VOL. 10, NO. 9, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

AN EMPIRICAL RESEARCH OF DYNAMIC CLUSTERING ALGORITHMS

S. Angel Latha Mary, D. Sivaganesan and R. Vinothkumar
Department of Computer Science Engineering, Karpagam College of Engineering, Coimbatore, Tamilnadu, India
E-Mail: xavierangellatha@gmail.com

ABSTRACT

Clustering and visualizing high dimensional dynamic data is a challenging problem in the data mining. Most of the existing clustering algorithms are based on the static statistical relationship among data. In the clustering process there are no predefined classes and no examples that would show what kind of desirable relations should be valid among the data. This paper gives existing work done in some papers related with dynamic clustering and incremental data clustering. Since most researchers will move and concentrate on solving the problem of using data mining dynamic databases.

Keywords: dynamic clustering, dynamic clustering algorithms, incremental clustering.

1. INTRODUCTION

A cluster is a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering is an important data mining technique used to find data segmentation and pattern information. It can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabelled data. Recently, the data are growing with unpredictable rate. Discovering knowledge in these data is a very expensive operation and difficult [1]. Most of the clustering algorithms are based on the static relationship among data. In the clustering process there are no predefined classes and no examples that would show what kind of desirable relations should be valid among the data [2]. The databases will dynamically change due to frequent insertions and deletions which changes clustering structure over time. Completely reapplying the clustering algorithm to detect the changes in the clustering structure and update the uncovered data patterns following such deletions and insertions is very expensive for large high dimensional fast changing dataset which increase the space and time complexity of of algorithms. Dynamic clustering is a mechanism to adopt and discover clusters where a data set is updated periodically through insertions and deletions [3]. Dynamic clustering is very useful to obtain high quality results in the field of time series analysis, telecommunications, mobile networking, nanotechnology, physics, chemistry, biology, health care, sociology and economics [4]. When there is a continuous update and huge amount of dynamic data, rescan and recluster the database is not possible. But this is possible in dynamic clustering [5].

The rest of the paper is organized as follows. Section 2 describes the related works in dynamic clustering algorithms. Section 3 presents the different dynamic clustering algorithms. Finally conclusions are drawn in section 4.

2. RELATED WORKS

Due to the continuous, unbounded, and high speed characteristics of dynamic data, there is a huge amount of data and there is not enough time to rescan the whole database or perform a rescan as in traditional data mining algorithms whenever an update occurs [6]. Ganti *et al.* 2002 examine mining of data streams. A block evolution model is introduced where a data set is updated periodically through insertions and deletions. In this model the data set consists of conceptually infinite sequence of data blocks D1, D2, ... that arrive at times 1, 2, ... where each block has a set of records. The authors highlight two challenges in mining evolving blocks of data, Initially change detection and data mining model maintenance. In change detection, the differences between two data blocks are determined. Next, a data mining model should be maintained under the insertions and deletions of blocks of the data according to a specified data span and block selection sequence [7].

Crespoa and Weberb presented a methodology for dynamic data mining using fuzzy clustering that assigns static objects to dynamic classes. Changes that they have studied are movement, creation and elimination of classes and any of their combination. Once a data mining system is installed and is being used in daily operations, the user has to be concerned with the system's future performance because the extracted knowledge is based on past behavior of the analyzed objects If future performance is very similar to past performance (e.g. if company customers files do not change their files over time) using the initial data mining system could be justified. If, however, performance changes over time (e.g. if hospital patients do not change their files over time), the continued use of the early system could lead to an unsuitable results and (as an effect) to an unacceptable decisions based on these results. Here dynamic data mining could be extremely helpful in making the right decision in the right time and affects the efficiency of the decision [5].

There are three strategies when user is applying data mining system in a changing environment.

- The user can neglect changes in the environment and keep on the initial system without any further updates. It is computationally cheap since no update to data mining system is performed and also it does not

require changes in subsequent processes. But current updates could not be detected.

- Depending on the application, a new system is developed using available data over a period of time. The user has always a system up-to-date due to the use of current data. This strategy increases the computational costs of creating a new system every time from scratch.
- Based on the initial system and "new data" an update of data is performed.

Chung and Mcleod proposed mining framework that supports the identification of useful patterns based on incremental data clustering, they focused their attention on news stream mining, and they presented a sophisticated incremental hierarchical document clustering algorithm using a neighborhood search [8].

In many situations, new information is more important than old information, such as in publication database, stock transactions, grocery markets, or web-log records. Consequently, a frequent itemset in the dynamic database is also important even if it is infrequent in the updated database.

Incremental clustering is the process of updating an existing set of clusters incrementally rather than mining them from the scratch on each database update. COBWEB was proposed by Fisher. It is an incremental clustering algorithm that builds taxonomy of clusters without having a pre-defined number of clusters [9]. Gennary *et al.* proposed CLASSIT which associates normal distributions with cluster nodes. The main drawback of both COBWEB and CLASSIT is that they results in highly unbalanced trees [10].

Charikar *et al.* introduced new deterministic and randomized incremental clustering algorithms while trying to minimize the maximum diameters of the clusters. The diameter of a cluster is its maximum distance among its points and is used in the restructuring process of the clusters. When a new point arrives, it is either assigned to one of the current clusters or it initializes its own cluster while two existing clusters are combined into one[11]. Ester *et al.* presented Incremental DBSCAN suitable for mining in a data warehousing environment. Incremental DBSCAN is based on the DBSCAN algorithm which is a density based clustering algorithm. It uses R* Tree as an index structure to perform region queries. Due to its density based qualities, in Incremental DBSCAN the effects of inserting and deleting objects are limited only to the neighborhood of these objects. Incremental DBSCAN requires only a distance function and is applicable to any data set from a metric space. However, the proposed method does not address the problem of changing point densities over time, which would require adapting the input parameters for Incremental DBSCAN over time [12]. Another limitation of the algorithm is that it adds or deletes one data point at a time. An incremental clustering algorithm based on SWARM intelligence is given in Chen and Meng [13].

3. DYNAMIC CLUSTERING ALGORITHMS

Dynamic FClust Algorithm summarizes the author's initial work in designing a simultaneous clustering and visualization algorithm [2] which is based on flocks of agents as a biological metaphor. This algorithm falls within the swarm based clustering family, which is unique compared to other approaches, because its model is an ongoing swarm of agents that socially interact with each other and is therefore inherently dynamic. Yucheng Kao, Szu-Yuan Lee presented new dynamic data clustering algorithm based on K-Means and combinatorial particle swarm optimization, called KCPSO. Unlike the traditional K-Means method, KCPSO does not need a specific number of clusters given before performing the clustering process and is able to find the optimal number of clusters during the clustering process. In each iteration of KCPSO, a discrete PSO(Particle Swarm Optimization) is used to optimize the number of clusters with which the K-Means is used to find the best clustering result. KCPSO has been developed into a software system and evaluated by testing some datasets. Encouraging results show that KCPSO is an effective algorithm for solving dynamic clustering problems [14].

CHAMELEON that measures the similarity of two clusters based on a dynamic model. In the clustering process, two clusters are merged only if the inter-connectivity and closeness between two clusters are comparable to the internal inter-connectivity of the clusters and closeness of items within the clusters. But this algorithm very much affected by the size of the dataset [15].

All these existing clustering algorithms use static dataset. These algorithms can give incorrect results if the choice of parameters in the static model is incorrect with respect to the data set being clustered, or the data consists of clusters that are of diverse shapes, densities, and sizes.

Elghazel Haytham, gives a dynamic version for the b-coloring based clustering approach which relies only on dissimilarity matrix and cluster dominating vertices in order to cluster new data as they are added to the data collection or to rearrange a partition when an existing data is removed. A real advantage of this method is that it performs a dynamic classification that correctly satisfies the b-coloring properties and the clustering performances in terms of quality and runtime, when the number of clusters is not pre-defined and without any exception on the type of data [16].

Ester presents an incremental clustering algorithm based on the clustering algorithm DBSCAN for mining in a data warehousing environment which is applicable to any database containing data from a metric space, e.g., to a spatial database or to a WWW-log database. Due to the density based nature of DBSCAN, the insertion or deletion of an object affects the current clustering only in the neighbourhood of that object. Thus efficient algorithms could be given for incremental insertions and deletions to an existing clustering [17]. Based on the formal definition of clusters, this incremental algorithm yields the same result as DBSCAN.

Incremental DBSCAN yields significant speed-up factors over DBSCAN even for large numbers of daily updates in a data warehouse. The authors were assumed that the parameter values *Eps* and *MinPts* of incremental DBSCAN did not change significantly when inserting and deleting objects**.**

4. CONCLUSIONS

It is observed that, there are some drawbacks of the above dynamic clustering algorithms. They are able to insert data objects one by one and then re-estimate the cluster IDs during every point, which is inserted and capable of creating, modifying and inserting clusters over time. It considered problems related with data insertion. But the important point is, during each step of insertion it does not considers the data points which are classified as noise(outliers) or border objects. These data points can be considered again as unclassified points and can be combined with the new data which is to be inserted. Also these algorithms are not able to handle batch insertion, which can reduce the run time of the algorithms.

REFERENCES

[1] Elena N. Benderskaya and Sofya V. Zhukova. 2010. "Dynamic Data Mining: Synergy of Bio-Inspired Clustering Methods", Knowledge-Oriented Applications in Data Mining.

[2] Saka E. and Nasraoui O. 2010. "On dynamic data clustering and visualization using swarm intelligence", In Data Engineering Workshops (ICDEW), 26th International Conference on IEEE, pp. 337-340.

[3] Crespoa F. and Weber R. 2005. "A methodology for dynamic data mining based on fuzzy clustering", Fuzzy Sets and Systems, Elsevier, Vol. 150, pp. 267-284.

[4] Xu R., Wunsh D. 2010. II, "Survey of clustering algorithms", IEEE Trans. on Neural Networks, Vol. 15, pp. 645-678.

[5] Zhang J., Tianrui Li, Da Ruan and Dun Liu. 2012. "Neighborhood Rough Sets for Dynamic Data Mining", International Journal of Intelligent Systems, Wiley Periodicals, Inc., Vol. 27, pp. 317-342.

[6] Hebah, H.O. Naseredd. 2009. "Stream Data Mining", in International Journal of Web Applications, Vol. 1, No. 4, pp. 183-190.

[7] Ganti V., Gehrke J., Ramakrishnan R. and Loh W. 2002. "Mining data streams under block evolution", In ACM SIGKDD Explorations, Vol. 3, no. 2, pp. 1-10.

[8] Chung S. and Dennis Mcleod. "Dynamic Pattern Mining: An Incremental Data Clustering Approach", www.sigmod.org/ dblp/db/indices/a-tree/m/McLeod:Dennis.html.

[9] Fisher. 1987. "Knowledge acquisition via incremental conceptual clustering", Machine Learning, Vol. 2, pp. 139-172.

[10] Gennary J., Langley P. and Fisher D. 1989. "Model of Incremental Concept Formation", Artificial Intelligence Journal, Vol. 40, pp. 11-61.

[11] Charikar M., Chekuri C., Feder T. and Motwani R. 1997. "Incremental clustering and dynamic information retrieval", 29th Symposium on Theory of Computing, pp. 626-635.

[12] Ester M. and Wittmann R. 1998. "Incremental Generalization for Mining in a Data Warehousing Environment", Proc. 6th Int. Conf. on Extending Database Technology, Valencia, Spain, 1998, in: Lecture Notes in Computer Science, Springer, Vol. 1377, pp. 135-152.

[13] Chen Z. and Meng Q.C. 2004. "An incremental clustering algorithm based on SWARM intelligence theory", Proc. of the 3rd Int. Conf. on Machine Learning and Cybernetics, Shanghai, 26-29 August.

[14] Yucheng Kao and Szu-Yuan Lee. 2009. "Combining K-Means and particle swarm optimization for dynamic data clustering problems", This paper appears in: Intelligent Computing and Intelligent Systems, ICIS 2009, IEEE International Conference on, Vol. 1, pp. 757-761.

[15] Karypis G., Eui-Hong (Sam) Han and VipinKumar. 1999. "Chameleon: Hierarchical Clustering Using Dynamic Modeling", IEEE Computers.

[16] Elghazel Haytham, Hamamache Kheddouci, Véronique Deslandres and Alain Dussauchoy. 2007. "A Partially Dynamic Clustering Algorithm for Data Insertion and Removal", Discovery Science ,Lecture Notes in Computer Science, Vol. 4755, pp. 78-90.

[17] Ester M., Kriegel H.-P., Sander J. and Xu X. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland.