Implementation Of Web Usage Mining Using Log Markup Language (LOGML) and Closed Association Rule Mining Algorithm (CHARM)

Gant Gaw Wutt Mhon, Nang Saing Moon Kham
University of Computer Studies, Yangon
wuttmhon@gmail.com

Abstract

Web usage mining is the discovery of user access patterns from Web usage logs. The paper presents two XML (Extensible Markup Language) 1.0 applications and a web data mining application which utilizes it to extract web data from web log files. The two XML 1.0 applications are: LOGML (Log Markup Language) is a web-log report description language and XGMML (Extensible Graph Markup and Modeling Language) is a graph description language. As a case study, the system implements a sample website for web graph information and usage information. The main goal of this paper is that web graph information transforms to XGMML document and then this graph information and cleaned usage information of finished user sessions transforms to LOGML document. The next goal is that these cleaned data with LOGML document are mined with CHARM (Closed Association Rule Mining) algorithm to implement the most frequently accessed pages from one site. CHARM is an efficient algorithm for mining all closed frequent itemsets (set of all subsets of items).

Keywords: Web usage Mining, XGMML, LOGML, CHARM.

.

1. Introduction

Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. There are four main mining techniques that can be applied to Web access logs to extract knowledge [3]: Sequential-pattern-mining-based, Association-rule-mining-based, Clustering-based and Classification-based. As the Association-rule-mining-based [7] from these mining techniques: frequent sets are to discover all frequent subsets in the database of collection of transactions which are simply subsets of primitive items.

In Web mining, Web usage mining applies data mining method to discover web usage pattern through web usage data. And recently XML has also gained wider acceptance in both commercial and research establishments. So this paper proposed the mining the most user accessed web pages from one site by using CHARM algorithm to mine from LOGML files and XML to display the results.

This system intends for users who need to reduce the time on using internet by knowing the most populated web pages from one website. From the side of business, they can know the most populated web pages from their website. From the system can study XML languages can be used in Web usage mining and know XGMML language can be used in web characterization. And the system can implement LOGML document extracted from the database of web access logs that can perform several mining tasks.

And then the proposed system can discover usage patterns from web data to understand and better serve the needs of web-based application. The next objectives are that the system can know CHARM is an efficient algorithm for enumerating the set of all frequent closed itemsets than other association algorithm and implement the most frequently accessed pages from one site. The system can also provide for the purpose of ease accessing and time saving

The paper is organized as follows. Section 2 provides the related work for the system. Section 3 describes the background theory. Section 4 presents the implementation of the system. Section 5 displays conclusion of the system.

2. Related Work

In past, many association mining have also been applied to web usage mining. Chen et al. [4] presented the term of mining traversal patterns which means the capturing of user access patterns in distributed information providing environments such as Web. But Chen et al. cannot make any distinction between references used for various purposes and then may discover too many sequences from the transactions identified using the MFR (Maximal Forward Reference) techniques.

Apriori algorithm or downward closure property [6] was the first efficient and scalable method for mining associations. It starts by counting frequent items, and during each subsequent pass it extends the current set of frequent itemsets by one more item, until no more frequent itemsets are found. Then Apriori uses a pure bottom-up search method and it can improve the efficiency of the level-wise generation of frequent itemsets and reduce the search space. But it can only examine all subsets of a frequent itemset but it is not possible to identify the closed frequent itemsets. The proposed system uses CHARM algorithm because it can find maximal frequent itemsets.

3. Web Usage Mining

Web mining involves a wide range of applications that aims at discovering and extracting hidden information in data stored on the Web. And then Web mining is to provide a mechanism to make the data access more efficiently and adequately. Web mining can be categorized into three different classes based on which part of the Web is to be mined [5]. These three categories are (i) Web content mining, (ii) Web structure mining and (iii) Web usage mining. In this paper, we focus on the Web usage mining.

Web usage mining is the task of discovering the activities of the users while they are browsing and navigating through the Web. And the phases of web usage mining are preprocessing, mining and applying mining results. In preprocessing phase, logfiles are transformed into a form that is suitable for mining.

In proposed system, web usage mining is implemented with LOGML generator as preprocessor to extract the set of nodes of finished user sessions and then these extracted data are mined with CHARM algorithm to provide the association rules.

3.1 Log Markup Language (LOGML)

LOGML is XML 1.0 application and it is a web-log report description language. Log reports are the compressed version of logfiles [1]. Web master need to generate LOGML reports of log files because of insignificant size of LOGML. So the system generated web-log reports with LOGML format for a web site from web log files and the web graph. The root element of a LOGML document is the logml element. LOGML document has three sections: the first section is a graph that describes the log graph of the visits of the users to web pages and hyperlinks. The second section is the additional information of log reports such as top visiting hosts, top user agents, top keywords, etc and the third section is the report of the user sessions.

The first and third sections can get from the XGMML. So the first section uses XGMML to describe the graph and the third section also uses XGMML to extract the finished user sessions. Each user session is a subgraph of the log graph. The subgraphs are reported as a list of edges that refer to the nodes of the log graph. The namespace for LOGML is: http://www.cs.rpi.edu/LOGML and the suffix for LOGML elements is lml.

3.1.1 LOGML Generator

LOGML generator [1] can read a common log file and generate a LOGML file. So it can also read all web log lines, only the finished user sessions are reported in the logml document. Therefore LOGML generator can output the pattern information with logml document which are the information of finished user sessions. For checking user session is finished or not, LOGML generator use User Manager. User Manager is a module in LOGML generator. User Manger is invoked for each web log line that is processed. And it has access to the container of the user sessions and the web graph of the web site of the web logs so the user manager can add user sessions and get metadata information from the web graph. There are six steps in the User Manager to create and finish user sessions.

3.1.2 Extensible Graph Markup and Modeling Language (XGMML)

XGMML [1] is an XML 1.0 application based on graph modeling language which is used for graph description. An XGMML document describes a graph structure. The main element of XGMML are : graph, node, edge, att and graphics. The graph element is the root element of an XGMML valid document and then these graph element may not be unique in the XGMML document.

The global attributes of XGMML are: id, name and label. For a graph, G= (V, E) is a set of nodes V and a set of edges E. Each edge is either an ordered (directed graph) or unordered (undirected) pair of nodes. Graphs can be described as data objects whose elements are nodes and edges. The system represented the web pages are nodes and the hyperlinks are edges in a graph. The node element describes a node a graph and the edge element describes an edge of a graph. The namespace for XGMML is: http://www.cs.rpi.edu/XGMML and the suffix for the XGMML elements is xgmml.

3.2 Closed Association Rule Mining Algorithm (CHARM)

CHARM algorithm [1] is an efficient algorithm for all closed frequent itemsets. The tasks of mining association rules consists of two main steps: find all frequent itemsets and generate strong association rules from the frequent itemsets. But CHARM implement that it is not necessary to mine all frequent itemsets in the first step. In CHARM, the exploration of both the itemsets and tidset space allows to quickly identity the closed frequent itemsets, instead of having to enumerate many non-closed subsets. CHARM uses a two-pronged pruning strategy and the fundamental operation of CHARM used is an union of two itemsets and an intersection of their tidsets.

CHARM uses depth-first methods and so it can find maximal frequent itemsets. Because this approach allows the frequent itemsets border to be detected more quickly than Apriori. And then the choices of vertical and horizontal representation can affect the I/O costs incurred when computing the support of candidate itemsets. CHARM uses a vertical data layout to store the lists of transaction identifiers (TID-list) associated with each item and so

this format implies to be successful for association mining and to lead to very good performance than horizontal data layout of Apriori algorithm. CHARM is fully scalable for large-scale database mining and it provides orders of magnitude improvement over existing methods for mining closed itemsets. Table 1 represents the CHARM algorithm.

The main computation of CHARM is depent on the four properties. It tests each of this properties of itemset-tidset pairs, extending existing itemsets, removing some subsumed branches from the current set of nodes, or inserting new pairs in the node set for the next (depth-first) step. So the main computation of CHARM relies on the following properties:

Property1: If $t(X_1) = t(X_2)$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) = t(X_1) = t(X_2)$. It means that $X_1 \cup X_2$ as a composite itemset.

Property2 : If $t(X_1) \subset t(X_2)$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) = t(X_1) \neq t(X_2)$. In this state, X_2 generates a different closure.

Propety3 : If $t(X_1) \supset t(X_2)$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) = t(X_1) \neq t(X_2)$. In this state, X_1 produces a different closure and it must be retained.

Property4 : If $t(X_1) \neq t(X_2)$, then $t(X_1 \cup X_2) = t(X_1) \cap t(X_2) \neq t(X_1) \neq t(X_2)$. So nothing can be eliminated and both X_1 and X_2 lead to different closures.

After mining with these four properties in CHARM algorithm, all closed frequent itemsets will be identified.

Table 1 CHARM Algorithm

CHARM ($\delta \subseteq I \times T$, *minsup*):
1. Nodes = {Ij × t(Ij) : Ij $\in$ I Λ | t(Ij) | $\geq$ *minsup*
2. CHARM-EXTEND (Nodes, C):
3. **for each** Xi × t(Xi) in Nodes
4. NewN = ϕ and X = Xi
5. **for each** Xj × t(Xj) in Nodes, with f(j) > f(i)
6. X = X $\cup$ Xj and Y = t(Xi) $\cap$ t(Xj)
7. CHARM-PROPERTY(Nodes, NewN)
8. **if** NewN ≠ ϕ **then** CHARM-EXTEND (NewN)
9. C = C $\cup$ X //if **X** is not subsumed
CHARM-PROPERTY (Nodes, NewN):
10. **if** (|Y| *minsup*) **then**
11. **if** t(Xi) = t(Xj) **then** //Property 1
12. Remove Xj from Nodes
13. Replace all Xi with X
14. **else if** t(Xi) ⊂ t(Xj) **then** //Property 2
15. Replace all Xi with X
16. **else if** t(Xi) ⊃ t(Xj) **then** //Property 3
17. Remove Xj from Nodes
18. Add X × Y to NewN
19. **else if** t(Xi) ≠ t(Xj) **then** //Property 4
20. Add X × Y to NewN

4. Web Mining System Using CHARM

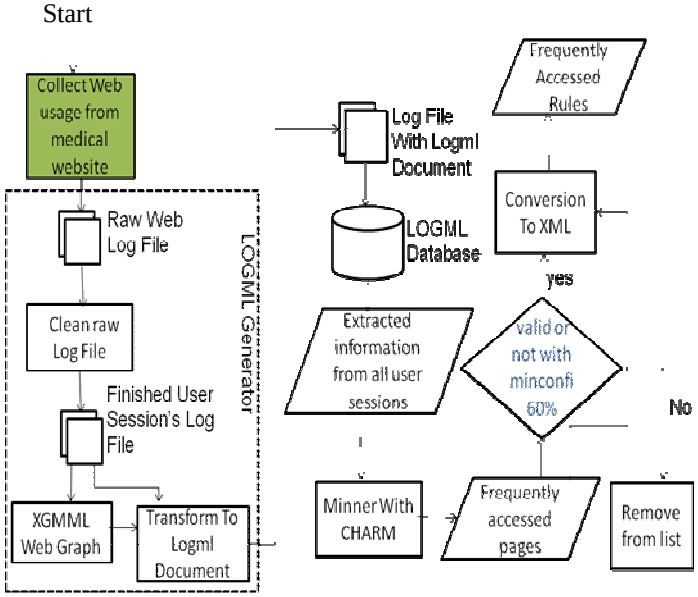

Figure 1: Proposed System Architecture

This system is implementation of the web usage mining processes responsible for web log reports with detailed information of each user session and then mining the frequently itemsets by using LOGML language and CHARM algorithm. The above Figure 1 represents the design for the proposed system. Just a case study, the medical website with 56 nodes (webpages) will be used for the implementation protocol of the proposed system. And then, the system can continue the three main steps of the proposed system according to the above Figure 1.

4.1 Implementation of the Proposed System

This phase is about the detail process of the system. The system applied the usage information of the users from the medical website as the protocol of the system. The usage information of finished and not finished user sessions will contain in raw log file because of "End session manually" button. The system is generated this button in the sample website for checking each user session is finished or not.

In the first step, the input log file is cleaned with LOGML generator. Because LOGML generator can also read all web log lines and only the finished user sessions are reported in the LOGML document. So the system check the the each user session is finished or not and produce the output file with the usage information of finished user sessions. For the three sections of the LOGML document, the system transform to XGMML format for web graph information of first and third section of the LOGML document and then combine with the rest of second section from LOGML file. In system, the session id identified with randomly id number and each web page in the medical website identified with id number

which are sorting with alphabetically. In system, the user can see the web graph with the web pages id which are the user's usage information (the visited web pages and hyperlinks) on the medical website. Table 2 is the example of web log report with LOGML format. Then the cleaned log file with LOGML format is displayed in the system and is stored in the LOGML database to extract the information which is shown in Table 3 as the implementation of the example of the LOGML database.

Table 2 Example of LOGML File

```
<?xmlversion="1.0"?>
<logml xmlns=http://www.cs.rpi.edu/LOGML
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-
instance"
xsi:schemaLocation="http://www.cs.rpi.edu/LOGML
http://www.cs.rpi.edu/puninj/LOGML/logml.xsd">
<graph
xmlns="http://www.cs.rpi.edu/XGMML"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-
instance"
xsi:schemaLocation="http://www.cs.rpi.edu/LOGML
http://www.cs.rpi.edu/puninj/LOGML/logml.xsd">
<node>
  <pageid>40</pageid>
  <label>Home.aspx</label>
  <sessionid>84</sessionid>
  <utime>7/19/2010 3:39:13 PM</utime>
  </node>
-------------------------------------------------
<edge sessionId="84" source="40" target="4" />
<edge sessionId="84" source="4" target="8" />
-------------------------------------------------
</graph>
<summary>
 <request>49</request>
 <sessions>5</sessions>
 <pages>35</pages>
</summary>
<userSessions  count="5"
max_edges="100" min_edges="3">
<userSession sessionId="35" startTime="7/22/2010
9:28:33 PM" endTime="7/22/2010 9:29:09 PM"
access_Count="7">
<path count="5">
  <uedge source="40" target="8" utime="7/22/2010 9:28:37
PM" />
  <uedge source="40" target="4" utime="7/22/2010 9:28:49
PM" />
  <uedge source="40" target="22" utime="7/22/2010
9:28:53 PM" />
  <uedge source="22" target="28" utime="7/22/2010
9:28:54 PM" />
  <uedge source="4" target="43" utime="7/22/2010 9:28:59
PM" />
  </path>
 </userSession>
-------------------------------------------------
</userSessions>
 </logml>
```

Second, the system use CHARM algorithm to discover the frequently accessed pages by mining the frequently user accessed patterns from LOGML database. To get the strong associated rules, the system processed the validation with minimum confidence (mincofi). In the processing of mining the frequently accessed pages, CHARM uses the union of itemsets (web pages from each user session) and intersection of tidsets (transaction or user sessions), so this can reduce the chances of nonclosed frequent itemsets being in a node. The detail processing of CHARM algorithm described in Table 1.

Third, the system converted the frequently accessed rules to human readable XML format. For strong association rules, the system generate these results with minconfi and can also convert to XML format. Because the frequently accessed rules with XML will allow the easier analysis of the results of every system and will get interesting information from XML document collections**.**

Table 3 Example OF LOGML database

User id	count (num of nodes accessed)	node list
84	4	40,4,9,12
35	3	4,8,12
48	4	40,4, 9,12
30	4	40,4, 8,12
19	5	40,4, 8,9,12
20	3	4,8,9

4.2 Example of Process of CHARM Algorithm

Figure 2 show how the above four basic properties of itemset-tidset pairs are developed in CHARM to mine the closed frequent itemsets with minsup= 3. For presenting the CHARM algorithm, let's assume that the system is processing five items and six user sessions from the example database which is represented in above Table 3. The system has identified seven closed frequent itemsets because of four properties in CHARM algorithm which are described in section 3.2.

In Figure 2, when two itemset-tidset pairs 40 and 4 combine, property 2 is true. This property can remove 40 and replace 40,4. Due to the property 4, combining 40,4 with 8 produces an infrequent set 40,4,8 because of less than minsup which is pruned. Combining with 9 produces the 40,4,9 because of property 4. Combining 40,4 with 12 produces the pair 40,4,12. It can observe that property 2 holds and can remove 40,4 then replace it with 40,4,12. And the next combining 40,4,12 with 40,4,9 produces 40,4,9,12 and property 2 is also true.

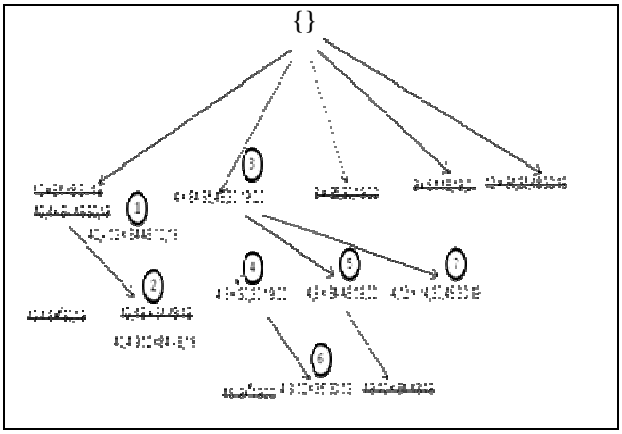

Figure 2 Example of the flow of CHARM Algorithm

For the rest of nodes, CHARM algorithm identified closed frequent itemsets with exactly the same computation until all branches have been processed. So Web data mining with CHARM algorithm can easily mined all closed frequent itemsets because CHARM adds a few seconds of additional processing time to the total execution time. By using LOGML database, web master can easily generate the web log reports for a large period of time in system. The system proposed the usage information of 100 user transactions from the sample website with 56 webpages which are generated the strong association rules depending on the desired minsup. In system, the usage information of the users are created with sample website so which cannot be analyzed. But this confirms the efficiency of the system by using with the other real websites. After running the CHARM algorithm, the system will produce the frequent itemsets with node ids and the name of web pages. This system can also describe the results with XML format.

5. Conclusion

The system can generate the web log report with LOGML format for a web site from web log files and the web graph. The next step is that CHARM algorithm is an efficient and effective closed association mining algorithm and it scales linearly in the number of transactions and is also linear in the number of closed itemsets found. Then the mined results are transformed into human-readable XML format. Therefore, the user can know the most populated web pages from one site by using XML languages and CHARM algorithm.

References

[1] John R. Punin, Mukkai S. Krishnamoorthy, Mohammed J. Zaki “Web Usage Mining – Languages and Algorithm”.

[2] Kosala and Blockeel, “Web mining research: A survey,” SIGKDD:SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM, Vol. 2, 2000.

[3] Mohammed J.Zaki and Ching-Jui Hsiao “CHARM: An Efficient Alogrithm For Closed Association Rule Mining”.

[4] M.S. Chen, J.S Park, and P.S. Yu.” Data mining for path traversal patterns in a web environment”. In International Conference on Distributed Computing Systems, 1999.

[5] Q. Yang and H. H. Zhang, “Web-log mining for predictive web caching’’. IEEE Trans. Knowl. Data Eng., Vol.15, N0.4, pp. 1050-1053,2003.

[6] R.Agrawal, H. Mannila, R.Srikant, H. Toivonen, and A. Inkeri Verkamo. ”Fast discovery of association rules”. In U. Fayyad and et al, editors, Advances in Knowledge Discovery and Data Mining, pages 307-328. AAAI Press, Menlo Park, CA, 1996.

[7] Yew-Kwong Woon, Wee-Keong Ng and Ee-Peng Lim “Web Usage Mining : Algorithm and Results”.