Neurocomputing 171 (2016) 9–22

Contents lists available at ScienceDirect

Neurocomputing

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

An efficient and scalable density-based clustering algorithm for datasets with complex structures

Yinghua Lv [a], Tinghuai Ma [b,*], Meili Tang [c], Jie Cao [d], Yuan Tian [e], Abdullah Al-Dhelaan [e], Mznah Al-Rodhaan [e]

[a] School of Computer & Software, Nanjing University of information science & Technology, Jiangsu, Nanjing 210-044, China
[b] Jiangsu Engineering Center of Network Monitoring, Nanjing University of information science & Technology, Nanjing 210-044, China
[c] School of Public Administration, Nanjing University of Information Science & Technology, Nanjing 210044, China
[d] School of Economics & Management, Nanjing University of Information Science & Technology, Nanjing 210044, China
[e] Computer Science Department, College of Computer and Information Science, King Saud University, Riyadh 11362, Saudi Arabia

ARTICLE INFO

Article history:
Received 14 December 2014
Received in revised form
23 March 2015
Accepted 5 May 2015
Communicated by Hung-Yuan Chung
Available online 26 June 2015

Keywords:
Density-based clustering
Locality sensitive hashing
The influence space
Border objects detecting

ABSTRACT

As a research branch of data mining, clustering, as an unsupervised learning scheme, focuses on assigning objects in the dataset into several groups, called clusters, without any prior knowledge. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the most widely used clustering algorithms for spatial datasets, which can detect any shapes of clusters and can automatically identify noise points. However, there are several troublesome limitations of DBSCAN: (1) the performance of the algorithm depends on two specified parameters, ε and *MinPts* in which ε represents the maximum radius of a neighborhood from the observing point and *MinPts* means the minimum number of data points contained in such a neighborhood. (2) The time consumption for searching the nearest neighbors of each object is intolerable in the cluster expansion. (3) Selecting different starting points results in quite different consequences. (4) DBSCAN is unable to identify adjacent clusters of various densities. In addition to these restrictions about DBSCAN mentioned above, the identification of border points is often ignored. In our paper, we successfully solve the above problems. Firstly, we improve the traditional locality sensitive hashing method to implement fast query of nearest neighbors. Secondly, several definitions are redefined on the basis of the influence space of each object, which takes the nearest neighbors and the reverse nearest neighbors into account. The influence space is proved to be sensitive to local density changes to successfully reduce the amount of parameters and identify adjacent clusters of different densities. Moreover, this new relationship based on the influence space makes the insensitivity to the ordering of inputting points possible. Finally, a new concept—core density reachable based on the influence space is put forward which aims to distinguish between border objects and noisy objects. Several experiments are performed which demonstrate that the performance of our proposed algorithm is better than the traditional DBSCAN algorithm and the improved algorithm IS-DBSCAN.

1. Introduction

Clustering is the most commonly used and more powerful unsupervised learning technique in data mining. It is a useful process which aims to organize the input dataset into a set of finite number of semantically consistent groups with respect to some similarity measure. Since the early 1950s, a plethora of clustering algorithms have been put forward [1]. These algorithms can be roughly classified into seven groups, namely partitional algorithms, hierarchical algorithms, density-based algorithms, graph-based algorithms, grid-based algorithms, model-based algorithms and combinational algorithms [2,3]. Several issues associated with the use of these clustering techniques are described in [4], emphasizing on some challenges of these algorithms. Among these kinds of algorithms, density-based algorithms are famous for their straightforward rationale and the relative easy implementation. Another two significant advantages of this kind of algorithms are that it is capable of discovering clusters of different shapes and different sizes even in noisy dataset and it does not require users to specify the number of clusters. In density-based algorithms, a cluster is defined as a connected dense component and grows in the

* Corresponding author.
E-mail address: thma@nuist.edu.cn (T. Ma).

http://dx.doi.org/10.1016/j.neucom.2015.05.109
0925-2312/

direction driven by the density. The purpose of density-based algorithms is identifying dense regions that are separated by low-density regions.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) proposed by Ester [5] is the first algorithm that implements the density-based strategy and it is one of the commonly used algorithms. It is popular because of the capability of discovering clusters with arbitrary shapes without any preliminary information about the groups present in a dataset. The rationale of this algorithm is to obtain high density regions as possible clusters ensuring that the density, represented by the number of objects in the neighborhood, exceeds certain specified thresholds. Although there are so many advantages of DBSCAN, there are several distinct drawbacks for DBSCAN. (1) The performance of clustering depends on two specified parameters. One is the maximum radius of a neighborhood from the observing point and the other is the minimum number of data points contained in such a neighborhood. It is difficult to estimate appropriate values of these two parameters for various dataset without any enough prior knowledge. (2) The computational complexity is high when dealing with high dimensional dataset. The complexity of DBSCAN is $O(n^2)$ without utilizing any index structure because of the calculation of the similarity measure between data points, although it only scans through the whole dataset one time. This characteristic directly results in the problems of scalability when the algorithm is applied on large dataset. (3) This algorithm is sensitive to the order of the inputting data. Different orderings of data points in the same dataset result in various consequences. (4) Adjacent clusters of different densities cannot be properly identified maybe due to the use of the global density parameters.

In this paper, in order to achieve a more robust performance, we propose an effective method based on DBSCAN to overcome both the drawbacks mentioned above and the identification of border points. We introduce an improved p-stable locality sensitive hashing algorithm on the basis of the approximate nearest neighbor scheme to reduce the time consumption of the neighbor query in DBSCAN. Usually, several data index structures are used to improve the time complexity of DBSCAN, such as kd-tree [6], R-tree [7], SR-tree [8], and PR-tree [9]. The complexity of DBSCAN can be reduced to $O(n \log n)$ after making use of these indexing methods. However, these index structures are efficient only when the dimension of objects is less than 10. They will break down in practice for high-dimensional data, maybe are slower than the brute-force, linear-scan approach [10]. The improved p-stable locality sensitive hashing algorithm is corroborated to be applicable to high dimensional and large scale data sets with less time consumption.

The problems of difficulty to identify the adjacent clusters of different densities, sensitivity to the input points, dependency on two parameters are all solved by the new relationship called the influence space. We use the influence space (IS) to obtain a better estimation of the neighborhoods density distribution to solve this deficiency. Since IS takes advantage of both the nearest neighbors (NNs) and reverse nearest neighbors (RNNs) which will be explained in detail in the following section, it outperforms other methods to highly sensitive to local density changes. Meanwhile, because this new neighborhood relationship is symmetric, our new proposed method can randomly select any object to start cluster expansion. At the same time, unlike the traditional DBSCAN algorithm, our proposed algorithm requires only one parameter: the number of k-nearest neighbors instead of two parameters.

The identification of border points are based on a new concept called core density reachable. The new concept is based on the characteristic of border points that they do not contribute to the expansion mechanism of density-reachable chains of objects. In our paper, this new concept is put forward based on the influence space while it is firstly raised on the basis of the ε-neighborhood. Experimental results show that border points and noisy points are well distinguished in the final step of our new algorithm.

The rest paper is organized as follows. Section 2 gives a summary of improved strategies about DBSCAN. In Section 3, the traditional DBSCAN is described in detail. In Section 4, we present our improved clustering algorithm called DBSCAN based on Influence Space and Detection of border points (ISB-DBSCAN for short) in detail. Several experimental results are shown to verify the superiority of our proposed algorithm in Section 5. Finally, Section 6 concludes the paper and sketches some future research directions.

2. DBSCAN related works

In this section, we focus on the DBSCAN algorithm under the density-based clustering algorithms. For convenience of understanding the content of this paper, Several approaches proposed previously to extend and improve the DBSCAN algorithm are discussed in detail.

DBSCAN is one of the most widely used algorithms in many applications, such as chemistry [11], spectroscopy [12], social science [13,14], civil engineering [15], anomaly detection [16,17], medical and biomedical image analysis [18]. As a pioneer of density-based clustering algorithms, DBSCAN has the same non-ignorable limitations as the traditional density-based algorithms which have been mentioned above. In term of these limitations, several methods have been proposed to enhance DBSCAN.

The main drawback of DBSCAN is the high computational complexity in the neighborhood query for each object to construct the similarity matrix. DBSCAN can efficiently cluster a low dimensional space while its performance degrades when dealing with high-dimensional and large-scale datasets. Facing this problem, three main strategies are adopted to achieve the computation efficiency: data indexing structures, parallel computing and dividing data sets. Data indexing structures are firstly used to reduce the time complexity $O(n^2)$ to $O(n \log n)$. Kd-tree [6], R-tree (Rectangle-tree) [7] (includes R*-tree [19]), SS-tree [20] and SR-tree (Sphere/Rectangle-tree) [8] are those previously often employed index structures. They are efficient when the dimension of objects is less than 10. Besides, some new index structures are proposed which can be embedding into DBSCAN to achieve the goal of handling with high-dimensional datasets, such as VA-file (vector approximation file) [21], iDistance [22,23], locality sensitive hashing (LSH) [25,25] and iPoc [26].

The second strategy to solve the high computational complexity is parallel computing. Three main methods to realize the parallel computing are distributed memory strategies, efficient processing units and multiprocessor computers [27–29,51]. Chen et al. [30] presented a novel parallel version of DBSCAN in distributed environment, called P-DBSCAN. Dai et al. [31] successfully embedded Hadoop framework into the DBSCAN algorithm to solve the scalability problem. Patwary et al. [32] proposed a new parallel DBSCAN algorithm (PDSDBSCAN) using graph algorithmic concepts. On the one hand, this method achieves a better-balanced workload distribution, on the other hand, the access sequentiality of DBSCAN is overcome by the disjoint-set data structure. G-DBSCAN proposed by Andrade et al. [33] implemented the parallelization of DBSCAN using graphics processing units. Although the time complexity of this algorithm is also $O(n^2)$ which is higher than DBSCAN with data index structures, this algorithm achieves a higher speedup with regard to the execution time.

The third solving strategy to reduce the time consumption is firstly dividing the large dataset into several small partitions, and afterwards applying DBSCAN on each partition. K-means [34] and