ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

A Profitable Macro Cluster Generation Utilizing Top-K Shared Micro Clusters in Information Streams

Nagamalleswararao Tadikonda
Department of Computer Science,
Programmer Analyst, Trans IT mpower labs pvt.ltd,
Bangalore, Karnataka-560029, India.

Abstract:
Now-a-days data streams or information streams are gigantic and quick changing. The usage of information streams can fluctuate from basic logical, scientific applications to vital business and money related ones. The useful information is abstracted from the stream and represented in the form of micro-clusters in the online phase. In offline phase micro-clusters are merged to form the macro clusters. DBSTREAM technique captures the density between micro-clusters by means of a shared density graph in the online phase. The density data in this graph is then used in reclustering for improving the formation of clusters but DBSTREAM takes more time in handling the corrupted data points In this paper an early pruning algorithm is used before pre-processing of information and a bloom filter is used for recognizing the corrupted information. Our experiments on real time datasets shows that using this approach improves the efficiency of macro-clusters by 90% and increases the generation of more number of micro-clusters within in a short time.

Index:
Data Stream Clustering, Density based Clustering.

1. I INTRODUCTION:

Clustering is a standard or imperative system of exploratory information mining, which isolates an arrangement of information into a few gatherings (additionally called clusters) such that items in same gathering are more comparable with each other in some sense than with the items in different gatherings. Data streams are the continuous flow of data and its size has no bounds [2][10]. Many applications produce this type of streaming data like GPS data from vehicles, web click stream data, computer network monitoring, readings from sensors etc. Data stream clustering is done for better understanding of data. Cluster algorithms and their parameter settings depend on the individual data sets. Data stream clustering algorithms process the data quickly by providing timely results, detects whether new clusters should appear or disappear and also identifies the outliers. Clustering of data streams can be done by using grid based algorithms like D-Stream [1] or density based algorithms like DBSTREAM [2] or partitioning based algorithms like k-means. The main or primary goal of this paper is to improve the quality of final clusters and to reduce the time in generating the micro-clusters.

2. RELATED WORK:

In the application point of view one-pass clustering algorithms are not useful as the outdated data makes the cluster quality poor. CluStream is an effective and efficient method characterizes the data streams in different time horizons. The micro-clusters are stored as snapshots in pyramidal time window [5]. But cannot find arbitrary shaped clusters and cannot handle outliers.[6]. Density based clustering algorithm, DBSCAN is used to find the clusters of arbitrary shapes in large spatial Databases with noise and it requires only one input parameter. It counts the number of data points and estimates its density by using eps, midpoints parameters and identifies the core, border and noise points.[3].

Cite this article as: Nagamalleswararao Tadikonda, "A Profitable Macro Cluster Generation Utilizing Top-K Shared Micro Clusters in Information Streams", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 4 Issue 11, 2017, Page 140-144.

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

The disadvantage is it cannot process huge amount of data. DenStream also discovers clusters of arbitrary shapes and it is insensitive to noise[7]. Required information for forming clusters is provided by core micro-clusters and outlier micro-clusters. The major drawback is the computational cost is more [8]. D-Stream is grid based clustering method. In the online phase each data record is mapped to a grid. It uses density decaying technique for adjusting the real time data streams and in the offline phase clusters the grids based on their density [1].For removing the inactive grids a sound technique is used due to this limited space and time are sufficient. D-Stream with attraction is the extension of D-Stream [1] for improving the quality of clusters and for generating clusters accurately [9]. It is used for high-speed data streams. LeaDen-Stream algorithm improves the cluster quality at the same time decreases the time complexity by using mini-micro and micro-leader clusters.[4]. Based on the distribution of the data in the micro leader clusters the data points are determined for offline clustering. It performs better than DenStream and CluStream. DBSTREAM is the online component captures the shared region between micro-clusters using the shared density graph. It improves the quality of clusters compared to the above data stream clustering methods. [2]

3. EXISTING SYSTEM:

In the existing system DBSCAN (Density based Spatial Clustering of Applications with Noise) is used for identifying the noise in the spatial databases. Eps, minimum point's parameter values are to be given by the user. These parameters are helpful in the formation of micro-clusters and in identifying the core, border and noise points[3]. In the offline stage micro-clusters are reclustered based on the low density estimates between the micro-clusters resulting in the improper formation of clusters. The drawback is DBSCAN does not store the data points in the overlap area of micro-clusters in the online phase and it is not applicable for high dimensional data. DBSTREAM (Density based STREAM clustering) is the first micro-cluster-based online clustering component explicitly captures the density between micro-clusters via a shared density graph in the online process. The density information in this graph is then used for reclustering based on actual density between adjacent micro-clusters. In the micro-clusters formation leader based clustering algorithm is used in which DBSTREAM represents each MC by a leader which is a data point defining the MC's center and the density in an area of a user-specified radius r i.e. threshold around the center[4]. This is similar to DBSCAN's concept of counting the points is an eps-neighbourhood, however, here the density is not estimated for each point, but only for each MC which can easily be achieved for streaming data. Disadvantages are when processing huge amount of data corruption of data points occurs and DBSTREAM takes time in handling the malformed request and it affects the remaining data points resulting in the formation of inaccurate clusters.

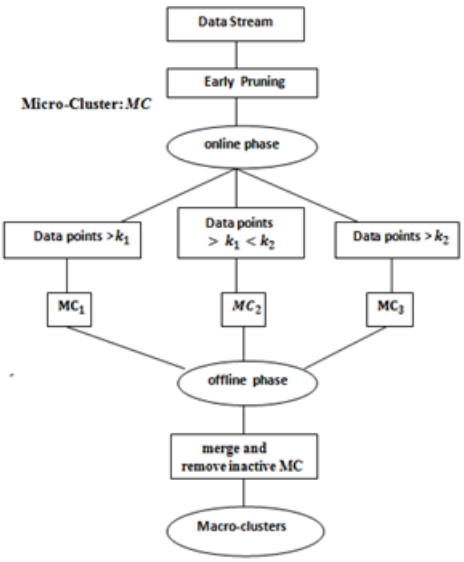

Figure1: Data flow in Data Stream Clustering

4. PROPOSED SYSTEM:

In this paper, we used an early pruning algorithm for removing the corrupted or invalid data points before pre-processing the data stream. Bloom filter is a probabilistic data structure used for finding the corrupted data points in less memory space. Bloom filter checks the input data stream if the data points belongs to the set it allows the data for the formation of micro-clusters. If the data point does not belong to the set then it is pruned or removed.

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

After pre-processing the data stream micro-cluster formation is done using DBSTREAM online component and shared density graph.[2]. In reclustering the density information collected from the shared density graph is helpful in merging the micro-clusters resulting in the formation of accurate macro or final clusters.

Algorithm 1.EarlyPruning (j, pil, pit)

Require: Bloom filter, bf
Input: Create S, T
index of 'S', j;
positional index, pil;
candidate positional index, pit for T;
1: int $index_j = [\log_2 PC_j]$
//($PC_j = \max\{C_k[j]\ (1 \le k \le m)\}$)
2: if $pil \le RC_j$ then
3: return false
//($RC_j = \min\{C_k[j]\ (1 \le k \le m)\}$)
4: end if
5: for k = 1 to m do
6: if j = = k then
7: continue
8: end if
9: boolean inflag = testInBF ($EGBFT_K$ ($index_j$), pit)
10: if (inflag) then
11: return false
12: end if
13: end for
14: return true

Functioning of Early Pruning Algorithm:

Pruning is a procedure of removing the unnecessary part of the data. Pruning is of two types' early pruning and late pruning. Early pruning is applied on the initial data that is before processing whereas late pruning is applied to the data after processing. In this paper we are using an early pruning algorithm. Data streams are stored initially in the created 'S' list. Bloom filter bf, is constructed which is a space efficent data structure initially sets all its values to '0' and sets the maximum and minimum index values. Hash set 'T' is created which contains hash values.

Positional index list, pil and candidate positional index list, pit are checked by using test In BF ($EGBFT_K$ ($index_j$), pit).Exponential Gap Bloom Filter Table stores the hash values and its size depends on the tuple number If positional index, pit belongs to the bloom filter it returns flase as there is no need to remove the data point and changes the bit '0' to '1'.If it returns true the data point is pruned. At the termination stage the resulted data points from the bloom filter are used to form the micro-cluster.

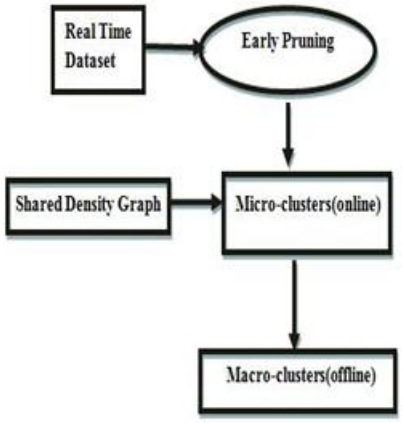

Figure 2: Schematic view of proposed approach

A. Real Time Dataset:

In this paper, we considered the GPS data of vehicles in the real time environment. The information generated is automatically converted into datasets with attributes like proximity, vehicle name, velocity, speed, log data etc. The records in the dataset are pre-processed before generating the micro-clusters.

B. Early Pruning:

Our proposed algorithm is applied on generated data. In this process, bloom filters identifies invalid data points and pruning is applied on them. The valid data points are considered for generation of micro clusters.

C. Micro-clusters:

After pruning the data points are grouped together to form the micro-clusters using DBSTREAM technique. The density in the overlapping region of micro-clusters is captured using shared density graph.

ISSN No: 2348-4845
International Conference on Advanced Computer Science & Software Engineering (ICACSSE)
November 12, 2017 - Hyderabad, India
Paper Published in IJMETMR, A Peer Reviewed Open Access

As invalid data points are removed in the pre-processing phase accurate micro-clusters will be formed.

D. Macro-clusters:

Final clusters or macro-clusters are created in offline phase by merging the micro-clusters based on shared density graph information resulting in the improved formation of final clusters and the inactive macro-clusters are removed using the fading factor.

5. RESULTS AND DESCRIPTION:

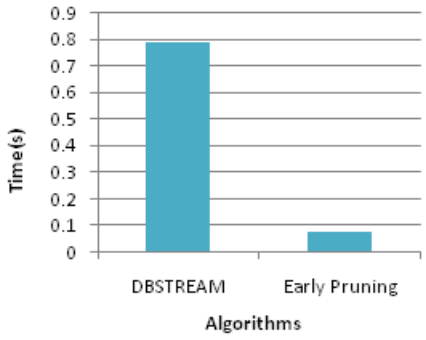

Figure 3: Performance comparison between Existing and Proposed algorithms

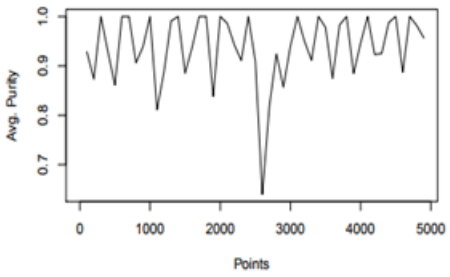

Figure 4 : Micro-cluster purity over an evolving data stream

Points	Purity
0	NA
1000	0.929
2000	0.873
3000	0.69
4000	0.929
5000	0.861

Table1: Purity reduces with increase in the number of data points

As illustrated in figure3 the time complexity for early pruning is less compared with DBSTREAM as it takes more time in handling the malformed request. Early pruning improves the cluster quality and also it takes less memory for processing the dynamic data. From figure4 we considered a new and larger set of 5000 evaluation data points from the stream. Purity of the micro-clusters is high since each micro-cluster only covers points from the same true cluster. However, the corrected index is low because several micro-clusters split the points from each true cluster. To evaluate how well a clustering algorithm can adapt to an evolving data stream, stream provides evaluate_cluster() for performing prequential evaluation within a given horizon. The first row in the results table contains NA in the purity measure. This is the case since we started evaluation with a new, empty clustering and purity values decreases with increase in the data points.

6. CONCLUSION:

Clustering data streams puts extra requirements on clustering algorithms. Data streams expect algorithms to influence a solitary disregard the data with limited memory and restricted preparing to time, while the stream might be exceptionally unique and changes after some time. A few clustering algorithms are presented for data streams that are separate based and can't deal with the entwined groups. In this paper, we considered the real-time data for data stream clustering and applied early pruning algorithm for identifying and removing the corrupted data points. DBSTREAM captures the shared density between micro-clusters formed by valid data points in the online phase and uses this information in reclustering. The advantage of using pruning algorithm resulted in improving performance, cost reduction, increase in the generation of micro-clusters in less time and formation of efficient macro-clusters.

REFERENCES:

[1] Y. Chen and L. Tu, “Density-based clustering for real-time stream data,” in Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2007, pp. 133–142.

[2] Michael Hahsler, "Clustering Data Streams Based on Shared Density between Micro-Clusters" in IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 6, June2016.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 1996, pp. 226–231.

[4] A. Amini and T. Y. Wah, "Leaden-stream: A leader density-based clustering algorithm over evolving data stream," J. Comput. Commun., vol. 1, no. 5, pp. 26–31, 2013.

[5] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in Proc. Int. Conf. Very Large Data Bases, 2003, pp. 81–92.

[6] Maryam Mousavi, Azuraliza Abu Bakar and Mohammadmahdi Vakilian "Data Stream Clustering Algorithms: A Review", Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015 ISSN 2074-8523.

[7] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in Proc. SIAM Int. Conf. Data Mining, 2006, pp. 328–339.

[8]]J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A.C.P.L.F.d.Carvalho, and J. A. Gama, "Data stream clustering: A survey," ACM Comput. Surveys, vol. 46, no. 1, pp. 13:1–13:31, Jul. 2013.

[9] L. Tu and Y. Chen, "Stream data clustering based on grid density and attraction," ACM Trans. Knowl. Discovery from Data, vol. 3, no. 3, pp. 1–27, 2009.