Neurocomputing 191 (2016) 34–43

Contents lists available at ScienceDirect

Neurocomputing

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

Incremental density-based ensemble clustering over evolving data streams

Imran Khan [a,*], Joshua Z. Huang [b], Kamen Ivanov [c]

[a] Shenzhen Key Laboratory of High Performance Data Mining. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[b] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[c] Shenzhen Key Laboratory for Low-cost Healthcare, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 10 June 2015
Received in revised form
31 December 2015
Accepted 12 January 2016
Communicated by Ning Wang
Available online 4 February 2016

Keywords:
Ensemble clustering
Data streams
Density-based clustering
Smart grid

ABSTRACT

The recent advances in smart meter technology have enabled for collecting information about customer power consumption in real time. The measurements are generated continuously and in some cases, e.g. in the industrial smart metering the data exchange rates are highly-fluctuating. The storage, querying, and mining of such smart meter streaming data with a large number of missing and sparse values are highly computationally challenging tasks. To address such matters, we propose a new method called incremental density-based ensemble clustering (IDEStream) for incremental segmentation of various kinds of factories based on their electricity consumption data. It exploits a gamma mixture model to suppress the influence of sparse data units in the data streams that sequentially arrive within a time window and then generates a clustering from the processed data of that window. IDEStream uses a unique incremental ensemble approach to incrementally aggregate the clusterings of subsequent time windows. Experimental results on data streams collected by smart meters from manufacturing factories in Guangdong province of China have shown that the proposed algorithm outperforms several state-of-the-art data stream clustering algorithms. The obtained segmentation can find numerous applications, an exemplar one being to define customer rates in a flexible way.

1. Introduction

Smart grid data often poses challenges such as very large size, high dimensionality, skewed distribution, sparsity and seasonal variations. It brings out the wealth and challenges to discover the segmentation of factories based on their electricity consumption data. It would not only bring benefits to electricity service providers by supplying them with demand response predictions but would also reveal the real economic structure. Such segmentation can be used for an integrated planning system, where the appropriate real-time selection among the available load-management alternatives will be critical to meet effectively the system demands [1]. The consumption patterns of electricity customers are also a source of valuable information to determine optimal tariff rates [2,1]. Load profiling is emerging as one of the most suitable methods to deal effectively with the data of customer power consumption, which is often presented in the form of load diagrams [3].

The exploration of the data streams over consecutive time windows allows for deep understanding of the clusters behavior. The real-time data arrive in a sequential form, where the underlying distribution may change over the time intervals. For instance, a variation in smart meter streaming data is to be expected depending on many factors e.g. production order, weather conditions, working hours, and price incentives. The smart meter readings received at an instant of time may have a dynamic distribution or contain a large number of sparse and missing values. The sparse values are usually considered as outliers that may degrade the performance of online streaming algorithms. The most existing works dedicated to resolving this problem have concentrated on algorithms that adapt to varying distributions either by rejecting old data or giving it lower weight [4]. When processing stream data in real time, particular challenges are the limited operational memory and the fact that only one processing pass over the input data is available. Traditional algorithms are not suitable for such type of data because they extract patterns from

* Corresponding author.
E-mail addresses: imran.khan@siat.ac.cn (I. Khan), zx.huang@szu.edu.cn (J.Z. Huang), kamen@siat.ac.cn (K. Ivanov).

http://dx.doi.org/10.1016/j.neucom.2016.01.009
0925-2312/

Table 1
Analysis of six data stream clustering algorithms.

Algorithm	Criteria				
	Data structure	Cluster algorithm	Outlier detection	Cluster shape	Cluster problem
CluStream	Feature vector	*k*-means	Statistical-based	Hyper-sphere	Object
ClusTree	Feature vector	*k*-means/DBSCAN	–	Arbitrary	Object
DStream	Grid	DBSCAN	Density-based	Arbitrary	Object
DenStream	Feature vector	DBSCAN	Density-based	Arbitrary	Object
HPStream	Feature vector	*k*-means	Statistical-based	Hyper-sphere	Object/feature
HDDStream	Feature vector	PreDeCon	Density-based	Arbitrary	Object/feature

data by considering the global properties, rather than undertaking the local ones. Moreover, they require the whole training data set.

In this paper, we suggest an algorithm that allows us to discover clusters in smart meter data in real time. We called it IDEStream, and it makes use of an incremental ensemble clustering approach for discovering of clusters in evolving data streams. The algorithm defines subsequent, non-overlapping time windows and performs incremental ensemble clustering on arrival of data within each time window. IDEStream comprises three phases. During the first phase, we utilize a gamma mixture model to identify the dense units in sequentially arriving data, and the sparse values are dissolved. In the second phase, the processed data of a single time window are collected, and clustering is performed over that data. Finally, in the third phase, the algorithm performs incremental ensemble clustering between obtained clusterings of two subsequent time windows. Then, the same process repeats and every time the clustering from the last time window is aggregated with the aggregated one from the previous time windows. Experimental results have been obtained using smart meter data sampled at 15-min intervals, collected at manufacturing industries located in Guangdong province of China. According to results, the proposed algorithm outperforms several well-known state-of-the-art data stream clustering algorithms.

The rest of the present work is structured as follows. In Section 2, a review of the data stream clustering techniques is provided. Section 3 provides an overview of smart grid data. In Section 4, a detailed description of IDEStream is presented. Section 5 shows a detailed evaluation of clustering results on the selected real-world data set. Section 6 contains the concluding remarks.

2. Related work

In this section, we review the related clustering algorithms for processing of evolving data streams. Recently, there are a lot of literature reviews that depict a large number of new mining methods [5–11]. However, all traditional clustering algorithms, such as k-means, Self-Organizing Maps (SOM) and Two-Step algorithm, are still widely used for load profiling. Traditional methods always operate over all feature spaces of an input data set to learn as much as possible. Thus, a lack to discover the hidden patterns in subspaces leads to a degraded performance.

The data stream clustering algorithms like CluStream [12], DenStream [13], DStream [14], and ClusTree[15] are being commonly utilized for mining of evolving data streams. JA Silva in [16] provided a detailed description of such data stream clustering algorithms. CluStream divides the clustering process into online and offline components. The online component incrementally maintains summary information of the data stream while the offline component considers the obtained information and user input to produce on-demand clustering results. CluStream generates only spherical-shaped clusters because it is based on the *k*-means clustering algorithm. Moreover, it is not suitable for noisy data and needs a constant predefined number of micro-clusters.

DenStream follows the online–offline rationale of CluStream, but in contrast to CluStream, it generates clusters of arbitrary shapes. The online component of DenStream focuses on the collection of micro-clusters, and incoming data points are assigned to the nearest micro-clusters. The offline component of DenStream uses these micro-clusters to generate on-demand clustering results. DStream is another grid-based stream clustering algorithm, which maintains the summary information into the grid cells. The grid cells are similar to micro-clusters. DStream uses a hash table for assigning the incoming data point to an appropriate grid cell. The offline component generates clustering results using the dense grids cells. Similar to DenStream, the ClusTree algorithm also utilizes a vector of summary information, which is kept in a hierarchical tree.

HPStream [17] is an extension of CluStream, and it is also a method for projected data stream clustering. It assumes a constant number of clusters over the whole lifetime of the stream. HDDStream [18] is another similar type of approach for projected data stream clustering, which adjusts the number of clusters over the time and generates clusters of arbitrary shapes. HDDStream works in online and offline modes; the online mode generates macro-clusters with projected clustering, while the offline mode produces micro-clusters. DenStream, DStream, HPStream, and HDDStream are based on the fading model of stream data. Table 1 provides a brief summary of the discussed algorithms.

In all online–offline algorithms, the offline component always contributes to the high computational cost of generating clusters. To find clusters that evolve over time, the online–offline algorithms always run their offline component multiple times, which is very time-consuming process. The data clustering performance of this kind of algorithms is also affected in many different ways. Such problems arise from the difficulties to derive summary information from the streaming data, intricacy of the data structure that are used for storing and managing the summary, and also, very often, the utilization of the offline component for cluster extraction. When processing real-time streaming data, due to the unavailability of the whole training data set, it is hard to predict the structure of clusters (arbitrary or spherical) in advance. Thus, it is also difficult to choose an appropriate data stream clustering algorithm.

IDEStream performs the following tasks. (1) Processes highly skewed scalable data in real time to dissolve dense information. (2) Maintains updated clustering information online of available data streams. (3) Adjusts the number of clusters over the time. (4) Extracts online the clusters that are embedded in the subspaces. (5) Generates consistent responses while operating within the available time and memory over a sequence of time windows.

3. Data description

In this section, we present a detailed description of the obtained smart meter data streams, collected at manufacturing factories located in Guangdong province of China.