A PLAUSIBLE COMPREHENSIVE WEB INTELLIGENT SYSTEM FOR INVESTIGATION OF WEB USER BEHAVIOUR ADAPTABLE TO INCREMENTAL MINING

V.V.R. Maheswara Rao[1], Dr. V. Valli Kumari[2] and Dr. K.V.S.V.N. Raju[2]

[1]Professor, Department of Computer Applications,
Shri Vishnu Engineering College for Women, Bhimavaram, Andhra Pradesh, India,
mahesh_vvr@yahoo.com
[2]Professor, Department of Computer Science & Systems Engineering,
College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India,
vallikumari@gmail.com, kvsvn.raju@gmail.com

ABSTRACT

With the continued increase in the usage of the World Wide Web (WWW) Web mining has been established as an important area of research. The WWW is a vast repository of unstructured information, in the form of interrelated files, distributed on numerous web servers over wide geographical regions. Web mining deals with the discovering and analyzing of useful information from the WWW. Web usage mining focuses on investigating the potential knowledge from the browsing patterns of users and to find the correlation between the pages on analysis. To proceed towards web intelligence, obviating the need for human interaction, need to incorporate and embed artificial intelligence into web tools. Before applying mining techniques, the data in the web log has to be pre-processed, integrated and transformed. The data pre-processing stage is the most important phase in the process of web mining and is critical and complex in successful extraction of useful data. The web log is non scalable, impractical and distributed in nature thus conventional data pre-processing techniques are proved to be not suitable as they assume that the data is static. Hence intelligent system is required for capable of pre processing weblog efficiently. Due to the incremental nature of the web log, it is necessary for web miners to use incremental mining techniques to extract the usage patterns and study the visiting characteristics of user, hence one can require a comprehensive algorithm which reduces the computing cost significantly.

This paper introduces an Intelligent System IPS for pre-processing of web log, in addition a learning algorithm IFP-tree model is proposed for pattern recognition. The Intelligent Pre-processing System (IPS) can differentiate human user and web search engine accesses intelligently in less time, and discards search engine accesses. The present system reduces the error rate and improves significant learning performance of the algorithm. The Incremental Frequent Pattern Tree (IFP-Tree) is to suit for continuously growing web log, based on association rule mining with incremental technique. IFP-Tree is to store user-specific browsing path information in a condensed way. The algorithm is more efficient as it avoids the generation of candidates, reduces the number of scans and allows interactive mining with different supports. The experimental results that prove this claim are given in the present paper.

KEYWORDS

Web usage mining, intelligent pre-processing system, incremental frequent pattern tree.

1. INTRODUCTION

Web mining deals with the application of data mining techniques to the Web for extracting interesting patterns and discovering knowledge. Web mining, though essentially an integral part of data mining, has emerged as an important and independent research direction due to the

DOI : 10.5121/ijdms.2010.2309

typical nature of Web, e.g., its diversity, size, heterogeneous and link-based nature. The Web is creating new challenges to different component tasks of Web mining as the amount of information on the Web is increasing and changing rapidly without any control. Usually every click corresponds to the visualization of a Web page. Thus, the Web log defines the sequence of the Web pages requested by a user. With the aim of predicting, possibly online, which pages will be seen, having seen a specific path of pages in the past. Such analysis can be very useful to understand for instance, what is the probability of seeing a page of interest coming from a specified page.

Mining of the Web data can be viewed in different levels: mining the Web content, its structure and its usage as shown in Figure 1.

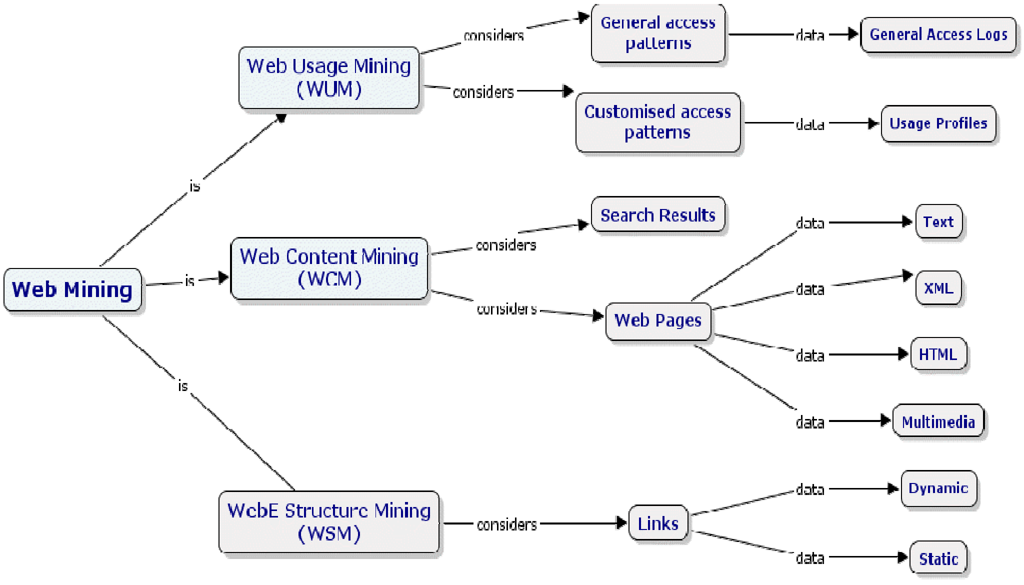


Figure 1. Stages of Web mining

The content of a Web page may be varied, e.g., text, images, HTML, tables or forms. Accordingly, the pages may be classified based on content, and the retrieval algorithms can be designed. Mining the search result is also an integral part of Web content mining.

Mining the structure of the Web involves extracting knowledge from the interconnections of the Hypertext documents in the WWW. This results in discovery of Web communities, and also pages that are authoritative. Moreover, the nature of relationship between neighbouring pages can be discovered by structure mining.

Web usage mining essentially consists of analyzing the data pertaining to the use of the Web. This can be viewed from two perspectives, viz., that of the user and that of the Website. On the client (or the user) side, usage data are collected through the browsing history, while on the server side, data are collected through the request logs. Web usage mining involves discovering the category of users accessing pages of a certain type, the time and duration of access, as well as the order of page references.

The general process of web mining includes (i) Resource collection: Process of extracting the task relevant data, (ii) Information pre-processing: Process of Cleaning, Integrating and Transforming of the result of resource collection, (iii) Pattern discovery: Process of uncovered general patterns in the pre process data and (iv) Pattern analysis: Process of validating the discovered patterns.

(i)Resource collection: In web mining techniques the nature of the data is incremental and is rapidly growing. The hyperlinks contain important information which can be utilized for efficient information retrieval. One has to collect the data from web which normally includes web content, web structure and web usage. Web content resource is collected from published data on internet in several forms like unstructured plain text; semi structured HTML pages and structured XML documents. The web structured data can be captured via inter-page-linkage among the pages of a website and site map of a website. The web usage data can be captured from web logs, click streams and database transactions. The above three resources are captured from a website or a group of related websites.

(ii)Information pre-processing: In web mining techniques the information pre-processing includes a) Content pre-processing, b) Structure pre-processing and c) Usage pre-processing. Content Pre-processing: Content pre-processing is the process of converting text, image, scripts and other files into the forms that can be used by the usage mining. Structure Pre-processing: The structure of a website is formed by the hyperlinks between page views. The structure pre-processing can be treated similar to the content pre processing. Usage Pre-processing: The inputs of the pre-processing phase may include the web server logs, referral logs, registration files, index server logs, and optional usage statistics from a previous analysis. The outputs are the user session files, transaction files, site topologies and page classifications.

iii) Pattern discovery: The goal of pattern discovery is the task of learning some general concepts from a given set of documents. In this phase, Pattern recognition and machine learning techniques, like classification, clustering and association rule mining, are usually used on the extracted information.

iv) Pattern analysis: The goal of pattern analysis is the task of understanding, visualizing, and interpreting the patterns once they are discovered in the Pattern Discovery phase.

Information pre processing technique can improve the quality of the data, Web logs contain user activity information of which some data is not closely relevant to the Usage mining in order to discover the knowledge one has to remove the irrelevant data without noticeably affecting the mining such as Web search engine access and all log image entries. Pre-processing is very important and difficult task. If this phase is not performed adequately, it is not possible for the mining algorithms to provide reliable results. Hence an intelligent system is required to process the web log more efficiently.

Association rule mining techniques can be used to discover correlation between pages found in a web log. Association rule mining is an iterative process, thus, the existing mining techniques have the limitations like, multiple scans of transactional Data base, huge number of candidate generation and burden of calculating the support. The web user behaviour may be changed with the rapid growth of web logs. Therefore, one must re-discover the user behaviour from the updated web log using incremental mining. The essence of incremental mining is that it utilizes the previous mining results and finds new patterns from the inserted or deleted part of the web log such that the mining time can be reduced.

The present paper introduces a) An Intelligent Pre-processing System of weblog (IPS) b) Incremental Frequent Pattern Tree which is suitable algorithm for mining the growing weblogs.

IPS is an intelligent system capable of pre-processing weblogs efficiently. It can identify human user and web search engine accesses intelligently, in less time. Web search engine is a software program that can automatically retrieve information from the web pages. Generally these programs are deployed by web portals. To analyze user browsing behaviour one must discard the accesses made by web search engines from web access logs. After discarding the search engine accesses from web access logs, the remaining data are considered as human accesses. This human access data pre processing includes cleansing, user identification, session identification, path completion and formatting. This collective work yields the data which is suitable for the next mining phase of Pattern Discovery.

IFP-Tree is to store user-specific browsing path information in a condensed way. Incremental mining techniques emerged to avoid algorithm re-execution and to update mining results when incremental data is added or old data is removed, ensuring a better performance in the web mining process and discovers interesting patterns in an effective manner. The IFP-Tree builds on divide and conquers strategy. It retains associated pages information and frequency of pages at each node except the root node with a single scan on web log. This algorithm minimizes the number of scans of web log, avoids generation of candidates. In addition, this algorithm interactively mines the data for different supports with a single scan of database and allows addition / deletion of new click streams in a finest granularity.

The present paper is organized as follows. The related work described in section 2. In next section 3, the overview of proposed work is introduced. In subsequent section 4, the theoretical analysis of proposed work is shown. In subsequent section 5, the experimental analysis of proposed work is shown. Finally in section 6 conclusions are mentioned.

2. RELATED WORK

Many of the previous authors have expressed the importance, criticality and efficiency of data preparation stage in the process of web mining. Most of the works in the literature do not concentrate on data preparation, and are not suitable for dynamically changing web log scenario.

Myra Spiliopoulou [1] suggests applying Web usage mining to website evaluation to determine needed modifications, primarily to the site's design of page content and link structure between pages. Such evaluation is one of the earliest steps, that adaptive sites automatically change their organization and presentation according to the preferences of the user accessing them. M. Eirinaki and M. Vazirgiannis.[2] proposed a model on web usage mining activities of an on-going project, called Click World, that aims at extracting models of the navigational behaviour of users for the purpose of website personalization. Path traversal pattern mining [8] is the technique that finds navigation behaviours for most of the users in the web environment. The website designer can use this information to improve the website design. Most of the researches focused on Full Scan algorithm and Selective Scan algorithm [9] etc. However, these algorithms have the limitations that they can only discover the simple path traversal pattern, i.e., a page cannot repeat in the pattern.

A number of Apriori-based algorithms [19, 20 and 21] have been proposed to improve the performance by addressing issues related to the I/O cost. A new data structure called H-struct [25] was introduced to deal with sparse data solely. Most of the researches focused on Full Scan algorithm and Selective Scan algorithm [27] etc. Show-Jane Yen and his colleagues introduced an Incremental data mining algorithm for discovering web traversal patterns.

To extract useful web information one has to follow an approach of collecting data from all possible server logs which are non scalable and impractical. Hence to perform the above there is a need of an intelligent system which can integrate, pre process all server logs and discard unwanted data. The output generated by the intelligent system will improve the efficiency of web mining techniques with respect to computational time. To discover useful patterns one has to concentrate on structurally complex and exponential growth of web log scenario along with I/O cost.

3. PROPOSED WORK

Web usage mining is a complete process, integrating various stages of data mining cycle, including web log Pre processing, Pattern Discovery & Pattern Analysis as shown in Figure 2. For any web mining technique, initially the preparation of suitable source data is an important task since the characteristic sources of web log are distributed and structurally complex. Before applying mining techniques to web usage data, resource collection has to be cleansed, integrated and transformed. It is necessary for web miners to utilize intelligent tools in order to find, extract, filter and evaluate the desired information. To perform this task it is important to separate accesses made by human users and web search engines. Later all the mining techniques can be applied on pre Processed web log. Pattern analysis is a final stage of web usage mining, which can validate the discovered patterns and identifies interested unique patterns.

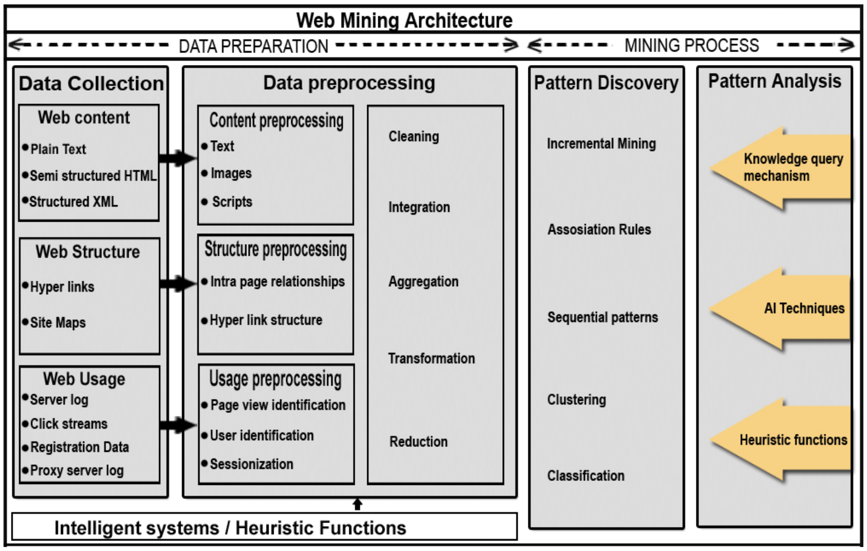


Figure 2. Web mining architecture

The authors in the present paper introduces a comprehensive model for pre Processing of Web log which includes (3.1) An Intelligent pre Processing System (IPS) and (3.2) learning algorithm Incremental Frequent Pattern Tree (IFP-Tree) is used to store user-specific browsing path information in a condensed way.

3.1. Intelligent Pre-processing System: IPS

The main goal of IPS is to separate the human user and Web search engine accesses. The intelligent system takes the raw web log as input and discards the search engine accesses automatically with less time. To analyze user browsing behaviour one must discard the accesses made by web search engines from web access logs. After discarding the search engine accesses from web access logs, the remaining data are considered as human accesses. This human access data pre processing includes cleansing, user identification, session identification, path completion and formatting as shown in Figure 3.

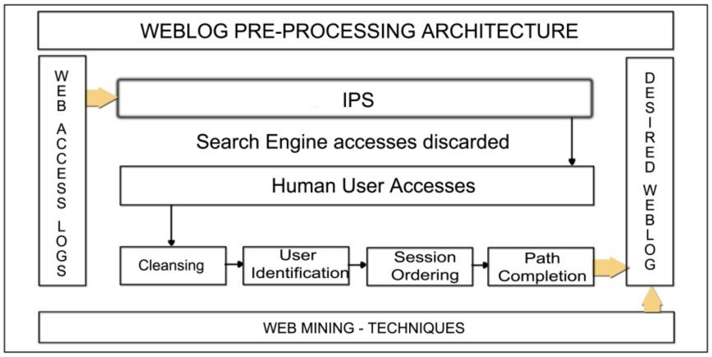


Figure 3. Web log pre-processing architecture

The intelligent systems can be broadly categorized into server side intelligent systems and client side intelligent systems. In web data preparation mostly client side intelligent systems are used. These client side systems are also called sessionization heuristic functions. The present paper introduces a learning heuristic function to separate human accesses and web search engine accesses from web access logs as a primary step of pre processing of web log data.

To separate the human user and web search engine accesses the IPS requires a learning capability. In order to get this capability any intelligent system acquires the knowledge from the knowledge base, where knowledge base is a “set of related facts”. These facts or characteristics are called Derived attributes, and form a knowledge base, which separates human user and web search engine accesses. This knowledge base can be used as training data to the IPS.

The present IPS model suggests the following characteristics, which helps to distinguish human user accesses and web search engine accesses.

- Accesses by web search engine tend to be more broad where as human accesses to be of more depth.
- Accesses by web search engines rarely contain the image pages whereas human user accesses contain all type of web pages.
- Accesses by web search engines contain large number of requested pages where as human user accesses contain less number of requested pages.
- Accesses by the web search engines are more likely to make repeated requests for the same web page, where as human users accesses often make repeated requests.

Table1. Example of Characteristics / Derived Attributes

Derived Attribute	Description
Total pages	Total pages retrieved in a web session
Image pages	Total number of image pages retrieved in a web session
Total Time	Total amount of time spent by website visitor
Repeated access	The same page requested more than once in a web session
GET	Percentage of requests made using GET method
Breadth	Breadth of the web traversal
Depth	Depth of the web traversal

All the records in the web logs are taken as testing data. The raw web usage data collected from different resources in web log includes IP address, unique users, requests, time stamp, protocol, total bytes and so on as shown below.

Table 2. Example of web server log.

No	IP Address	Unique Users	Requests	Time Stamp	Protocol	Total Bytes
1	125.252.226.42	1	4	11/22/2009 12:30	HTTP\1.1	14.78 MB
2	64.4.31.252	1	69	11/22/2009 13:00	HTTP\1.1	782.33 KB
3	125.252.226.81	1	41	11/22/2009 13:30	HTTP\1.1	546.71 KB
4	125.252.226.83	1	19	11/22/2009 14:00	HTTP\1.1	385.98 KB
5	125.252.226.80	1	20	11/22/2009 14:30	HTTP\1.1	143.44 KB
6	58.227.193.190	1	18	11/22/2009 15:00	HTTP\1.1	108.99 KB
7	70.37.129.174	1	4	11/22/2009 15:30	HTTP\1.1	86.66 KB
8	64.4.11.252	1	2	11/22/2009 16:00	HTTP\1.1	52.81 KB
9	208.92.236.184	1	17	11/22/2009 16:30	HTTP\1.1	32.13 KB
10	4.71.251.74	1	2	11/22/2009 17:00	HTTP\1.1	25.82 KB

To label the web sessions the IPS takes the training data as characteristics of session identification. A web session is a sequence of request made by the human user or web search made during a single visit to a website. This paper introduces a learning tree known as IPS to accomplish above task.

3.1.1 Design of IPS:

The IPS learning tree can be constructed from a set of derived attributes from knowledge base. The IPS learning tree consists of root node, internal node and leaf of terminal node. A root node that has no incoming edges and two or more outgoing edges. Any internal node has exactly one incoming edge and two or more outgoing edges. The leaf or terminal node each of which has exactly one incoming edge and no outgoing edges. In IPS learning tree, each leaf node is assigned with a class label. The class labels are human user access session and web search engine access sessions. The root node and other internal nodes are assigned with the characteristics of the session. The IPS tree works on a repeatedly posing series of questions about the characteristics of the session identification and it finally yields the class labels. Based on the tree traversal there are two notable features namely depth and breadth. Depth determines the maximum distance of a requested page where distance is measured in terms of number of hyperlinks from the home page of website. The breadth attribute determines the possible outcomes of each session characteristics.

3.1.2 Development of IPS: Learning Algorithm

An efficient IPS learning tree algorithm has been developed to get reasonably accurate learning to discard web search engine accesses from web log accesses. The algorithm is developed based on the characteristics of session.

```
TreeExtend(DA, TA)
If ConditionStop(DA, TA) = True then
    TerminalNode = CreateNewNode( )
    TerminalNode.Label = AssignedLabel(DA)
    Return TerminalNode
Else
    Root = CreateNewNode( )
    Root.ConditionTest = DeriveBestSplit(DA, TA)
    Let V ={v /v is a possible outcome of ConditionTest()}
```

```
09: For each v Є V do
10: DAv ={da / Root.ConditionTest(da) = v and d Є DA}
11: Child = TreeExtend(DAv, TA)
12: Add Child as descendant of root and label the edge as v
13: End for
14: End if
15: Return root
```

The input to the above algorithm consists of Training data DA and Testing data TA. The algorithm works by recursively selecting DeriveBestSplit( ) (step 7) and expanding the leaf nodes of the tree (Step 11 & 12) until condition stop is met (Step1). The details of methods of algorithm are as follows

CreateNewNode( ): This function is used to extend the tree by creating a new node. A new node in this tree is assigned either a test condition or a class label.

ConditionTest( ): Each recursive step of TreeExtend must select an attribute test condition to divide into two subsets namely human user accesses and search engine accesses. To implement this step, algorithm uses a method ConditionTest for measuring goodness of each condition.

ConditionStop(DA, TA): This function is used to terminate the tree extension process by testing whether all the records have either the same class label or the same attribute values. Another way of stopping the function is to test whether the number of records have fallen below minimum value.

AssignLabel ( ): This function is used to determine the class label to be assigned to a terminal node. For each terminal node t, Let p(i/t) denotes the rate of training records from class i associated with the node t. In most of the cases the terminal node is assigned to the class that has more number of training records.

DeriveBestSplit( ): This function is used to determine which attribute should be selected as a test condition for splitting the training records. To ensure the goodness of split, the Entropy and Gini index are used.

3.1.3 Example for IPS:

The main idea of IPS is to label the human user accesses and search engine accesses separately. The intelligent system acquires the knowledge from the derived characteristics of web log as shown in Table 1. Using IPS algorithm the derived characteristics are assigned to root node and intermediate nodes of the tree as shown in Figure 4. The leaf nodes are labelled with human user or search engine accesses.

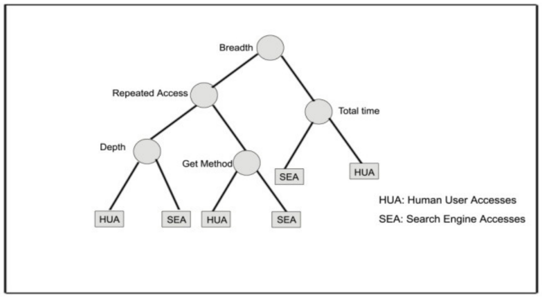


Figure 4. Example of IPS Learning Tree

After discarding search engine accesses, the web log consist only human user accesses. To complete the further stages of pre Processing, the standard techniques like Data Cleansing, User Identification, Session Identification and Path Completion are used as given below.

Data Cleansing:

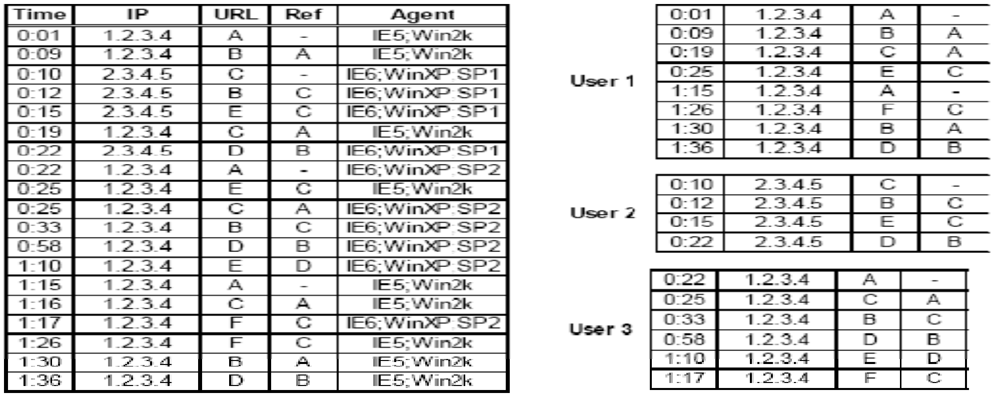
The next phase of pre processing is data cleansing. Data cleansing is usually site-specific, and involves tasks such as, removing extraneous references to embedded objects that may not be important for the purpose of analysis, including references to style files, graphics, or sound files. The cleaning process also may involve the removal of at least some of the data fields (e.g. number of bytes transferred or version of HTTP protocol used, etc.) that may not provide useful information in analysis or data mining tasks.

Table3. Example of web log with different extensions

No	Object Type	Unique Users	Requests	Bytes In	% of Total Bytes In
1	*.gif	1	46	89.00 KB	0.50%
2	*.js	1	37	753.95 KB	4.40%
3	*.aspx	1	34	397.05 KB	2.30%
4	*.png	1	31	137.67 KB	0.80%
5	*.jpg	1	20	224.72 KB	1.30%
6	Unknown	1	15	15.60 KB	0.10%
7	*.ashx	1	15	104.79 KB	0.60%
8	*.axd	1	13	274.81 KB	1.60%
9	*.css	1	8	71.78 KB	0.40%
10	*.dll	1	7	26.41 KB	0.20%
11	*.asp	1	4	1.26 KB	0.00%
12	*.html	1	3	2.17 KB	0.00%
13	*.htm	1	2	69.87 KB	0.40%
14	*.pli	1	2	24.92 KB	0.10%

User Identification:

The task of user identification is, to identify who access web site and which pages are accessed. The analysis of Web usage does not require knowledge about a user's identity. However, it is necessary to distinguish among different users. Since a user may visit a site more than once, the server logs record multiple sessions for each user. The phrase user activity record is used to refer to the sequence of logged activities belonging to the same user.



Time	IP	URL	Ref	Agent
0:01	1.2.3.4	A	-	IE5;Win2k
0:09	1.2.3.4	B	A	IE5;Win2k
0:10	2.3.4.5	C	-	IE6;WinXP;SP1
0:12	2.3.4.5	B	C	IE6;WinXP;SP1
0:15	2.3.4.5	E	C	IE6;WinXP;SP1
0:19	1.2.3.4	C	A	IE5;Win2k
0:22	2.3.4.5	D	B	IE6;WinXP;SP1
0:22	1.2.3.4	A	-	IE6;WinXP;SP2
0:25	1.2.3.4	E	C	IE5;Win2k
0:25	1.2.3.4	C	A	IE6;WinXP;SP2
0:33	1.2.3.4	B	C	IE6;WinXP;SP2
0:58	1.2.3.4	D	B	IE6;WinXP;SP2
1:10	1.2.3.4	E	D	IE6;WinXP;SP2
1:15	1.2.3.4	A	-	IE5;Win2k
1:16	1.2.3.4	C	A	IE5;Win2k
1:17	1.2.3.4	F	C	IE6;WinXP;SP2
1:26	1.2.3.4	F	C	IE5;Win2k
1:30	1.2.3.4	B	A	IE5;Win2k
1:36	1.2.3.4	D	B	IE5;Win2k

User 1	0:01	1.2.3.4	A	-
	0:09	1.2.3.4	B	A
	0:19	1.2.3.4	C	A
	0:25	1.2.3.4	E	C
	1:15	1.2.3.4	A	-
	1:26	1.2.3.4	F	C
	1:30	1.2.3.4	B	A
	1:36	1.2.3.4	D	B
User 2	0:10	2.3.4.5	C	-
	0:12	2.3.4.5	B	C
	0:15	2.3.4.5	E	C
	0:22	2.3.4.5	D	B
User 3	0:22	1.2.3.4	A	-
	0:25	1.2.3.4	C	A
	0:33	1.2.3.4	B	C
	0:58	1.2.3.4	D	B
	1:10	1.2.3.4	E	D
	1:17	1.2.3.4	F	C

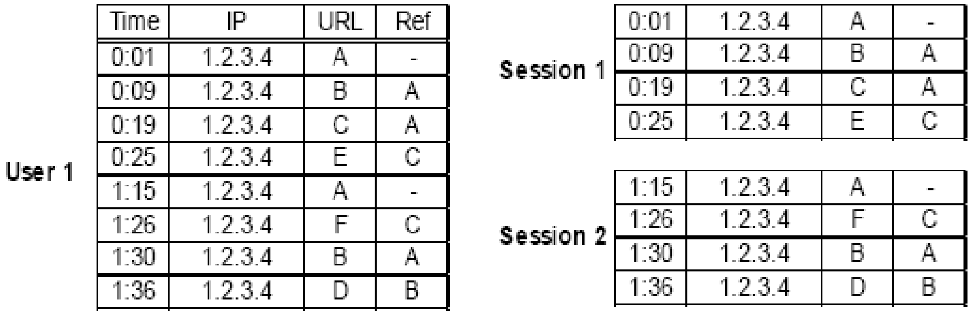
Figure 5. Example of user identification using IP + Agent

Consider, for instance, the example of Figure 5. On the left, depicts a portion of a partly pre processed log file (the time stamps are given as hours and minutes only). Using a combination of IP and Agent fields in the log file, one can partition the log into activity records for three separate users (depicted on the right).

Session Ordering:

Sessionization is the process of segmenting the user activity record of each user into sessions, each representing a single visit to the site. Web sites without the benefit of additional authentication information from users and without mechanisms such as embedded session ids must rely on heuristics methods for sessionization. The goal of a sessionization heuristic is to re-construct, from the click stream data, the actual sequence of actions performed by one user during one visit to the site.

Generally, sessionization heuristics fall into two basic categories: time-oriented or structure-oriented. Many authors in the literature survey have been identified various heuristics for sessionization. As an example, time-oriented heuristic, h1: Total session duration may not exceed a threshold θ. Given t0, the timestamp for the first request in a constructed session S, the request with a timestamp t is assigned to S, iff t – t0 ≤ θ. In Figure 6, the heuristic h1, described above, with θ = 30 minutes has been used to partition a user activity record (from the example of Figure 5) into two separate sessions.



User 1

Time	IP	URL	Ref
0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C
1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

Session 1

0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C

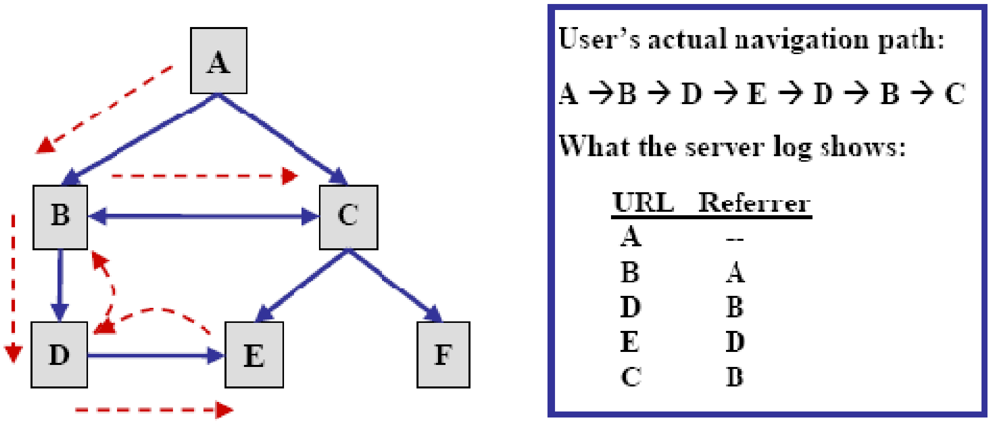
Session 2

1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

Figure 6. Example of sessionization with a time-oriented **h1** heuristic

Path Completion:

Another potentially important pre-processing task which is usually performed after sessionization is path completion. Path completion is process of adding the page accesses that are not in the web log but those which be actually occurred. Client or proxy-side caching can often result in missing access references to those pages or objects that have been cached. For instance, if a user returns to a page A during the same session, the second access to A will likely result in viewing the previously downloaded version of A that was cached on the client-side, and therefore, no request is made to the server. This results in the second reference to A not being recorded on the server logs. Missing references due to caching can be heuristically inferred through path completion which relies on the knowledge of site structure and referrer information from server logs. In the case of dynamically generated pages, form-based applications using the HTTP POST method result in all or part of the user input parameter not being appended to the URL accessed by the user. A simple example of missing references is given in Figure 7.



User's actual navigation path:

A →B → D → E → D → B → C

What the server log shows:

URL	Referrer
A	--
B	A
D	B
E	D
C	B

Figure 7. Identifying missing references in path completion

3.2 Incremental Frequent Pattern Tree: IFP

The IFP-Tree is used to keep track of patterns identified during the web usage mining process. It retains associated pages information and frequency of pages at each node except the root node with a single scan on web log. The IFP-Tree structure is suitable for interactive frequent pattern mining, builds on divide and conquers strategy. Using IFP-Tree one has to navigate from any node to any other node and also its size grows dynamically. Hence linked list representation is followed.

3.2.1 Design of IFP: Incremental Tree

The IFP-Tree can be constructed based on recursive procedure using formatted Web log as input. It consists of root node, which is assigned by null value and all other nodes including intermediate and leaf nodes, holds the information of associated pages and respective frequencies as integers. On reading the first session, repeatedly new nodes are created and assigned with respective page numbers with their frequencies till the end of the pages in that session. On reading new session first page is compared with the previously created immediate nodes of the root node. If it is found to be the same, the frequency of the matched node is updated and moved to the next node or new node that depends on the next page in the same session. The above step is repeated till the end of the formatted log. Traversal of the tree starts from root node to the desired node for interactive mining.

The IFP-Tree structure is more convenient to update incrementally. However, depending on the applications, one can apply a minimum support or to limit the size of the tree. To do so, two possible methods are proposed. Method I: An effective approach for limiting the size of the IFP-Tree is to set a maximum number of nodes and prune it. Method II: One can apply a minimum support and insert only supported sessions into the tree. If the data do not change drastically, the size of the tree will remain reasonably small.

3.2.2 Development of IFP: Incremental algorithm

The IFP-tree algorithm emerged incremental mining techniques to avoid algorithm re-execution and to update mining results when data is added or old data is removed. This algorithm minimizes the number of scans of web log, avoids generation of candidates. In addition, this algorithm interactively mines the data for different supports with a single scan of database and allows addition / deletion of new click streams in a finest granularity.

```
AddNewNode (TN, PS)
while LS do
    If PS $\neq \varphi$ then
        PP ← first page of PP
        while last page in PS do
            if PP $\notin$ TN.CN then
                TN.CN ← New TN (PP, TN)
            end if
            TN.CN (PP).frequency ++
            AddNewNode (TN.CN (PP), PS)
            PP ← next page in PS
        end of while
    else
        return
    end if
    PS ← next session
end of while
```

Where TN: Root Node, CN: Child Node, PS: Present Session, LS: Last Session, PP: Present Page and LP: Last Page. The above recursive procedure is used in construction of IFP-Tree.

3.2.3 Example for IFP:

An example of formatted web log as shown in Table 4 is taken as input, a tree is constructed by reading the sessions one by one and shown in Figure 8.

Table 4: An Example of formatted web log

Session Id	Requested Page	Session Id	Requested Page
1	a, b	6	a, b, c, d
2	b, c, d	7	a
3	a, c, d, e	8	a, b, c
4	a, d, e	9	a, b, d
5	a, b, c	10	b, c, e

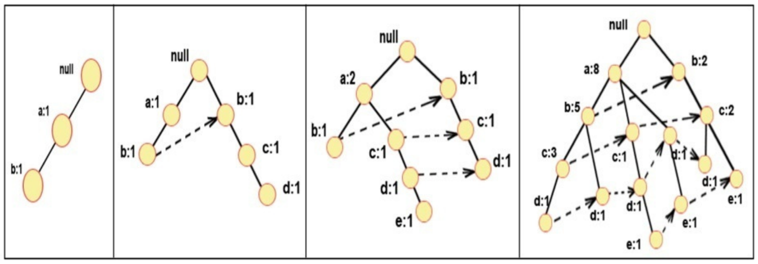


Figure 8. After reading the session id = 1, 2, 3 and 10

4. THEORETICAL ANALYSIS

The authors in the present paper present the mathematical model for the proposed comprehensive model. It consists of 4.1 Mathematical model for IPS - To estimate the training data over the testing data and 4.2 Mathematical model for IFP-Tree – To steady the relationship among the sessions and estimate the stickiness among the frequently visited pages.

4.1 Mathematical model for IPS:

The learning performance of any algorithm is proportionate on the training of algorithm, which directly depends on the training data. As testing data is continuously growing the training data is also continuous. Hence to estimate the training data one can use predictive modelling technique called regression. The goal of regression is to estimate the testing data with minimum errors.

Let S denote a data set that contains N observations,

$$S = \{(D_i, T_i) / i = 1,2,3,.....,N\}$$

Suppose to fit the observed data into a linear regression model, the line of regression D on T is

$$D = a + bT \tag{1}$$

Where a and b are parameters of the linear model and are called regression coefficients. A standard approach for doing this is to apply the method of least squares, which attempts to find the parameters (a, b) that minimize the sum of squared error say E.

$$E = \sum_{i=1}^{n} (D_i - a - bT_i)^2 \tag{2}$$

The optimization problem can be solved by taking partial derivative of E w.r.t a and b, equating them to zero and solving the corresponding system of linear equations.

$$\frac{\partial E}{\partial a} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} D_i = na + b\sum_{i=1}^{n} t_i \tag{3}$$

$$\frac{\partial E}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} D_i t_i = a\sum_{i=1}^{n} t_i + b\sum_{i=1}^{n} t_i^{\,2} \tag{4}$$

Equations (3) and (4) are called normal equations. By solving equations (3) and (4) for a given set of Di, Ti values, we can find the values of 'a' and 'b', which will be the best fit for the linear regression model. By dividing equation (3) by 'N' we get

$$\overline{D} = a + b\bar{t} \tag{5}$$

Thus the line of regression D on T passes through the point ($\overline{D}$, $\overline{T}$)

We can define,

$$\mu_{11} = Cov(D,T) = \frac{1}{n}\sum_{i=1}^{n} D_i T_i - \overline{D}\overline{T}$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} D_i T_i = \mu_{11} + \overline{D}\overline{T} \tag{6}$$

Also

$$\frac{1}{n}\sum D_i^{\,2} = \sigma_d^{\,2} + \overline{D}^2 \tag{7}$$

From equations (4),(6) and (7) we get

$$\mu_{11} + \overline{D}\,\overline{T} = a\overline{D} + b({\sigma_d}^2 + \overline{D}^2) \tag{8}$$

And on simplifying (8), we get

$$\mu_{11} = b{\sigma_d}^2 \Rightarrow b = \frac{\mu_{11}}{{\sigma_d}^2} \tag{9}$$

b is called the slope of regression D on T and the regression line passes through the point ($\overline{D}$, $\overline{T}$). The equation of the regression line is

$$D - \overline{D} = b(T - \overline{T}) = \frac{\mu_{11}}{{\sigma_d}^2}(T - \overline{T})$$

$$D - \overline{D} = r\frac{\sigma_d}{\sigma_t}(T - \overline{T})$$

$$\Rightarrow D = \overline{D} + r\frac{\sigma_d}{\sigma_t}(T - \overline{T}) \tag{10}$$

The linear regression coefficient ‘r’ is used to predict the error between testing data and training data. It can also used to study the nature of the relationship between training data and testing data. The learning performance can also be expressed in terms of training error rate of the learning algorithm. The training error rate is given by the following equation,

$$\text{Training Error Rate} = \frac{\text{Number of wrong characteristic definitions}}{\text{Total number of characteristic definitions}} \tag{11}$$

4.2 Mathematical Model for IFP:

The IFP-Tree discovered pages must be validated to eliminate irrelevant pages and extract interesting pages. In the present paper, a) Mathematical relationship among sessions of the same browser b) Stickiness among the frequently visited pages together, has been estimated.

Relationship among the sessions of the same browser:

To identify the user behavior, it is essential to study the relationship among the set of sessions of the same user. This can be incorporated with a simple correlated technique. Let the set D be defined as data generated by N unique users from formatted web log,

$$D = \{D_i \,/\, i = 1,2,...,N\} \tag{12}$$

Where Di is a session set of ith user, $1 \le i \le N$

For each $D_i = \{S_{ij} \,/\, j = 1,2,..., M\}$ (13)

Where Sij is jth session of ith user, $1 \leq j \leq M$. Hence an ith user may consist of finite multiple sessions. In any session the user may browse a set of pages,

$$S_{ji} = \{P_{ijk} / k = 1,2,....,L\} \tag{14}$$

Where Pijk is kth page of jth session of ith user. Let Pixk and Piyk be the set of pages of Xth and Yth sessions browsed by ith user. Rxy denotes the relationship between the sessions x,y and is defined as,

$$R_{xy} = \frac{Cov(x,y)}{\sigma x \sigma y} \tag{15}$$

Based on the value of Rxy one can identify the user behaviour. Suppose Rxy is approaching to 1, there is a high degree of correlation between the pages of sessions x and y. If Rxy is approaching 0, there is a less degree of correlation between the pages of the sessions x, y. In the similar procedure one can identify the correlation among the set of sessions of an unique user.

Stickiness among the pages:

The stickiness can be expressed as follows:
Stickiness, $\mathrm{S = F * D * TSR}$ (16)

Where S = Stickiness, F = Frequency, D = Duration and TSR = Total Site Reach

$$F = \frac{\text{Number of visits in time period T}}{\text{Number of Unique users who visited in T}} \tag{17}$$

$$D = \frac{\text{Total amount of time spent viewing all pages}}{\text{Number of visits in time period T}} \tag{18}$$

$$TSR = \frac{\text{Number of unique users who visited in T}}{\text{Total number of unique users}} \tag{19}$$

Therefore,

$$S = \frac{\text{Total amount of time spent in viewing all pages}}{\text{Total number of unique users}} \tag{20}$$

5. EXPERIMENTAL ANALYSIS

A) Learning performance of IPS: The server side web log data is experimented over a period of six months under standard execution environment. The experiments are proven intelligent pre processing systems are required to reduce the human intervention at the pre processing stage. The error rate between the testing data and training data is almost minimized in IPS and is found to be 0.2 on the average. Hence the experimental study is in line with the theoretical analysis of, goal of regression. The nature of relation between testing data and training data is studied and both are proven as continuous as shown in Figure 9.

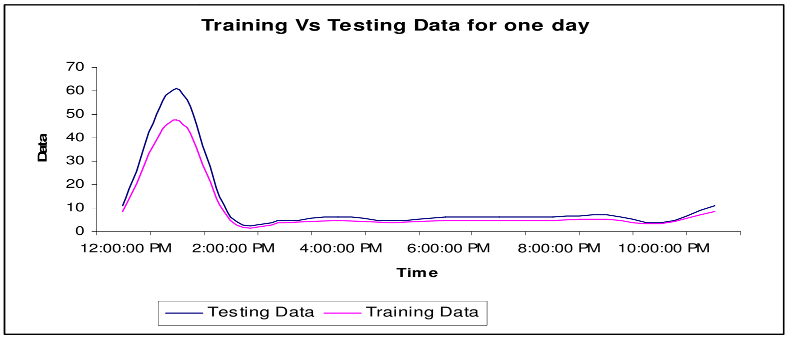


Figure 9. Testing Vs Training Data

B) Processing Performance of IPS: Several experiments are conducted in a standard environment with respect to the processing time of both present IPS and i-Miner. The results clearly indicate that IPS is essentially taking less processing time when compared with i-Miner. As the web log data grows incrementally with the time interval IPS is consistently taking less processing time than i-Miner as shown in Figure 10.

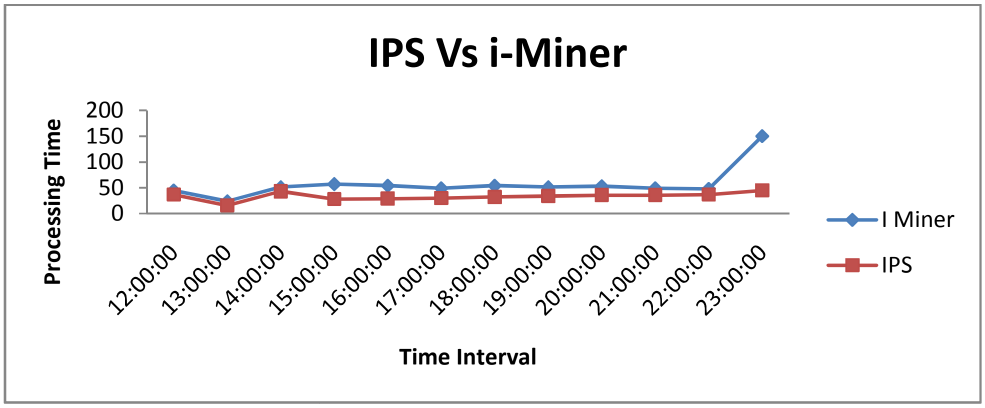


Figure 10. Processing performance

C) Memory Comparison of IFP-Tree: A set of experiments are conducted on a synthetic web log data for IFP-Tree, Cached Apriori, Feline and FP Tree for memory comparison. Among all the present IFP-Tree is essentially taking less memory. As the web log grows rapidly, still the IFP Tree is behaving constantly. For 20000 users the IFP-Tree is occupying only 40 MB as shown in Figure 11.

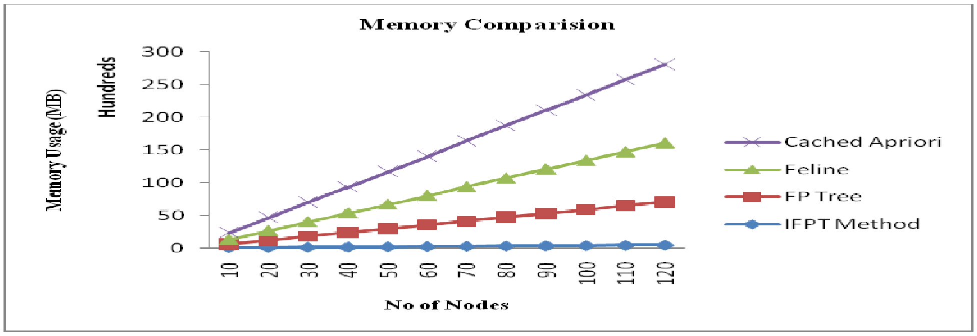


Figure 11. Memory comparison

D) Efficiency Comparison of IFP-Tree: A set of experiments are conducted on a synthetic web log data for IFP-Tree, Cached Apriori, Feline and FP Tree for efficiency comparison. The processing speed of IFP-Tree is almost twice the Cached Apriori and 40% betterment is observed when compared with other techniques. The results are shown in Figure 12.

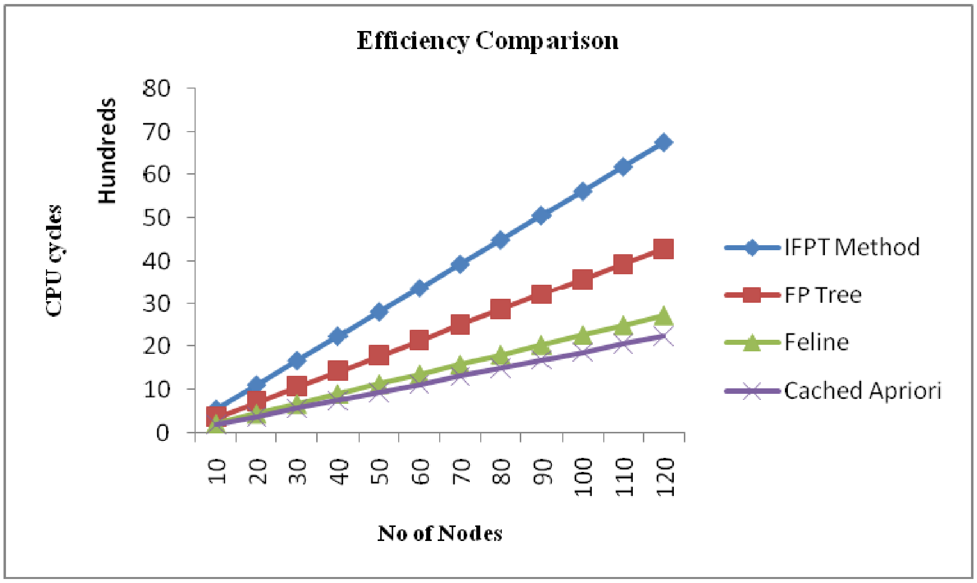


Figure 12. Efficiency comparison

E) Identification of Web user Usage Interests: In addition, the collective model IPS and IFP-Tree are experimented on educational domain (SVECW weblog data), the web user usage interests are identified with respective Time Interval as shown in Figure 13.

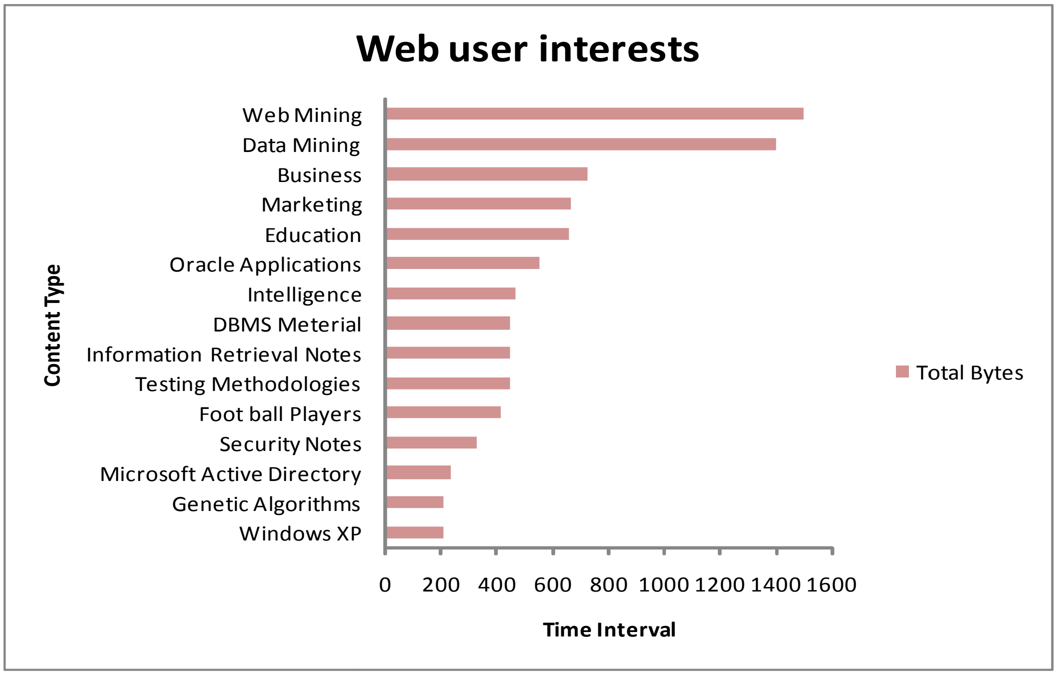


Figure 13. Web User Usage Interests

6. CONCLUSIONS

The work presented in the present paper belongs to the data mining as applied to the data on the web. Web usage mining has emerged as the essential tool for realizing more user friendly and personalized web services. Applying intelligent data pre processing techniques, modelling and advanced mining techniques, helped in resulting many successful applications in the weblog scenario. Usage patterns discovered through web usage mining are effective in capturing user-to-user relationships and similarities in browsing behaviour. Thus the present frame work focuses on both pre processing (IPS) and pattern discovery (IFP-Tree).

The IPS presented in the collective frame work, concentrated on the criticality of weblog pre processing. There are many advantages with IPS. 1) It improves the efficiency of pre processing of web log. 2) It separates human user and web search engine accesses automatically, in less time. 3) It reduces the error rate of learning algorithm. 4) The work ensures the goodness of split by using popular measures like Entropy and Gini index.

The IFP-Tree presented in the collective frame work, can handle the rapidly growing log data. There are many advantages with IFP-Tree. 1) Once the tree is built, it allows interactive mining with different supports. 2) It limits the number of scans of database to one. 3) It eliminates the generation of candidates. 4) It limits the size of the tree by removing the outdated instances. 5) It is a simple, effective and easy to apply to any application in identifying web user usage behaviour.

In conclusion, the results proven that, the employment of collective frame work of IPS and IFP-Tree gives promising solutions in the dynamic weblog scenario. This issue is becoming crucial as the size of the weblog increases at breath taking rates.

ACKNOWLEDGEMENTS

The authors recorded their acknowledgements to Shri Vishnu Engineering College for Women, Bhimavaram, Andhra University, Visakhapatnam and Acharya Nagarjuna University, Guntur authorities for their constant support and cooperation.

REFERENCESS

[1] M. Spiliopoulou, “Web Usage Mining for Site Evaluation,” Comm. ACM, , vol. 43, no. 8, 2000, pp. 127–134.

[2] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. ACM Transactions on Internet Technology (TOIT), 3(1):1_27, 2003.

[3] J.M. Kleinberg. Authoritatve sources in a hyperlinked environment. In ACM-SIAM symposium on Discrete Algorithms, 1998

[4] T. Kamdar, Creating Adaptive Web Servers Using Incremental Weblog Mining, masters thesis, Computer Science Dept., Univ. of Maryland, Baltimore, C0–1, 2001

[5] sYan Wang,Web Mining and Knowledge Discovery of Usage Patterns, February, 2000.

[6] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the World Wide Web. In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’97), 1997.

[7] J. Srivastava, P. Desikan, and V. Kumar, “Web Mining: Accomplishments and Future Directions,” Proc. US Nat’l Science Foundation Workshop on Next-Generation Data Mining (NGDM), Nat’l Science Foundation, 2002.

[8] R. Kumar et al., “Trawling the Web for Emerging Cybercommunities,” Proc. 8th World Wide Web Conf., Elsevier Science, 1999.

[9] Y. Manolopoulos et al., “Indexing Techniques for Web Access Logs,” Web Information Systems, IDEA Group, 2004.

[10] R. Armstrong et al., “Webwatcher: A Learning Apprentice for the World Wide Web,” Proc. AAAI Spring Symp. Information Gathering from Heterogeneous, Distributed Environments, AAAI Press, 1995.

[11] M.-S. Chen, J.S. Park, and P.S. Yu., “Efficient Data Mining for Path Traversal Patterns,” IEEE Trans. Knowledge and Data Eng., vol. 10, no. 2, 1998.

[12] ChengYanchun. Research on Intelligence Collecting System[J]. Journal of Shijiazhuang Railway Institute(Natural Science), 2008.

[13] Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, 1999.

[14] M. S. Chen, J. S. Park and P. S. Yu, “Efficient Data Mining for Path Traversal Patterns in a Web Environment”, IEEE Transaction on Knowledge and Data Engineering, 1998.

[15] Wang Shachi, Zhao Chengmou. Study on Enterprise Competitive IntelligenceSystem.[J]. Science Technology and Industrial, 2005.

[16] John E Prescott. Introduction to the Special Issue on Fundamentals of Competitive Intelligence. 10th Anniversary Retrospective Edition [C]. New York: John WiIey & Sons, Inc., 1999.

[17] M. Craven, S. Slattery and K. Nigam, “First-Order Learning for Web Mining”, In Proceedings of the 10th European Conference on Machine Learning, Chemnitz, 1998.

[18] Tsuyoshi, M and Saito, K. Extracting User’s Interest for Web Log Data. Proceeding of IEEE/ACM/WIC International Conference on Web Intteligence (WI’06), 2006.

[19] Savasere, A., Omiecinski, E., and Navathe, S. An Efficient Algorithm for Mining Association Rules in Large Databases. Proceedings of the VLDB Conference. 1995.

[20] Agrawal, R. and Srikant, R. Fast algorithms for mining association rules. VLDB, 487-499. 1994.

[21] Brin, S., Motwani, R., Ullman Jeffrey D., and Tsur Shalom. Dynamic itemset counting and implication rules for market basket data. SIGMOD. 1997.

[22] Han, J., Pei, J., and Yin, Y. Mining Frequent Patterns without Candidate Generation. SIGMOD, 1-12. 2000.

[23] Pei, J., Han, J., Nishio, S., Tang, S., and Yang, D. H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. Proc.2001 Int.Conf.on Data Mining. 2001.

[24] C. M. Brown, B. B. Danzig, D. Hardy, U. Manber, and M. F. Schwartz. The harvest information discovery and access system. In Proc. 2nd International World Wide Web Conference, 1994.

[25] W. B. Frakes and R. Baeza-Yates. Infomation Retrieval Data Structures and Algorithms. Prentice Hall, Englewood Cliffs, NJ, 1992.

[26] Lieberman, H. Letizia: An Agent that Assists Web Browsing. in Proc. of the 1995 International Joint Conference on Artificial Intelligence. 1995, p. 924-929, Montreal, Canada: Morgan Kaufmann.

[27] Mobasher, B., et al., Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. Data Mining and Knowledge Discovery, 2002. 6(1): p. 61-82.

[28] Hofmann, T. Probabilistic Latent Semantic Analysis. in Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval. 1999, p. 50-57, Berkeley, California, USA: ACM Press.

[29] Blei, D.M., A.Y. Ng, and M.I. Jordan, Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003(3): p. 993-1022.

[30] Jin, X., Y. Zhou, and B. Mobasher. A Unified Approach to Personalization Based on Probabilistic Latent Semantic Models of Web Usage and Content. in Proceedings of the AAAI 2004 Workshop on Semantic Web Personalization (SWP'04). 2004, San Jose.

Authors

Prof. V.V.R. Maheswara Rao received his Master of Computer Applications Degree from Osmania University, Hyderabad, India. He is working as Professor in the Dept of Computer Applications at SVECW , Bhimavaram, AP, India. He is currently pursuing his Ph.D. in Computer Science at Acharya Nagarjuna University, Guntur, India. His Research interests include Webmining, Artificial Intelligence.

Dr. V. Valli Kumari holds a PhD degree in Computer Science and Systems Engineering from Andhra University Engineering College, Visakhapatnam and is presently working as Professor in the same department. Her research interests include Security and privacy issues in Data Engineering, Network Security and E-Commerce. She is a member of IEEE and ACM.

Dr. KVSVN Raju holds a PhD degree in Computer Science from IIT, Kharagpur, and is presently working as Professor in the department of Computer Science and Systems Engineering at Andhra University Engineering College, Visakhapatnam. His research interests include Software Engineering Data Engineering and Data Security.