INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume3, Issue5)

Available online at: www.ijarnd.com

Frequent pattern mining on big data using Apriori algorithm

Lakshminarayanan

Student, Anna University, Chennai, Tamil Nadu

ABSTRACT

Frequent Pattern Mining is one of the most important tasks to extract meaningful and useful information from raw data. This task aims to extract item-sets that represent any type of homogeneity and regularity in data. Although many efficient algorithms have been developed in this regard, the growing interest in data has caused the performance of existing pattern mining techniques to be dropped. The goal of this paper is to propose new efficient pattern mining algorithms to work in big data. The existing pattern mining algorithms are based on homogeneity and regularity of data. With the dramatic increase on the scale of datasets collected and stored with cloud services in recent years, it takes more computation power for mining process in the cloud. Amount of work also transferred the approximate mining computation into the exact computation, where such methods not improve the accuracy also not enhance the efficiency. The proposed algorithm uses Hadoop distributed file server for frequent pattern mining. The Hadoop distributed file server improves the performance of the system. The Iterative apriori algorithm can be used to extract the frequent pattern from the dataset. In this approach, candidate itemsets are extracted from the initial dataset. The candidate itemsets are generated from the previous iteration. The support count is calculated for each candidate itemset. The support value is the frequency of items. The confidence value should be calculated for finding the dependency between itemsets. The threshold value is calculated and based on this value pruning is performed.

Keywords: *Frequent Pattern Mining, Big Data, Pruning, Support Count, Confidence Score, Map Reduce, Hadoop.*

1. INTRODUCTION

Big Data

Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating and information privacy.Lately, the term "big data" tends to refer to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. "There is little doubt that the quantities of data now available are indeed large, but that's not the most relevant characteristic of this new data ecosystem. Analysis of data sets can find new correlations to "spot business trends, prevent diseases, and combat crime and so on. Scientists, business executives, practitioners of medicine, advertising and governments alike regularly meet difficulties with large data-sets in areas including Internet search, fintech, urban informatics, and business informatics. Scientists encounter limitations in e-Science work, including meteorology, genomics, connectomics, complex physics sim- ulations, biology and environmental research.

Volume

Organizations collect data from a variety of sources, in-cluding business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies (such as Hadoop) have eased the burden.

Characteristics of Big Data

Velocity

Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors, and smart metering are driving the need to deal with torrents of data in near-real time.

Variety

Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents,

email, video, audio, stock ticker data and financial transactions.

At SAS, we consider two additional dimensions when it comes to big data:

Variability

In addition to the increasing velocities and varieties of data, data flows can be highly inconsistent with periodic peaks. Is something trending in social media? Daily, seasonal and event-triggered peak data loads can be challenging to manage. Even more so with unstructured data.

Complexity

Today's data comes from multiple sources, which makes it difficult to link, match, cleanse and transform data across systems. However, it's necessary to connect and correlate relationships, hierarchies and multiple data linkages or your data can quickly spiral out of control.

Big Data Technologies

Big data technologies are important in providing more accurate analysis, which may lead to more concrete decision-making resulting in greater operational efficiencies, cost re-ductions, and reduced risks for the business. To harness the power of big data, you would require an infrastructure that can manage and process huge volumes of structured and unstructured data in realtime and can protect data privacy and security. There are various technologies in the market from different vendors including Amazon, IBM, Microsoft, etc., to handle big data. While looking into the technologies that handle big data, we examine the following two classes of technology.

Analytical Big Data

This includes systems like Massively Parallel Processing (MPP) database systems and MapReduce that provide an-alytical capabilities for retrospective and complex analysis that may touch most or all of the data.MapReduce provides a new method of analyzing data that is complementary to the capabilities provided by SQL and a system based on MapReduce that can be scaled up from single servers to thousands of high and low-end machines. These two classes of technology are complementary and frequently deployed together.

Hadoop

Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. This brief tutorial provides a quick introduction to Big Data, MapReduce algorithm, and Hadoop Distributed File System.

Hadoop Architecture

Hadoop framework includes following four modules: Hadoop Common

These are Java libraries and utilities required by other Hadoop modules. These libraries provide filesystem and OS level abstractions and contain the necessary Java files and scripts required to start Hadoop.

Hadoop YARN this is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFSTM)

A distributed file system that provides high-throughput access to application data.

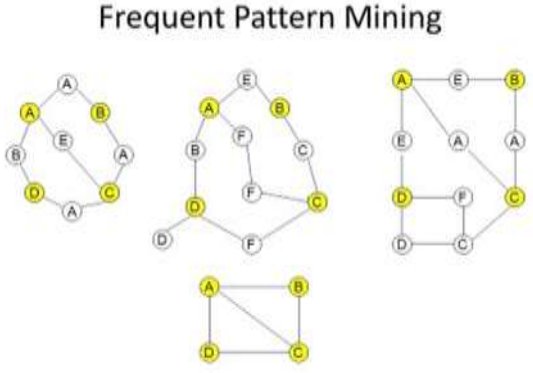
Hadoop MapReduce

This is a YARN-based system for parallel processing of large data sets.

Hadoop Distributed File System

Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS). The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more slave DataNodes that store the actual data. A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of reading and writes operation with the file system. They also take care of block creation, deletion, and replication based on an instruction given by NameNode.

Frequent Pattern Mining



Frequent patterns are itemsets, subsequences, or substruc-tures that appear in a data set with a frequency no less than a user-specified threshold. For example, a set of items, such as milk and bread that appear frequently together in a transaction

data set is a frequent itemset. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern. A substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently in a graph database, it is called a (frequent) structural pattern. Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data indexing, classification, clustering, and other data mining tasks as well. Frequent pattern mining is an important data mining task and a focused theme in data mining research. An abundant literature has been dedicated to this research and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemset mining in transaction databases to numerous research frontiers, such as sequential pattern mining.

The existing pattern mining algorithms are based on ho- homogeneity and regularity of data. With the dramatic increase on the scale of datasets collected and stored with cloud services in recent years, it takes more computation power for mining process in the cloud. Amount of work also transferred the approximate mining computation into the exact computation, where such methods do not improve the accuracy also not enhance the efficiency. With the increasing importance of data in every application, the amount of data to become unman- ageable, and the performance of such techniques could be dropped. No pruning techniques are used in existing approach. Discovering clusters in subspaces, or subspace clustering and related clustering paradigms is a research field where we find many frequent pattern mining related influences. In fact, as the first algorithms for subspace clustering were based on frequent pattern mining algorithms, it is fair to say that frequent pattern mining was at the cradle of subspace clustering—yet, it quickly developed into an independent research field. In this chapter, we discuss how frequent pattern mining algorithms have been extended and generalized towards the discovery of local clusters in high-dimensional data. In particular, we discuss several example algorithms for subspace clustering or projected clustering as well as point out recent research questions and open topics in this area relevant to researchers in either clustering or pattern mining. The Computational time and performance is not sufficient in this approach.No pruning techniques are used in this approach.

The proposed algorithm uses Hadoop distributed file server for frequent pattern mining. The Hadoop distributed file server improves the performance of the system. The Iterative apriori algorithm can be used to extract the frequent pattern from the dataset. In this approach, candidate itemsets are extracted from the initial dataset. The candidate itemsets are generated from the previous iteration. The support count is calculated for each candidate itemset. The support value is the frequency of items. The confidence value should be calculated for finding the dependency between itemsets. The threshold value is calculated and based on this value pruning is performed. Apriori uses a "bottom-up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k k from item sets of length k 1 k-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.The computational and storage problems are overcomes in this approach.Improved Pruning Strategy is used for mining Maximal Frequent Pat- terns.Performance of the System is improved by using Hadoop distributed file system.

Data Selection and Upload in DFS

In this module, input data is selected and Features are extracted from the input data. Feature values are uploaded into Distributed File Server. It is suitable for the distributed storage and processing. Hadoop provides a command interface to interact with HDFS. Streaming access to file system data. HDFS holds the very large amount of data and provides easier access. To store such huge data, the files are stored on multiple machines. These files are stored in a redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available for parallel processing.

Generation of Candidate Itemset

In this module, generate candidate itemsets by partitioning data into k- itemsets. This is the first step of our algorithm. This Candidate itemsets would be generated by means of preprocessing of data. This operation generates new candidate k-itemsets based on the frequent (k-1)-itemsets found in the previous iteration.

Measuring support count

In this module, support count is calculated for candidate itemsets. Support count is a frequency of each item in candidate itemset. Support is an indication of how frequently the itemset appears in the dataset. The support count of an itemset is always calculated with the respect to the number of items which contains the specific itemset. The absolute support of A, i.e. the absolute number of transactions which contains A, is 2. The relative support of A, i.e. the relative number of transactions which contains A, is 2/2=1.

MeasuringConfidence Interval

In this module, the confidence value is calculated for itemsets based on support value of itemsets. Confidence is an indication of how often the rule has been found to be true. The confidence is a value of a rule, X Y with respect to a set of Itemsets T, which is the proportion of the itemsets. This confidence interval value can be used to remove the infrequent patterns from the itemsets.

Pruning Items

In this module, we have calculated the threshold value based on confidence interval value. The confidence interval is above in the threshold value should be removed as the infrequent items. The confidence interval of itemsets which are below in the threshold value should be considered as the frequent itemsets. The threshold value is considered as a classifier for classifying itemsets.

Mining Frequent Itemsets

In this module, we will get the frequent patterns from the candidate itemsets. Frequent patterns are itemsets, sub-sequences, or substructures that appear in a data set with

frequency is less than a specified threshold. A substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data.

2. CONCLUSION

In this paper, we have proposed new efficient pattern mining algorithms to work in big data. All the proposed models are based on the well-known Apriori algorithm. This algorithm has been also proposed for mixing condensed representations of frequent patterns. Pruning the search space by means of anti-monotone property. Two additional algorithms have been proposed with the aim of discovering any frequent pattern available in data. In Future, We will use the Top – K Ranking Algorithm to find the top k frequent patterns from the given dataset. Ranking functions are evaluated by a variety of means; one of the simplest is determining the precision of the first k top-ranked results for some fixed k; Frequently, computation of ranking functions can be simplified by taking advantage of the observation that only the relative order of scores matters, not their absolute value; hence terms or factors that are independent of the features may be removed, and terms or factors that are independent of the feature may be precomputed and stored with the dataset.

3. ACKNOWLEDGMENT

First and foremost, I would like to thank god almighty for being the unconditional guiding light throughout my endeavor- our.

We express our gratitude to Founder Lion. Thiru R.PANDIAN, M.A. It is my great privilege to thank our college Chairman Mr.Pa.THANAVELAN, B.E.,M.B.A., I would like to express my gratitude to Dr.S.BASKARAN, M.Sc.(Engg.), Ph, D., Principal of our college for providing me with support and best facilities for the completion of this work.

I am thankful to Mr.M.JOHN BASHA, M.E.,(Ph.D.)., Assistant Professor and Head, Department of Computer Science & Engineering for the permission and encouragement to carry out this thesis work and also I would like to express my deep sense of gratitude and sincere thanks for his constant support, encouragement and motivation over the entire period of my thesis and his excellent guidance, valuable suggestions and constant support to bring the work into a shape.

My special thanks to the Project Coordinator Ms..VIJAYALAKSHMI ,M.E., Assistant Professor, Department of Computer Science & Engineering for the involvement and commitment shown towards successful progress of this work and timely support.I also thank all the members of faculty and lab technicians of CSE Departments for their kind co-operation in completion of the project.

4. REFERENCES

[1] T.-M. Choi, H. K. Chan, and X. Yue, "Recent development in big data analytics for business operations and risk management," IEEE Trans.Cybern., vol. 47, no. 1, pp.81–92,Jan.2017.

[2] J. M. Luna, "Pattern mining: Current status and emerging topics,"Progr.Artif. Intell., vol. 5, no. 3, pp. 165–170, 2016.

[3] C. C. Aggarwal and J. Han, Frequent Pattern Mining,1sted.Cham,Switzerland: Springer, 2014.

[4] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Currentstatus and future directions," Data Min. Knowl. Disc., vol. 15, no. 1,pp. 55–86, 2007.

[5] J. M. Luna, J. R. Romero, C. Romero, and S. Ventura,"On the use of genetic programming for mining comprehensible rules in subgroup discovery," IEEE Trans. Cybern.,vol. 44, no. 12, pp. 2329–2341, Dec. 2014. [Online]. Available:http://dx.doi.org/10.1109/TCYB.2014.2306819

[6] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: Aperformance perspective,"IEEE Trans. Knowl. Data Eng., vol. 5, no. 6,pp. 914–925, Dec. 1993.

[7] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach,"Data Min.Know. Disc., vol. 8, no. 1, pp. 53–87, 2004.

[8] S. Zhang, Z. Du, and J. T. L. Wang, "New techniques for mining frequent patterns in unordered trees," IEEE Trans. Cybern.,vol. 45, no. 6, pp. 1113–1125, Jun. 2015. [Online]. Available:http://dx.doi.org/10.1109/TCYB.2014.2345579

[9] J. M. Luna, J. R. Romero, and S. Ventura, "Design and behaviorstudy of a grammar-guided genetic programming algorithm for mining association rules," Knowl. Inf. Syst., vol. 32, no. 1, pp. 53–76,2012.

[10] R. Agrawal, T. Imielí nski, and A. Swami, "Mining association rulesbetween sets of items in large databases," in Proc. ACM SIGMODInt. Conf. Manag. Data (SIGMOD), Washington, DC, USA, 1993,pp. 207–216.

[11] J. Liu, K. Wang, and B. C. M. Fung, "Mining high utility patterns inone phase without generating candidates,"IEEE Trans. Knowl. DataEng., vol. 28, no. 5, pp. 1245–1257, May 2016. [Online]. Available:http://dx.doi.org/10.1109/TKDE.2015.2510012

[12] S. Ventura and J. M. Luna, Pattern Mining With EvolutionaryAlgorithms, 1st ed. Cham, Switzerland: Springer, 2016.

[13] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining forbig data," inProc. IEEE Int. Conf. Big Data (IEEEBigData), SiliconValley, CA, USA, 2013, pp. 111–118.

[14] J. M. Luna, A. Cano, M. Pechenizkiy, and S. Ventura, "Speeding-upassociation rule mining with in- verted index compression,"IEEE Trans.Cybern., vol. 46, no. 12, pp. 3059–3072, Dec. 2016. [Online]. Available:http://dx.doi.org/10.1109/TCYB.2015.2496175

[15] S. Sakr, A. Liu, D. M. Batista, and M. Alomari, "A survey of large scaledata management approaches in cloud environments,"IEEE Commun.Surveys Tuts., vol. 13, no. 3, pp. 311–336, 3rd Quart., 2011.X Y

