International Journal of Computer Applications Technology and Research
Volume 8–Issue 05, 176-178, 2019, ISSN:-2319–8656

Frequent Itemset in Sequential Pattern Matching Using Bigdata

M. Ilakkiya
M.Tech Computer Science
PRIST University
Thanjavur, India

D. Vinotha
Head and Assistant Professor, Department of CSE
PRIST University
Thanjavur, India

Abstract: A novel frequent item set mining algorithm, namely Horizontal parallel-Apriori (HP-Apriori), is proposed that divides database both horizontally and vertically with partitioning mining process into four sub-processes so that all four tasks are performed in parallel way. In addition, the HPApriori tries to speed up the mining process by an index file that is generated in the first step of algorithm. The proposed algorithm has been compared with Count Distribution (CD) in terms of execution time and speedup criteria on the four real datasets. Experimental results demonstrated that the HPApriori outperforms over CD in terms of minimizing execution time and maximizing speedup in high scalability. We deal with the problem of detecting frequent items in a stream under the constraint that items are weighted, and recent items must be weighted more than older ones. This kind of problem naturally arises in a wide class of applications in which recent data is considered more useful and valuable with regard to older, stale data. The weight assigned to an item is, therefore, a function of its arrival timestamp.

Keywords: Horizontal parallel-Apriori (HP-Apriori), Count Distribution (CD).

1. INTRODUCTION

With the fast development of networking, data storage, and the data collection capacity, the size of databases is rapidly growing in all domains. According to a report from International Data Corporation (IDC), in 2011, the overall created and replicated data volume in the world was 1.8ZB (≈ 1021B), which increased by nearly nine times within five years. Big data typically includes masses of unstructured data that requires more real-time analysis. Big data analytic by machine learning and data mining techniques has become an important research problem. Mining with big data is very difficult problem when the current data mining methodologies tools with a single personal computer are used to deal with very large datasets due to their large size and complexity. Big Data has special characteristics that make it an extreme challenge for discovering useful knowledge. These characteristics including, large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data. The Association Rule Mining (ARM) finds regularities in the transaction data that can lead to develop and implement better strategic business decisions. The ARM contains two phases, (1) mining the Frequent Item sets (FIs), and (2) Association Rule (AR) extraction. The Apriori algorithm is considered as one of the most influential data mining algorithms to extract knowledge in the forms of ARs or FIs. Others well-known AR mining algorithms include FPGrowth, Eclat, and D-CLUB. When the size of datasets becomes very large, the traditional ARM algorithms cannot deliver the results in a reasonable amount of time. Especially the Apriori algorithm that uses a breadth-first search to count the support count of item sets; therefore, it is far less efficient than the other algorithms. In the ARM from big data, the complexity of FI mining is more than the AR extraction. This is due to the fact the one database scanning is needed to compute the support count of each item set; on the other hand, the calculating the confidence of the strong ARs is performed simply. Therefore, frequent item set mining step requires very more execution time than AR mining step. The parallel paradigm can be considered to tackle this big data problem in an efficient way.

2. METHODS AND MATERIAL

Literature Survey

Data Mining With Big Data

Big Data concern large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences. This paper presents a HACE theorem that characterizes the features of the Big Data revolution, and proposes a Big Data processing model, from the data mining perspective. This data-driven model involves demand-driven aggregation of information sources, mining and analysis, user interest modeling, and security and privacy considerations. We analyze the challenging issues in the data-driven model and also in the Big Data revolution.

Mining Ars in Big Data with Ngep

Big data is a data with large size means it has large volume, velocity and variety. Now a day's big data is expanding in a various science and engineering fields. And so there are many challenges to manage and analyse big data using various tools. This paper introduces the big data and its Characteristic concepts and Next section elaborates about the Challenges in Big data. In Particular, wed discuss about the technologies used in big data Analysis and Which Tools are mainly used to analyze the data. As big data is growing day by day there are lot of application areas where we need to use any of the

technology and tools discussed in paper. Mainly this paper focuses on the Challenges, Technologies, Tools and Applications used for big data Analysis.

Big Data Mining With Parallel Computing: A Comparison of Distributed and Map Reduce Methodologies

Applications of machine learning are widely used in the real world with either supervised or unsupervised learning process. Recently emerged domain in the information technologies is Big Data which refers to data with characteristics such as volume, velocity and variety. The existing machine learning approaches cannot cope with Big Data. The processing of big data has to be done in an environment where distributed programming is supported. In such environment like Hadoop, a distributed file system like Hadoop Distributed File System (HDFS) is required to support scalable and efficient access to data. Distributed environments are often associated with cloud computing and data centres. Naturally such environments are equipped with GPUs (Graphical Processing Units) that support parallel processing. Thus the environment is suitable for processing huge amount of data in short span of time. In this paper we propose a framework that can have generic operations that support processing of big data. Our framework provides building blocks to support clustering of unstructured data which is in the form of documents. We proposed an algorithm that works in scheduling jobs of multiple users. We built a prototype application to demonstrate the proof of concept. The empirical results revealed that the proposed framework shows 95% accuracy when the results are compared with the ground truth.

Parallel Mining of ARS

One of the important and well-researched problems in data mining is mining association rules from transactional databases, where each transaction consists of a set of items. The main operation in this discovery process is computing the occurrence frequency of the interesting set of items. i.e., Association Rule mining algorithms search for the set of all subsets of items that frequently occur in many database transactions. In practice, we are usually faced with large data warehouses, which contain a large number of transactions and an exponentially large space of candidate itemsets, which have to be verified. A potential solution to the computation complexity is to parallelize the mining algorithm. In this paper, four parallel versions of a novel sequential mining algorithm for discovery of frequent item sets are proposed. The parallelized solutions are compared analytically and experimentally, by considering some important factors, such as time complexity, communication rate, and load balancing

Large-Scale Parallel Data Mining

The explosive growth in data collection in business and scientific fields has literally forced upon us the need to analyze and mine useful knowledge from it. Data mining refers to the entire process of extracting useful and novel patterns/models from large datasets. Due to the huge size of data and amount of computation involved in data mining, high-performance computing is an essential component for any successful large-scale data mining application. This chapter presents a survey on large-scale parallel and distributed data mining algorithms and systems, serving as an introduction to the rest of this volume. It also discusses the issues and challenges that must be overcome for designing and implementing successful tools for large-scale data mining.

Existing Process

As a consequence, whilst in traditional frequent item mining applications we need to estimate frequency counts, we are instead required to estimate decayed counts. These applications are said to work in the time fading model. Two sketch-based algorithms for processing time-decayed streams have been recently published independently near the end of 2016. The Filtered Space Saving with Quasi-Heap (FSSQ) algorithm, besides a sketch, also uses an additional data structure called quasi-heap to maintain frequent items.

Proposed Methodology

Forward Decay Count-Min Space Saving (FDCMSS), our algorithm, cleverly combines key ideas borrowed from forward decay, the Count-Min sketch and the Space Saving algorithm. Therefore, it makes sense to compare and contrast the two algorithms in order to fully understand their strengths and weaknesses. We show, through extensive experimental results, that FSSQ is better for detecting frequent items than for frequency estimation. The use of the quasi-heap data structure slows down the algorithm owing to the huge number of maintenance operations. Therefore, FSSQ may not be able to cope with high-speed data streams. FDCMSS is better suitable for frequency estimation; moreover, it is extremely fast and can be used in the context of high-speed data streams and for the detection of frequent items as well, since its recall is always greater than 99%, even when using an extremely tiny amount of space. Therefore, FDCMSS proves to be an overall good choice when considering jointly the recall, precision, average relative error and the speed.

3. RESULTS AND DISCUSSION

Users

When users access the system through Portal Direct Entry, they are considered guests until they log in. The Login Module is a portal module that allows users to type a user name and password to log in. This module can be placed on any module tab to allow users to log in to the system.

Search Engine

Search engine optimization (“SEO”) ensures that the website can be properly indexed by popular search engines, such as Google, Bing, and Yahoo. Search engine optimization can provide higher levels of website traffic and reduced digital advertising costs with programs such as Google Ad words. For these reasons, search engine optimization can be a key success factor for many projects.

Frequent Items Measure

In Find item sets by you can set criteria for item set search: Minimal support: a minimal ratio of data instances that must support (contain) the item set for it to be generated. For large data sets it is normal to set a lower minimal support (e.g. between 2%-0.01%). Max. Number of item sets: limits the upward quantity of generated item sets. Item sets are generated in no particular order.

Filter item sets:

If you're looking for a specific item or itemsets, filter the results by regular expressions. Separate regular expressions by comma to filter by more than one word.

Forward Decay

It is a different model of decay satisfying the forward decay is computed on the amount of time between the arrival of an item and a fixed point L, known as the landmark.

Server

A server is a computer program or a device that provides functionality for other programs or devices, called "clients". This architecture is called the client–server model, and a single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities

Ranking

The ranks of the modules in an exact sequence. Satisfy the equality. The rank of a free module over an arbitrary ring (cf. free module) is defined as the number of its free generators.

Conclusion

To overcome the drawbacks of traditional algorithms, we proposed an algorithm, namely HP-Apriori, which mines knowledge in the form of FIs using performing four sub-tasks in the parallel way. The HP-Apriori performs similarly as the CD with the only difference that the HP-Apriori partitions the data horizontally and uses two processes in each partition for calculating the support counts of local itemsets instead of one process, so that the execution time is reduced significantly. We examined the feasibility and effectiveness of HP-Apriori to solving the ARs mining problems in the big data. The experimental results showed that computational complexity for HP-Apriori is reduced than CD.

Future Enhancement

Priority Based Selection: In Cognitive Radio network the users are classified into Licensed Primary Users and Unlicensed Secondary Users and there is no dedicated channel to send data, sensors need to negotiate with the neighbors and select a channel for data communication in CR-WSNs. This is a very challenging issue, because there is no cooperation between the PUs and SUs. PUs may arrive on the channel any time. If the PU claims the channel, the SUs have to leave the channel immediately. Therefore, data channels should be selected intelligently considering the PU's behavior on the channel and using some Priority Based Selection algorithms. Therefore USFR has been shown to effectively improve self-coexistence jointly in spectrum utilization, power consumption, and intra-cell fairness.

4. REFERENCES

[1] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data", IEEE Transactions on Knowledge and Data Engineering, vol. 26, no.1, pp. 97-107, 2014.

[2] M. Chen, S. Mao, and Y. Liu, "Big data: a survey", Mobile Network Application, vol. 19, pp.171-209, 2014.

[3] Y. Chen, F. Li, and J. Fan, "Mining ARs in big data with NGEP", Cluster Computing, vol. 18, no. 2, pp. 577-585, 2015.

[4] C. F. Tsai, W. C. Lin, and S. W. Ke, "Big data mining with parallel computing: a comparison of distributed and MapReduce methodologies", Journal of Systems and Software, vol. 122, pp. 83-92, 2016.

[5] W. Fan, and A. Bifet, "Mining big data: current status, and forecast to the future", ACM SIGKDD Exploration, vol. 14, no. 2, pp. 1-5, 2012.

[6] R. Agrawal, T. Imielinski, and A. Swami, "Mining ARs between sets of items in large databases", The International Conference on Management of Data, p. 207-216, 1993.

[7] Y. Le Bras, P. Lenca, and S. Lallich, "Opt monotone measures for optimal rule discovery", Computational Intelligence, vol. 28, no. 4, pp. 475-504, 2012.

[8] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", ACM SIGMOD International Conference on Management of Data, p. 1-12, 2000.

[9] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "New algorithms for fast discovery of ARs", Technical Report TR651, University of Rochester, 1997.

[10] J. Li, A. Choudhary, N. Jiang, and W. K. Liao, "Mining frequent patterns by differential refinement of clustered bitmaps", International Conference on Data Mining, 2006.