iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** III **Month of publication:** March 2019

DOI: http://doi.org/10.22214/ijraset.2019.3181

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

Mining Frequent Pattern On Big Data Using Map Reduce

Mr. M. Ramesh Kumar[1], Gokul D[2], Gokul M[3], Gokulakannan R[4]
[1]*Assistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur.*
[2, 3, 4]*Department of Computer Science and Engineering, Adhiyamaan College of Engineering. Hosur*

Abstract*: 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 work in big data. To this aim, a series of algorithms based on the Map Reduce framework and the Hardtop open-source implementation have been proposed. The proposed algorithms can be divided into three main groups.
Keywords: Mining, algorithms, dropped, anti-monotone, frequent patterns

I. INTRODUCTION

Data analysis has a growing interest in many fields like business intelligence, which includes a set of techniques to transform raw data into meaningful and useful for business analysis purposes. With the increasing importance of data in every application, the amount of data to deal with has become unmanageable, and the performance of such techniques could be dropped. The term big data is more and more used to refer to the challenges and advances derived from the processing of such high-dimensional datasets in an efficient way. Pattern mining is considered as an essential part of data analysis and data mining. Its aim is to extract subsequences, substructures or item-sets that represent any type of homogeneity and regularity in data, denoting intrinsic and important properties. This problem was originally proposed in the context of market basket analysis in order to find frequent groups. of products that are bought together ..

A. Ease Of Use

1) *Existing System:* Pair wise constraints are often defined as the must-link constraints and the cannot-link constraints. The must-link constraint means that two feature vectors should be assigned to the same cluster, while the cannot-link constraints means that two feature vectors cannot be assigned to the same cluster. Most of the cluster ensemble methods cannot achieve satisfactory results on high dimensional datasets. Not all the ensemble members contribute to the result.

II. LITERATURE SURVEY

Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites.

A. Proposed System

Cluster ensemble, is referred to as consensus clustering, one of the important research directions in ensemble learning, that can be divided into two stages: the first stage aims at generating a set of various ensemble members, while the objective of the second stage is to choose a suitable consensus function to summarize the ensemble members and search for an optimal unified clustering solution. To attain these objectives, we use a knowledge reuse framework which integrates multiple clustering solutions into a unified one. While there are various kinds of cluster ensemble techniques, some of them consider how to handle high dimensional data clustering and how to make use of prior knowledge of the given data set.

High dimensional datasets have too many attributes relative to the number of samples, which will lead to the over fitting problem. Most of the conventional cluster ensemble methods do not consider how to handle the over fitting problem, and cannot obtain satisfactory results when handling high dimensional data. Our method uses the random subspace technique to generate the new datasets in a low dimensional space, which will evaluate this problem. In summary, most of the cluster ensemble approaches only consider using a similarity score or feature selection technique to remove the redundant ensemble members, and few of them study how to apply an optimization method to search for a suitable subset of ensemble members.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

III. MAP REDUCE

Map Reduce is an emerging paradigm that has become very popular for intensive computing. The programming model offers a simple and robust method for writing parallel algorithms. To have recently described the significance of the Map Reduce framework for processing large datasets, leading other parallelization schemes such as message passing interface. This emerging technology has been applied to many problems where computation is often highly demanding, and pattern mining is one of them. Proposed first methods based on Map Reduce to mine item-sets on large datasets. These methods were properly implemented by means of Hardtop, which is considered as one of the most popular open-source software frameworks for distributed computing on very large data sets. Considering previous studies and proposals, the aim of this paper is to provide research community with new and more powerful pattern mining algorithms for big data.

These new proposals rely on the Map Reduce framework and the open-source implementation, and they can be classified as follows.

1) No pruning strategy. Two algorithms [Apriori Map Reduce (AprioriMR) and iterative AprioriMR (IApriori MR)] are properly designed to extract patterns in large datasets. These algorithms extract any existing item-set in data regardless their frequency.

2) Pruning the search space by means of the ant monotone property. Two additional algorithms [space pruning AprioriMR (SP AprioriMR) and top AprioriMR (Top AprioriMR)] are proposed with the aim of discovering any frequent pattern available in data.

3) Maximal frequent patterns. A last algorithm [maximal AprioriMR (Max AprioriMR)] is also proposed for mining condensed representations of frequent patterns, i.e., frequent patterns with no frequent supersets. To test the performance of the proposed models, a series of experiments over a varied collection of big data sets has been carried out, comprising up to 3 · 1018 transactions and more than 5 million of singletons (a search space close to 25 267 646 − 1). Additionally, the experimental stage includes comparisons against both well-known sequential pattern mining algorithms and Map Reduce proposals. The ultimate goal of this analysis is to serve as a framework for future researches in the field. Results have demonstrated the interest of using the Map Reduce framework when big data is considered. They have also proved that this framework is unsuitable for small data, so sequential algorithms are preferable. The rest of this paper is organized as follows. Section II presents the most relevant definitions and related work. Section III describes the proposed algorithms. Section IV presents the datasets used in the experiments and the results obtained. Finally, some concluding remarks are outlined in Section V.

A. Applying Multiple Reducers

Under these circumstances, the use of Map Reduce in pattern mining is meaningless since the problem is still the memory requirements and the computational cost. To overcome this problem, we propose the use of multiple reducers, which represents a major feature of the algorithms proposed in this paper. Each reducer works with a small set of keys in such a way that the performance of the proposed algorithms is improved. The wrong use of multiple reducers in Map Reduce implies that the same key k might be sent to different reducers, causing a loss of information.

Thus, it is particularly important to previously fix the set of keys that will be analyzed in each reducer. However, this redistribution process cannot be manually carried out due to the keys are not known beforehand and, besides, it is especially sensitive since those reducers that receive a huge number of keys will slow down the whole process. In order to reduce the difference in number of k, v pairs analyzed by each reducer, and considering that the singletons that form the item-sets are always in the same order, i.e., I = {i1, i2,...,in} ⇔ 1 < 2 < ⋯ < n, a special procedure is proposed as follows. In a first reducer, any item-set that comprises the item i1 is considered. As a matter of example, let us consider a dataset comprising 20 singletons and three different reducers. Here, the first reducer will combine a maximum of 219 = 524 288 pairs of k, v ; the second reducer will combine a maximum of 218 = 262 144 pairs; and, finally, the third reducer will combine a maximum of 217 + 216 +⋯+ 20 = 262 143 pairs.

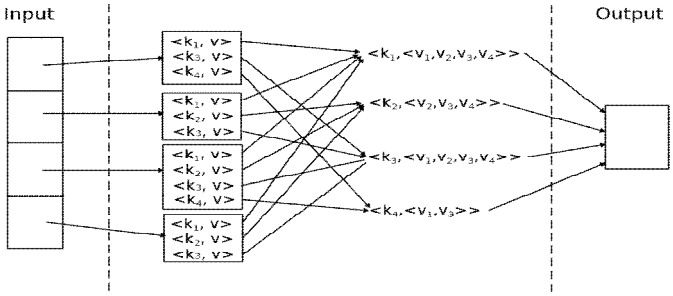


Diagram1.1 Multiple Map Reduce Framework

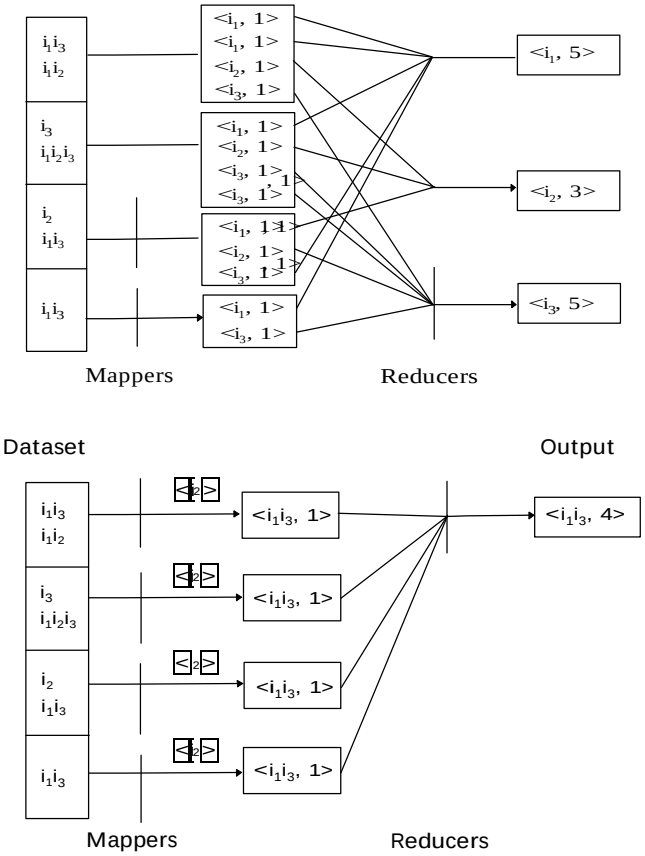
B. Apriori Versions Without Space Pruning

First Apriori version is based on the extraction of any item-set available in data. In this well-known algorithm, proposed to generate all the feasible item-sets in each transaction and to assign a support of one to them. Then, the algorithm checks whether each of the new item-sets were already generated by previous transactions and, if so, their support is increased in a unity. The same process is repeated for each transaction giving rise to a list L of patterns including their support or frequency of occurrence. As shown, the higher the number of both transactions and singletons, the higher the computational complexity and the memory requirements. It is noteworthy that a significant number of transactions implies a huge number of iterations and, therefore, a drop in the runtime. At the same time, a high number of singletons entails a huge number of candidate item-sets in C increasing both the computational complexity and the memory requirements.

C. Apriori Versions With Space Pruning

The task of finding all patterns in a database is quite challenging since the search space exponentially increases with the number of single items occurring in the database. As described in previous sections, given a dataset comprising n singletons, a maximum number of $2^n - 1$ different patterns can be found in that dataset. Additionally, this dataset might contain plenty of transactions and the process of calculating the frequency for each pattern might be considered as a tough problem. All of this highlights the importance of pruning the search space, given rise to the paradigm of constraint-based mining [3]. A really important pruning strategy in the pattern mining field is anti-monotone property, which determines that any sub-pattern of a frequent pattern is also frequent, and any super-pattern of an infrequent pattern will be never frequent.

Apriori may also include a space pruning strategy by using a level-wise paradigm in which all the frequent patterns of size $|P| = s$ is generated by using all the frequent patterns of size $s-1$. Thus, the main characteristic of Apriori is that every subset of a frequent pattern is also frequent, so it follows the anti-monotone property. In order to obtain new patterns of size $|P| = s$.



Apriori Versions With Space Pruning

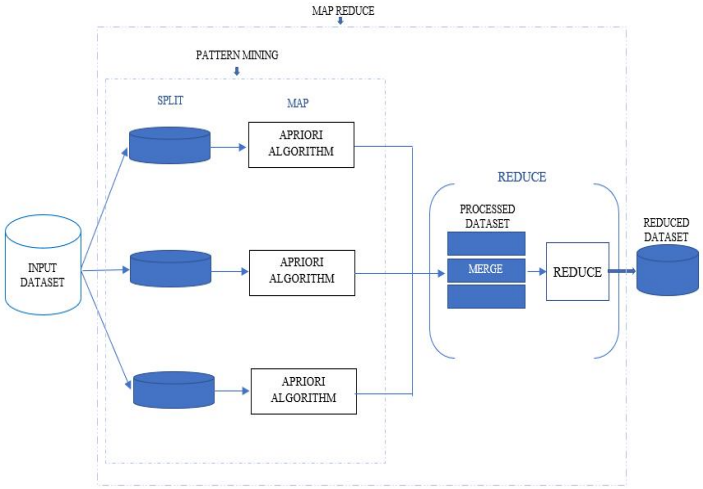
Produces any pattern P of size $|P| = 2$ that can be derived from the specific transaction $t_l \in T$. The final step of this second iteration is the reducer procedure, producing h{i_1,i_3},4i as a frequent pattern.

IV. SYSTEM ARCHITECTURE DESIGN

System Design is a solution, how to approach to creation of a new system. This important phase is composed of several steps. It provides the understanding and procedural details for implementing the system recommended infeasibility study. Stress in on translating performance requirement into design specification design goes through logical physical stages of development. Logical design reviews the present physical, prepare input and output specification..

1) Problem definition.
2) Input output specification.
3) Data based designed.
4) Modular program design.
5) Preparation of source code.
6) Testing and debug.

The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases. Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.



1.3 Architecture design

A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only. The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.
Choosing the right output method for each user is another objective in designing output. Much output now appears on display screens, and users have the option of printing it out with their own printer. The analyst needs to recognize the trade-offs involved in choosing an output method. Costs differ; for the user, there are also differences in the accessibility, flexibility, durability, distribution, storage and retrieval possibilities, transportability, and overall impact of the data.

A. Designing Output To Serve The Intended Purpose

During the information requirements determination phase of analysis, the systems analyst finds out what user and organizational purposes exist. Output is then designed based on those purposes. You will have numerous opportunities to supply output simply because the application permits you to do so. Remember the rule of purposiveness, however. If the output is not functional, it should not be created, because there are costs of time and materials associated with all output from the system.

B. Designing Output To Fit The User

With a large information system serving many users for many different purposes, it is often difficult to personalize output. On the basis of interviews, observations, cost considerations, and perhaps prototypes, it will be possible to design output that addresses what many, if not all, users need and prefer. Generally speaking, it is more practical to create user-specific or user-customizable output when designing for a decision support system or other highly interactive applications such as those using the Web as a platform. It is still possible, however, to design output to fit a user's tasks and function in the organization, which leads us to the next objective.

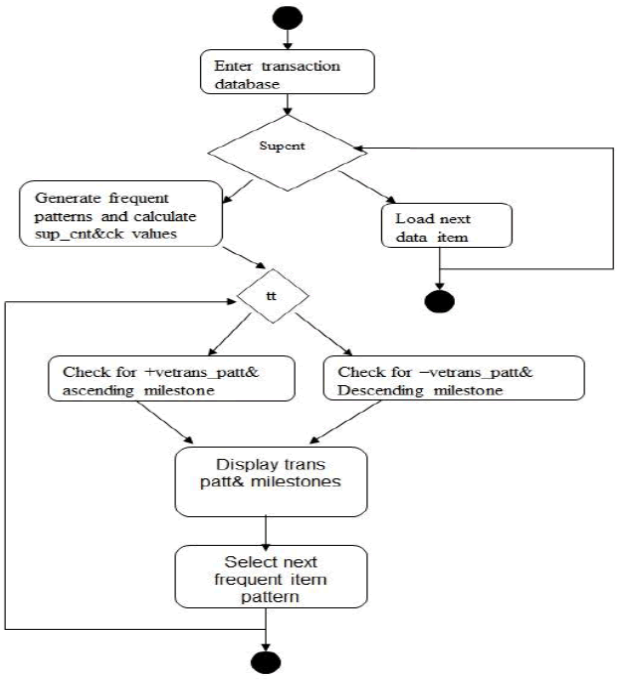
C. Delivering The Appropriate Quantity Of Output

Part of the task of designing output is deciding what quantity of output is correct for users. A useful heuristic is that the system must provide what each person needs to complete his or her work. This answer is still far from a total solution, because it may be appropriate to display a subset of that information at first and then provide a way for the user to access additional information easily. The problem of information overload is so prevalent that it is a cliché, but it remains a valid concern. No one is served if excess information is given only to flaunt the capabilities of the system. Always keep the decision makers in mind. Often, they will not need great amounts of output, especially if there is an easy way to access more via a hyperlink or drill-down capability.

V. SYSTEM ORGANIZATION

A. Flow Chart Diagram

They define different states of an object during its

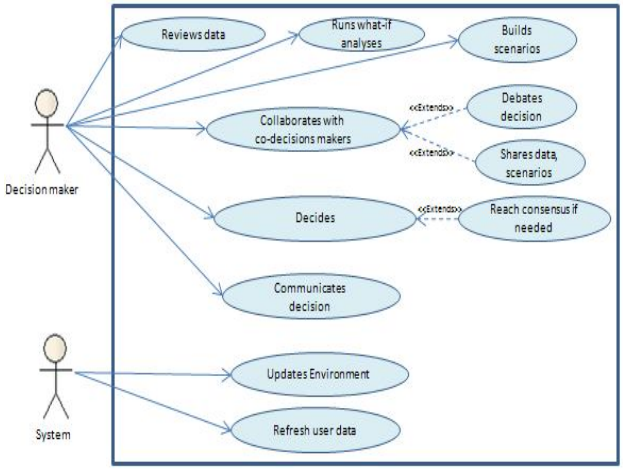


Lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination. Chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

B. Use Case Diagram

The most important step to fully understanding the requirements and scope of this system was the creation and subsequent refinement of this use case diagram. Many changes were made to the initial diagram before a consensus was met on the final design used.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

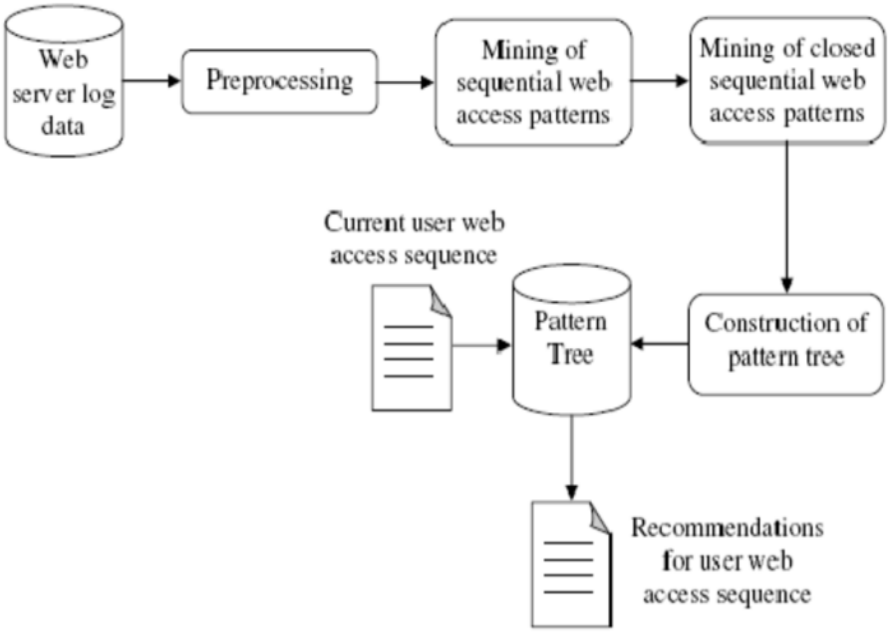


Use case diagram

C. Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved

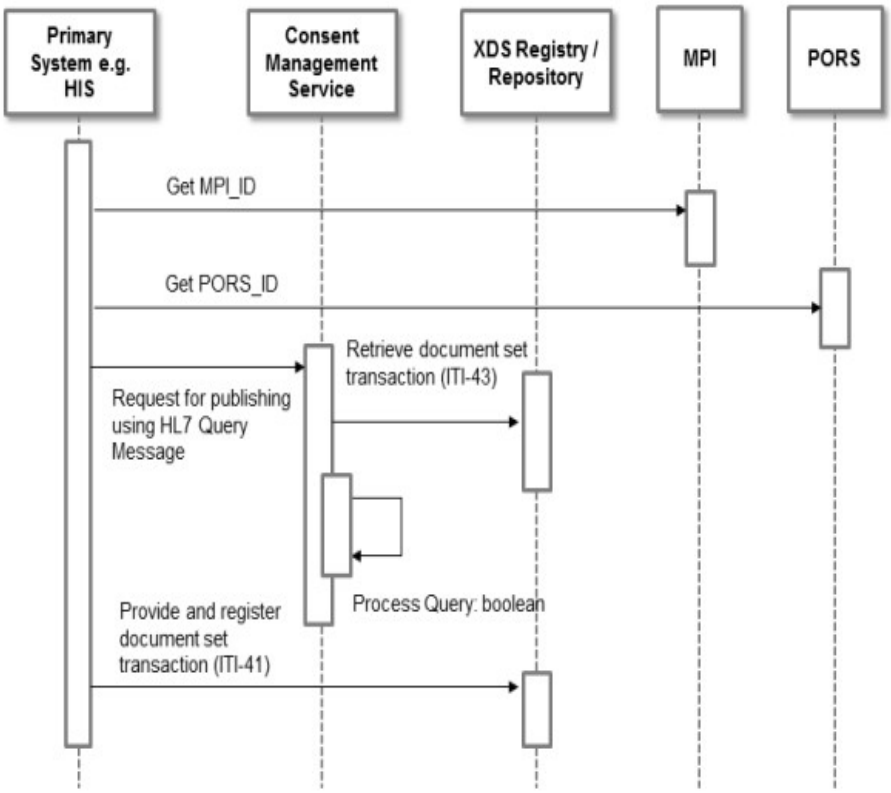
A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.



Collaboration diagram

D. Sequence Diagram

Sequence diagrams are used to help quickly picture how the objects in a use case interact during the sequence of events in a use case. They do this by showing the behavior of the objects in the use case and the messages they pass. The main strength of sequence diagrams is the clarity with which they show what objects are making what calls and to whom, and which objects are doing what processing.



Desiging Plan

1) Many of the core principles of the DSDM come about from the idea that the designing and building of a system should be an incremental process.
2) To benefit most from the methodology being used it was decided that the implementation phase of the development would be carried out in an incremental fashion and to do this the system would have to be split into distinct sections.
3) Due to the manner in which the requirements were gathered it was very easy to view the system in such a way. It was decided that the best way to split the system up would be by the main tasks that the system was required to achieve and it was hoped that by doing so each increment of the build would be kept to a manageable level.

VI. MODULES

A. *Pattern Mining*

1) Pattern mining used to find statistically relevant patterns between data examples where the values are delivered in a sequence. It is usually presumed that the values are discrete, and thus time series mining is closely related, but usually considered a different activity. Sequential pattern mining is a special case of structured data mining.
2) The term pattern is defined as a set of items that represents any type of homogeneity and regularity in data, denoting intrinsic and important properties of data
3) It is noteworthy the support of a pattern is monotonic i.e., none of the super-patterns of an infrequent pattern can be frequent.

B. *Map Reduce*

1) Map Reduce is a recent paradigm of parallel computing. It allows writing parallel algorithms in a simple way, where the applications are composed of two main phases defined by the programmer: 1) map and 2) reduce. In the map phase, each map per processes a subset of input data and produces key-value (*k*, *v*) pairs.
2) The flowchart of a generic Map Reduce framework is depicted. There are many Map Reduce implementations but Hadoop is one of the most widespread due to its open-source implementation, installation facilities and the fundamental assumption that hardware failures are common and should be automatically handled by the platform. Furthermore, Hadoop proposes the use of a distributed files Item, (HDFS). HDFS replicates data into multiple storage nodes that can concurrently access to the data.

VII.DATABASE

A database is a collection of information related to a particular subject or purpose such as tracking customer orders or maintaining a music collection. Using Microsoft access, you can manage all your information from a single database file. With the file, data is divided into separate storage containers called tables, views. One may add, and update table data using online forms, find and retrieve just the data you want using queries, and analyze or data in a specific layout using reports.

To store your data, create one table for each type of information you track. To bring the data from multiple tables together in a query, from, or report, you have defined relationship between the tables. To find and retrieve just the data the meets conditions you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built- in or custom calculations on your data, To easily view, enter and change data directly in a table create a form. When you open a form, Microsoft access retrieves the data from one or more table and tables and displays it on screen using the layout chosen in the form wizard or using a layout that you created from scratch.

VIII. IMPLEMENTATION AND RESULTS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

The following chapter looks at how the system implementation progressed over the course of the build. It starts off by detailing the plan that was used to order how the implementation would progress. The chapter then covers some of the important aspects of implementation that were carried out throughout the building of the system. The data-tier and then the web tier will be looked at in turn, and for the web-tier the implementation of each aspect of the Model-View-Controller will be considered. Finally, the testing that was carried out on the system will be examined and the effect the results had on the system will be looked.

IX. CONCLUSION & FUTURE WORK

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 and the Map Reduce framework

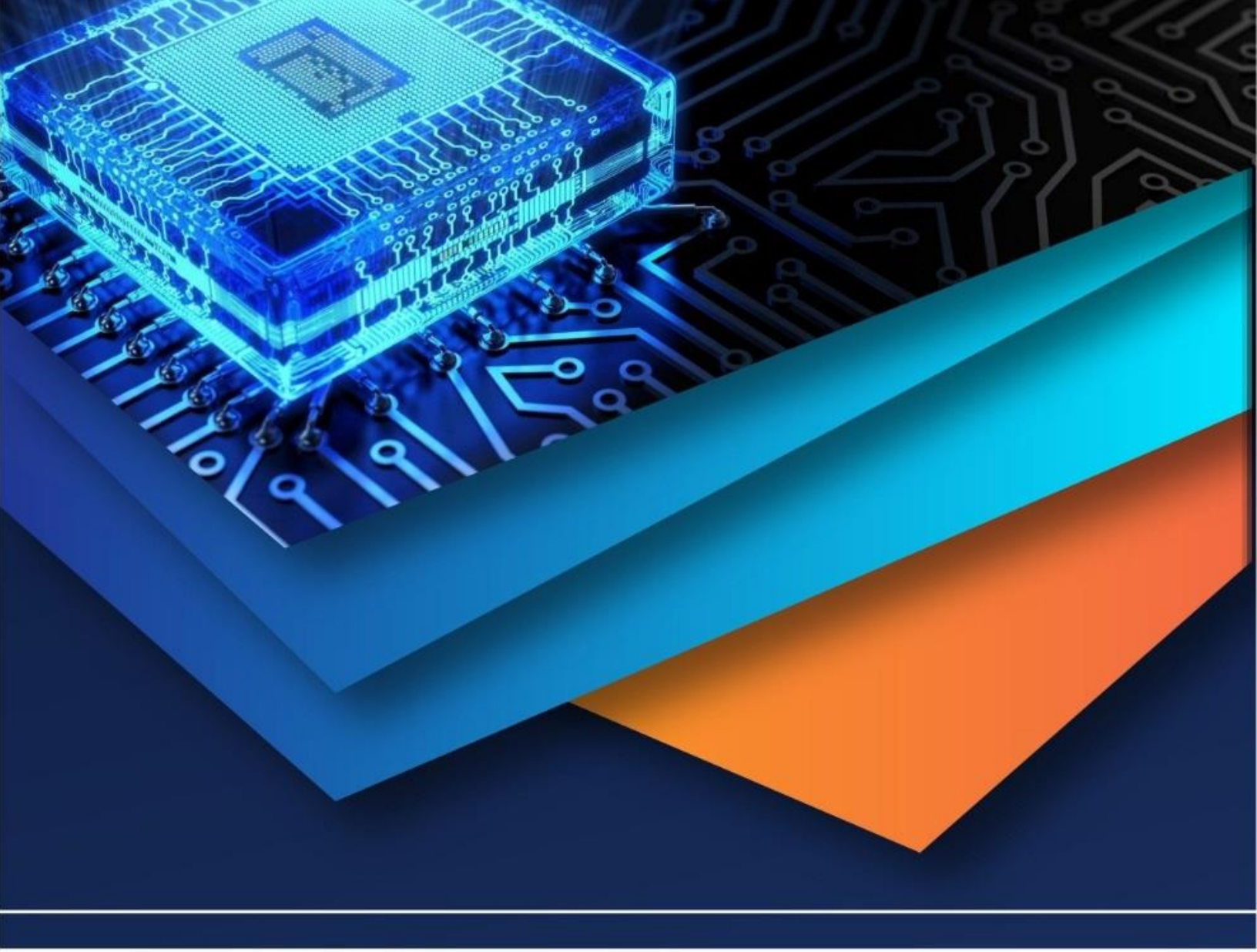
A. No pruning strategy. Two algorithms (AprioriMR and AprioriMR) for mining any existing pattern in data have been proposed.
B. Pruning the search space by means of antimonotone property. Two additional algorithms (SPAprioriMR and TopAprioriMR) have been proposed with the aim of discovering any frequent pattern available in data.
C. Maximal frequent patterns. A last algorithm (MaxAprioriMR) has been also proposed for mining condensed representations of frequent patterns.

All the algorithms have been compared to highly efficient algorithms in the pattern mining field. The experimental stage has revealed that our proposals perform really well for huge search spaces. Results have also revealed the unsuitability of MapReduce frameworks when small datasets are considered.

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, 1st ed. Cham, Switzerland: Springer, 2014.
[4] J. Han, H. Cheng, D. Xin, and X. Yan, “Frequent pattern mining: Current status 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: A performance 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 behavior study 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. Imielinski, and A. Swami, "Mining association rules ´ between sets of items in large databases," in Proc. ACM SIGMOD Int. 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 in one phase without generating candidates," IEEE Trans. Knowl. Data Eng., 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 Evolutionary Algorithms, 1st ed. Cham, Switzerland: Springer, 2016.
[13] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data," in Proc. IEEE Int. Conf. Big Data (IEEEBigData), Silicon Valley, CA, USA, 2013, pp. 111–118.
[14] J. M. Luna, A. Cano, M. Pechenizkiy, and S. Ventura, "Speeding-up association rule mining with inverted 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 scale data management approaches in cloud environments," IEEE Commun. Surveys Tuts., vol. 13, no. 3, pp. 311–336, 3rd Quart., 2011
[16] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.
[17] I. Triguero, D. Peralta, J. Bacardit, S. García, and F. Herrera, "MRPR: A MapReduce solution for prototype reduction in big data classification," Neurocomputing, vol. 150, pp. 331–345, Feb. 2015.
[18] C. Lam, Hadoop in Action, 1st ed. Stamford, CT, USA: Manning, 2010.
[19] B. Ziani and Y. Ouinten, "Mining maximal frequent itemsets: A Java implementation of FPMAX algorithm," in Proc. 6th Int. Conf. Innov. Inf. Technol. (IIT), Al Ain, UAE, 2009, pp. 330–334.
[20] M. J. Zaki, "Scalable algorithms for association mining," IEEE Trans. Knowl. Data Eng., vol. 12, no. 3, pp. 372–390, May/Jun. 2000.
[21] J. Pei et al., "Mining sequential patterns by pattern-growth: The PrefixSpan approach," IEEE Trans. Knowl. Data Eng., vol. 16, no. 11, pp. 1424–1440, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2004.77
[22] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, 1st ed. Boston, MA, USA: Addison-Wesley, 2005.
[23] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang, "Mars: A MapReduce framework on graphics processors," in Proc. 17th Int. Conf. Parallel Archit. Compilation Tech. (PACT), Toronto, ON, Canada, 2008, pp. 260–269.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY