# AN ENHANCED PRE-PROCESSING RESEARCH FRAMEWORK FOR WEB LOG DATA USING A LEARNING ALGORITHM

V.V.R. Maheswara Rao[1] and Dr. V. Valli Kumari[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 growth and proliferation of Web services and Web based information systems, the volumes of user data have reached astronomical proportions. Before analyzing such data using web mining techniques, the web log has to be pre processed, integrated and transformed. As the World Wide Web is continuously and rapidly growing, it is necessary for the web miners to utilize intelligent tools in order to find, extract, filter and evaluate the desired information. The data pre-processing stage is the most important phase for investigation of the web user usage behaviour. To do this one must extract the only human user accesses from weblog data which is critical and complex. The web log is incremental in nature, thus conventional data pre-processing techniques were proved to be not suitable. Hence an extensive learning algorithm is required in order to get the desired information.This paper introduces an extensive research frame work capable of pre processing web log data completely and efficiently. The learning algorithm of proposed research frame work can separates human user and search engine accesses intelligently, with less time. In order to create suitable target data, the further essential tasks of pre-processing Data Cleansing, User Identification, Sessionization and Path Completion are designed collectively. The framework reduces the error rate and improves significant learning performance of the algorithm. The work ensures the goodness of split by using popular measures like Entropy and Gini index. This framework helps to investigate the web user usage behaviour efficiently. The experimental results proving this claim are given in this paper.

**KeyWords:**Web usage mining, intelligent pre-processing system, cleansing, sessionization and path completion.

## 1. INTRODUCTION

Over the last decade, with the continued increase in the usage of the WWW, web mining has been established as an important area of research. Whenever, the web users visit the WWW, they leave abundant information in web log, which is structurally complex, heterogeneous, high dimensional and incremental in nature. Analyzing such data can help to determine the browsing interest of web user. To do this, web usage mining focuses on investigating the potential knowledge from browsing patterns of the users and to find the correlation between the pages on analysis. The main goal of web usage mining is to Capture, Model and Analyze the web log data in such a way that it automatically discovers the usage behaviour of web user.

Nabendu Chaki et al. (Eds.): NeTCoM 2010,CSCP 01, pp. 01–15, 2011.

DOI : 10.5121/csit.2011.1101

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 as shown in Figure 1.

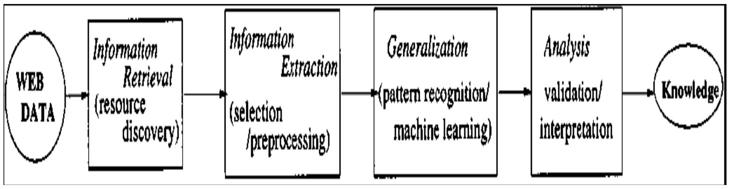

WEB DATA
Information Retrieval (resource discovery)
Information Extraction (selection /preprocessing)
Generalization (pattern recognition/ machine learning)
Analysis validation/ interpretation
Knowledge

Figure1. Web Usage Mining

**i) Resource collection:** The conventional data mining techniques assumes that the data is static, and is retrieved from the conventional databases. In web mining techniques the nature of the data is incremental and is rapidly growing. 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.

**ii) Information pre-processing:** In conventional data mining techniques information pre processing includes data cleaning, integration, transformation and reduction. 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:** All the data mining techniques can be applied on pre-processed data. Statistical methods are used to mine the relevant knowledge.

**iv) Pattern analysis:** The goal of pattern analysis is to eliminate the irrelative rules and to extract the interesting rules from the output of patterns discovery process.

As the World Wide Web is continuously and rapidly growing, it is necessary for users to utilize intelligent tools in order to find, extract, filter, and evaluate the desired information and resources.

This paper introduces an Intelligent System Web Usage Pre-processor (ISWUP), which works based on a learning algorithm. The main idea behind ISWUP is to separate the human user accesses and web search engine accesses of web log data. This ISWUP acquires the knowledge from the derived characteristics of web sessions. It discards the web search engine accesses from the web log. 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 creates the more suitable target data which helps in investigation of the web user usage behaviour.

This paper is organized as follows. In section 2, we described related work. In next section 3, we introduced the overview of proposed work. In subsequent section 4, we expressed the study of theoretical analysis. In section 5, the experimental analysis of proposed work is shown. Finally in section 6 we mention the conclusions.

## 2. RELATED WORK

Many of the previous authors are expressing 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.

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. However, these algorithms have the limitations that they can only discover the simple path traversal pattern.

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.

## 3. PROPOSED WORK

The web usage mining is a task of applying data mining techniques to extract useful patterns from web access logs. These patterns discover interesting characteristics of site visitors. Generally the web access logs are incremental, distributed and rapidly growing in nature. It is necessary for web miners to utilize intelligent tools / heuristic functions in order to find, extract, filter and evaluate the desired information.

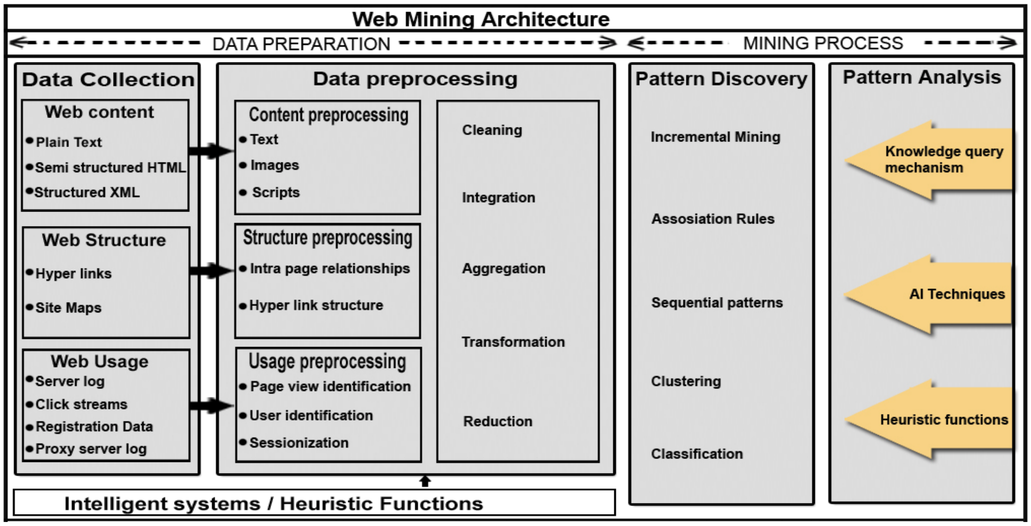


Figure 2. Web Mining Architecture

Before applying web mining techniques to web usage data, the web usage resource collection has to be cleansed, integrated and transformed. To perform the same first it is important to separate accesses made by human users and web search engines. 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. In order to create suitable target data, the further essential tasks of pre-processing Data Cleansing, User Identification, Sessionization and Path Completion are designed collectively.

## 3.1 Intelligent systems

The intelligent system takes the raw web log as input and discards the search engine accesses automatically with less time. It generates desired web log consists of only human user accesses. The web log pre-processing architecture shown in Figure3.

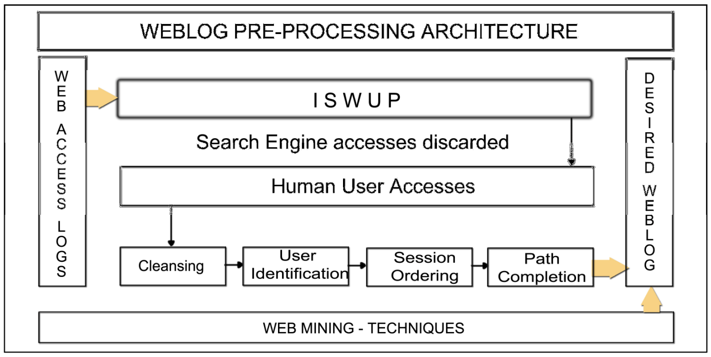


Figure 3. Web log pre-processing architecture

### 3.1.1 Working of ISWUP

The main goal of ISWUP is to separate the human user and search engines accesses. To perform this task any intelligent system requires a learning capability. Any intelligent system acquires the knowledge from the knowledge base, where knowledge base is a “set of related facts”. All the records in the web access logs are taken as testing data. Derived attributes from web logs can be considered as characteristics, which separate human user and web search engine accesses.

Total pages, inner pages, total time spent, repeated access, get, post and so on are called derived attributes. The derived attributes can be taken as a set of facts and form a knowledge base. This knowledge base can be used as training data to the ISWUP. The web usage pre processing includes cleansing, user identification, session identification, path completion and formatting. The raw web usage data collected from different resources in web access log includes IP address, unique users, requests, time stamp, protocol, total bytes and so on as shown in Table1.

| S.No | IP Address | Unique Users | Requests | Time Stamp | Protocol | Total Bytes |
|---|---|---|---|---|---|---|
| | | | | | | |

Table 1. Sample web log

To label the web sessions the ISWUP 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 ISWUP to accomplish above task.

The ISWUP 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 ISWUP 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. It works on a repeatedly posing series of questions about the characteristics of the session identification and it finally yields the class labels.

### 3.1.2 Modelling of ISWUP

The ISWUP learning tree can be constructed from a set of derived attributes from knowledge base. An efficient ISWUP learning tree algorithm has been developed to get reasonably accurate learning to discard web search engine accesses from web log accesses.

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 out comes of each session characteristics. The proposed model suggests the following characteristics 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.

### 3.1.3 Algorithm for Intelligent System ISWUP

```
TreeExtend(DA, TA)
01: If ConditionStop(DA, TA) = True then
02:     TerminalNode = CreateNewNode( )
03:     TerminalNode.Label = AssignedLabel(DA)
04:     Return TerminalNode
05: Else
06:     Root = CreateNewNode( )
07:     Root.ConditionTest = DeriveBestSplit(DA, TA)
08:     Let V ={v /v is a possible outcome of ConditionTest()}
09:     For each v Є V do
10:         $DA_v$ ={da / Root.ConditionTest(da) = v and d Є DA}
11:         Child = TreeExtend($DA_{v,}$ 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.

| No | IP Address | 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 |

Table 2. Example of web server log.

| 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 |
| Error request | Errors in requesting for web pages |
| GET | Percentage of requests made using GET method |
| Breadth | Breadth of the web traversal |
| Depth | Depth of the web traversal |
| Multi IP | Session with multiple IP addresses |

Table 3. Example of Characteristics / Derived Attributes

The main idea of ISWUP 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 3. Using ISWUP algorithm the derived characteristics are assigned to root node and intermediate nodes of the tree as shown in figure 3. The leaf nodes are labeled with human user or search engine accesses.

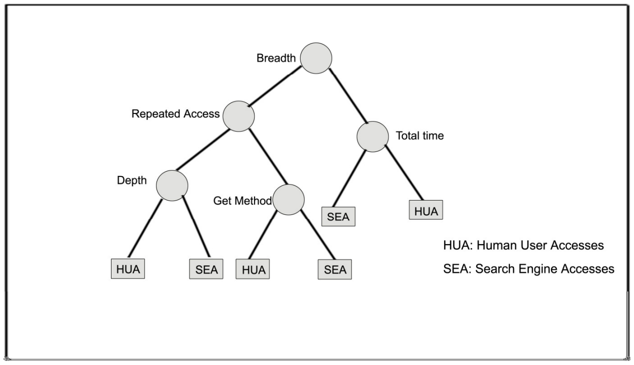


Figure 3. Example of Learning Tree

### 3.2.Data Cleansing:

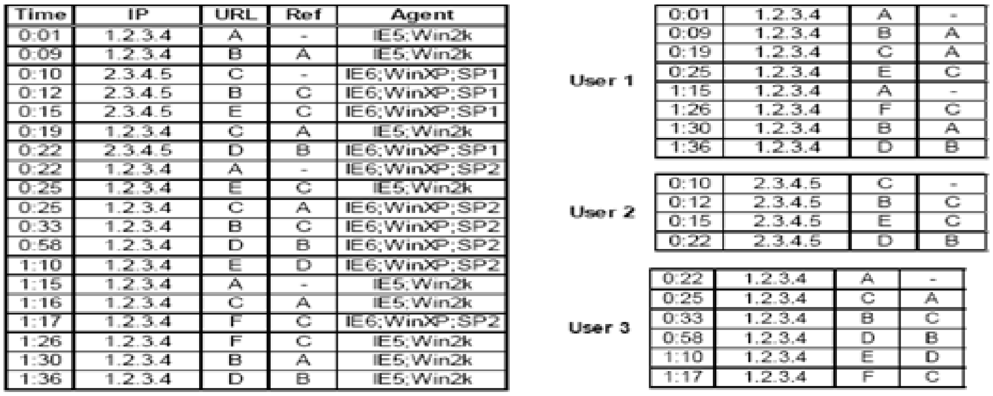
The next phase of pre processing is data cleansing. Data cleansing is usually site-specific, and involves tasks such as, removing extraneous references to embedded objects that may not be important for the purpose of analysis, including references to style files, graphics, or sound files. The cleaning 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% |

### 3.3. 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 |

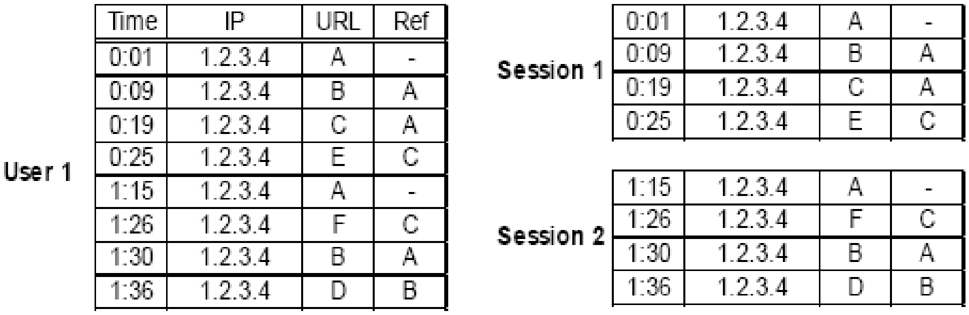
Figure 5. Example of user identification using IP + Agent

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).

### 3.4. 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 $t - t_0 \leq \theta$. 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 |

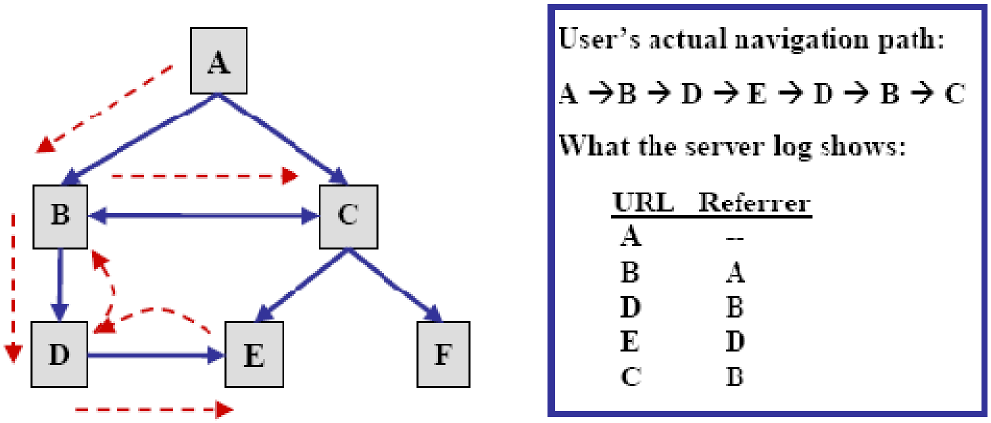
Session 2

| | | | |
|---|---|---|---|
| 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

### 3.5. 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

## 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

$$D = a + bT \qquad (1)$$

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.

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

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.

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

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

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

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

## 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.

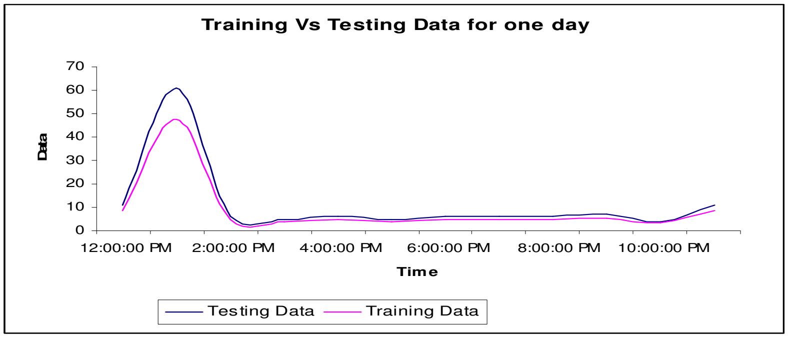


Figure 9. Testing Vs Training Data

**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.

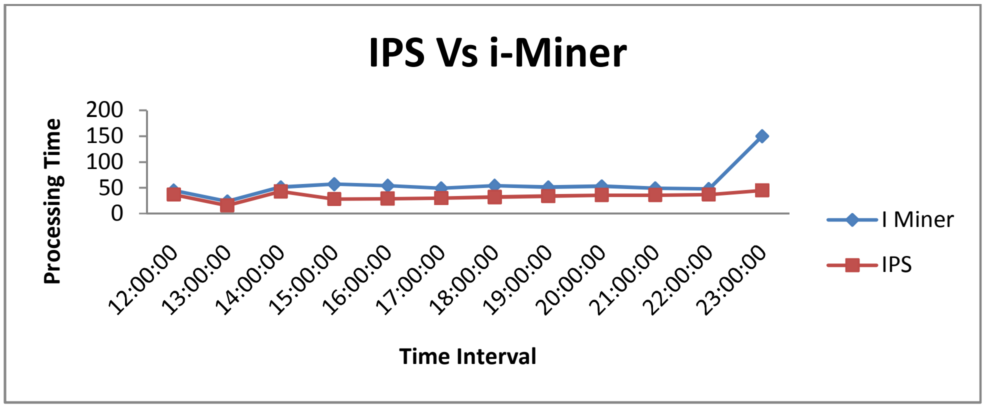


Figure 10. Processing performance

## 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 pre processing (IPS).

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.

In conclusion, the results proven that, the employment of proposed frame work of IPS 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 & Engineering at Acharya Nagarjuna University, Guntur, India. His Research interests include Web Mining, 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.