Neurocomputing 219 (2017) 502–517

Contents lists available at ScienceDirect

Neurocomputing

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

Supervised Adaptive Incremental Clustering for data stream of chunks

Laiwen Zheng, Hong Huo*, Yiyou Guo, Tao Fang

Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, China

ARTICLE INFO

Communicated by Prof. Liu Guangcan

Keywords:
Automatic clustering
Adaptive update
Classification
Data stream of chunks
Supervised clustering

ABSTRACT

Many supervised clustering algorithms have been developed to find the optimal clusters for static datasets by presetting some parameters, but they are seldom suitable for dynamic datasets, such as the data stream of chunks. To find the optimal clusters of the data stream of chunks, a novel Supervised Adaptive Incremental Clustering (SAIC) algorithm is proposed. SAIC can cluster dynamic datasets of arbitrary shapes and sizes automatically. It includes learning and post-processing phases. In the learning phase, each cluster updates adaptively according to its learning rate that is calculated from its counter value. All data points are shuffled at each iteration in order to make SAIC insensitive to the input order of data points. In the post-processing phase, the outliers or boundary points are eliminated according to the counter value of each cluster and the number of iterations. Four synthetic datasets and fourteen UCI datasets are used to evaluate the performance of SAIC, respectively. The experiments on UCI datasets show that SAIC reaches to or outperforms some other supervised clustering algorithms and several unsupervised incremental clustering algorithms. In addition, three data stream of chunks are used to evaluate SAIC from different aspects, which shows SAIC has the scalability and incremental learning ability for the clustering of data streams of chunks.

1. Introduction

In the real-world applications, more and more labeled and unlabeled data have been collected nowadays, such as surveillance video streams, credit card transactional flows, social text streams, photo streams [1] and so on [2]. Clustering data stream has been an active research area with advances in data collection techniques [3], e.g., database systems, data mining [4], information retrieval [5] and etc. Although unsupervised clustering [6] and semi-supervised clustering [7,8] have been proposed for data streams, unfortunately, there is little research on the supervised clustering of data stream so far [9]. In unsupervised clustering, the class labels of all data points are unknown, so clustering is only performed by similarity information; in semi-supervised clustering, in addition to the similarity information used by unsupervised clustering, the class labels of a small amount of data points are known and are used to "guide" the clustering process [10]; in supervised clustering, data attributes and class labels take part in grouping data points into clusters, and class labels are mainly used to ensure that the clusters of a given class have high probability density, and cluster purity is as high as possible [11,12].

For making best use of the labeled data to process a large amount of unlabeled data, the supervised clustering has been attracted wide attention in recent years [13]. For labeled data, the attribute and label information of the data points are known, but we do not know which of them are redundant data points or outliers. Therefore, we can discover the distribution of data points in the attribute space by clustering, thereby to find the "optimal" set of clusters to describe the data [14], or to detect outliers. However, the optimal clustering of a dataset has been proved to be an NP-complete problem [15].

Many supervised clustering algorithms [11,16–18,12] have been developed to find the optimal clusters of a dataset, but most of them are mainly tailored to the clustering of static dataset, not of data stream. However, it is very necessary to develop supervised clustering algorithms that could handle data stream of chunks due to the following two reasons. First, nowadays the data in many applications is not static, which is often continuously coming as data chunks in a data stream; Second, because of the limitation of computer memory, it is impossible to load a large static dataset completely [19], so some researchers proposed to divided a large static dataset into data stream of chunks for clustering [20,21]. Unlike static dataset with the fixed number of data points, in data stream of chunks, the number of data points is continuously increasing. The data points in data stream of chunks can not be accessed at one time, compared to a one-time accessing of all data points in static dataset supposed that the computer memory is large enough. Moreover, the number of data points and the number of categories in each chunk are all random, so it is hard to know what kind

* Corresponding author.
E-mail address: huohong@sjtu.edu.cn (H. Huo).

http://dx.doi.org/10.1016/j.neucom.2016.09.054
Received 23 January 2016; Received in revised form 20 September 2016; Accepted 26 September 2016
Available online 28 September 2016

of data will be presented in the next time. Therefore, the clustering of data stream of chunks is more difficult than that of static data.

To cluster the data stream of chunks, the incremental learning might be a suitable solution, as the information about the new clusters can be obtained from the new input training data, while the information about the old clusters can be retained simultaneously. So it doesn't need to keep a lot of training samples for incremental learning [22]. However, many challenges and issues [20] will be encountered and must be solved for data stream clustering, three of these issues are very important for clustering algorithms for data stream. First, the algorithms should have the scalability and incremental learning ability. Because the data points that take part in clustering are constantly changing for clustering of data streams, the obtained clusters should have the ability to save the knowledge of all data points that have been learned. In addition, the scalability and incremental learning ability of an algorithm will be affected if too many parameters need to be set in the algorithm. Second, how to reduce the impact of the input order of data points on clustering results. Many researchers regard the "incremental clustering" or "online-clustering" of data streams as one-pass clustering [20,23]. In our opinion, these two terms refer to the same thing in the study of clustering of data streams, but, only the emphasis point is different from different aspects. The term "online-clustering" emphasizes that the data points are constantly arriving when clustering, while the term "clustering incremental" emphasizes that each time only a portion of data points take part in clustering. However, one-pass clustering will inevitably lead to the "local bias of input order" problem [17], i.e., different input orders of training data points may produce quite different cluster structures. Therefore, how to reduce the impact of the input order of data points on clustering results is a research hotspot in clustering of data streams. Third, due to the inevitable existence of outliers in the data streams, how to detect and remove outliers is also very important for supervised clustering of data streams.

To automatically find the optimal clusters of the data stream of chunks, a novel Supervised Adaptive Incremental Clustering (SAIC) algorithm is proposed. SAIC includes learning and post-processing phases. The learning phase iteratively generates candidates clusters, and post-processing phase removes outliers or boundary points. In order to address the above three issues that encountered in clustering of data stream of chunks, the contributions of this paper are as follows: (1) In order to realize incremental clustering, a counter is set for each cluster to record its winning times in the learning process. As a result, each cluster has three pieces of information: class label, attributes, and a counter. These three pieces of information can save the knowledge of all data points that have been grouped into a cluster, so the learned data chunks can be discarded, which makes SAIC can be used for the incremental clustering of data stream of chunks. (2) In learning phase, the clusters of data chunks are automatically found. The data points in one chunk are shuffled and learned for multi-pass to improve the robustness of clustering results, and the number of iterations can be automatically determined. The learning rate calculated automatically by the winning times of a cluster can be used to realize the adaptive update of a cluster. (3) In the post-processing phase, SAIC utilizes the counter value of each cluster and the number of iterations to eliminate outliers or boundary points. Four synthetic datasets and fourteen UCI datasets are used to evaluate the performance of SAIC. Because a static dataset can be regarded as one chunk of data stream, namely the clustering of static dataset is a special case of that of data stream of chunks. Therefore, the datasets used in experiments include both static datasets and data streams. The experimental results on UCI datasets have shown that the performance of SAIC not only reaches or outperforms some other supervised clustering algorithms that aim to finding the optimal clusters of a dataset, but also outperforms several unsupervised incremental clustering algorithms. In addition, one synthetic data stream of chunks and two real data stream of chunks are also used to evaluate the performance of SAIC, which shows SAIC has the scalability and incremental learning ability for the clustering of data streams of chunks.

The organization of the rest of this paper is as follows. Section 2 reviews the existing supervised clustering algorithms on static datasets and incremental clustering algorithms on data streams. Section 3 presents the proposed SAIC. The complexity of SAIC is analyzed, and the classification by using the obtained clusters is also discussed. In Section 4 the performance of SAIC is evaluated by comparing with several other supervised clustering algorithms and unsupervised incremental clustering algorithms. The last section is the conclusion and the future work.

2. Related work

In this section, both the supervised clustering algorithms for static dataset and the incremental clustering algorithms for data stream are reviewed, and three issues that are important to the incremental clustering algorithms are also discussed.

2.1. Supervised clustering algorithms for static dataset

According to the representations of clusters, these supervised clustering algorithms for static dataset can be further roughly divided into the following four subcategories: the representative-based, the prototype-based, the centroid-based and the grid-based supervised clustering algorithms.

Representative-based Supervised Clustering Algorithms are applied to find a set of representatives for a dataset [11]. The well known algorithms include Supervised Partitioning Around Medoids (SPAM), Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Restart Algorithm (SRIDHCR) and Supervised Clustering using Evolutionary Computing Algorithm (SCEC) [11,14]. The fitness function is used to measure the performance of these three algorithms.

SPAM consists of two sub-algorithms: *SBUILD* and *SSWAP*. *SBUILD* intelligently selects the initial representatives, *SSWAP* tries to improve the clustering by using a non-representative to replace a representative. The algorithm terminates until no replacement occurs. The number of representatives is pre-defined and cannot be changed during its implementation. To overcome the limitation of SPAM, SRIDHCR [14] allows to change the number of representatives, such as a non-representative example added into the current set of representatives or a representative removed from it. The fitness function that measures the solution quality in SRIDHCR is not only used to maximize the class purity, but also to minimize the number of clusters. When the class purity cannot be improved further, SRIDHCR doesn't stop. But, the number of clusters is reduced by retaining the class purity. Therefore, the number of clusters obtained by SRIDHCR is not fixed. SCEC [14] clusters a dataset by evolving a population of solutions over a pre-defined number of generations. The initial generation is randomly created, and the best solution to the last generation is chosen as the optimal set of data points. Its subsequent generations are created through three different genetic operators: *Mutation*, *Crossover* and *Copy*. The weakness of SCEC is that the number of generations and the size of the population also need to be pre-defined.

In addition, there are still some other algorithms to find representative examples from the dataset. Pan et al. [24] used information-theoretic measures, such as mutual information and relative entropy, to search for good representatives from massive data; Bagherjeiran et al. [25] proposed an adaptive clustering that uses the reinforcement learning to search for good clusters. But none of the aforementioned algorithms are guaranteed to find optimal clusters. Some algorithms need to predefine the number of optimal clusters that usually may not be attainable [26].

Prototype-based Supervised Clustering Algorithms aim to find the prototypes of a dataset, such as Supervised Growing Neural Gas (SGNG) and Robust Supervised Growing Neural Gas (RSGNG) [26].