International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 2, Issue 5 (March 2013) PP: 16-21

Web Mining Patterns Discovery and Analysis Using Custom-Built Apriori Algorithm

Latheefa.V[1], Rohini.V[2]
[1, 2] *Department of Computer Science, Bangalore, India,*
Christ University,

Abstract*: Mining web data in order to extract useful knowledge from it has become vital with the wide usage of the World Wide Web. Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from web data. In this paper, a Custom-Built Apriori Algorithm is proposed for the discovery of frequent patterns in the web log data. This study has also developed a tool to discover the frequent patterns, association rules in the web log data. In this study, Web log data of Christ University web site is analysed. The experiments conducted in this study proved that Custom-Built Apriori Algorithm is efficient than Classical Apriori Algorithm as it takes less time.*

Key words*: Web Usage Mining, Custom-Built Apriori, FrequentPatterns, Association Rules.*

I. Introduction

The web is an open medium. With this openness web has grown enormously. One can find any information on web which is its main strength. As web and its usage continue to grow, there is an opportunity to mine the web data and extract useful knowledge from it. Web mining is the application of data mining techniques to find interesting and potentially useful knowledge from the web data. Web mining can be broadly divided into 3 distinct categories based on the kinds of data to be mined: 1.Web Content Mining is the process of extracting useful information from the contents of web documents. 2. Web Structure Mining can be regarded as the process of discovering structure information from the web.3.Web Usage Mining is understanding user behaviour in interacting with the web or with a web site[1]. One of the aims is to obtain information that may assist web site reorganization or assist site adaptation to better suit the user needs.

The applications of Web Usage Mining are [2]:

- By determining the access behaviour of users, needed links can be identified to improve the overall performance of future accesses
- Web usage patterns are used to gather business intelligence to improve customer attraction, and sales.
- Personalization for a user can be achieved by keeping track of previously accessed pages. These pages can be identified to improve the overall performance of future accesses.

In this paper, web log file of size 70MB is preprocessed and thenanalysed. The contents of the paper are organized as follows: Section IIbriefs the structure of web log file, Section III briefs the methodology, Section IV discusses the experiment results and finally conclusion and future work are mentioned in Section V.

II. Web log file

A web log file is a file to which the webserver writes information each time a user requests a resource from that particular website[3]. Web usage information takes the form of web server log files, or web logs. For each request from a user's browser to a web server, a response is generated automatically. This response takes the form of a simple single-line transaction record that is appended to an ASCII text file on the web server. This file is called as web server log.Figure 1 shows the fragment from webserver log of Christ University website.

115.119.146.168 - - [14/Dec/2011:16:37:59 +0530] "GET /block/block.html HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows NT 5.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1"122.167.80.114 - - [14/Dec/2011:16:37:59 +0530] "GET /webadmin/depgallery/_654.jpeg HTTP/1.1" 200 283307 "http://www.christuniversity.in/Management%20Studies/deptcourses.php?division=Commerceand Management&dept=23" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SearchToolbar 1.2; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; MDDC;)

Fig.1 Fragment of web log file

A. Common Log Format

Web logs come in various formats, which vary depending on the configuration of the web server. The common log format is supported by a variety of web server applications and includes the following seven fields:

- Remote host field
- Identification field
- Auth-user field
- Date/time field
- HTTP request
- Status code field
- Transfer volume field

III. Methodology

The methodology contains the following steps:

- Data Pre-processing
- Developing custom built apriori algorithm
- Identifying web patterns using custom built apriori algorithm
- Analysing the discovered web patterns.
- Developing a standalone application for web mining.

The general flow of project methodology is diagrammatically represented in the figure 2.

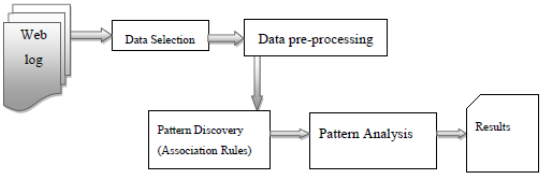

Fig.2Methodology flow

A. Data Preprocessing

An important task in any data mining application is the creation of a suitable target data set to which data mining algorithms can be applied. This is particularly important in web usage mining due to characteristics of clickstream data and its relationship to other related data collected from multiple sources and across multiple channels.

In this analysis WUMPrep tool is used for web log data preparation. WUMPrep contains a set of Perl scripts for cleaning web log file of irrelevant and automatic requests and creating sessions in it[3]. In this study web log data is pre-processed to make it suitable for mining. Data preprocessing consists of sequence of following tasks:

- Data Cleaning.
- Sessionization.
- Robot Detection.
- Converting Web log File to ARFF (Attribute Relation File Format).

B. Further Data Preprocessing

In this process the requests that are made by the users are replaced with the folders to which they belong. This process is implemented by a java application developed in this thesis. The fragment of weblog file generated by the java application is shown in the figure 3

*241393:19|180.215.27.123 - - [14/Dec/2011:16:38:41 +0530] "GET **Commerce** HTTP/1.1" 200 9589*
*241393:28|117.230.171.53 - - [14/Dec/2011:16:38:53 +0530] "GET **Psychology** HTTP/1.1" 200 11991*
*241393:29|59.145.142.36 - - [14/Dec/2011:16:39:27 +0530] "GET **School%20of%20Education** HTTP/1.1" 200 8140*

Fig.3 Folder request

C. Web log to ARFF

The proposed apriori algorithm accepts input files in a format called ARFF (Attribute Relation File Format).

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two distinct sections. The first section is the **Header** information, which is followed the **Data** information. Lines that begin with a % are comments.

The **Header** of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. The Header contains three parts **@RELATION**, **@ATTRIBUTE** and **@DATA.** All these declarations are case insensitive.
The relation name is defined as the first line in the ARFF file. The format is:
@relation <relation-name>
Attribute declarations take the form of an ordered sequence of **@attribute** statements. Each attribute in the data set has its own **@attribute** statement which uniquely defines the name of that attribute and its data type. The order the attributes are declared indicates the column position in the data section of the file. The format for the **@attribute** statement is:
@attribute <attribute-name><data type>
where the <*attribute-name*>must start with an alphabetic character,
<data type > is Boolean in the analysis.
The ARFF Data section of the file contains the data declaration line and the actual instance lines.
The **@data** declaration is a single line denoting the start of the data segment in the file. The format is: @data
Each instance is represented on a single line, with carriage returns denoting the end of the instance.
{ 0 t, 1 t, 2 t, 3 t }
There are two possible representatives of data in the Arff file – *dense and sparse*. In both formats a web page must be considered as a binary attribute that takes the values true or false in each transaction, depending on whether the page occurs in the transaction or not. Since the occurrence of web pages in transactions (user sessions) is scarce, the sparse format is chosen for presenting the sessions in weblog data. Since WUMPrep does not contain any script that converts web log data into either sparse or dense Arff file format, a utility application is developed in java for this purpose. The code for this application is found in appendix. A fragment of Arff file that is generated from this application is shown in fig 4.

```
@relation christweblog
@attribute Biotechnology {f,t}
@attribute Botany {f,t}
@attribute Chemistry {f,t}
@attribute Economics {f,t}
@data
{ 0 t, 1 t } { 0 t }
{ 0 t }
{ 0 t, 1 t, 2 t, 3 t }
{ 2 t, 3 t }
```

Fig.4 web log to ARFF

D. Apriori Algorithm

Apriori algorithm is proposed by R.Agrawal and R.Srikant for finding the association rules. This algorithm can be considered to consist of two parts. In the first part, those itemsets that exceed the minimum support requirement are found. Such itemsets are called frequent itemsets. In the second part, the association rules that meet the minimum confidence requirement are found from the frequent itemsets[4].

Problems with Apriori Algorithm

- The number of candidate itemsets grows quickly and can result in huge candidate sets.
- The Apriori algorithm requires many scans of the database. If n is the length of the longest itemset, then (n+1) scans are required.
- Many trivial rules are derived and it can often be difficult to extract the most interesting rules from all the rules derived.

The Apriori algorithm assumes sparsity since the number of items in each transaction is small compared with the total number of items. The algorithm works better with sparsity. Some applications produce dense data (i.e. many items in each transaction) which may also have many frequently occurring items. Apriori is likely to be inefficient for such applications.

E. Custom-Built Apriori Algorithm

In this research custom-built apriori is proposed to improve the performance of the Apriori algorithm. The changes done in the proposed algorithm are

- Pruning operation is performed only on the candidate itemsets whose size > 2.
- For generation of frequent itemsets of size k only the transactions whose size >= k are considered.

Custom-Built Apriori Algorithm:-

Input:

- D, a database of transactions
- Minimum support count threshold

Output: L, Frequent itemsets in D

Method:

1. C_1 := All the 1-item sets
2. Read the database to count the support of C1 to determine L_1
3. L_1:={frequent 1-itemsets}
4. K:=2
5. While($L_{k-1} \neq$ □) do

begin

5.1 C_k:=gen_candidate_itemsets with the given L_{k-1}
5.2 If(k>2) then
5.3 Prune(C_k)
5.4 For all candidates in C_k do
5.5 count the number of transactions of at least length k that are common in each item □ C_k
5.6 L_k:=All candidates in C_k with minimum support
5.7 k:=k+1

end

6. Answer:=$U_k L_k$

F. Web Usage Association Rule Discovery Software

For the purpose of this research, an independent windows desktop application for the association rule discovery in the web log data is developed. The application is implemented using java programming language. It is an object oriented application specialized for the discovery of association rules in web log data, rather than in the general relational database data. Figure 4.shows the user interface of the software.

When an input file containing web log usage data is opened, the web log data is transformed into directories mentioned in the properties file. This file will be converted into ARFF format when the user clicks on the generateARFFFile button. Once the ARFF is prepared, the web log data is ready for execution and obtain the association rules. For Rules to obtain there is a need to satisfy constraints as support and confidence which can be entered in the text fields.

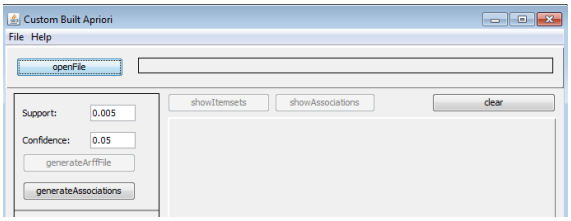

Fig. 5 Custom built software to analyse association rules based on support and confidence

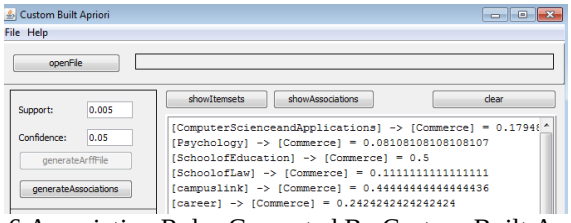

Fig. 6 Association Rules Generated By Custom-Built Apriori

IV. Results and Discussion

The analysis has been conducted on the web log data of size ~70MB. The first step in the web log mining process is to clean web log data of irrelevant and automatic web requests prior to running association rule discovery algorithm, as well as to group web requests into visitor sessions. The WUM Prep tool has been used for web log data.

Perl Script	Resultant File Size	% of File Reduction
logFilter.pl	6230KB	89
sessionized.pl	7550KB	79
detectRobots.pl	5858KB	83

Fig. 7WUMPrepTool Statistics

A. Itemset Level Generations

In Apriori Algorithm, the number of Candidate itemsets grows quickly and can result in huge candidate sets. The proposed apriori algorithm will reduce the number of transactions that need to be verified at each level of large item set generation.

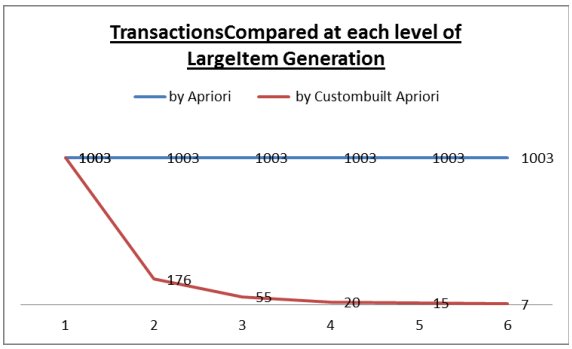

Fig. 8 Transactions compared at each level of Large Item generation

A number of tests have been conducted with various support levels to find the range of support levels in which the satisfactory and meaningful association rules are generated.

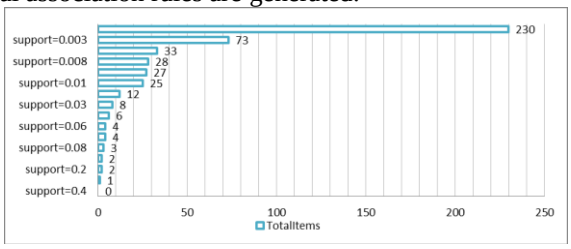

Fig 9 Large Items Generation Based On Support Level

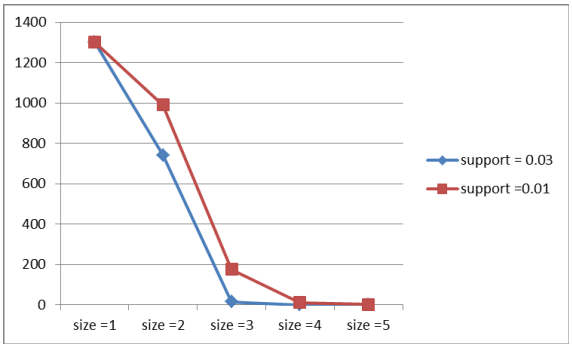

Fig.10Behaviour of Accepted candidate Items over different Supports

V. Conclusion

The analysis was performed with an objective of knowledge discovery in web data by applying data mining techniques. The analysis was carried out in four steps: data collection, data preprocessing, pattern discovery using custom-built apriori algorithm, pattern analysis.

In this analysis Custom-built apriori algorithm is developed. The proposed algorithm has successfully discovered the frequent patterns from web data.A tool has been developed in java for implementing the custom-built apriori algorithm.This tool is specialized for the discovery of association rules in web log data rather than in the general relational database data.For converting preprocessed web log data to ARFF java program has been

developed.The generated patterns are analysed by executing the custom-built apriori algorithm with different support and confidence values

A.Future Work: This analysis can be extended by implementing other interesting measures such as lift in order to fine tune the results. The proposed work can also be implemented by the adoption of more advanced techniques/tools such as fuzzy associations, rule mining algorithm using the interesting measures such as fuzzy support, fuzzy confidence, fuzzy interest and fuzzy conviction etc.

References

[1] R. Cooley, B. Mobasher and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web," IEEE, 1997.
[2] K. R. Suneetha and Dr. R. Krishnamoorthi, "Identifying User Behavior by Analyzing Web ServerAccess Log File, " International Journal of Computer Science and Network Security, , vol. 9, no. 4, 2009.
[3] M. Dimitrijevic and Z. Bošnjak, "Discovering Interesting Association Rules in the Web Log Usage Data," Interdisciplinary Journal of Information, Knowledge, and Management, vol. 5, 2010.
[4] G.K.Gupta, Introduction to Data Mining with Case Studies, New Delhi: EEE PHI , 2011.
[5] C. A. R. C. Goswami D.N., "An Algorithm for Frequent Pattern Mining Based On Apriori," International Journal on Computer Science and Engineering, vol. 02, 2010.
[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in 20th Int. Conf. Very Large Data Bases.
[7] Han and Kamber, Data Mining Concepts and Techniques, Secoond ed., New Delhi: Elsevier, 2006.