Web Usage Mining:

An Implementation

A thesis report
Submitted in partial fulfillment of the requirements for the award of the degree

Bachelor of Technology
In
Computer Science & Engineering

Submitted By:
Aniket Dash -10606013
Liju Robin George-10606023

National Institute of Technology, Rourkela
Rourkela-769008, Orissa, India

.

Web Usage Mining:

An Implementation

A thesis report
Submitted in partial fulfillment of the requirements for the award of the degree

Bachelor of Technology
In
Computer Science & Engineering

Submitted By:

Aniket Dash -10606013

Liju Robin George-10606023

Under the Guidance of:
Prof. Korra Sathya Babu
Assistant Professor
Department of Computer Science & Engineering
NIT Rourkela

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Rourkela-769008, Orissa, India

Declaration

This is to certify that the work in this Thesis Report entitled "**Web Usage Mining – An Implementation**" by **Aniket Dash** and **Liju Robin George** has been carried out under my supervision in partial fulfillment of the requirements for the degree of **Bachelor of Technology**, in Computer Science during session **2009-2010** in the Department of Computer Science and Engineering, **National Institute of Technology Rourkela**, and this work has not been submitted elsewhere for a degree.

Place : Rourkela

Date : 30/04/2010

Prof. Korra Sathya Babu
Assitant Professor
Department of C S E
NIT Rourkela

Acknowledgement

It is with great satisfaction and pride that we present our thesis on the project undertaken under the “Research Project” paper during 7th and 8th semesters, for partial fulfillment of our Bachelor of Technology degree at NIT Rourkela. We are thankful to Assistant Prof. K. Sathya Babu for being our mentor in this journey. He has been the corner stone of our project and has guided us during our troubles. We are also thankful to all the professors of our department for being a constant source of inspiration and motivation during the course of the project. We offer our heartiest thanks to our friends for their constant inspirations and also Mr. M. Madhan, Librarian, NIT Rourkela central Library for providing us with the DSpace Server Logs. Last but not the least, we want to acknowledge the contributions of our parents and family members, for their constant motivation, inspirations and their belief in us that we could make it happen.

Aniket Dash

Liju Robin George

Contents

Abstract

Web usage mining is the area of data mining which deals with the discovery and analysis of usage patterns from Web data, specifically web logs, in order to improve web based applications. Web usage mining consists of three phases, preprocessing, pattern discovery, and pattern analysis. After the completion of these three phases the user can find the required usage patterns and use these information for the specific needs.

In this project, the DSpace log files have been preprocessed to convert the data stored in them into a structured format. Thereafter, the general procedures for bot-removal and session-identification from a web log file, have been written down with certain modifications pertaining to the DSpace log files, in an algorithmic form. Furthermore, analysis of these log files using a subjective interpretation of a recently proposed algorithm EIN-WUM has also been conducted. This algorithm is based on the artificial immune system model and uses this model to learn and extract information present in the web data i.e server logs. This algorithm has been duly modified according to DSpace@NITR Website structure.

List Of Figures

Chapter 1: Introduction

Chapter 1

Introduction

1.1 Introduction

Web Usage Mining is a part of Web Mining, which, in turn, is a part of Data Mining. As Data Mining involves the concept of extraction meaningful and valuable information from large volume of data, Web Usage mining involves mining the usage characteristics of the users of Web Applications. This extracted information can then be used in a variety of ways such as, improvement of the application, checking of fraudulent elements etc.

Web Usage Mining is often regarded as a part of the Business Intelligence in an organization rather than the technical aspect. It is used for deciding business strategies through the efficient use of Web Applications. It is also crucial for the Customer Relationship Management (CRM) as it can ensure customer satisfaction as far as the interaction between the customer and the organization is concerned.

The major problem with Web Mining in general and Web Usage Mining in particular is the nature of the data they deal with. With the upsurge of Internet in this millennium, the Web Data has become huge in nature and a lot of transactions and usages are taking place by the seconds. Apart from the volume of the data, the data is not completely structured. It is in a semi-structured format so that it needs a lot of preprocessing and parsing before the actual extraction of the required information.

In this project, we have taken up a small part of the Web Usage Mining process, which involves the Preprocessing, User Identification, Bot-removal and Analysis of the Dspace@NITR Web Server Logs.

1.2 Motivation

In the current era, we are witnessing a surge of Web Usage around the globe. A large volume of data is constantly being accessed and shared among a varied type of users; both humans and intelligent machines. Thus, taking up a structured approach to control this information exchange, has what made Web Mining one of the hot topics in the field of Information Technology. This very reason motivated us to take up this topic as our B. Tech project. Another consideration, albeit minor, was the fact there has been a minimal amount of research work on Web Usage Mining in Department of CSE, NIT Rourkela.

1.3 Thesis Layout:

The entire Thesis is divided into the following Chapters

1. Introduction
2. Web Usage Mining – An Overview
3. Key Constraints and Solutions
4. Implementation
5. Conclusion

Chapter 2: Web Usage Mining : Overview

Chapter 2

Overview

2.1 WEB DATA[1]

In Web Usage Mining, data can be collected in server logs, browser logs, proxy logs, or obtained from an organization's database. These data collections differ in terms of the location of the data source, the kinds of data available, the segment of population from which the data was collected, and methods of implementation.

There are many kinds of data that can be used in Web Mining.

1. Content: The *visible* data in the Web pages or the information which was meant to be imparted to the users. A major part of it includes text and graphics (images).
2. Structure: Data which describes the organization of the website. It is divided into two types. *Intra-page* structure information includes the arrangement of various HTML or XML tags within a given page. The principal kind of *inter-page* structure information are the hyper-links used for site navigation.
3. Usage: Data that describes the usage patterns of Web pages, such as IP addresses, page references, and the date and time of accesses and various other information depending on the log format.

2.2 Data Sources[1]

The data sources used in Web Usage Mining may include web data repositories like:

1. **Web Server Logs** – These are logs which maintain a history of page requests. The W3C maintains a standard format for web server log files, but other proprietary formats exist. More recent entries are typically appended to the end of the file. Information about the request, including client IP address, request date/time, page requested, HTTP code, bytes served, user agent, and referrer are typically added. These data can be combined into a single file, or separated into distinct logs, such as

an access log, error log, or referrer log. However, server logs typically do not collect user-specific information. These files are usually not accessible to general Internet users, only to the webmaster or other administrative person. A statistical analysis of the server log may be used to examine traffic patterns by time of day, day of week, referrer, or user agent. Efficient web site administration, adequate hosting resources and the fine tuning of sales efforts can be aided by analysis of the web server logs. Marketing departments of any organization that owns a website should be trained to understand these powerful tools.

2. **Proxy Server Logs[1]** - A Web proxy is a caching mechanism which lies between client browsers and Web servers. It helps to reduce the load time of Web pages as well as the network traffic load at the server and client side. Proxy server logs contain the HTTP requests from multiple clients to multiple Web servers. This may serve as a data source to discover the usage pattern of a group of anonymous users, sharing a common proxy server.

3. **Browser Logs[1]** – Various browsers like Mozilla, Internet Explorer etc. can be modified or various JavaScript and Java applets can be used to collect client side data. This implementation of client-side data collection requires user cooperation, either in enabling the functionality of the JavaScript and Java applets, or to voluntarily use the modified browser. Client-side collection scores over server-side collection because it reduces both the bot and session identification problems.

2.3 Information Obtained[3]

1. **Number of Hits**: This number usually signifies the number of times any resource is accessed in a Website. A hit is a request to a web server for a file (web page, image, JavaScript, Cascading Style Sheet, etc.). When a web page is uploaded from a server the number of "hits" or "page hits" is equal to the number of files requested. Therefore, one page load does not always equal one hit because often pages are made up of other images and other files which stack up the number of hits counted.
2. **Number of Visitors:** A "visitor" is exactly what it sounds like. It's a human who navigates to your website and browses one or more pages on your site.

3. **Visitor Referring Website:** The referring website gives the information or url of the website which referred the particular website in consideration.

4. **Visitor Referral Website:** The referral website gives the information or url of the website which is being referred to by the particular website in consideration.

5. **Time and Duration:** This information in the server logs give the time and duration for how long the Website was accessed by a particular user.

6. **Path Analysis:** Path analysis gives the analysis of the path a particular user has followed in accessing contents of a Website.

7. **Visitor IP address:** This information gives the Internet Protocol(I.P.) address of the visitors who visited the Website in consideration.

8. **Browser Type:** This information gives the information of the type of browser that was used for accessing the Website.

9. **Cookies:** A message given to a Web browser by a Web server. The browser stores the message in a text file called cookie. The message is then sent back to the server each time the browser requests a page from the server. The main purpose of cookies is to identify users and possibly prepare customized Web pages for them. When you enter a Web site using cookies, you may be asked to fill out a form providing such information as your name and interests. This information is packaged into a cookie and sent to your Web browser which stores it for later use. The next time you go to the same Web site, your browser will send the cookie to the Web server. The server can use this information to present you with custom Web pages. So, for example, instead of seeing just a generic welcome page you might see a welcome page with your name on it.

10. **Platform:** This information gives the type of Operating System etc. that was used to access the Website.

2.4 Possible Actions[3]

1. **Shortening Paths of High visit Pages:** The pages which are frequently accessed by the users can be seen as to follow a particular path. These pages can be included in an easily accessible part of the Website thus resulting in the decrease in the navigation path length.

2. **Eliminating or Combining Low Visit Pages:** The pages which are not frequently accessed by users can be either removed or their content can be merged with pages with frequent access.

3. **Redesigning Pages to help User Navigation:** To help the user to navigate through the website in the best possible manner, the information obtained can be used to redesign the structure of the Website.

4. **Redesigning Pages For Search Engine Optimization:** The content as well as other information in the website can be improved from analyzing user patterns and this information can be used to redesign pages for Search Engine Optimization so that the search engines index the website at a proper rank.

5. **Help Evaluating Effectiveness of Advertising Campaigns:** Important and business critical advertisements can be put up on pages that are frequently accessed.

2.5 Web Usage Mining Process[1]:

The main processes in Web Usage Mining are:

Preprocessing: Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Commonly used as a preliminary data mining practice, data preprocessing transforms the data into a format that will be more easily and effectively processed for the purpose of the user. The different types of preprocessing in Web Usage Mining are:

1. Usage Pre-Processing: Pre-Processing relating to Usage patterns of users.
2. Content Pre-Processing: Pre-Processing of content accessed.
3. Structure Pre-Processing: Pre-Processing related to structure of the website.

Pattern Discovery: Web Usage mining can be used to uncover patterns in server logs but is often carried out only on samples of data. The mining process will be ineffective if the samples are not a good representation of the larger body of data. The following are the pattern discovery methods.

1. Statistical Analysis
2. Association Rules

3. Clustering
4. Classification
5. Sequential Patterns
6. Dependency Modeling

Pattern Analysis [1]: This is the final step in the Web Usage Mining process. After the preprocessing and pattern discovery, the obtained usage patterns are analyzed to filter uninteresting information and extract the useful information. The methods like SQL (Structured Query Language) processing and OLAP (Online Analytical Processing) can be used.

2.6 Web Usage Mining Areas[1]

1. Personalization
2. System Improvement
3. Site Modification
4. Business Intelligence
5. Usage Characterization

2.7 Web Usage Mining Applications[1]

1. Letizia[4]

Letizia is an application that assists a user browsing the Internet. As the user operates a conventional Web browser such as Mozilla, the application tracks usage patterns and attempts to predict items of interest by performing concurrent and autonomous exploration of links from the user's current position. The application uses a best-first search augmented by heuristics inferring user interest from browsing behavior.

2. **WebSift[1]**

The WebSIFT (Web Site Information Filter) system is another application which performs Web Usage Mining from server logs recorded in the extended NSCA format (includes referrer and agent fields), which is quite similar to the combined log format which used in case of DSpace log files. The preprocessing algorithms include identifying users, server sessions, and identifying cached page references through the use of the referrer field. It identifies interesting information and frequent item sets from mining usage data.

3. **Adaptive Websites**

An adaptive website adjusts the structure, content, or presentation of information in response to measured user interaction with the site, with the objective of optimizing future user interactions. Adaptive websites are web sites that automatically improve their organization and presentation by learning from their user access patterns. User interaction patterns may be collected directly on the website or may be mined from Web server logs. A model or models are created of user interaction using artificial intelligence and statistical methods. The models are used as the basis for tailoring the website for known and specific patterns of user interaction.

2.8 Analysis of Web Server Logs

We used different web server log analyzers like Web Expert Lite 6.1 and Analog6.0 to analyze various sample web server logs obtained.

The key information obtained was:

Total Hits, Visitor Hits, Average Hits per Day, Average Hits per Visitor, Failed Requests, Page Views Total Page Views, Average Page Views per Day , Average Page Views per

Visitor, Visitors Total Visitors Average Visitors per Day, Total Unique IPs , Bandwidth, Total Bandwidth , Visitor Bandwidth , Average Bandwidth per Day, Average Bandwidth per Hit, Average Bandwidth per Visitor.

Access Data like files, images etc., Referrers, User Agents etc.

Analysis of above obtained information proved Web Usage Mining as a powerful technique in Web Site Management and improvement.

Chapter 3: Key Constraints and Solutions

3.1 Collection of DSpace Log Files

3.2 Analysis of Log File

3.3 Bot Removal

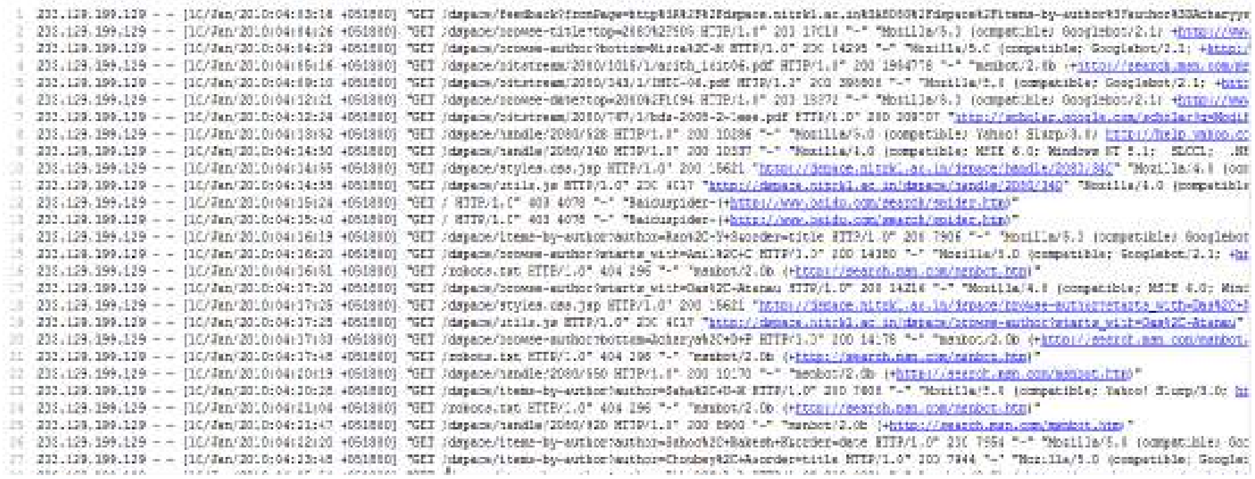
3.4 User Identification

Chapter 3

Key Constraints and Solutions

3.1. Collection of Dspace Log Files

Below is a snapshot of the collected log Dspace server log file



```
232.129.199.129 - - [10/Jan/2010:04:03:18 +051800] "GET /dspace/feedback?fromPage=http%3A%2F%2Fdspace.nitrkl.ac.in%3A8080%2Fdspace%2Fitems-by-author%3Fauthor%3DAcharya
232.129.199.129 - - [10/Jan/2010:04:04:26 +051800] "GET /dspace/browse-title?top=2080%2F506 HTTP/1.1" 200 17018 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.
232.129.199.129 - - [10/Jan/2010:04:04:29 +051800] "GET /dspace/browse-author?bottom=Misra%2C+N HTTP/1.0" 200 14295 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http:/
232.129.199.129 - - [10/Jan/2010:04:06:16 +051800] "GET /dspace/bitstream/2080/1016/1/arith_icit06.pdf HTTP/1.1" 200 1964778 "-" "msnbot/2.0b (+http://search.msn.com/ms
232.129.199.129 - - [10/Jan/2010:04:09:10 +051800] "GET /dspace/bitstream/2080/343/1/IMEC-06.pdf HTTP/1.1" 200 298808 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http:
232.129.199.129 - - [10/Jan/2010:04:12:21 +051800] "GET /dspace/browse-date?top=2080%2F1094 HTTP/1.1" 200 18272 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.
232.129.199.129 - - [10/Jan/2010:04:12:24 +051800] "GET /dspace/bitstream/2080/787/1/bds-2008-2-1eee.pdf HTTP/1.0" 200 209707 "http://scholar.google.com/scholar?q=Modifi
232.129.199.129 - - [10/Jan/2010:04:13:52 +051800] "GET /dspace/handle/2080/528 HTTP/1.1" 200 10286 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; http://help.yahoo.co
232.129.199.129 - - [10/Jan/2010:04:14:50 +051800] "GET /dspace/handle/2080/340 HTTP/1.1" 200 10337 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;  SLCC1;  .NE
232.129.199.129 - - [10/Jan/2010:04:14:55 +051800] "GET /dspace/styles.css.jsp HTTP/1.0" 200 15621 "http://dspace.nitrkl.ac.in/dspace/handle/2080/340" "Mozilla/4.0 (com
232.129.199.129 - - [10/Jan/2010:04:14:55 +051800] "GET /dspace/utils.js HTTP/1.0" 200 4017 "http://dspace.nitrkl.ac.in/dspace/handle/2080/340" "Mozilla/4.0 (compatible
232.129.199.129 - - [10/Jan/2010:04:15:24 +051800] "GET / HTTP/1.0" 403 4078 "-" "Baiduspider+(+http://www.baidu.com/search/spider.htm)"
232.129.199.129 - - [10/Jan/2010:04:15:40 +051800] "GET / HTTP/1.0" 403 4078 "-" "Baiduspider+(+http://www.baidu.com/search/spider.htm)"
232.129.199.129 - - [10/Jan/2010:04:16:19 +051800] "GET /dspace/items-by-author?author=Rao%2C+Y+S&order=title HTTP/1.0" 200 7906 "-" "Mozilla/5.0 (compatible; Googlebot
232.129.199.129 - - [10/Jan/2010:04:16:20 +051800] "GET /dspace/browse-author?starts_with=Ani1%2C+C HTTP/1.0" 200 14380 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +ht
232.129.199.129 - - [10/Jan/2010:04:16:51 +051800] "GET /robots.txt HTTP/1.0" 404 296 "-" "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
232.129.199.129 - - [10/Jan/2010:04:17:20 +051800] "GET /dspace/browse-author?starts_with=Das%2C+Atanu HTTP/1.0" 200 14216 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Wind
232.129.199.129 - - [10/Jan/2010:04:17:25 +051800] "GET /dspace/styles.css.jsp HTTP/1.0" 200 15621 "http://dspace.nitrkl.ac.in/dspace/browse-author?starts_with=Das%2C+A
232.129.199.129 - - [10/Jan/2010:04:17:25 +051800] "GET /dspace/utils.js HTTP/1.0" 200 4017 "http://dspace.nitrkl.ac.in/dspace/browse-author?starts_with=Das%2C+Atanu"
232.129.199.129 - - [10/Jan/2010:04:17:33 +051800] "GET /dspace/browse-author?bottom=Acharya%2C+B+P HTTP/1.0" 200 14178 "-" "msnbot/2.0b (+http://search.msn.com/msnbot.
232.129.199.129 - - [10/Jan/2010:04:17:48 +051800] "GET /robots.txt HTTP/1.0" 404 296 "-" "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
232.129.199.129 - - [10/Jan/2010:04:20:19 +051800] "GET /dspace/handle/2080/550 HTTP/1.1" 200 10170 "-" "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
232.129.199.129 - - [10/Jan/2010:04:20:28 +051800] "GET /dspace/items-by-author?author=Saha%2C+S+K HTTP/1.0" 200 7808 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; ht
232.129.199.129 - - [10/Jan/2010:04:21:04 +051800] "GET /robots.txt HTTP/1.0" 404 296 "-" "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
232.129.199.129 - - [10/Jan/2010:04:21:47 +051800] "GET /dspace/handle/2080/820 HTTP/1.1" 200 8900 "-" "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
232.129.199.129 - - [10/Jan/2010:04:22:20 +051800] "GET /dspace/items-by-author?author=Sahoo%2C+Rakesh+&order=date HTTP/1.0" 200 7554 "-" "Mozilla/5.0 (compatible; Goo
232.129.199.129 - - [10/Jan/2010:04:23:45 +051800] "GET /dspace/items-by-author?author=Choubey%2C+A&order=title HTTP/1.0" 200 7844 "-" "Mozilla/5.0 (compatible; Googleb
```

Figure 3.1

Features of Dspace Log Files

1. **Combined Log Format**: Combined Log Format is quite similar to the NCSA Common log format which contains only basic HTTP access information. The NCSA Combined Log is the second of three logs in the NCSA Separate log format.

The Combined log contains the requested resource and a few other pieces of information like referral, user agent, and cookie information. The information is contained in a single file.

The fields in the Combined log file format are: host rfc931 username date:time request statuscode bytes referrer user agent .The following example shows these fields populated with values in a common log file record:

```
125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html
HTTP/1.0" 200 1043 “http://www.example.com“ "Mozilla/4.0 (compatible;
MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0;
.NET CLR 3.5.30729; .NET CLR 3.0.30618; InfoPath.2)"
```

The following is a description of the fields in the fields in the Combined log format:

host (125.125.125.125 in the example)
The IP address or host/subdomain name of the HTTP client that made the HTTP resource request.

rfc931 ("-" in the example) :
The identifier used to identify the client making the HTTP request. If no value is present, a "-" is substituted.

username (dsmith in the example) :
The username, (or user ID) used by the client for authentication. If no value is present, a "-" is substituted.

date:time timezone ([10/Oct/1999:21:15:05 +0500] in the example) :
The date and time stamp of the HTTP request.

The fields in the date/time field are:

[dd/MMM/yyyy:hh:mm:ss +-hhmm]

where the fields are defined as follow:
dd is the day of the month
MMM is the month
yyy is the year
:hh is the hour

:mm is the minute

:ss is the seconds

+-hhmm is the time zone

In practice, the day is typically logged in two-digit format even for single-digit days. For example, the second day of the month would be represented as 02. However, some HTTP servers do log a single digit day as a single digit. When parsing log records, you should be aware of both possible day representations.

request ("GET /index.html HTTP/1.0" in the example)

The HTTP request. The request field contains three pieces of information. The main piece is the requested resource (index.html). The request field also contains the HTTP method (GET) and the HTTP protocol version (1.0).

statuscode (200 in the example)

The status is the numeric code indicating the success or failure of the HTTP request.

bytes (1043 in the example)

The bytes field is a numeric field containing the number of bytes of data transferred as part of the HTTP request, not including the HTTP header.

The rest fields though provided are of little importance as far as this implementation is concerned except that the last field is used to identify whether the entry is a bot entry or not.

2. **Not much Variation in IP**: As we are considering the dsapce log files, which is specific to NIT Rourkela, it is observed that there is not much variation in IP addresses in the entries recorded in the log file

    3. **Username and aliases not provided**: The second and third entries in the common log format are the usernames and aliases which are mainly recorded in a login based website. These information are not provided in the dspace log files.

4. **Crawlers[5]**:

A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Other terms for Web crawlers are ants, automatic indexers, bots, and worms or Web spider, Web robot, or Web scutter[5].

This process is called Web crawling or spidering[5]. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for spam).

A Web crawler is one type of bot[5], or software agent. In general, it starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies

The following are the types of crawlers found in the DSpace Log Files

A. MSN Bots

B. Yahoo Slurps

C. Google Bots

D. Baidu Spiders etc.

Problems Faced

1. Bots Removal
2. User Identification

3.2. Analysis of the Log File

First Part of Analysis was preprocessing. Preprocessing segregated all the details provided in the log file into a structured form.

Language used: JAVA

Data Structures used: Linear Arrays: ip[], time[], content[], httpmethod[], httpstatus[], bandwidth[], browser[] etc.

The preprocessing program collected the details in the appropriate data structures and also identified whether an entry is a bot entry or a valid user entry.

3.3. Bot Identification

After much research and analysis of bot identification and removal, we came up with a method specific to Dspace log file to do the same. The pseudocode for the method is as follows.

3.3.1 Pseudocode for BotID

```
begin
        while(!EOF)
        begin
                readLine();
                Check for keywords (bot, slurp, spider) in browser[] array
                if the array contains keyword
                begin
                    botflag=true;
                    botcounter++;
                end
                else
                    botflag=false;
        end
end
```

The bot entries are not considered as valid entries while defining user sessions.

3.4. Identification of User Sessions

User or Sessions in Web Usage Mining generally refers to the usage or access of any content of the website from a fixed IP over a fixed period of time.

The period of time is subjective to the analyzer.

Considering the above requirements, we came up with a method specific to Dspace log file to identify user sessions in the log file. The pseudocode for the method is as follows:

3.4.1 Pseudocode for SessionID

```
begin
    while(!EOF)
    begin
```

```
        i=1;
        add first not bot entry to session i;
        for each (next entry)
        begin
            if(entry != bot)
                if(IP == Previous IP)
                    if(time[this entry] - time[this entry -1] < x)
                        add entry to session i;
                    else
                        begin
                            i++;
                            add entry to session i;
                        end
                else
                begin
                    i++;
                    add entry to session i;
                end
        end
end
```

Chapter 4: Implementation

Chapter 4

Implementation

4.1 EIN-WUM[2]

A brief pseudo Code of the algorithm is as follows (as taken from [2])

```
1- Initialize antibodies using some of input data (sessions)
2- Construct neighborhoods using a simple clustering method.
3- Set the neighborhood threshold to average dissimilarity between cluster prototypes.
4- For each antigen (in coming sessions)
    4-1 Calculate danger level of antigen (interestingness of session), if danger level of antigen is more than a threshold continue, else go to 4.
    4-2 Present antigens to cluster prototypes.
    4-3 Choose the most activated neighborhood.
    4-4 If affinity between antigen and selected neighborhood prototype is less than neighborhood threshold add a new neighborhood with a copy of antigen to the network, update neighborhood information and go to step 4.
    4-5 Else calculate stimulation level of antibodies in the selected neighborhood, update neighborhood information and update antibodies' vector according to flaw or supplementary state.
    4-6 Clone antibodies.
    4-7 After processing every $T$ antigen, mutate antibodies, add the new antibodies to the network.
    4-8 Delete excess antibodies with least stimulation level, move stagnated antibodies to second memory.
```

4.2 Interpretation

Our Interpretation of the algorithm is as follows

- Limit value of no. of antibodies to 6 (based on the category from Dspace Website.
- We define the category of each entry in the Server Log by assigning it a number (0 through 6). The numbers signify

  0 – Default value
  1 - Content searched by title
  2 - Content searched by author
  3 - Content searched by date
  4 - Content searched by author
  5 – Content accessed by handle
  6 - Content accessed by bitstream

- The antibodies are initialized from the first 10 sessions. For each session an entry goes to the corresponding number of antibody as its category is. So ech antibody contains only one category of server log entry.
- For each in coming session

```
    Compare with each existing antibody
    If (Similarity of antibody > threshold)
        Replace old session with new session
    Else if( similarity < threshold)
        Update antibody with most similarity
```

- Put a limit on the size of antibody
- If (antibody crosses limit)

```
        Delete old entries.
```

Here Danger level[2] or interestingness is based on content.

4.3 Utility

Utility of the above Interpretation

a) At the end of the program, the ten most interesting antibodies will remain.

b) The contents accessed in the antibodies will be the most frequently accessed contents in the whole website.

c) Based on **(b)** the following changes can be brought to the concerned site

- **i)** Improvements on frequently accessed pages
- **ii)** Deletion or merging of unused pages
- **iii)** Improvement of content
- **iv)** Improve interaction with referral sites.

4.4 Results

The results obtained from our analysis include

A. Preprocessed log files information e.g

```
1      true     203.129.199.129   10/Jan/2010:04:04:26       GET        200          17013B
0                        /dspace/browse-title?top=2080%2F905

2      true     203.129.199.129   10/Jan/2010:04:04:29       GET        200          14295B
0                        /dspace/browse-author?bottom=Misra%2C+M
```

The data separated by spaces in the above examples denote the various details about the user hits. The second data item represents whether the hit has come from a bot or a human user.

B. Summary of the log file giving overall details as an example shown below

```
***************Summary****************************
number of hits = 14274
number of visitor hits= 7923
number of spider hits= 6351
Number of days= 5
Average hits per day = 2854
Total Bandwidth used = 1494419892 Bytes
Avenrage Bandwidth= 298883978 Bytes
*******************************************************
```

C. The sessions and the different log file entries that constitute of the sessions as shown below

```
session 1   182
session 1   183
session 1   191
session 2   193
session 2   194
session 2   195
session 2   196
session 2   197
session 2   198
```

D. The different frequently accessed contents in the Dspace@NITR website for example

```
16 0 1 /dspace/browse-author?starts_with=Das%2C+Atanau
92 0 1 /dspace/browse-author?bottom=Wai%2C+P+K+A
181 0 1 /dspace/browse-author?starts_with=Verghese%2C+L
227 1 1 /dspace/browse-author?top=Joshi%2C+Avinash
364 1 1 /dspace/browse-author?top=Joshi%2C+Avinash
```

```
527 5 1 /dspace/browse-author
530 5 1 /dspace/browse-author?starts_with=C
532 5 1 /dspace/browse-author?top=Chatterjee%2C+Saurav
536 5 1 /dspace/browse-author?starts_with=S
569 7 1 /dspace/browse-author?starts_with=S
571 7 1 /dspace/browse-author?top=Chatterjee%2C+Saurav
715 8 1 /dspace/browse-author?top=Bal%2C+S
748 8 1 /dspace/browse-author?starts_with=Das%2C+B+M
831 8 1 /dspace/browse-author?starts_with=Karanam%2C+U+M+R
```

This sample result has been taken from the final status of the antibody 2 (with 2nd category entries. For categories see section 4.2)

Chapter 5: Future Work and Conclusion

Chapter 5

5.1 Future Work

The research and implementation that we have presented in this thesis is in a nascent stage and is purely DSpace@NITR Website specific. However this subjective interpretation of the algorithm EIN-WUM [2] is very ingenious and proposes a lot of scope to be extended on to other problem domains. Furthermore, anyone interested in this field can take a similar approach and modify these methods to expand them to a general scenario or to shift their use to a different area.

5.2 Conclusion

The proposed methods were successfully tested on the log files for bot removal and user sessions identification. The results which were obtained after the analysis were satisfactory and contained valuable information about the Log Files.

The subjective interpretation and implementation of the EIN-WUM [2] algorithm produced results which depicted the usage patterns (frequently accessed contents) of the DSpace@NITR website users.

References

[1]. Jaideep Srivastava , Robert Cooley, Mukund Deshpande, Pang-Ning Tan, *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data,* SIGKDD Explorations, Volume 1, Issue 2- Pages 12-23.

[2]. Adel T. Rahmani and B. Hoda Helmi, *EIN-WUM an AIS-based Algorithm for Web Usage Mining*, Proceedings of GECCO’08, July 12–16, 2008, Atlanta, Georgia, USA, ACM978-1-60558-130-9/08/07 (Pages 291-292)

[3]. Dr. G. K. Gupta, *Introduction to Data Mining with Case Studies*, , PHI Publication

[4]. http://web.media.mit.edu/~lieber/Lieberary/Letizia/Letizia-Intro.html

[5]. http://en.wikipedia.org/wiki/Web_crawler