An Intelligent System for Web Usage Data Preprocessing

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

[1] Professor, Department of Computer Applications,
Shri Vishnu Engineering College for Women, Bhimavaram, W.G. Dt, 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. Web mining is an application of data mining technologies for huge data repositories. Before applying web mining techniques, the data in 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 preprocessing 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 incremental in nature, thus conventional data pre-processing techniques were proved to be not suitable as they assume that the data is static. The web logs are non scalable, impractical and are distributed in nature. Hence we require a comprehensive learning algorithm in order to get the desired information.

This paper introduces an intelligent system, capable of preprocessing web logs efficiently. It can identify human user and web search engine accesses intelligently, in less time. The system discussed reduces the error rate and improves significant learning performance of the learning algorithm. The work ensures the goodness of split by using popular measures like Entropy and Gini index. The experimental results proving this claim are given in this paper.

Keywords: ISWUP, Human user accesses, Search engine accesses, session identification.

1 Introduction

Web mining is an application of data mining technology for huge web data repositories. The web mining can be used to discover hidden patterns and relationships with in the web data. The web mining task can be divided into three general categories, known as Web Content Mining, Web Structure Mining and Web Usage Mining.

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.

N. Meghanathan et al. (Eds.): CCSIT 2011, Part I, CCIS 131, pp. 481–490, 2011.

(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 Preprocessing: Content preprocessing is the process of converting text, image, scripts and other files into the forms that can be used by the usage mining. Structure Preprocessing: The structure of a website is formed by the hyperlinks between page views. The structure preprocessing can be treated similar to the content pre processing. Usage Preprocessing: The inputs of the preprocessing 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 preprocessed 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 Preprocessor (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.

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 behavior of users for the purpose of website personalization. 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. 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.

3 Overview of the 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.

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 behavior 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.

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 preprocessing architecture shown in Figure 1.

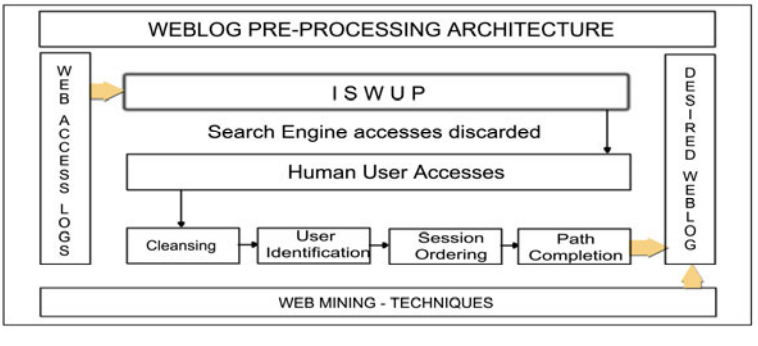

Fig. 1. Web log preprocessing architecture

3.1.1 Types of Intelligent Systems

The intelligent systems can be broadly categorized into server side intelligent systems and client side intelligent systems. The client side heuristic functions are further categorized into resource collection and information pre processing 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.

3.1.2 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.

Table 1. Sample web log

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

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

3.1.3 Modeling 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. The algorithm is developed based on the characteristics of session.

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.

3.1.4 Algorithm for Intelligent System ISWUP

TreeExtend(DA, TA)

```
1:  If ConditionStop(DA, TA) = True then
2:  TerminalNode = CreateNewNode( )
3:  TerminalNode.Label = AssignedLabel(DA)
4:  Return TerminalNode
5:  Else
6:  Root = CreateNewNode( )
7:  Root.ConditionTest = DeriveBestSplit(DA, TA)
8:  Let V ={v /v is a possible outcome of ConditionTest()}
9:  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.

3.1.5 Example for ISWUP

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 2. The leaf nodes are labeled with human user or search engine accesses.

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

Table 3. 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
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

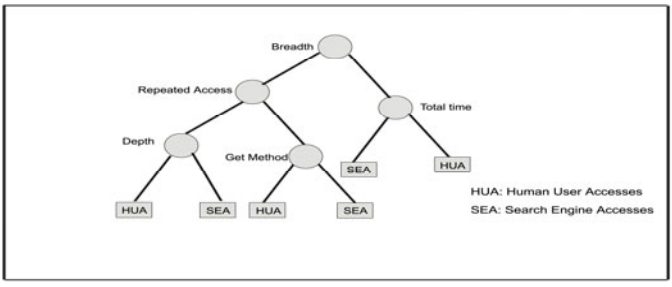

Fig. 2. Example of Learning Tree

3.2 Path Completion

Path completion is process of adding the page accesses that are not in the web log but those which be actually occurred. A mechanism is developed to infer the missing pages and generate the time stamp for missing pages.

4 Theoretical Analysis

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 modeling 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.

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

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

Equations (3) and (4) are called normal equations. By solving equations (3) and (4) for a given set of D_i, T_i 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. 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,

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

5 Experimental Study

The experiments are conducted on one day web log data. The results are analyzed and are shown in Figure 3. The error rate between the testing data and training data is almost minimized 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. The execution performance of the algorithm is constantly improving up to certain training data, from that it improves drastically as indicated in Figure 4.

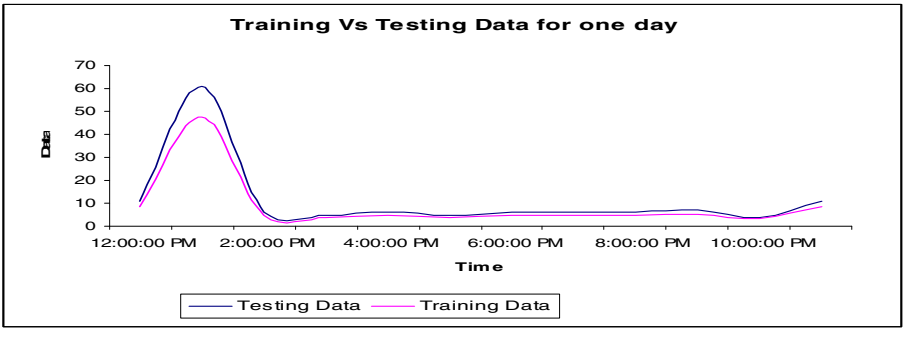

Fig. 3. Training Vs Testing Data

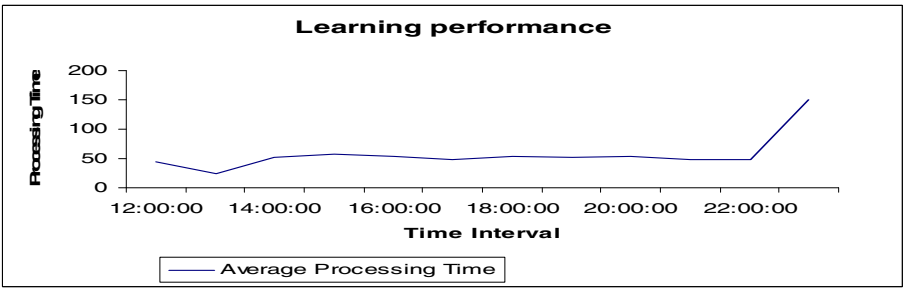

Fig. 4. Learning performance

6 Conclusion

Web mining is a new and promising research issue to help users in gaining insight into overwhelming information on the World Wide Web. The present paper discusses about the importance and criticality of web log pre-processing, including the definition, architecture and all stages of it. And the present paper introduces an intelligent

system, which improves the efficiency of preprocessing of web logs. It separates human user and web search engine accesses automatically, in less time. It reduces the error rate learning algorithm. It ensures the goodness of split by taking widely using measures like Entropy and Gini index.

References

1. Spiliopoulou, M.: Web Usage Mining for Site Evaluation. Comm. ACM 43(8), 127–134 (2000)
2. Eirinaki, M., Vazirgiannis, M.: Web mining for web personalization. ACM Transactions on Internet Technology (TOIT) 3(1), 1–27 (2003)
3. Kleinberg, J.M.: Authoritatve sources in a hyperlinked environment. In: ACM-SIAM Symposium on Discrete Algorithms (1998)
4. Kamdar, T.: Creating Adaptive Web Servers Using Incremental Weblog Mining, masters thesis, Computer Science Dept., Univ. of Maryland, Baltimore, C0–1 (2001)
5. Wang, Y.: Web Mining and Knowledge Discovery of Usage Patterns (February 2000)
6. Cooley, R., Mobasher, B., Srivastava, J.: 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 1997 (1997)
7. Srivastava, J., Desikan, P., Kumar, V.: Web Mining: Accomplishments and Future Directions. In: Proc. US Nat'l Science Foundation Workshop on Next-Generation Data Mining (NGDM). Nat'l Science Foundation (2002)
8. Kumar, R., et al.: Trawling the Web for Emerging Cybercommunities. In: Proc. 8th World Wide Web Conf., Elsevier Science, Amsterdam (1999)
9. Manolopoulos, Y., et al.: Indexing Techniques for Web Access Logs. Web Information Systems, IDEA Group (2004)
10. Armstrong, R., et al.: Webwatcher: A Learning Apprentice for the World Wide Web. In: Proc. AAAI Spring Symp. Information Gathering from Heterogeneous, Distributed Environments. AAAI Press, Menlo Park (1995)
11. Chen, M.-S., Park, J.S., Yu, P.S.: Efficient Data Mining for Path Traversal Patterns. IEEE Trans. Knowledge and Data Eng. 10(2) (1998)
12. Yanchun, C.: 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. Chen, M.S., Park, J.S., Yu, P.S.: Efficient Data Mining for Path Traversal Patterns in a Web Environment. IEEE Transaction on Knowledge and Data Engineering (1998)