ISSN:2231-0711
www.ijcset.net

An Efficient Algorithm for Frequent Pattern Mining using Web Analysis Approach

[1]Monika Verma, [2]Shikha Pandey
[1]*Research Scholar*,[2]*Assistant Professor*
Department of Computer Science & Engineering
Rungta College of Engineering and Technology, Bhilai

[1]monika04verma@gmail.com,
[2]shikhamtech2008@gmail.com

***Abstract*- In this paper a complete structure with new modified algorithm for mining and finding web usage patterns from a Web Application is presented. The Web Application can be a real Web site that has all the challenging aspects of real-life Web usage mining, including evolving user profiles and external data describing ontology of the Web content. A name is given to each phase used in this framework i.e. Data Collection & Preprocessing, Pattern Discovery and Pattern Analysis. Our research work is started by invoking pages of our application over Internet, user may navigate from that to his/her intended destination. In this we recognize and analyze different patterns (page sequence) invoked in intra page navigation by different users then we will count the frequency of a particular pattern by the means of number of hits. In Pattern Analysis we have developed a modified algorithm for mining frequent pattern that consume less time and memory usage as compare to traditional frequent pattern mining algorithm.**

***Keywords*- Web usage pattern, Web Application, Data Collection & Preprocessing, Pattern Discovery, Pattern Analysis, Frequent pattern mining.**

1. INTRODUCTION

1.1 Pattern Discovery in Web Usage Mining

Web mining can be categorized into three different classes based on which part of the Web is to be mined. These three categories are (i) Web content mining, (ii) Web structure mining and (iii) Web usage mining. [1]

Table 1: MaxMind Geo IP

Hostname
Country Code
Country Name
Region
Region Name
City
Postal Code
ISP
Organization
Metro Code
Area Code

1.1.1 Web usage mining

It refers to the automatic discovery and analysis of patterns in clickstream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. The goal is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages, objects, or re-sources that are frequently accessed by groups of users with common needs or interests.

1.2 Web Usage Mining Process

Web usage mining is a powerful tool to analyzing, designing and modifying a Web site structure as well as it is also useful to understanding and analyzing the site visitor's behavior in two aspects: i) The interest and information one access. ii) The way to access this information. Web usage mining activities pacify two different aspects: how designers expect to be used the site by the visitors and the way visitors effectively using the site.

Web usage mining can be divided in at least three different phases namely [2]

i) Data preparation,
ii) Pattern discovering
iii) Pattern analysis and visualization.

1.2.1 Data Preparation

In Data Preparation phase the web log data must be cleaned, filtered, integrated and transformed in such a way that the irrelevant and redundant data can be removed, user session and transaction can identified.

1.2.2 Pattern Discovery

After data preparation phase, the pattern discovery method should be applied. This phase consists of different techniques derived from various fields such as statistics, machine learning, data mining, pattern recognition, etc. applied to the Web domain and to the available data.

1.2.3 Pattern Analysis

This is the final step in the Web Usage Mining process. After the preprocessing and pattern discovery, the obtained usage patterns are analyzed to filter uninteresting information and extract the useful information. [7]

A lot of work has been done in this direction. Different researcher proposed their view to find out the web usage mining areas for different types of approach and work. In the field of Web Mining more specifically Web Usage Mining an extraction of pattern sequence of navigations and session tracking of any user using Web Analysis Approach is

difficult and along with the pattern sequence if we have to identify other details of any particular user like geographical location, IP address, name of web browser and operating system being used it become more tedious task. [1]

2. WORK DONE EARLIER

In the sequence of this research work our first paper tries to generalize problem by targeting the following points

A. To find out the different areas concerning to the different users like
   I. Date: The date of visit.
   II. Page: The page that is visited.
   III. Client Info: The information of the client.
   IV. Client IP: IP address of the client.
   V. Platform: OS that is used by the Users.
   VI. Web Browser: The type of Browser used by the user.
B. To find out the numbers of hit for the particular page separately.
C. To find out the browsing patterns of different users of the web sites.

2.1 Solution on these targeted points has been given by our first paper as mentioned bellow:

2.1.1 Scripts make use of the MaxMind GeoIP Javascript, which is freely usable as long as you adhere to the Terms of Use below.

These scripts may be used to control access to your web site based upon where the user is located. This process has obvious limitations: in order for it to work, the user must have a javascript-enabled browser. That's not to say this is not a reasonably good control mechanism for your web site; but this is by no means a 100% solution to filtering traffic.

2.1.2 MaxMind GeoIP JavaScript Web Service

GeoIP JavaScript is a service offered by MaxMind to return the Country, Region, City, Latitude, and Longitude for your web visitors. It uses JavaScript and is very easy to program, works on both static and dynamically served web pages. If you only need to display the country, use GeoIP Country JavaScript service. In order to use this JavaScript on our website, a link back to the www.maxmind.com website should be provided, or a JavaScript attribution-free license can be purchased for $250/year.

Example:

Country Code: IN
Country Name: India
Region:
Region Name:
City:
Postal Code:
Latitude: 20.0000
Longitude: 77.0000

2.1.3 Source code

For example to find the country code the snap of the source code for this is

```
<script language="JavaScript" src=
"http://j.maxmind.com/app/geoip.js">
</script>
<br>Country Code:
<script language= "JavaScript">
document.write(geoip_country_code());
</script>
```

2.1.4 GeoIP Geo-location Products

MaxMind GeoIP products deliver information on the geographic location and connection type of Internet visitors. APIs and associated documentation for databases can be found in the GeoIP Support Center. We can purchase multiple databases with one order by adding them to the shopping cart before checking out. [1]

3. TRADITIONAL METHODOLOGY

3.1 Apriori Algorithm

Apriori Algorithm [10] is for finding the association rules was first proposed in 1994. This algorithm may consist of two parts. In the first part, those itemset that exceed the minimum support requirement are found, such itemset are called frequent itemsets. In the second part, the association rules that meet the minimum confidence requirement are found from the frequent itemsets.

3.1.1 Limitation of Apriori Algorithm

1. The algorithm is of low efficiency, such as firstly it needs to repeatedly scan the database, which spends much in I/O.
2. Needs several iterations of the data.
3. Huge number of candidate set.
4. Uses a uniform minimum support threshold.
5. It creates a large number of 2-candidate itemsets during outputting frequent 2-itemsets.
6. Difficulties to find rarely occurring events.
7. It doesn't cancel the useless itemsets during outputting frequent k- itemsets.

3.2 Methods to Improve Apriori's Efficiency

- **Hash-based itemset counting:** A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent.
- **Transaction reduction:** A transaction that does not contain any frequent k-itemset is useless in subsequent scans.
- **Partitioning:** Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB.
- **Sampling:** mining on a subset of given data, lower support threshold + a method to determine the completeness.
- **Dynamic itemset counting:** add new candidate itemsets only when all of their subsets are estimated to be frequent.

4. PROPOSED METHODOLOGY

4.1 Dynamic Candidate Generation algorithm

It is an extension to Apriori algorithm and taking the concept of Dynamic itemset counting, which is a method of improving efficiency of Apriori algorithm.

- Used to reduce number of scans on the dataset.
- Alternative to Apriori Itemset Generation
- Itemsets are dynamically added and deleted as transactions are read
- Relies on the fact that for an itemset to be frequent, all of its subsets must also be frequent, so we only examine those itemsets whose subsets are all frequent.

A dynamic candidate generation technique is proposed by taking the concept of Dynamic Itemset counting and Apriori algorithm, in which the database is partitioned into blocks marked by start points. In this variation, new candidate itemsets can be added at any start point, unlike in Apriori, which determines new candidate itemsets only immediately prior to each complete database scan. The technique is dynamic in that it estimates the support of all of the itemsets that have been counted so far, adding new candidate itemsets if all of their subsets are estimated to be frequent. The resulting algorithm requires two database scans. Further, dynamic itemset counting algorithm, an extension to Apriori algorithm used to reduce number of scans on the dataset. It was alternative to Apriori Itemset Generation .In this, itemsets are dynamically added and deleted as transactions are read .It relies on the fact that for an itemset to be frequent, all of its subsets must also be frequent, so we only examine those itemsets whose subsets are all frequent. Both Apriori and DCG are based on candidate generation.

4.1.1 Advantage over Apriori Algorithm

There are some advantages of using Dynamic Candidate Generation Algorithm over Apriori algorithm.

- DCG algorithm can dynamically incorporate new itemsets to be added, it is not necessary to wait.
- Nodes can proceed to count the itemsets they suspect are candidates and make adjustments as they get more results from other nodes
- DCG algorithm reduces the number of scans required by combining the counting for the number of itemset as soon as it appears and might be necessary to count it.
- Itemsets are dynamically added and deleted as transaction are.

4.2 Pseudocode of Algorithm

Algorithm:

1. Mark the empty itemset with a solid square. Mark all the 1-itemsets with circles. Leave all other itemsets unmarked.
2. While any circled itemsets remain:
   A. Read *M* transactions (if we reach the end of the transaction file, continue from the beginning). For each transaction, increment the respective counters for the itemsets that appear in the transaction and are marked with circle.
   B. If a dashed circle's count exceeds *minsupp*, turn it into a square. If any immediate superset of it has all of its subsets as solid or dashed squares, add a new counter for it and make it a circle.
   C. Once a square itemset has been counted through all the transactions, make it marked and stop counting it.

4.3 Example of Algorithm

Itemset lattices:

An itemset lattice contains all of the possible itemsets for a transaction database. Each itemset in the lattice points to all of its supersets. An itemset lattice can help us to understand the concepts behind the DCG algorithm.

Example: *minsupp* = 25% and M = 2

Table 2: Transaction Database

TID	*A*	*B*	*C*	*D*	*E*	*F*
T1	1	1	0	1	1	1
T2	0	0	1	1	0	1
T3	1	0	1	0	1	0
T4	1	1	1	0	1	0
T5	0	0	1	1	1	1

STEP 1: Itemset lattice for the above transaction database for three itemset before any Transaction are read.

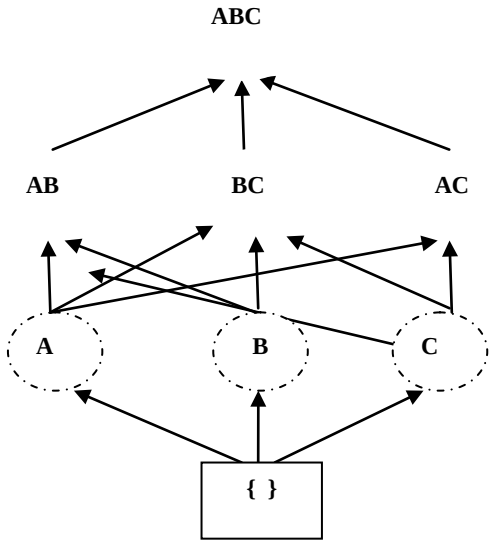

Figure 1: Itemset Lattice 1

Counters: A=0, B=0, C=0 Empty itemset is marked with a solid box. All 1-itemsets are marked with dashed circles.

STEP 2: After M transactions are read

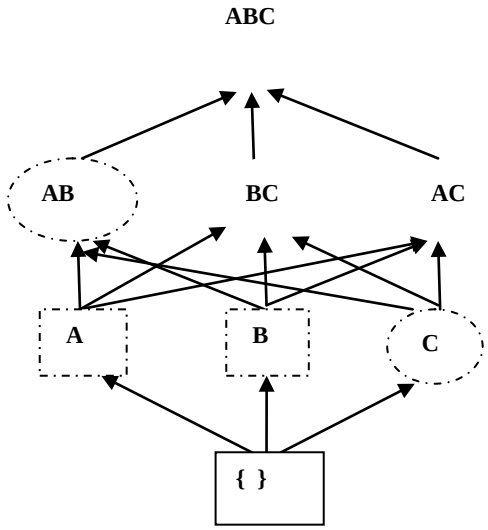

Figure 2: Itemset Lattice after M transactions read

Counters: A = 2, B = 1, C = 0, AB = 0, we change A and B to dashed boxes because their counters are greater than minsup (1) and add a counter for AB because both of its subsets are boxes.

STEP 3: Counters: A = 2, B = 2, C = 1, AB = 0, AC = 0, BC = 0

C changes to a square because its counter is greater than minsup A, B and C have been counted all the way through so we stop counting them and make their boxes solid. Add counters for AC and BC because their subsets are all boxes.

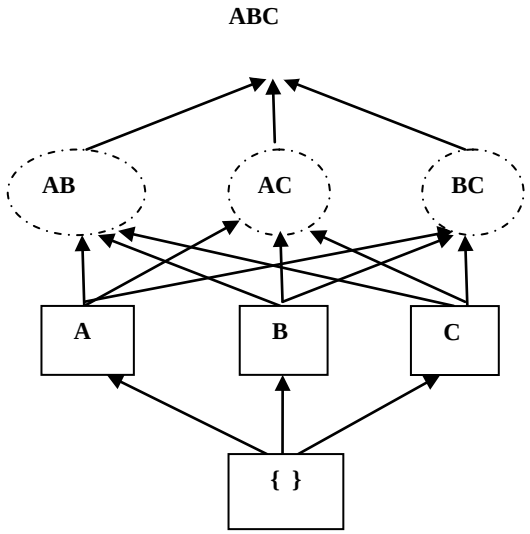

Figure 3: Some Processed Item in Itemset Lattice

STEP 4: Counters: A = 2, B = 2, C = 1, AB = 1, AC = 0, BC = 0

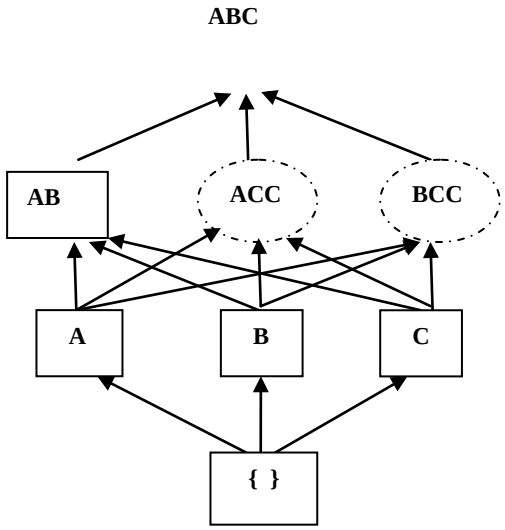

Figure 4: Next Itemset lattice

AB has been counted all the way through and its counter satisfies minsup so we change it to a solid box. BC changes to a dashed box.

STEP 5: Counters: A = 2, B = 2, C = 1, AB = 1, AC = 0, BC = 1

AC and BC are counted all the way through. We do not count ABC because one of its subsets is a circle. There are no dashed itemsets left so the algorithm is done.

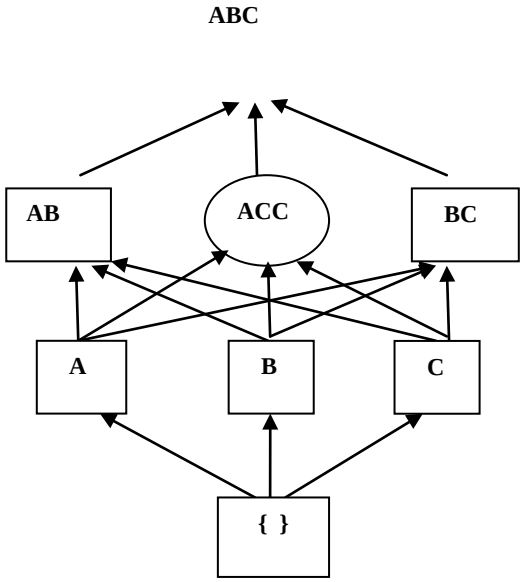

Figure 5: Final Itemset Lattice

5. Experimental Results and Analysis

Here we have given Analysis and comparison of Apriori algorithm and dynamic candidate generation algorithm in terms of time and memory it consumes.

Table 3: Result Based on Dataset

PARAMETER	ALGORITHM	
	APRIORI ALGORITHM	DYNAMIC CANDIDATE GENERATION ALGO.
TIME	0.641 seconds	0.046 seconds
MEMORY USAGE	858672 kb	337568 kb

We have found that our algorithm takes less time for execution as well as consumes less memory in comparison with Apriori algorithm.

5.1 Snapshot of Apriori Algorithm

Figure 6: Output of Apriori Algorithm

5.2 Snapshot of Dynamic Candidate Generation algorithm

Figure 7: Output of Dynamic Candidate generation Algorithm

6. Conclusions

A lot of people have implemented and compared several algorithms that try to solve the frequent itemset mining problem as efficiently as possible. In this paper a new method is introduced by us. This newly implemented algorithm is given a name as “Dynamic Candidate Generation” algorithm. Further to access its efficiency and performance for large dataset it has been compared with Apriori algorithms. DCG i.e. Dynamic Candidate Generation algorithm has been proved more efficient with respect to the existing algorithm in terms of time and memory consumption. DCG (Dynamic Candidate Generation) algorithm consumes less time and memory than the Apriori algorithm

REFERENCES

[1] Monika Verma, Mridu Sahu, "Extraction of Pattern Sequence and Geographical Location of User using Web Analysis Approach" IJCSET February 2012 Vol 2, Issue 2,877-879.

[2] Shahnaz Parvin Nina, Md. Mahamudur Rahaman, Md. Khairul Islam Bhuiyan, Khandakar Entenam Unayes Ahmed, "Pattern Discovery of Web Usage Mining", 2009 International Conference on Computer Technology and Development IEEE.

[3] Olfa Nasraoui, Maha Soliman, Esin Saka, Antonio Badia and Richard Germain, "A Web Usage Mining Framework for Mining Evolving User Profiles in Dynamic Web Sites", IEEE Transactions On Knowledge And Data Engineering, Vol. 20, No. 2, February 2008.

[4] Mahesh Thylore Ramakrishna, Latha Kolal Gowdar, Malatesh Somashekar Havanur, Banur Puttappa Mallikarjuna Swamy," Web Mining: Key Accomplishments Applications and Future Directions", 2010 International Conference on Data Storage and Data Engineering IEEE.

[5] Craig P. Oosthuizen, Janet Wesson & Charmain Cilliers," Processing Web Logs in order to Mine Web Usage Patterns".

[6] Maximilian Viermetz, Carsten Stolz K., Vassil Gedov, Michal Skubacz, "Relevance and Impact of Tabbed Browsing Behavior on Web Usage Mining", Proceedings of the International Conference on Web Intelligence, 2006.

[7] Qingtian Han, Xiaoyan Gao, Wenguo Wu," Study on Web Mining Algorithm Based on Usage Mining", 2008 IEEE.

[8] Cheng Zheng Yong Fang Yaping Shen,"The Implementation of the Web Mining Based on XML Technology", 2009 International Conference on Computational Intelligence and Security.

[9] Yassine Mrabet, Khaled Khelif, Rose Dieng-Kuntz,"Recognising Professional-Activity Groups and Web Usage Mining for Web Browsing Personalisation", 2007 International Conference on Web Intelligence.

[10] Liping Sun, Xiuzhen Zhang, "Efficient Frequent Pattern Mining on Web Log Data", Advanced Web Technologies and Applications: Sixth Asia-Pacific Web Conference, APWeb 2004, Berlin, 15 March 2004.

[11] Jaideep Srivastava, Prasanna Desikan, Vipin Kumar," Web Mining - Accomplishments & Future Directions.