PATTERN ANALYSIS AND KNOWLEDGE DISCOVERY ALGORITHMS – DIFFERENT PERSPECTIVES

Dr. B. Lavanya and Ms. P. Devipriya

ABSTRACT

In recent years many approaches to discovering unexpected and useful patterns from massive datasets are always being the most important problem in data mining. Over the past years, many researchers develop different frequent pattern mining algorithm. Many of the early pattern mining algorithm [5] to discover the frequent relevant pattern and subsequence's from the given sequences. At the same time, several studies have been done on effective algorithms for mining frequent sequential patterns which satisfies the user-specified constraint. Sequential Pattern Mining (SPM) is widely used data mining technique for discovering sequential patterns with a host of application domains including medicine, shopping sequences, sales record analysis, stock market data, the symptoms of a patient, DNA sequences telecommunications, and the WWW, etc. This survey paper aim is to analyze different existing algorithms to develop a new effective pattern discovery algorithm to discover knowledge from the dataset. Frequent patterns and frequent sequential pattern mining algorithms like Apriori, FP-growth, and GSP, SPIRIT a family of novel algorithms have studied The survey of these algorithms are collected from different perspectives of research. A detailed survey of these entire algorithms is presented in this paper. This paper studies and analysis and presents a comparison of different algorithms various types, methods, and data mining techniques for Frequent pattern mining.

Index Terms *Pattern discovery, Frequent pattern mining, Constraint, MapReduce, Parallel.*

Reference *to this paper should be made as follows**:** Dr. B. Lavanya and Ms. P. Devipriya (2020), "Pattern Analysis and Knowledge Discovery Algorithms – Different Perspectives" Int. J. Electronics Engineering and Applications, Vol. 8, No. 2, pp. 63-68.*

Biographical notes:

Dr. B. Lavanya *completed his bachelors in Computer Science and Engineering from Anna Univeristy (Coimbatore), in 2010 and Masters in Computer Cognition and Technology from University of Mysore, India, during 2013.*
Soon after his bachelor, he has served in a product based company named Amphisoft Technologies Private Limited for more than a period of 2 years. After his masters, he rejoined his previous employer in Research & Development division, Amphisoft Technologies. His areas of interest include Image Processing, Video Processing, Virtual Reality, Pattern Recognition, Web, Mobile and Related Technologies.

P. Devipriya *completed his bachelors in Electrical Engineering from SJCE Mysore, India and Masters from BITS Pilani, India.*

Published by: International Associations of Professionals and Technical Teachers (IAPATT)

I. INTRODUCTION

Discovering pattern is an important data mining task with broad applications. In the previous year, there have been many studies on efficient pattern mining and its applications. The mining process mainly focused on discovering useful, interesting and unexpected pattern in databases. The first algorithm of frequent pattern mining was introduced by Agarwal and Srikant for finding frequent itemsets and then extracting the association rules [5]. Frequent Pattern mining
This paper aims to discover the frequent relevant pattern and the sub subsequence from the database [5].
This paper presents an outline of all the aspects of pattern mining method and this work also highlighted techniques and algorithms used in pattern discovery and analysis. Present work identified gaps that are present in several existing pre-processing and pattern discovery algorithms.

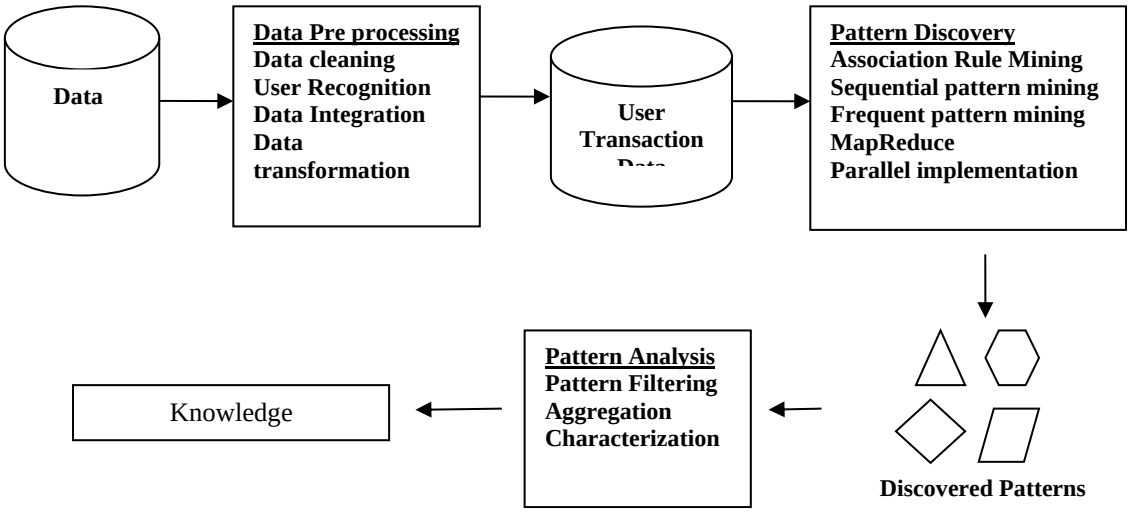

Fig. 1 Architecture of Data analysis and Knowledge Discovery

II. LITERATURE REVIEW

Akshita Bhandari et al. [2014] presents fp - tree based apriori algorithm which leads to a lot of memory space reduction by using the parallel algorithm. In this improved algorithm which reduces the limitations of the original Apriori algorithm's repeated scan of the database, by using the parallel algorithm and the concept of partitioning. The data structure which is introduced in this paper is the frequent pattern tree which is used for finding out the frequent itemsets and also used for generating the conditional patterns. The analysis has shown that the time consumed in improved Apriori in each group of the transaction is less than the original Apriori, and the difference increases more and more as the number of transactions increases. The memory space is reduced by using the partitioning approach which partitions the clusters initially and selects one particular cluster out of this. It is an improvement as earlier the algorithm took exponential space but now it is reduced greatly [6].
M. N. Garofalakis [1999] developed a family of SPIRIT algorithm for mining frequent sequential pattern. The basic idea of this algorithm is constraint specification use of the regular expression at

the flexible tool [2]. He considered two important parts to select the regular expression as a constraint specification tool. Discover the simple and natural syntax for specification of families of sequential is the first part and to specifying the interesting pattern constraints are the second part and most powerful for a huge range of data.

C. Antunes and A L. Oliveira [2004] proposed a new pattern mining approach consists on the use of a constraint relaxation (expressed as a formal language [Garofalakis 1999] [Antunes 2002]). The new concept behind this algorithm is constraint approximations based on SPIRIT algorithm using regular languages as constraints to discover unknown information (expressed as deterministic finite automata). This relaxation is close to the original constraint and makes the finding possible unknown information. The new concept behind e-accepts is to ignore the generation of possible sequences, given that this performance consumes considerable time in sequential pattern mining algorithms. Generating the potential sequences, considering possible errors, if each sequence element performs a valid transition in the automaton, beginning on the initial state. Whenever an element does not correspond to a valid transition, the algorithm tries to replace it (which corresponds to apply to Replacement), and if this fail, then tries to ignore (which corresponds to a Deletion) and finally by trying to introduce a valid transition (which corresponds to an Insertion) [3].

J.M. Luna, F et al.[4] have proposed new efficient pattern mining algorithms based on the Apriori algorithm and the MapReduce framework. All of them depended on the MapReduce structure and the Hadoop open-source implementation. Two of(AprioriMR and IAprioriMR) these algorithms with no pruning strategy are proposed to enable any existing pattern to be discovered. Two additional algorithms (SPAprioriMR and TopAprioriMR) using the strategy of pruning for mining frequent patterns. Were finally proposed new effective maximal frequent pattern on MapReduce (MaxAprioirMR) algorithm is proposed.

Sheetal Rathi and C.A. Dhote proposed a new model to implement a parallel FP Growth algorithm that makes the use of removing process by FP Growth algorithm without generating the actual tree (or multiple smaller trees). This approach improves the performance of the algorithm at the same time results in more efficient memory usage. The proposed algorithm Accelerated Frequent Itemset Mining (AFIM) makes use of multiple GPU systems [7].

III. CATEGORIES OF PATTERN MINING ALGORITHMS

Apriori algorithm

Apriori algorithm proposed by R. Agrawal and R. Srikant in 1994[5] for frequent itemset mining and association rule learning. Apriori algorithm is one of the most important in data mining. The mining process performs based on two important concepts: minimum support and minimum confidence.
Apriori employs an iterative approach known as a level wise search, where k-itemsets are used to explore (k+1) itemsets. All nonempty subsets of a frequent itemset must also be frequent. A two-step process is followed: Apriori followed two important steps to discover frequent pattern mining. First, it joins candidate generating and the next one is pruning step, this process was doing upto find the total records are empty. This Apriori algorithm candidate set generation- and test approach fails when the dataset is very large.

FP-Growth algorithm

Frequent Pattern Growth (FP-Growth) (Han *et al*.2000) is an algorithm which is widely used for frequent pattern mining is an improved version of the Apriori Algorithm. Aprior has two major drawbacks: the first one is candidate sets have to be built each step and the second one is the algorithm has to repeatedly scan the database to build the candidate sets. The two drawbacks are to be overcome by this algorithm to implements a divide-and-conquer technique for interchanging the frequent items into Frequent pattern Tree (FP-Tree). Two major steps are followed to find the frequent pattern:

Design a simple data structure, FP Tree, which stores more data in less space.

FP-tree based pattern growth strategy to reveal frequent pattern design frequently

SPM Algorithm

The problem of sequential pattern mining was first addressed by Agrawal and Srikant in 1995[1]. Sequential pattern mining is the mining of frequent sequences or subsequences in the given database as patterns. There are two important issues in research to discover the sequential pattern mining one is improving the efficiency of sequential pattern mining process and the second one is Extending the sequential pattern which are associated to time constraint.

As per the research done till date on the sequential pattern mining algorithms mainly differ in two way, one is to reduce the very large number of sequences into the most interesting sequential patterns so that to reduce the storage cost. The second one is to remove any database record or data structure that has to be maintained over time for support of counting purposes only. In general, Sequential pattern mining algorithm can be broadly categorized into two groups based on these criteria's: Apriori Based and Pattern Growth Based.

GSP (Generalized Sequential Pattern)

In this algorithm proposed by Agrawal and Shrikant, it dived the data into multiple passes. This algorithm is faster than the AprioriAll algorithm because of reducing the number of scan concept. This algorithm classified into two major parts: one is candidate generation and candidate pruning.

Candidate Generation is used to merge pairs found in K-1^{th} pass w1 and w2 can be merged if subsequences obtained by removal of the first element of w1 and the last element of w2 are same. Candidate Pruning is used to Prune candidates that contain a subsequence which is infrequent in k-1 subsequences. To create a new pass in the database it needs support count (Candidate elimination involves thresholding based on minimum support).

MapReduce Algorithm

MapReduce is the current model of parallel computing designed for processing of large volumes of data use the parallel processing Introduced by Google. In this algorithm categorized based on two important phases defined by the programmer: namely, map and reduce which runs on all machines in a Hadoop cluster. The input and output of each phase must be in form of (key, value) pair.

Hadoop presents the use of a distributed file system, its convert the data into multiple storage nodes that accessed all at once and easy to scale data processing is the major advantage of MapReduce. The data processing primitives are called mappers and reducers. Large scale distributed data processing use MapReduce concept to design an efficient, scalable and simplified programming model [8].

Parallel Algorithm

A parallel algorithm that can execute several instructions simultaneously on different processing devices (CPU)& then combine all individual outputs to produce the final result. Distributed framework method is used in Parallel algorithm.

Parallel algorithms are designed to improve the computation speed of a compute

- To analyze a parallel algorithm, the following parameters are considered.
  (1) Time Complexity (Execution time)
  (2) Total number of processors used
  (3) Total cost

1. Time Complexity

Execution time is measured based on the time taken by the algorithm to solve a problem.

- Total execution time is calculated from the moment when the algorithm stats executing to the moment it stops.
- In parallel computing, if all the processors do not start or end execution at the same time, then the total execution of the algorithm is the moment the first processor started its execution to the moment when the last processor stops its execution

Speed up of an Algorithm:-

The performance of a parallel algorithm is determined by calculating its speedup.

$$\text{Speedup} = \frac{\text{worst case execution time of the fastest known sequential algorithm for a particular problem.}}{\text{Worst case execution time of parallel algorithm}}$$

2. Number of processors used:-

- Here the cost of buy, maintain and run the computers are calculated.
- Larger the number of processors used by an algorithm to solve a problem, more costly becomes the obtained result.
-

3. Total Cost

Total cost = Time complexity X Number of processors based.

$$\text{Efficiently of parallel algorithm} = \frac{\text{Worst case execution time of sequential algorithm}}{\text{Worst case execution time of parallel algorithm}}$$

IV. CONCLUSION

In this paper, we discussed many existing frequent pattern mining and various types of their algorithms. This paper presented an outline of all the aspects of pattern mining method. This survey paper aim is to analyze different existing algorithms to develop a new effective pattern algorithm to discover knowledge from the dataset. This paper also studies and analyzed frequent patterns and frequent sequential pattern mining algorithms like Apriori, FP-growth, and GSP, SPIRIT Regular Expression, Approximated Constraints, MapReduce, Parallel Implementation. The survey of these algorithms collected from different perspectives of research.

REFERENCES

[1] Agrawal R, and Srikant R., Mining Sequential Patterns. In Proc. Of the 11th International Conference on Data Engineering, March 1995

[2] M. N. Garofalakis, R. Rastogi, and K. Shim. SPIRIT: Sequential Pattern Mining with Regular

Expression Constraint". Bell Labs Tech. Memorandum BL0112370-9902 23TM, February 1999.

[3] C. Antunes, Arlindo L. Oliveira, "Sequential Pattern Mining With Approximated Constraints"

[4] J.M. Luna, Member, IEEE, F. Padillo, M. Pechenizkiy, Member, IEEE, and S.Ventura, Senior Member, IEEE "Apriori versions based on MapReduce for Mining Frequent Patterns on Big Data", IEEE TRANSACTIONS ON CYBERNETICS. VOL. X. NO. X. MONTH 2015.

[5] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules". In Proc. Of the 20th Intl. Conf. On very Large Data Bases, September 1994.

[6] Akshita Bhandari, Ashutosh Gupta, Debasis Das 2014. "Improvised apriori Algorithm using frequent pattern tree for real time applications in data mining".

[7] S Rathi and C.A. Dhote "PARALLEL IMPLEMENTATION OF FP GROWTH ALGORITHM ON XML DATA USING MULTIPLE GPU". In: Information Systems Design and Intelligent Applications pp 581-589.

[8] Anoop Joyti Sahoo, and Rajesh Kumar Tiwari "A Novel Approach for Hiding Secret data in Program Files" International Journal of Information and Computer Security. Volume 8 Issue 1, March 2016,

[9] Abu Salim, Sachin Tripathi and Rajesh Kumar Tiwari "A secure and timestamp-based communication scheme for cloud environment" Published in International Journal of Electronic Security and Digital Forensics, Volume 6, Issue 4, 319-332.

[10] Rajesh Kumar Tiwari and G. Sahoo, "A Novel Watermark Scheme for Secure Relational Databases" Information Security Journal: A Global Perspective, Volume 22, Issue 3, July 2013