyzing the speed of the data streams, the algorithm adjusts the sliding window size dynamically. Finally, the simulations evaluate the value range of Running time, Similarity, Purity, Entropy, and F-measure of our proposed method and the relevant existing clustering methods.

The experimental results indicate that the similarity calculation method of PML documents based on the Bayesian network outperforms the other related methods in terms of accuracy and speed. It implicates that the Bayesian network can be used to cluster PML documents instead of variables and has the enormous potential to be used to calculate the similarity between data streams within IoT networks. Besides, the clustering method based on the dynamic sliding window can deal with evolutionary underlying data streams while still maintains comparable high clustering accuracy. Moreover, our method provides computational advantages for distributions that can be represented with a PML document structure.

In future work, we will focus on the optimization of its performance:

1. Incomplete-document loss of the data clustering was not considered in this work. It is necessary to improve the fault tolerance of the algorithm.

2. As the pattern that data streams present may change significantly over time, our algorithm needs to be optimized to respond faster to sudden changes in underlying data distributions.

Acknowledgments

This work is supported by the following projects:

(1) 2019 Zhongshan Social Public Welfare Science and Technology Research Project (Project Number: 2019B2082).

(2) 2019 Guangdong Provincial Department of Education General Colleges and Universities Scientific Research Project (Project Number: 2019GKQNCX138).

References

[1] OECD, The Internet of Things-Seizing the Benefits and Addressing the Challenges. <https://www.oecd-ilibrary.org/science-and-technology/the-internet-of-things_5jlwvzz8td0n-en>, 2016 (accessed 07.06.16).

[2] Q. Zhang, C. Zhu, L. -T. Yang, P. Li, An incremental CFS algorithm for clustering large data in industrial Internet of Things, IEEE Transactions on Industrial Informatics 99(2017) 1193-1201.

[3] X. Yao, J. Wang, M. Shen, H. Kong, H. Ning, An improved clustering algorithm and its application in IoT data analysis, Computer Networks 159(2019) 63-72.

[4] Z. Liu, Q. Zheng, Z. Ji, W. Zhao, Sparse Self-Represented Network Map: A fast representative-based clustering method for large dataset and data stream, Engineering Applications of Artificial Intelligence 68(2018) 121-130.

[5] F. Gao, D. Chen, K. Zhou, W. Niu, H. Liu, A fast clustering method for identifying rock discontinuity sets, KSCE Journal of Civil Engineering 23(5)(2018) 556-566.

[6] A. Abid, S. Jamoussi, A. B. Hamadou, AIS-Clus: A Bio-Inspired method for textual data stream clustering, Vietnam Journal of Computer Science 06(02)(2019) 223-256.

[7] C. Fahy, S. Yang, M. Gongora, Ant Colony Stream Clustering: A fast density clustering algorithm for dynamic data streams, IEEE Transactions on Cybernetics 99(2018) 1-14.

[8] M. M. Ferdaus, M. Pratama, S. G. Anavatti, M. A. Garratt, PALM: An incremental construction of hyperplanes for data stream regression, IEEE Transactions on Fuzzy Systems 99(2018) 2115-2129.

[9] V. Ignatjev, D. Stankevich, A fast estimation method for the phase difference between two Quasi-harmonic signals for real-time systems, Circuits Systems & Signal Processing 36(9)(2017) 3854-3863.

[10] I. Ceema, M. Kavitha, G. Renukadevi, G. sripriya, S. RajeshKumar, Clustering web documents using hierarchical method for efficient cluster formation, International Journal of Advanced Research in Computer Science and Electronics Engineering 1(5)(2012) 127-131.

[11] C. C. Aggarwal, J. Han, J. Wang, P. S. Yu, A framework for clustering evolving data streams, in: Proc. 29th international conference on Very large data bases, 2003.

[12] A. Kumar, A. Singh, R. Singh, An efficient hybrid-clustream algorithm for stream mining, in: Proc. 13th International Conference on Signal-Image Technology and Internet-Based Systems, 2017.

[13] J. A. Hartigan, M. A. Wong, Algorithm AS 136: A k-means clustering algorithm, Jproposednal of the Royal Statistical Society, Series C (Applied Statistics) 28(1)(1979) 100-108.

[14] F. Cao, M. Estert, W. Qian, A. Zhou, Density-based clustering over an evolving data stream with noise, in: Proc. 6th SIAM International Conference on Data Mining, 2006.

[15] R. Nayak., Fast and effective clustering of XML data using structural information, Knowledge and Information Systems 14(2008) 197-215.

[16] H. Yazid, K. Kalti, N. E. Benamara, A new similarity measure based on Bayesian Network signature correspondence for brain tumors cases retrieval, International Journal of Computational Intelligence Systems 7(6)(2014) 1123-1136.

[17] G. Chowdhury, Social information retrieval systems: Emerging technologies and applications for searching the web effectively, Journal of the American Society for Information Science and Technology 61(12)(2010) 2587-2588.

[18] M. Garofalakis, A. Kumar, XML stream processing using tree-edit distance embeddings, ACM Transactions on Database Systems 30(1)(2005) 279-332.