ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Extracting Frequent Sequences from Web Log Data using Sequence Tree Algorithm

Priyanka Baraskar[1], Supriya Chavan[2], Dipshree Dhage[3], Samruddhi Giri[4], Jayashree Jha[5]

Student, Pursuing B.E., Information Technology, Atharva College of Engineering, Mumbai, India [1,2,3,4]

Professor, Information Technology, Atharva College of Engineering, Mumbai, India[5]

Abstract: Sequential Pattern mining is the process of applying data mining techniques to large web data repositories.With the extensive use of Internet, discovery and analysis of useful information from the World Wide Web becomes a practical necessity. Data mining techniques are applied to a sequential database to discover the correlation relationships that exists among the ordered list of events. In this kind of mining, hidden data is extracted to get useful information which helps in knowing the browsing patterns of the users. Web usage mining is a data mining method that can be used in recommending the web usage patterns with the help of users' session and behaviour. The aim of discovering frequent sequential patterns in Web log data is to obtain information about the access behaviour of the users. It helps to understand the buying pattern of the existing customers. This paper focuses on the performance of the sequence tree algorithm which is better than the Generalized Sequential Pattern (GSP) algorithm. This paper emphasizes on the running time of sequence tree algorithm and its ability to discover more number of patterns than the standard GSP algorithm.

Keywords*:* Sequential Pattern Mining, Web usage mining, Generalized Sequential Pattern (GSP), Sequence tree algorithm.

I. INTRODUCTION

With the advancement of the Information Technology, usage of World Wide Web is increasing day by day, which is becoming today's necessity. Innumerable visitors interact daily with the web throughout the world. Different kinds of data have to be organized in a way that they can be accessed by many users effectively**.** Web Mining is the application of data mining technologies which is being used to extract huge Web data repositories. Web mining can be broadly classified into three major parts: Web Contents Mining, Web Usage Mining and Web structure mining.

Web content mining is the extraction and integration of useful data from Web page. It is defined as the automatic search and retrieval of information and resources available from millions of sites and on-line databases through search engines. Web structure mining is the process of extracting hyperlink patterns from the web. Web usage mining is an important application of sequential pattern mining. It is mainly concerned with browsing patterns from the web.

Sequential patterns are the most frequently occurring subsequence's in sequences of sets of items. It is concerned with finding navigational patterns on the web by extracting knowledge from web logs, where frequent sequence of events in the database are composed of single items and not sets of items.

Sequential Pattern mining involves discovering frequent sequences from a database where data to be mined is in some sequential order, this was first introduced by Agrawal and Srikant [1], as follows: given a sequence database where each sequence is a list of transactions ordered by transaction action time and each transaction consists of a set of items. It finds all sequential patterns with a user specified minimum support, where the support is the number of data sequences that contain the pattern. The need for sequential pattern mining occurred during market analysis. It focused on retrieving frequent patterns in the sequences of products purchased by customers through time ordered transactions. Further its application was extended to complex applications like DNA research, network detection, telecommunication etc. Many algorithms were proposed. The very first was Apriori algorithm, which was put forward by the founders themselves. Later more scalable algorithms for complex applications were developed. E.g. GSP, Spade, PrefixSpan etc.

II. RELATED WORK

Surveys looked at different mining methods applicable to web logs [Srivastava et al. 2000; Facca and Lanzi 2003; Masseglia et al. 2005; Facca and Lanzi 2005; Iv´ancsy and Vajk 2006]. The drawbacks of previous methods are as follows:

(i) It does not provide complete solution for sequential patterns;
(ii) It does not provide proper classification.
(iii)It does not give brief insight of the techniques and theories used in mining sequential patterns

Shrikant and Agrawal [2] generalized their definition of sequential patterns in [2] to include time constraints, sliding time window and user-defined taxonomy and developed a generalized sequential pattern mining algorithm, GSP, which outperforms their AprioriAll algorithm [2]..

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

In general, there are two main research issues in sequential pattern mining.

1. The first is to improve the efficiency in sequential pattern mining process
2. Extending the mining of sequential pattern which are related to time constraints.

Sequential Pattern Mining Algorithms mainly differ in two ways [3]:

(1) The way in which candidate sequences are generated and stored. The main goal here is to minimize the number of candidate sequences generated so as to minimize I/O cost.

(2) The way in which support is counted and how candidate sequences are tested for frequency. The key strategy here is to eliminate any database or data structure that has to be maintained all the time for support of counting purposes only.

Based on these criteria's sequential pattern mining can be divided broadly into two parts:

- Pattern Growth Based
- Apriori Based

A. Pattern-Growth Algorithms:

The pattern growth method emerged as a solution to the problem of generate and test. The key idea is to intensify the search on a limited portion of the initial database and to avoid the candidate generation step altogether. The partitioning of search space plays a vital role in pattern-growth. Every pattern growth algorithm starts by representation of the database to be mined. It further provides partitioning of the search space and then generates few candidate sequences as possible by growing on the already mined frequent sequences. It applies apriori property as the search space which is being recursively traversed searching for frequent sequences.

i) *FREESPAN*: FreeSpan stands for Frequent Pattern-Projected Sequential Pattern Mining. FreeSpan algorithm was developed to mainly reduce the expensive candidate generation and Apriori testing, without disturbing its basic features. It integrates frequent items to recursively project the sequence database into projected databases while growing subsequence fragments in each projected database. Each projection partitions the database and confines further testing to progressively smaller and more manageable units. The trade-off is a considerable amount of sequence duplication as the same sequence could appear in more than one projected database [4]. However, the size of each projected database generally decreases rapidly with iterations. The major drawback of FreeSpan is that it generates much non trivial number of projected databases. If a particular pattern occurs in each sequence of a database, its projected database does not shrink.It is observed by analysis that it incurs higher costs for handling projected databases. This is happens because every possible combination of a likely candidate sequence is checked.

ii) *PREFIXSPAN:* The PrefixSpan stands for Prefix-projected Sequential Pattern Mining algorithm which represents the pattern-growth methodology. It finds the frequent items after scanning the sequence database once. The focus of this method is the projection only on frequent prefixes rather than projecting sequence databases by considering all possible occurrences of frequent subsequences. This is because by growing a frequent prefix any frequent subsequences can easily be found. Further the complete set of sequential patterns is obtained by iteratively growing subsequence fragments in each projected database. The PrefixSpan algorithm successfully discovered patterns employing the divide-and-conquer strategy, the cost of memory space might be high due to the creation and processing of huge number of projected sub-databases [4].

iii) *WAP-MINE:* WAP-Mine algorithm is a pattern growth and tree structure-mining technique with its WAP-tree structure. It is one of algorithms for mining frequent web access patterns from web access database [5]. It generates frequent web access patterns by recursively mining the web access pattern trees by use of WAP-tree [5].Here the sequence database is scanned two times to construct the WAP-tree from frequent sequences along with their support; a header table is maintained to point at the first occurrence for each item in a frequent item set, which is later tracked in a threaded way to mine the tree for frequent sequences, building on the suffix. Since it scans the database only twice, it can avoid the problem of generating explosive candidates as in apriori-based methods. WAP-Mine generates many intermediate data which lowers efficiency. Memory consumption is major problem of WAP-mine algorithm, as it iteratively reconstructs numerous intermediate WAP-trees during mining, and in particular, as the number of mined frequent patterns increases [4].

B. Apriori-Based Algorithms:

The Apriori [Agrawal and Srikant 1994] and AprioriAll [Agrawal and Srikant 1995] set the basis for a breed of algorithms that depend largely on the apriori property and use the Apriori-generate join procedure to generate candidate sequences. The apriori property states that "All nonempty subsets of a frequent item set must also be Frequent". It is also described as downward-closed. This refers to a sequence in which if a sequence could not pass the minimum support test then its entire super sequences would cause the test to fail.

The following principles are used for pruning candidate item sets

- If an item set is a sequence then all of its subsets must also be sequential.
- If an item set is in sequential then all of its supersets must also be insequential [6].

i) *GSP:* The GSP algorithm described by Agrawal and Shrikant [7] makes multiple database passes. This algorithm is not a main-memory algorithm. If the candidates do not occupy space in memory, the algorithm generates as many candidates as possible

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

which fits in memory and the data is scanned to count the support of these candidates. Frequent sequences resulting from these candidates are marked to disk; while the one without minimum support are deleted. This procedure is repeated until all the candidates have been counted [4].GSP algorithm scans the sequence database multiple times. Initially it scans to find all the frequent 1-items and forms the set of 1-element frequent sequences. In the next scans it generates step-wise longer candidate sequences from the set of frequent sequences and check their supports. GSP is efficient when the sequences are not long and the transactions are not large [9]. However, when the length of the sequences increases and the transactions are large, the number of candidate sequences generated may grow exponentially and GSP encounters difficulties.

ii) *SPADE:* SPADE stands for stands for Sequential Pattern Discovery using Equivalent Class. This algorithm proposed by M.J.Jaki [7] is an Apriori based vertical format sequential pattern mining algorithm i.e. the sequences are given in vertical order instead of horizontal format. It completes the mining in three passes of database scanning.This algorithm makes use of ID List technique to minimize the cost for computing support counts. It includes ID-List pairs where the first value refers customer sequence and the second value denotes transaction in it. The algorithm uses a breadth first or a depth first search method for finding new sequences. The transformation of a database from horizontal format to vertical format results into additional computing time. This therefore creates a need for additional storage space greater than the original sequence database [4].

iii) *SPIRIT:* The SPIRIT (Sequential Pattern Mining with Regular Expression Constraints) algorithm is to use regular expressions as flexible constraint specification tool [7]. It involves a generic user-specified regular expression constraint on the mined patterns. In order to push the constraining inside the mining process, in practice the algorithm uses an appropriately relaxed, that is less restrictive, version of the constraint. There exist several different versions of the algorithm that differs in the degree to which the constraints are enforced to prune the search space of pattern during computation. Choice of regular expressions (REs) as a constraint specification tool is motivated by two important factors. First, REs provide a simple, natural syntax for the succinct specification of families of sequential patterns. Second, REs possess sufficient expressive power for specifying a wide range of interesting, non-trivial pattern constraints [10].

III. PROPOSED ALGORITHM

SEQUENCE TREE ALGORITHM:

Sequential tree algorithm is used to find the frequent sequences in the log file. It involves two stages-construction of the sequence tree and mining of the sequence tree for finding frequent sequences. It takes as input the sequential database records which specify the various sequences of pages visited by the user in a particular session and also threshold. The frequent sequences that have been identified from the log file based on the minimum support threshold are obtained as output. The sequence tree algorithm is described in section 4.1 from steps (1) to (6).

Algorithm Sequence Tree

(1) Read the sequential database and create a map with key-value pairs. Key refers to the unique page name that was visited and sequence value refers to the frequency or number of time the page occurs.

(2) Sort the map in descending order of frequency of the keys

(3) Create a sequence tree using the following steps

A root node is termed as null

For every row of sequential database that is read

Attach the sequence as a branch to the root.

If the sequence is already present

increment the counter

else

a new branch is created with that sequence.

(4) Include a header table with number of rows equal to sequences with value one.

Each row of header table is a linked list of nodes which specify the position where nodes are present.

(5) Perform mining process

For every row in the header table

If frequency < minimum support threshold

Ignore the row of header table

else

For every node in the particular linked list

If frequency < minimum support threshold

Ignore the path

else

Traverse the tree up till the root and store the path in the file as frequent sequences.

EndFor

EndFor

(6) The frequent sequences identified from the algorithm are stored in a file

CONCLUSION

In this paper a small attempt is made to provide some information of the increasing use of Web mining and how the various techniques of pattern discovery helps in building business plans especially in the area of e-business. Web usage mining techniques is applied to large web repositories to extract usage patterns. The sequence tree algorithm is one such web usage mining technique which extracts frequent sequential patterns by formation of a tree. The running time and number of patterns generated are examined. The results show that the Sequence Tree algorithm shows faster running time than the standard GSP algorithm It also discovers more patterns than the standard GSP algorithm.

ACKNOWLEDGMENT

Our sincere thanks to all the people who have contributed in accomplishing this paper work.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

REFERENCES

[1] Agrawal R and Srikant R, "Mining Sequential Patterns", in Int'l. Conf. Data Engineering (ICDE 95), (1995) 3-14

[2] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements,"Proc. Fifth Int'l Conf.Extending Database Technology (EDBT '96), pp. 3-17, Mar. 1996

[3] Vishal S. Motegaonkar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2486-2492

[4] Manisha Valera, Kirit Rathod "A Novel Approach of Mining Frequent Sequential Pattern from Customized Web Log Preprocessing" (IJERA) Vol. 3, Issue 1, January -February 2013, pp.269-380

[5] Sen Yang Fudan Univ,Shanghai Jiankui Guo ; Yangyong Zhu "An Efficient Algorithm for Web Access Pattern Mining"

[6] R. Sridevi et al Int. Journal of Engineering Research and Applications ISSN: 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.829-83

[7] Chetna Chand, Amit Thakkar, Amit Ganatra, "Sequential Pattern Mining: Survey and Current Research Challenges", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012

[8] M. Zaki, "SPADE: An efficient algorithm for mining frequent sequences", Machine Learning, 2001.

[9] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Hua zhu "Mining Access Patterns from Web logs"

[10] K Suneetha , Dr M. Usha Rani "A survey onimproving the efficiency of prefix spansequential pattern mining algorithm"- International Journal of Conceptions on Computing and Information TechnologyVol.2, Issue 3, March' 2014; ISSN: 2345 98081|2 6