Knowledge-Based Systems 70 (2014) 3–14

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Hyper-ellipsoidal clustering technique for evolving data stream

Muhammad Zia-ur Rehman, Tianrui Li *, Yan Yang, Hongjun Wang

School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

ARTICLE INFO

Article history:
Available online 8 December 2013

Keywords:
Data mining
Decision support systems
Hyper-ellipsoidal clustering
Evolving data stream
Data clustering

ABSTRACT

Data mining has become a key ingredient in establishing intelligent decision support systems. As one of main branches in data mining, data stream clustering has received much attention over the past decade. Most existing data stream clustering techniques count on Euclidean distance metric for finding similar objects and hence produce spherical clusters which are not always suitable to represent the data. Moreover, in most of the real world problems, we come across the data of varying density which cannot be handled by density-based clustering techniques. In this paper, we introduce a new clustering technique called Hyper-Ellipsoidal Clustering for Evolving data Stream (HECES) based on the recently proposed HyCARCE algorithm. In HECES, a few modifications in the HyCARCE algorithm are made for handling stream clustering problem: sliding window model is used to handle incoming stream of data to minimize the impact of the obsolete information on recent clustering results; shrinkage technique is used to avoid the singularity issue in finding the covariance of correlated data; a novel technique for merging the initial ellipsoids is used to obtain the final clusters instead of a computationally intensive process of expansion and adjustment. HECES relies on Mahalanobis distance metric to cluster the data points and hence results in ellipsoidal shaped clusters. It can successfully handle data of varying density. Experiments on various synthetic and real datasets for clustering streaming data provide a comparative validation of our approach.

1. Introduction

Advances in computer technology provide a broad range of applications which recast the data mining techniques. Automated data processing has revolutionized the data handling techniques and new data mining techniques are emerged [1,2]. Data mining is the process of identifying novel, effective knowledge from large datasets and becomes a key ingredient in establishing intelligent decision support systems [3]. As one of the main techniques of data mining, clustering is an effective approach to discover hidden groups called clusters within the data (observations and features/dimensions) based on similarity. Objects in a cluster are more similar to each other than they are to an object belonging to other cluster. Certain distance function is used to determine similarity or proximity between a pair in a cluster. A variety of distance measures are discussed in literature having inherent assumptions about cluster shapes [4], such as Euclidean distance is the most commonly used distance metric for proximity check. It performs well for the clusters which are well separated and spherical shape. Another well-known distance metric is called Mahalanobis distance metric, which is capable of predicting much broader range of data distribution from linear to spherical. Moreover, Mahalanobian prediction is based on whole data therefore less sensitive to outliers [5]. As we know, clustering is an unsupervised machine learning technique, and does not require a learning set or prior information about data to establish the grouping of similar objects into distinct clusters. Therefore, the usage of distance metric having implicit assumption of the spherical cluster may produce erroneous results.

Numerous new problems come forth in different scientific projects, such as human genetics, medical imaging, social network analysis, software evolution and evolutionary algorithms, which can be efficiently handled by robust clustering techniques [6,7]. Most of the above mentioned real world applications produce a huge amount of continuous data, called data stream. Data stream mining is relatively a new area of research in the data mining community to better support our decision making. Patterns to study in data streams are volatile as one might have a limited time to discover them without storing the incoming data. Many data stream mining approaches have been proposed such as association rule mining, sequential pattern mining, text mining, classification and clustering. Clustering is the process of discovering useful patterns in data by grouping similar data elements using a given similarity metric [8,9]. Finding groups of similar points is a nontrivial problem and it becomes even harder while dealing with massive, continuous and agile sequence of data items in the form of data

* Corresponding author. Tel.: +86 28 86466426.
E-mail addresses: moh.zia@gmail.com (M. Zia-ur Rehman), trli@swjtu.edu.cn (T. Li), yyang@swjtu.edu.cn (Y. Yang), wanghongjun@swjtu.edu.cn (H. Wang).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.11.022

stream. Moreover, clusters in data stream evolve with time and the number of clusters cannot be defined before hand. In the streaming environment, not only the number of clusters can change but also their center points, shapes and orientation can vary abruptly with time. Thus the clustering for streaming data becomes an even more aggravated problem.

Problem of clustering for streaming data has been recently studied extensively [6,10–12]. Most of popular algorithms, e.g., variants of *K*-means, for data stream mining tasks are very sensitive to the initial seeds and the selected number of clusters. Resultant clusters are always in spherical shapes because the Euclidian distance metric is used, which cannot always flexibly represent the incoming streaming data. The Mahalanobis distance is also used in some of the algorithms for data stream mining to find the clusters for evolving data but they are fuzzy clustering algorithms [13,14].

In this paper, we introduce an efficient stream clustering algorithm called Hyper-Ellipsoidal Clustering for Evolving data Stream (HECES) by a few modifications in the recently proposed HyCARCE algorithm [15], which is a powerful clustering technique, specially designed for low dimensional static data. The HyCARCE has a very low computational cost, which makes it attractive for applying on streaming data. But we cannot apply it directly because there are several technical issues to be addressed first.

The main contributions of our work are summarized as follows: (1) The HyCARCE has been designed for static data clustering and it must be adopted for online handling of streaming data. We use a sliding window model for discovering most recent clustering results without storing the data itself (Section 3.3). (2) Dividing the input space into a set of fixed sized grid-Cells may produce grids having linearly correlated or insufficient data with a badly scaled covariance. Covariance shrinkage estimation is used to overcome this problem (Section 3.3.3). (3) Instead of using computationally expensive procedure of expansion and readjustment, we use the heuristic technique for merger of initial clusters to obtain final clusters (Section 3.3.4). (4) Performance evaluation using real and synthetic streaming data confirms that results discussed here are clearly superior to other competitive algorithms in all aspects, including the compactness of clusters and the cluster matching accuracy (Section 5).

The remaining of the paper is organized as follows. Section 2 gives a brief overview of related work. Key concepts and the hyper-ellipsoidal clustering technique for evolving stream clustering together with an illustration are given in Section 3. The HECES algorithm as well as its complexity analysis is outlined in Section 4. Experimental setup and evaluations with real and artificial datasets are given in Section 5. The paper ends with conclusions and future research work in Section 6.

2. Related work

Nowadays, the volume of information continues to grow at an unparalleled rate. Such information often evolves over time. Clustering is a distinct data mining task. In this paper, we study the clustering problem for multidimensional data in the form of a stream. Data stream mining in general and data stream clustering in particular posses unique challenges such as single pass clustering, limited time and memory for analysis, lack of prior information on shape, size and number of clusters, volatile nature of stream under concept drift and concept shift. A good clustering algorithm must have to fullfil all the above challenges simultaneously.

Clustering data stream is a very well-studied problem in data mining domain. The earliest data stream clustering algorithm is BIRCH [16]. It is a heuristic clustering approach which summarizes the incoming data into a so-called Clustering Feature (CF) vector in an online fashion, then CFs are clustered together using a memory efficient height balance tree structure called CF tree. Other earliest data stream clustering techniques use divide and conquer schemes [17] and detect clusters using *K*-means or its variants as the base algorithm [12,18]. There is no optimal solution of these algorithms and the final solution is the approximation of the optimal solution. When the size of the streaming data increases, these algorithms recursively call themselves which deteriorate the quality of the solution. StreamKM++ is another *K*-means clustering algorithm proposed by Ackermann et al. [11]. It organizes the data in a small number of samples and merges the samples when the number of input points in two or more samples are equal. New samples are created from the combined sample using a coreset tree structure, which is used to estimate the new sample point.

A series of grid-based clustering algorithms has been long proposed for streaming data such as STING [19], WaveCluster [20], DStream [21] and D3 [22]. Grid-based clustering approaches maintain the distribution and statistics of incoming streaming data without storing the data itself. One of the key steps in grid-based techniques is the partitioning of the input data space into grid-Cells of the equal length in all dimensions. These grid-Cells are the starting point for the clustering process. Grid-based algorithms are robust against the noise with almost linear time complexity and are very successful for datasets with a high density. However, the number of grids increases exponentially with the increasing dimension of data.

Density-based clustering methods are non-parametric techniques for defining clusters over dense areas. Unlike other clustering methods, density-based methods do not require the number of clusters as an input. However, several parameters are needed to be provided in advance such as the weight and radius of core micro-clusters, where micro-cluster is the representation of summary for data in the stream. One of the most popular density-based clustering methods for static data is DBSCAN [23] proposed by Easter et al. A few years later, they proposed the incremental version named incDBSCAN [24], which is able to regroup the clusters after receiving new data in the data warehouse environment. Both of these methods are not suitable for data stream framework, where the entire historical data cannot be revisited.

There are some density-based clustering algorithms which are specially designed to handle data stream. Most of them generate the summary of incoming data stream and use this summary to perform clustering. In CluStream [6], the concept of micro-clusters is proposed based on the concept of CF introduced in BRICH [16], which is originally designed for warehouse. CluStream is divided into online and offline phases, in which micro-clusters are basically the data structure used to store the summary of incoming data items in the online phase. In the offline phase, the clustering algorithm is applied on micro-clusters to find out the final clusters called macro-clusters. DenStream [25] uses the framework of CluStream to develop a density-based clustering algorithm for evolving data streams. It is also divided into two phases and can detect clusters with arbitrary shapes. It introduces the concept of outlier micro-clusters and potential core micro-clusters to accommodate the possible future clusters because an outlier may possibly become a core micro-cluster and vice versa. rDenStream [26] is a three-step data stream clustering algorithm based on DenStream. It introduces an additional third step, named retrospect step, for clustering data stream. In this step, the discarded micro-clusters are held little longer to improve the clustering accuracy in the future. C-DenStream [27] is another density-based clustering algorithm. It is a semi-supervised clustering algorithm. It exploits the background knowledge in the form of instance level constraints (must-link and cannot-link constraints) to perform the clustering on micro-clusters. It is basically an extension of the static semi-supervised clustering algorithm C-DBSCAN [28].