Load Shedding using Transactional Algorithm in Data Stream

RujutaVartak
Department of Computer Engineering,
Mumbai University,
Shree L.R. Tiwari College of Engineering and Technology,Mira Road, India
rujutavvartak@gmail.com

Dr.MadhuriRao
Department of Information Technology,
Mumbai University,
Thadomal College of Engineering and Technology,Mumbai, India
my_rao@yahoo.com

Prof. Jayamalini k
Department of Computer Engineering,
Mumbai University,
Shree L.R. Tiwari College of Engineering and Technology,Mira Road, India
jayamalini.k@slrtce.in

Abstract- Many Companies and organization have huge database,that grow to the limit of millions of records per day. Data stream refer to the data that flow into system in vast volumes. The most important difficulty while processing the data stream is handling overloaded continuous data. This paper studies Transactional mining Algorithm that handle overloaded continuous data and this method is known as load shedding. Using load shedding continuous and high speed data stream can easily consume system resources.

Keywords: Data stream; Load shedding

*****

I. INTRODUCTION

There are many commercial field presented their data in the form of continues emitted stream, namely data streams. A stream data occur continuously with high speed and have a data distribution that often changes with time. Hence there is need to load shedding using an efficient frequent pattern mining algorithm.Load shedding refers to the process of eliminating a group of subsequent data streams during periods of overload.

During peak period, the data transfer rate of a stream exceeds the data processing rate of system is overload and unable to handle all received data elements properly in a time unit. Further, an overloaded system may work abnormally or even come into a crash. Accordingly, it is essential for a data-stream mining system to adequately deal with data overload.

Frequent patterns meansthat appear frequently in a data set. Frequent pattern mining find outreoccurring relationships in a given data set. It plays vital role in mining associations and correlation analysis among data, is very important data mining task. This work focuses on discovering frequent item sets in data stream which may suffer from data overload problem.

The rest of this paper is organized as follows. In Section 2, related work regarding data-stream mining and load shedding is described. Section 3 introduces data stream approaches In Section 4, the load-shedding mining system Architecture is illustrated. Section 5 shows experimentalresults with simple analyses. Finally, Section 6 concludes this research.

II. RELATED WORK

Demonstrated that graph-oriented approach is more efficient, for load-shedding semantic data stream by [1].Indeed, this approach allows improving the quality of queries results of these systems by protecting the semantics and consistency of the semantic data streams.

Proposed load adaptation techniques are designed considering various conditions of nodes and tasks [2], and classified as inter-node load balancing, inter-task load balancing, increasing processing capability, and finally load shedding for the last way to avoid entire system halt due to the data explosion.

Heuristics Miner algorithm is proposed, called StrProM [3], builds prefix–trees to extract sequential patterns of events from the stream. This algorithm uses batch based approach to continuously update and prune these prefix -trees.

Proposed an efficient algorithm DSM-Miner for mining maximal frequent patterns over data streams [4]. Sliding Window Maximum frequent pattern Tree SWM-Tree can effectively maintain the latest pattern’s information that users need in data stream. Also perform mining maximal frequent and appropriate pruning operations.INSTANT algorithm, based on operators on itemsets, intended todiscover frequent itemsets from data streams [7]. INSTANThas good time and space efficiencies both with increases of the data stream scale and decreases of the minimum support value.

Using Lossy counting algorithm load shedding schema is proposed for mining system [5]. The load shedding schema is compatible with the mining algorithm on the same data structure and the cost of switch between both of them is found to be small. By running the scheme (in place of its own mining algorithm) when necessary, the mining system can obtain an increase in its throughput and thus protect itself from the overload.

Using Loadstar dynamically adjusts to changes in available system resources[6].This proves that an intelligent load shedding scheme can make a stream management system more robust to external disturbances.

Novel algorithms for computing approximate frequency counts of elements in a data stream [8]. Solve iceberg queries, frequent itemsets, association rules, and packet flow identification.

III. DATA STREAM MINING APPROACH

Research problems and challenges that have been arisen in mining data streams can be solved to some extent using well established statistical and computational approaches. Here main focus is data and the task to be performed on data. When data is the key focus, idea is to examine only a subset of the whole dataset or to transform the data vertically or horizontally to an approximate smaller size data representation. Sampling, load shedding and sketching techniques etc., are some of the data based approaches. We use load shedding technique.Load Shedding: As we know data streams are processes which create large volumes of incoming data, they lead to several challenges in both processing the data as well as applying traditional database operations. For example, when the incoming rate of the data streams is higher than that processing capacity of the system, techniques are required in order to selectively pick data points from the stream, without losing accuracy.

IV. SYSTEM ARCHITECTURE

System work flows are shows in the following architecture diagram.

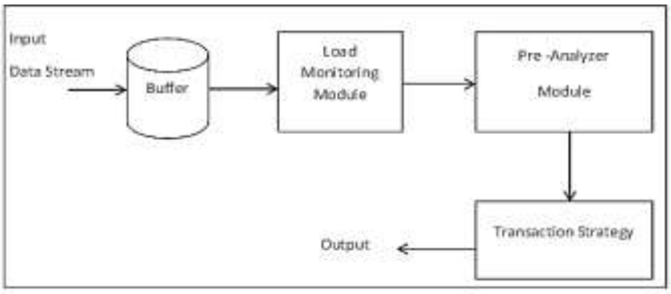


Fig.1. System architecture

Load Monitoringdealswithmonitoringtheinputdatastream,which issendtoPre-Analyzer. The inputfilecanbe JSON type. Input sends to the buffer. The Pre-Analyzerhelpsbyprovidingperformancemeasuresfor transactionstrategy.

Theworkingofthestrategyisexplainedbelow.

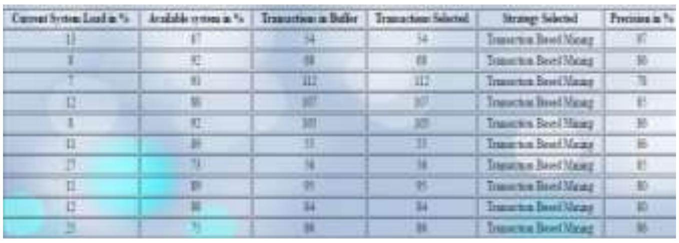
A. *Transaction-Based*

Intheacceptedandrawitemsets,thetransactionbasedalgorithm discarded individual items,fromthe individual item setdependingonthesystemload,i.e.,ifthesystemcanallow20itemsfromanitemset,whichisof23items,thetransactionbasedwouldremovethelast3itemsfromtheitemset. Theseremoved items from theitem setsare then back tothe bufferfornext processingcycle.

B.Algorithm of Transaction based strategy

Step1: JSON file is used to store the data stream.
Step2: This data stream is given as input to load monitoring module and stored in the buffer.
Step3: The load monitoring module calculates the system load.
Step4: Depending on the system load, a set of transaction is sent to transaction strategy module. This set of transaction is stored in an array (L).
Step5: The support count of all the reoccurring items are calculated for transaction based strategy.
Step6: The step 5 is repeated for all the items in the item sets.
Step7: Since, pruning is performed on the item sets with lowest support count. hence, items in L are sorted in increasing order of support count.
Step8: Create duplicate of all item in the item sets. The clone of these item sets are created for sending to the buffer. The cloning operation is carried out for those item sets, which are pruned.
1) Step9: Send these pruned items back to buffer for next processing cycle.
Step10: The load monitoring module checks the system load. Depending on the system load, the module checks, if the system can process more transactions, then got to step 5, else stop.
Step11: Depending on number of transaction selected, the transaction strategy module calculates the precision value.

V. EXPERIMENTAL RESULT



Current System Load in %	Available system in %	Transactions in Buffer	Transactions Selected	Strategy Selected	Precision in %
13	87	54	54	Transaction Based Mining	87
8	92	68	68	Transaction Based Mining	90
7	93	112	112	Transaction Based Mining	78
12	88	107	107	Transaction Based Mining	87
8	92	103	103	Transaction Based Mining	90
11	89	77	77	Transaction Based Mining	86
27	73	58	58	Transaction Based Mining	87
11	89	95	95	Transaction Based Mining	80
12	88	84	84	Transaction Based Mining	80
25	75	88	88	Transaction Based Mining	86

Fig 2: Load balancing for stream mining

In this table calculate current system load,it depends upon that particular system. Thenbasis of current system load calculate percentage of available system load for transaction. Then depending upon number of transaction percentage of precision value is calculated.Precision value is defined as the ratio of number of transactions retained to the number of transactions the system can process.Using this transaction strategy data is controlled in overload situation.

VI. CONCLUSION

The load-shedding schemes can indeed lighten the workload of an overloaded system as well as preserve the mining accuracy at an acceptable level. User can make choice depending on the mining and load administrable is proposed. Moreover, Transaction load shedding strategyis constructed and unified in the system based on the various concepts.

REFERENCES

[1] Fethi Belghaouti, Amel Bouzeghoub, Zakia Kazi Aoul, Raja Chiky, "Graph-Oriented Load-Shedding For Semantic Data Stream Processing",ComputationalIntelligence for Multimedia Understanding (IWCIM), October 2015.

[2] Myungcheol Lee, Miyoung Lee, Sung Jin Hur, Ikkyun Kim, "Load Adaptive Distributed Stream Processing System for Explosive Stream Data", ICACT ,pp.753-757, 2015.

[3] Marwan Hassani, Sergio Siccha, Florian Richterand Thomas Seidl, "Efficient Process Discovery From Event Streams Using Sequential Pattern Mining" IEEE Computational Intelligence, IEEE Symposium Series ,pp.1366-1373,2015.

[4] Junrui Yang , Yanjun Wei , Fenfen Zhou, "An efficient algorithm for mining maximal frequent patterns over data streams" 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, IEEE, pp.444-447,2015.

[5] Kuen-Fang Jea, Chao-Wei Li, Chih-Wei Hsu, Ru-PingLin, Ssu-Fan Yen, "A Load Shedding Scheme for Frequent Pattern Mining in Transactional Data Streams" , IEEE Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp.1294-1299,2011.

[6] Y. Chi, H. Wang, and P. S. Yu, "Loadstar: Load Shedding in Data Stream Mining," Proceedings of the 31st VLDB Conference, Trondheim, Norway, pp.1302-1305, 2005.

[7] G Mao, X Wu, X Zhu, et al, "Mining maximal frequent itemsets form data streams," Journal of Information Science, vol. 33(3), pp. 251-262, 2007.

[8] G. S.Manku and R. Motwani, "Approximate Frequency Counts over Data Streams," Proceedings of the 28th VLDB Conference, Hong Kong, pp. 346-357, China, 2002.