WEBANALYZER: A TOOL FOR EFFECTIVE WEB USER BEHAVIOR MODELING

by

NAVEED AHMED

(Under the Direction of Eileen T. Kraemer)

ABSTRACT

The World Wide Web has become a major source of information dissemination for academia, business and government organizations. Hence, the usability and effectiveness of these websites is increasingly important. User behavior modeling is an important element of such evaluations. We have developed a tool, *WebAnalyzer,* that lets website administrators select the "best" parameters (number of clusters, distance measures) for clustering user sessions, representations of user behavior while interacting with a web site. Clustering of labeled session data is performed, and both running times and cluster quality measures such as sensitivity and specificity are reported. Website administrators can then select the parameters that achieve the most desirable combination of clustering quality and running time for the labeled data, and apply these parameters to similar but unlabeled datasets to form high-quality user models that permit improved evaluation of website effectiveness.

INDEX WORDS: web site effectiveness, web usage mining, user profiling using web logs

WEBANALYZER: A TOOL FOR EFFECTIVE WEB USER BEHAVIOR MODELING

by

NAVEED AHMED

B.E., Osmania University, Hyderabad, India

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2010

WEBANALYZER: A TOOL FOR EFFECTIVE WEB USER BEHAVIOR MODELING

by

NAVEED AHMED

Major Professor:	Eileen T. Kraemer
Committee:	John Miller Khaled Rasheed

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2010

DEDICATION

I would like to dedicate this thesis to my parents and my sister who have been supportive of me all through my life.

ACKNOWLEDGEMENTS

I would like to thank wholeheartedly Dr. Eileen T. Kraemer and Kelly N. Storm who have helped me with their expert guidance and patient feedback from the start of my research. I would also like to thank my committee members Dr. John Miller and Dr. Khaled Rasheed for taking their valuable time to review and offer suggestions for my thesis. I thank all the staff/faculty members of the department of computer science, who have helped in one way or the other.

In the end, I would like to thank my family for their moral support from time to time and without whose invaluable help and unwavering motivation, this thesis would not have been possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

Chapter 1

INTRODUCTION

1.1 Background

Since the invention of World Wide Web technology by Tim Berners-Lee at CERN in 1989 [1], the web has become a primary source of "information dissemination" by academia, business and government organizations [25]. Moreover, the web also serves as an e-commerce platform for many business sectors. Hence, the usability and effectiveness of these websites is important, as these websites are the primary revenue source for many companies. Site administrators make use of user models and usage data in evaluating and promoting web effectiveness and in redesigning and refining a website to best meet user needs [22]. Web effectiveness is defined as how well a "website meets its business goals as well as user goals" [48]. User models are analyzed when evaluating the impact of website changes on particular user groups. Redesigning of websites is necessary to make them more intuitive and user friendly, ultimately making them more effective for their end users [22]. As website designers build more complex websites using new and sophisticated technologies, we need a method to model user behavior accurately. The work described in this thesis addresses the problem of determining the characteristics of groups of web site users by creating a methodology to more accurately model user behavior. Hence our problem falls under the larger domain of making websites more effective.

Traditionally, website administrators make use of usability guidelines [2], user feedback and user testing in order to design and refine websites that meet user needs [3, 4]. Another

approach used by website administrators and usability researchers is the analysis of web server logs [5, 6, 7]. Web servers generate logs of records for each transaction they process. These transactions are known as HTTP (hyper text transfer protocol) transactions and they occur in response to user input (clicks). Each user click or page visit is recorded in the web log. A group of such page visits within a specified period of time is called a session. When we group together web sessions from similar users we can infer information about the user group. For example, users who are browsing a site to research a product may interact with the site in one way, while users who interact with the site to make a purchase may behave in another way. By placing all of the buyers in one group and the browsers in another group, we can analyze the user group behaviour and learn typical interaction patterns for buyers versus browsers, and perhaps modify the site design and content to better meet the target user's needs.

While the problem of modeling user behavior is well known in the area of data mining [11, 25], good methods for user group profiling, capturing the essential characteristics of the users in those groups, are less well-defined in the web effectiveness domain.. The clustering of user sessions typically involves using algorithms such as *k*-means clustering [8], hierarchical clustering [9], or other clustering algorithms [10] to come up with a good model of how the users interacted with the site. Our approach in this thesis is to analyze the weblogs of a dataset for which labeled data exists, extract parameters including the number of clusters and the distance measure to be used, and then make use of these parameters to label unlabeled datasets from that site or similar sites.

1.2 Approach

In this thesis, we have developed *Webanalyzer* software that takes as its input 1) a sessionized web server log (split into collections of records corresponding to individual user sessions) 2) the maximum number of clusters to be considered, and a 3) set of distance measures. The software then performs *k*-means clustering [8] of the sessions for each of the distance measures, producing groups of similar user sessions. Also, given a labeled set of web sessions that we believe has similar properties to an unlabeled set that we wish to label (one for which we know the user group characteristics before hand), the tool calculates the sensitivity (proportion of actual group members found) and specificity (proportion of predicted members that are correct) [50] of each clustering result produced by the software for each distance measure and identifies the clustering result having the highest weighted average of sensitivity and specificity for each distance measure. We also keep track of the time required to execute the clustering algorithm for each distance measure. In this way, users can select the best parameters for the given data set taking into consideration execution time, sensitivity , specificity and the weighted average of sensitivity and specificity.

To evaluate our tool and this methodology for clustering users, we applied this approach to data collected from the website www.alumni.uga.edu. This particular site was chosen for its well defined site structure and a user base whose membership can be determined using auxiliary data.

1.3 Results Overview

Using this methodology, we were able to select a distance measure that provides the best weighted average of sensitivity and specificity (Euclidean distance measure followed by Manhattan distance measure). However, we also found that this distance measure did not result

in the shortest running time. In the case that running time is an important parameter to consider, our methodology showed that the Manhattan distance measure provided good results in terms of sensitivity and specificity, but with the shortest running time among all the distance measures.

Using our methodology, site administrators are able to make an informed choice about the distance measures when clustering unlabeled data from this site or similar sites. In general, site administrators can employ our methodology to help select parameters when clustering for user behavior from weblog data.

1.4 Outline

Chapter 2 provides a broad overview of the different approaches that have been used to address the problem of profiling web users in the domain of web user mining. Additionally, this chapter also gives a brief snapshot of the latest research being performed in the area of clustering website users. Chapter 3 gives a brief overview of how to use the software we developed. Chapter 4 discusses the various distance measures used. Chapter 5 presents the experiments we have conducted and a general discussion on the results obtained. Chapter 6 contains the conclusions of our work and provides the motivation for future work.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Introduction

The World Wide Web has become a preferred medium for "information dissemination" [25] and business for many companies. Therefore, it is very important that website users are able to extract the "relevant information" from the website [25], that the websites are effective at supporting users in performing selected tasks [22]. User modeling is an important element in evaluating and promoting website effectiveness [51]. Hence, good user modeling (profiling) relates to the larger research area of effective website design and management, and also to the issue of mass customization [25]. One major approach to solve the above problem of making effective websites is to make use of web mining techniques. We concern ourselves with the approach of web mining in this thesis. We briefly discuss the area of web mining and in particular web usage mining. We also present background work on clustering, which is our method for grouping users. We present prior work used in solving our particular problem of profiling users based on web logs.

2.2 Overview of Web Mining

Web mining is defined as "discovery and information extraction from web documents and services by making use of data mining techniques" [25]. Hence the area of web mining is a very broad research area which comprises of three sub areas: 1) Web content mining, 2) Web structure mining and 3) Web usage mining [25].

2.2.1. Web content mining

Web content mining is defined as "discovery of useful information from any web content" [25]. Web content can be any "text, documents, images, videos and other links" [25]. Web content data may consist of "unstructured data (free text), semi-structured data (Hyper Text Markup Language [HTML] pages) or structured data (dynamically generated HTML pages)" [25].

2.2.2 Web structure mining

Web structure mining is defined as finding the "underlying structure of the website by making use of the hyperlinks" [25]. Hyperlinks are elements of a web page that refer to other sites from within a webpage. For example a web address such as http://www.google.com is an example of a hyperlink. The website structure is used for "web page categorization" and may be used to determine the level of "similarity or the dissimilarity" between different websites [25]. Algorithms such as PageRank [42] and HITS [40] are used in analyzing the web site structure [25]. PageRank [42] is the well known algorithm used by Google to search the Internet.

2.2.3. Web usage mining

Web usage mining is defined as the study of "user interaction with the web" [25]. It can also be understood as "usage patterns discovered from web data by using data mining techniques" [11]. To obtain usage patterns of users of our site of interest, we must have an appropriate data set. For that purpose, we require web logs of the website. A typical example of web log record is as follows [44]:

192.168.10.11 -- frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326.

The fields of the above HTTP log record are as follows [44]:

- Host -- client IP address - (192.168.10.11)
- User id – person requesting the web document – (frank)

- Request time -- the time at which request was made. – (10/Oct /2000 13:55:36 -700)
- Request Type – the type of client request made - (GET)
- Resource name – name of the resource which was requested - (/apache_pb.gif)
- Protocol – the protocol used for the request - (HTTP/1.0)
- Status code – status code of the web document - (200)
- Size -- size of the web document - (2236)

Web log data may be found in server logs, client-side logs and in proxy server logs if a proxy server is involved [11]. Server log files contain records of accesses by multiple users [11]. The records are interleaved, according to the order in which the accesses occurred. However, the log may not serve as a complete record of user behavior because some user requests may be fulfilled through client–side caches of responses previously received from the server. Information about cached web pages is not recorded in the server web logs [11]. Caching is the process of storing previously visited pages on the client side in order to decrease the website access time in case the same pages are visited by the user. Therefore, the server logs may not represent the complete behavior of the user on the website. The problem of analysis in the case of caching can be overcome by making use of client side logs [11].

Logs related to the user are also generated on the client side. Javascript and Java applets are the preferred modes of data collection on the client side [11]. If caching is used, we may not record all the web page accesses by the user in the server side log. Hence, client side logs are helpful in understanding the complete behavior of the user since they will have records for web pages that are served only from the cached pages. However, client-side log collection mechanisms require explicit user permissions [43].

Many websites make use of proxy level servers in order to speed up web site access and also “reduce network load” [43]. They can be visualized as serving as a kind of middle ground “between the client and the server” by caching web pages [43]. So, proxy level server logs will contain information about “multiple users accessing single / multiple websites” [43]. Hence they may provide valuable information about the behavior of multiple users using a proxy server to access a website [43]. Even though it can provide valuable information about multiple users, one faces added difficulty in accessing web proxy server logs [43].

For our experiments we have made use of web server log data. After an overview of the various data sources, we analyze the various tasks involved in web usage mining. Web usage mining can be broken down into 3 major steps: 1) Pre-processing Data 2) Pattern Discovery, and 3) Pattern Analysis [6].

2.2.3.1 Pre-processing Data

Pre-processing refers to all the steps taken to process the log data into information about the users and sessions [25]. Pre-processing is an essential step in web usage mining [43]. The web profiles which are obtained at the end of the web usage mining process will be helpful only if we extract useful information related to the user behavior on the web site from the web logs [43]. Pre processing consists of two sub tasks [43]: 1) Data Cleaning and 2) Sessionization.

1) Data Cleaning

A web server log contains some extraneous information that is not required for our task of web usage mining. There are some actions which are performed by the user on the site which are not required when trying to model the user behavior. For example, error codes, data downloads and image file accesses can be ignored from the log file since they do not give us any useful information about the user behavior [43]. Also, we see many non-human agents that make web requests like search crawlers, bots and spiders [43]. Crawlers and spiders are used by search

engines (for example www.google.com) to index information about the website. A web log containing these non-human agent records will not be useful to us, because these website accesses are not performed by a human user. Hence they do not contribute in our attempt to model the user behavior. Therefore these records need to be removed from the web logs by making use of IP addresses of "known crawlers" [43].

2) Sessionization

The next important step in pre-processing relates to the grouping of the web server requests into sessions. A *session* is defined as a sequence of web page visits by a unique user within a "defined period of time" [43]. In order to understand how the users have interacted with the website, we need to group the web log records into logical units called sessions. Normally, a session starts with the first access by a user and includes each request by the user until the last access by a user [49]. The "precise end point of a user session" might be difficult to pinpoint since a user can keep a browser open without performing any activity [49]. Hence, a session timeout parameter is specified on most servers [49]. A session will be closed after detecting no activity from the user during this time period [49]. There are a few issues with correctly sessionizing web logs. A major challenge while trying to sessionize logs is to uniquely identify a user [43]. Normally, ISPs (Internet service providers) deploy web servers along with proxy servers for the purpose of faster website accesses [43]. In such a setup, only a single IP address will be recorded even though there may be different users accessing the website [43]. Also, in some cases, each request of a website user might have a different IP address because the ISPs might periodically assign a new IP address from a pool of addresses for each request [43]. Hence a single user session will have different IP addresses which might be interpreted incorrectly as different users [43]. It is also possible that there are different users accessing the site from the

same IP address or a user can visit the website from different computers or IP addresses [11]. Identifying a unique user session will be difficult because of the above issues. Client side tracking through the use of cookies is one of the ways of overcoming the above problem [43]. Another way to uniquely identify user sessions is by making "use of combination IP address, machine name, and browser agents "[43]. A combination of both the above methods might also be used.

2.2.3.2 Pattern discovery

After cleaning the data to include only the required information and sessionizing the web server logs, we proceed with the task of pattern discovery. Pattern discovery makes use of techniques from areas including "machine learning, artificial intelligence, data mining, pattern recognition and statistics" [11]. This task makes use of techniques such as statistical analysis, association rules, classification, sequential patterns, and clustering to come up with meaningful patterns [11]. A *cluster* is defined as a "grouping of data into similar objects" [28]. For our dataset, we employ clustering as we are trying to group users based on their behavior and hence clustering is an effective technique to produce high quality user profiles [11]. We will briefly describe the steps in clustering. The task of clustering a dataset can be broken down into 5 steps as follows [37]:

1. *Data representation*

Data representation "refers to choosing the format of the data, number of clusters, classes and features" that will become an input to the clustering algorithm [37]. For our problem, our data is represented as a bag of URLs (uniform resource locator) [24]. Each session is represented by 1 (present) or 0 (absent) for each of the URLs that can be accessed at that site. An example of our session data in our input file is as follows:

URL	1	2	3	4	5
Session_1	0.0	1.0	1.0	0.0	1.0

This indicates that during session 1, the user visited URLs 2, 3 and 5. The number of URLs considered is 5.

2. *Selection of an appropriate distance measure*

Differences between sessions are quantified using distance measures such as Pearson correlation coefficient [15], Pearson correlation coefficient (absolute value) [15], Uncentered correlation [13], Uncentered correlation (absolute value) [13], Spearman's rank correlation [16], Euclidean distance [17] and Manhattan distance [18]. The distance measures are employed during the clustering process.

3. *Clustering or grouping*

Algorithms like *k*-means [8] and *k*-medoids [19] are used for grouping data. For our task we have made use of the *k*-means algorithm, pseudo code [8, 13] for which can be seen in Figure 1.The *k*-means algorithm is one of the most popular algorithms used to cluster data [8]. The *k*-means [8] algorithm uses the squared error criterion first formulated by Lloyd's Algorithm [20]. One of the major issues with *k*-means is that it the final result is dependent on the initial cluster assignments, since we select the initial cluster points randomly [8]. Also one of the other problems is to know beforehand, the number of iterations *k*-means algorithm has to be run before it arrives at a reasonably optimal solution [8]. If one repeats the algorithm far below the number of optimal runs required, one may end up with sub optimal clusters [8]. Hence one may have to run the algorithm for quite a few times before one knows the optimal number of runs [8].

1. Start with a random number of sessions and choose the number of clusters to form.
2. Choose random sessions as the cluster centroids (number of clusters chosen must be equal to the random sessions chosen as centroids).
3. Select random sessions to cluster and calculate the distance between that session and the centroid session.
4. The session is then reassigned to the cluster to which it is nearest.
5. Step 3, 4 are repeated for each session.
6. Steps 3 to 5 are repeated until we arrive at stable clusters (there is negligible reassignment of sessions).

Figure 1 - Pseudo code for *k*-means algorithm

4. Data abstraction

Data abstraction step is to abstract a concise representation of data [37]. The final clusters that are formed can be represented by the cluster centroid. The centroid of the final clusters can be represented as follows.

URL	1	2	3	4	5
Centroid A	0.25	0.65	0.35	0.45	0.05
Centroid B	0.35	0.15	0.55	0.85	0.95

Here we observe that the two centroids are represented by values which are between 0 and 1. The fraction for each URL represents the frequency of that URL being present in

the sessions which are present in that cluster. Thus, just by plain observation we notice that centroid for cluster B contains a high probability of sessions with URL's 4 and 5.

5. *Output interpretation*

   The output from the clustering algorithm is taken and interpreted to give meaningful information [37]. We can look at the profiles of the individual clusters to come up with interesting results. By looking at the clustering result, we are able to identify the characteristics that make up that particular user group. For our experiment, we observed that users clustered in the two clusters had mostly similar page visits.

2.2.3.3 Pattern analysis

The final step in web usage mining after discovering the patterns (clusters) or models is to analyze them in order to extract meaningful information from them [11 ,43]. Visualization tools like Web Viz tool developed by Pitkow [45] will further analyze the patterns to make sense of them. Also, On-Line Analytical Processing (OLAP) [45] is another tool useful for analyzing patterns. We do not perform any visual pattern analysis for our experiments in this thesis. However, our results are used as an input to the Ajalytics tool [31] to generate customer behavior model graphs (CBMG) which are analyzed for coming up with interesting insights into the behavior of the users clustered.

2.3 History

The problem of characterizing web site users falls under the category of web usage mining categorization and specifically under web user modeling. The problem of characterizing web site users by mining web logs was worked on by a few researchers whose approaches are discussed in this section.

An approach for profiling web site users is to make use of user access patterns. The sequence of accesses by a user in a web log is known as access pattern [23]. Draier [27] used

sequences of user actions for profiling web users by considering parameters like "number of events, session duration, time spent on each page, number of sequences per session," etc [12]. Li [27] approached the issue by using a web access pattern [WAP] tree to store these long user sequences and a WAP–mine, an algorithm to mine the web access patterns from weblogs [12]. Analysis of web logs can be done by making use of traversal patterns [34]. Yang et al. [34] have proposed a sequential mining algorithm (LAPIN_WEB) to analyze the web logs and the results are interpreted using a visualization tool.

More recently, the approach of extracting user profiles by mining web access logs and coming up with session profiles [5, 24, 47] is being employed [12]. In [5], the user sessions were clustered using a new distance measure known as a "dissimilarity measure" which "encapsulates both the "URLs as well as the site structure". The clusters formed were analyzed by generating a "session profile vector" which encapsulated the "typical session in each cluster". Similarly Nasraoui et al. have used a CARD (Competitive Agglomeration of Relational Data) algorithm [24] to cluster relational data while using non-Euclidean distance measures. In [47], Nasraoui has described a Hierarchical Unsupervised Niche Clustering (H-UNC) algorithm to cluster web sessions. This algorithm employs non-metric distance measures to produce different session profiles and discovers contextual associations between different URL addresses [47]. As the software was not available to use outside the Nasraoui lab, we wanted to have a tool which could give us a range of clustering parameters to choose from while having a metric for evaluating the quality and performance of the clustering result. This formed the main basis for the development of our tool.

Chapter 3

HIGH LEVEL APPROACH

3.1 Introduction

In this chapter, we provide an overview of how our clustering tool, *Webanalyzer*, can be used by web site administrators and web usage mining researchers. Web site administrators may wish to evaluate the impact of changes to a site on the different groups of site users. *Webanalyzer* is designed to support site administrators in identifying these user groups and their members. This chapter provides a description of how site administrators can use *Webanalyzer* to select good parameters based on a labeled dataset (one for which we know the cluster profile already) and then use those parameters to identify groups and group membership for a similar unlabeled dataset in a way that is efficient and employs the desired balance between sensitivity and specificity.

As discussed earlier, user interactions with a web site are captured in the web server logs. After the web server logs are cleaned, they are then sessionized using Sessionizer [12]. Sessionizer processes the web logs into logical user sessions. The output of Sessionizer [12] is a database dump file. This database dump file contains information such as the web URL, request time, the user IP address accessed, and the entry and the exit times of the users, in database tables. Sessions having fewer than 3 requests are removed, in order to consider only significant user interactions. *Webanalyzer* takes the session information from the database and prepares an input file for the Cluster 3.0 software [53], which assigns the sessions to clusters.

As shown in Figure 2, the input file to the Cluster 3.0 software [53] consists of a matrix for all the sessions and URL's. For any given session we have a vector of URLs such as:

URL	1	2	3	4	5
Session_1	1.0	0.0	0.0	1.0	0.0

The value of 1.0 signifies a visit and the value of 0.0 means "not visited". The above sample vector gives us the information that in session_1, the user visited URLs 1 and 4 from a group of 5 URLs. Therefore, the whole input file contains a matrix of all possible sessions considered vs. the URLs for the particular website considered.

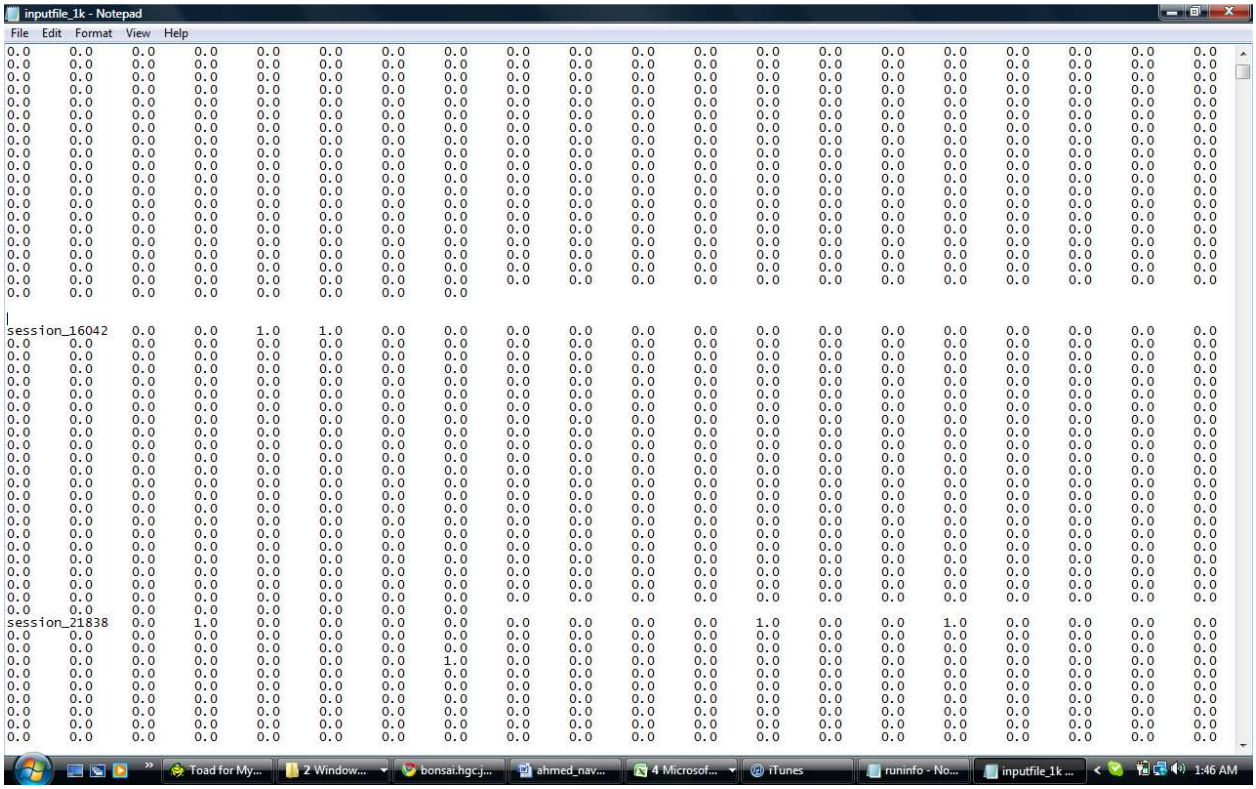


Figure 2- Input to the Cluster 3.0 file. Each row represents a session and each column represents a URL. This file contains 446 columns and 10000 rows.

.

Webanalyzer invokes the Cluster 3.0 [53] software iteratively, once for each combination of distance measure and number of clusters considered. For each combination, Cluster 3.0 [53] produces an output file, which lists the cluster assignment for each session, as show in Figure 2.

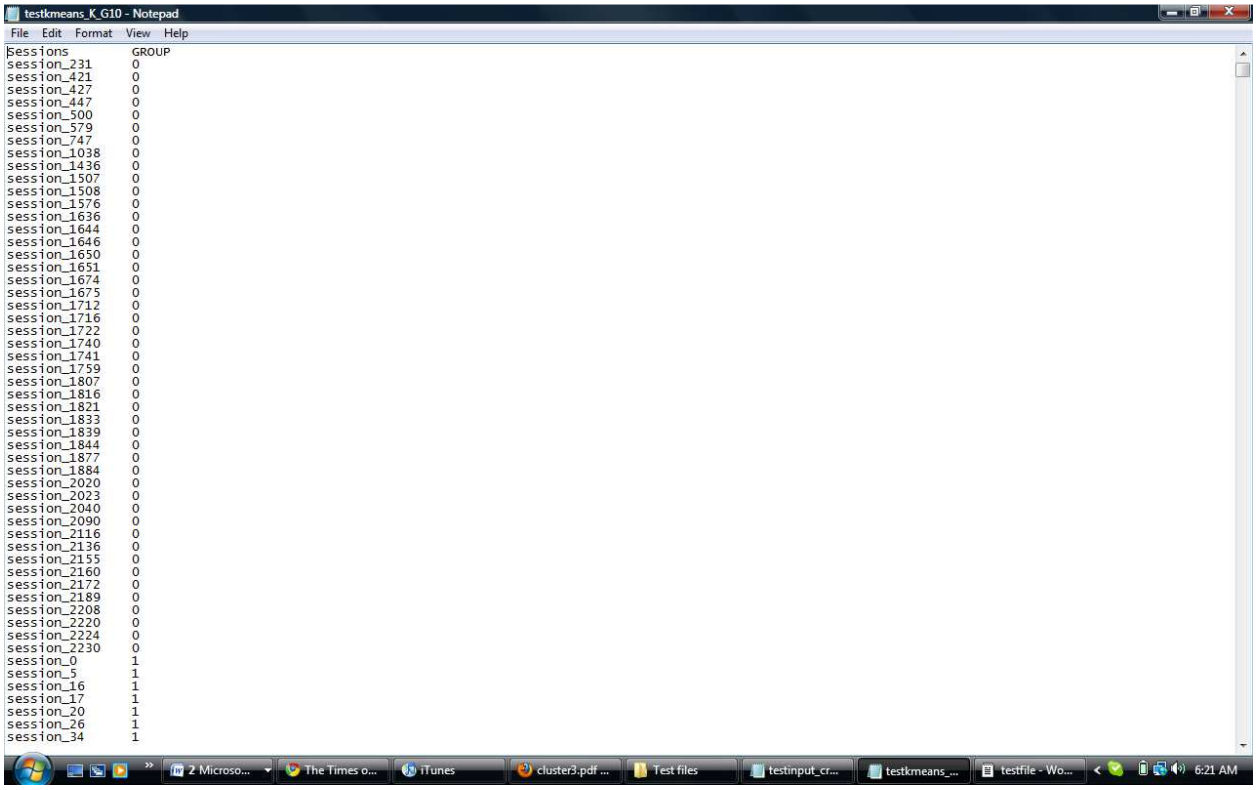


Figure 3 - Output file from Cluster 3.0 software. The file contains the cluster assignment (0, 1, etc.) for each of the sessions given in the input file.

The cluster assignments that are obtained from the Cluster 3.0 [53] output are then processed by *Webanalyzer* to form an output as in Figure 4 which lists the id's of the sessions assigned to each cluster. Then this output, along with the labeled data set is analyzed by our tool *Webanalyzer*, to evaluate the "goodness" of the clustering in terms of how well formed the clusters are.

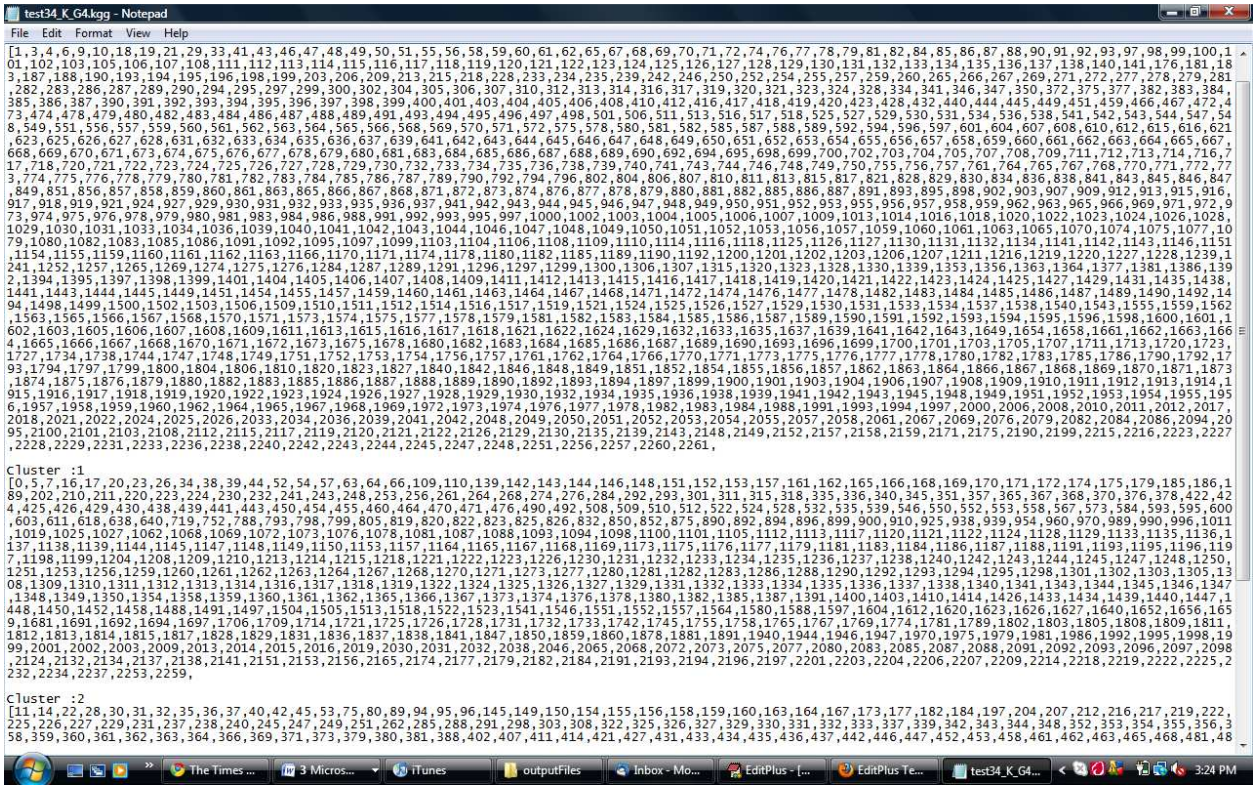


Figure 4 -Output Data processed by out tool *Webanalyzer*. Each of the individual clusters with its constituent sessions is present in the above file

.

As seen in Figure 5 of Chapter 3, the administrator provides a set of sessionized web logs and specifies a range of the number of clusters. For each number of clusters, clustering is performed using each of 7 different distance measures, Running time is recorded for each of the clustering results, as are the metrics for "goodness" of clustering. The user can then see which parameters (number of clusters, distance measures) result in the most desirable clustering/run-time combination.

3.1 Detailed Breakdown of Tasks

The *Webanalyzer* tool can be used for achieving two major tasks as follows:

1) Training on a labeled dataset - Selecting a distance measure and number of clusters.

2) Clustering an unlabelled dataset- Using the distance measure and number of clusters from above.

***Task I: Selecting an optimum distance measure and number of clusters by training on a* labeled dataset**

Inputs: 1) Sessionized logs

2) Annotated file indicating "user type" for sessions

Outputs: distance measure, number of clusters

Steps:

1. Pre-process sessions (sessions having only a significant number of URL's are considered).

2. Generate a run for each combination of distance measures and desired number of clusters.

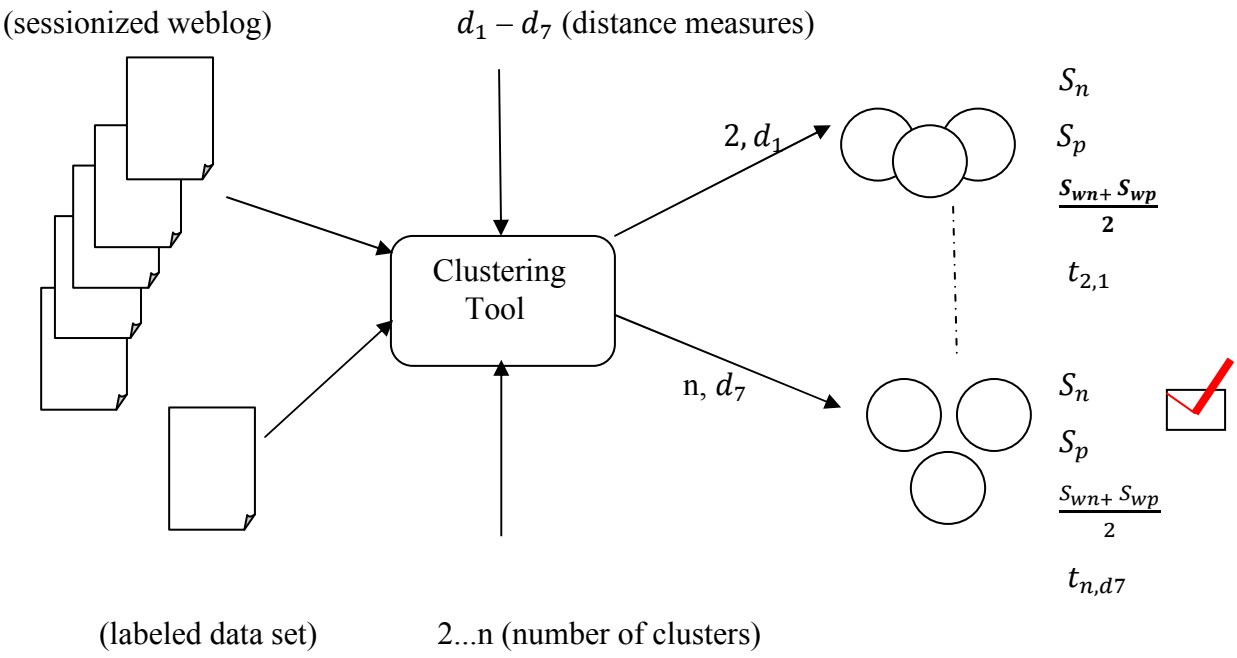
3. For each run

   a) Run cluster algorithm with selected number of clusters and distance measure.

b) Assess cluster membership. This means we evaluate all the possible permutations [52] of the label assignments (1, 2, 3) possible for each clustering result. For example, suppose there are 3 clusters and 3 assignments (novice, intermediate and advanced). We evaluate each permutation (1= novice, 2= intermediate , 3 = advanced) , (1= intermediate , 2 = novice, 3 = advanced) , ...etc.

c) Calculate Sensitivity (S_n) and Specificity (S_p) and the weighted averages for each of the above clustering results.

4. Report the best distance measure and number of clusters from the above calculated statistics, including the running time for each result.



d_n= Distance Measure

S_n= Sensitivity

S_p= Specificity

$\frac{S_{wn+}\, S_{wp}}{2}$ =Average of Sensitivity and Specificity

$t_{2,1}$ = running time taken for that clustering result.

Figure 5- Flow Diagram of Task I

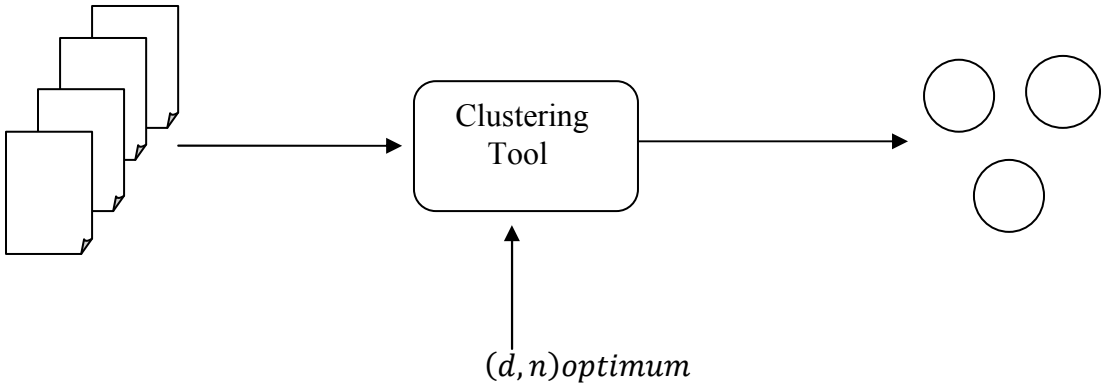
Task II: Cluster and label sessions

Inputs: An unlabelled data set of sessions (sessionized log), distance measure and desired number of clusters selected using Task I.

Output: labeled sessions

Steps:

1. Preprocess sessions (Sessions having only a significant number of URLs are considered).
2. Generate clusters using the selected distance measure and number of clusters.
3. Site administrator looks at aggregate properties of each cluster and assigns a label to each cluster.



(Sessionized web logs) (Optimal clustering result)

Figure 6-Flow Diagram of Task II

In the above tasks, we have discussed calculating sensitivity and specificity. Let us see an example of this. As seen in Figure 5, statistics for sensitivity, specificity and an average of sensitivity and specificity are calculated for each clustering result. Each clustering result is a way of dividing up the sessions into clusters, a mapping from session ids to cluster ids. We wish to evaluate the quality of these clustering results on known data and then use the "good parameters" on other similar data sets for which the cluster assignment is not known a priori.

The *sensitivity* of a cluster is defined as the actual number of sessions of a particular assignment type that have been assigned to the cluster as a proportion of the total number of sessions of that type that should have been assigned to that cluster [50]. *Specificity* for a cluster is defined as the actual number of sessions that are correctly assigned to a cluster as a proportion of the total number of sessions assigned in that particular cluster [50].

Suppose we have 75 sessions, 25 of type A, 25 of type B, 25 of type C as seen in Figure 7.

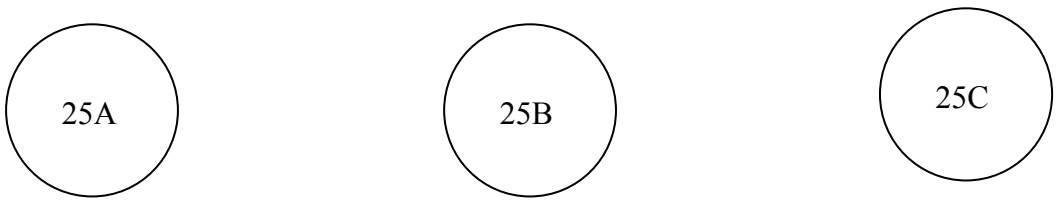


Figure 7-Example of Sensitivity and Specificity calculation

Imagine that, as a result of the run of a clustering algorithm, these 75 sessions are assigned to 3 clusters (0, 1, 2) as seen in Figure 8. Imagine further that we let Cluster 0 represent type A, cluster 1 represent type B, and Cluster 2 represent type C.

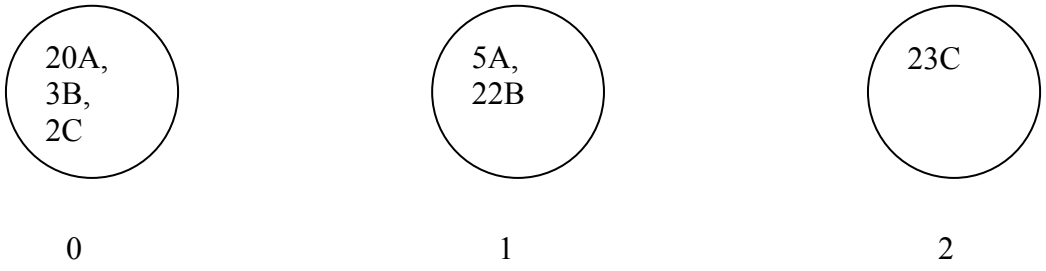


Figure 8-- An example of Sensitivity and Specificity calculation with cluster assignments

Then we calculate the Sensitivity and Specificity of the above cluster assignments as follows.

Cluster 0 (A)	Cluster 1 (B)	Cluster 2 (C)
$S_n = \frac{20}{25} = 0.8$	$S_n = \frac{22}{25} = 0.88$	$S_n = \frac{23}{25} = 0.92$
$S_p = \frac{20}{25} = 0.8$	$S_p = \frac{22}{27} = 0.81$	$S_p = \frac{23}{23} = 1.0$
$S_{avg} = (S_n + S_p)\ /2 = 0.8$	$S_{avg} = (S_n + S_p)\ /2 = 0.845$	$S_{avg} = (S_n + S_p)\ /2 = 0.96$

Also we can calculate a weighted average for each clustering result.

S_{wn} is the weighted average value of S_n and S_{wp} is the weighted average value of S_p and they are calculated as follows:

$$S_{wn} = \frac{[(25 * 0.8) + (25 * 0.88) + (25 * 0.92)]}{75} = 0.866$$

$$S_{wp} = \frac{[(25 * 0.8) + \ (27 * 0.81) + \ (23 * 1.00)]}{75} = 0.866$$

Chapter 4

AN INDEPTH VIEW

4.1 Distance Measures

Earlier, we talked about the different distance measures which are used as a parameter while running the Cluster 3.0 [53] software. We have used seven different distance measures which we are explained in detail below.

4.1.1 Pearson Correlation

The Pearson correlation coefficient [15] is a distance measure used to "evaluate dependence between two variables" [15]. It's defined as

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

Equation 4.1.1

where x is a vector of values x_i ..... x_n , $\bar{x}$ is the average and σ_x is the standard deviation of the values x_i ..... x_n .Similarly $\bar{y}$ is the average and σ_y is the standard deviation of the values y_i ..... y_n.

It can be used to calculate how "closely two vector of values x and y are co-related" [13]. The value of the coefficient of correlation, r, varies "between 1 and -1" [13]. An r value of 0 signifies that the series x and y are "completely uncorrelated" while r value of 1 tells us that they "are identical", and an r value of -1 tells us that these vectors are "perfect opposites" [13].

For example, if we have a set of four observations for variables x and y,

x	0	0	1	0
y	1	1	0	0

We calculate the Pearson correlation as follows:

$\bar{x} = 0.25$, $\sigma_x = 0.5$, $\bar{y} = 0.5$ and $\sigma_y = 0.577$, n =4

$$r = \frac{1}{4-1}\left[\left(\frac{0-0.25}{0.5}\right)\left(\frac{1-0.5}{0.577}\right) + \left(\frac{0-0.25}{0.5}\right)\left(\frac{1-0.5}{0.577}\right) + \left(\frac{1-0.25}{0.5}\right)\left(\frac{0-0.5}{0.577}\right) + \left(\frac{0-0.25}{0.5}\right)\left(\frac{0-0.5}{0.577}\right)\right]$$

$r =$ - 0.577

4.1.2 Uncentered correlation

Uncentered correlation [13] is defined as "the cosine of the angle of two n-dimensional vectors x and y, each representing a vector in n-dimensional space that passes through the origin" [13]. Essentially, the formula reduces to the above formulae 4.1.1 for Pearson correlation with mean 0 [13]. The difference between Uncentered correlation and Pearson correlation can be explained in the following way. Let us suppose there are "two vectors x and y with identical shape," but which are offset relative to each other by a fixed value [13]. They will not have an "Uncentered correlation of 1 but will have a standard Pearson correlation of 1" [13]. Uncentered Correlation makes use of the equations below for calculating the correlation coefficient [13].

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i}{\sigma_x{}^{(0)}}\right)\left(\frac{y_i}{\sigma_y{}^{(0)}}\right)$$

Equation 4.1.2.1

in which

$$\sigma_x{}^{(0)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i)^2}$$

Equation 4.1.2.2

$$\sigma_y{}^{(0)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i)^2}$$

Equation 4.1.2.3

where x is a vector of values x_i ..... x_n , $\sigma_x{}^{(0)}$ is the standard deviation of the values x_i ..... x_n assuming mean is 0 .Similarly $\sigma_y{}^{(0)}$ is the standard deviation of the values y_i ..... y_n assuming mean is 0.

Let's take the above example again for calculation of uncentered correlation.

x	0	0	1	0
y	1	1	0	0

$\sigma_x{}^{(0)} = 0.5$, $\sigma_y{}^{(0)} = 0.707$, and n =4

$$r = \frac{1}{4-1}\left[\left(\frac{0}{0.5}\right)\left(\frac{1}{0.707}\right) + \left(\frac{0}{0.5}\right)\left(\frac{1}{0.707}\right) + \left(\frac{1}{0.5}\right)\left(\frac{0}{0.707}\right) + \left(\frac{0}{0.5}\right)\left(\frac{0}{0.707}\right)\right]$$

$r = 0$

Uncentered correlation might be used in datasets where the data has been moved around the mean so that the average of values is 0 [15]. Cluster 3.0 [53] uses two other distance measures that are obtained by using only absolute values of the above distance measures [13]. Absolute distance measures may be used if we want two vectors to be considered as similar when they

have exactly opposite values [13]. The traditional correlation value will be -1 for the above values but the absolute values of the correlation will be 1 [13].

4.1.3 Spearman's Rank Correlation

Sometimes our data values may have outliers (values which are way outside of the normal range of values) which might skew our calculation of the correlation coefficients [13]. In such scenarios, we can use the Spearman's rank correlation [16]. The value "calculates the correlation between the ranks of the data values in the two vectors" [13]. Therefore, Spearman's correlation can be simply be thought of as the "Pearson's correlation coefficient between the two ranked variables" [16]. It's calculated as follows:

$$\rho = 1 - \frac{6 \sum {d_i}^2}{n(n^2 - 1)}$$

Equation 4.1.3

where $d_i = x_i - y_i$ and x_i, y_i are the respective ranks of the two variables of vectors x and y. Suppose we have x = {2.5, 3.5, 1.5, .001} and y = {20.5, 7.8, 100, 1000}. Then ranking the data respectively we have x = {3, 4, 2, 1} and y = {2, 1, 3, 4}

$$\rho = 1 - \left(\frac{6}{4^2}\right)\left(\frac{1^2 + 3^2 + 1^2 + 3^2}{4^2 - 1}\right)$$

$$\rho = 0.5$$

4.1.4 Euclidean distance

The Euclidean distance [17] is the most commonly used distance measure in everyday life. Euclidean distance between any two points is defined as the shortest distance between them [13]. If $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ are two points in Euclidean space, then the distance from x to y is defined as [17]:

$$d\,(x, y) = \sqrt{\sum_{i=1}^{n}((x_i - y_i)^2)}$$

Equation 4.1.4

This distance measure should be used when the data is normalized and both x and y values are present [13]. For example, using the same example as we have used earlier.

x	0	0	1	0
y	1	1	0	0

$$d\,(x, y) = \sqrt{[(0-1)^2 + (0-1)^2 + (1-0)^2 + (0-0)^2]}$$

$$d = 1.732$$

4.1.5 Manhattan distance or the City block distance

Manhattan distance originates from city blocks of Manhattan in New York City [18]. It is defined as the distance one travels between two different points when travelling along the edge of the city block [18]. The city block distance is the "sum of distances along each dimension" [13] and is mathematically defined as

$$d = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|$$

Equation 4.1.5

Taking the above vector example, we can calculate the Manhattan distance as follows

x	0	0	1	0
y	1	1	0	0

$$d = \frac{1}{4}[|0-1| + |0-1| + |1-0| + |0-0|]$$

$$d = 0.75$$

Chapter 5

EXPERIMENTS

5.1 Experiments

In order to evaluate our *WebAnalyzer* tool, we performed a case study using the Apache logs of www.alumni.uga.edu (an External Affairs website at the University of Georgia). We performed a case study using the Apache server logs for the site. The logs cover the May-June 2010 time period that included over 20,000 sessions. We then proceeded to manually annotate these clusters for the purpose of our program, i.e. we clustered sessions from these logs to obtain "predicted labels". Using a special purpose program that relies on the labeling of sessions with state and city parameters, we annotated each session as coming from 1) Inside Georgia (GA) versus Outside Georgia (NGA) and 2) Inside Athens (ATH) versus Outside Athens (NATH). We also labeled each session based on additional external information in the session (location information) in order to obtain "actual" labels. In practice, website administrators might want to look at other characteristics and we are able to cluster based on those characteristics. A good example of such a characteristic is browsers versus buyers or recent alumni versus former alumni. Moreover, as described in Storm's thesis [31], the web administrators could provide filters to select such characteristics.

We then evaluated the performance of the *k*-means clustering algorithm and distance measures using quality measures such as sensitivity and specificity and their weighted average. We looked at 7 different measures: Uncentered correlation, Pearson correlation, Uncentered correlation (absolute value), Pearson correlation (absolute value), Spearman's rank correlation,

Euclidean distance and Manhattan distance. We ran the sessions for varying size subsets of the total sessions. We applied a filter to remove sessions containing fewer than 2 accesses. We then randomly selected 1000, 2000, 4000, 8000 and 10000 sessions to create input files of corresponding size. These input files were then clustered using the Cluster 3.0 [53] software. We compared the cluster assignments that resulted with those obtained by direct labeling (actual), using quality measures such as sensitivity and specificity. We also recorded the running time for each clustering result. We then selected the best clustering methodology based on the highest weighted average of sensitivity and specificity. Website administrators may want to take into account the running time as well as the weighted averages when considering the best clustering methodology for their dataset. The choice of distance measure may also be affected by the size of their datasets.

5.2 Results

The following are the results for Inside Georgia (GA) versus Outside GA (NGA) categorization.

a) Running Time (sec) vs. Number of Sessions (#): In the graph (Figure 8), we have plotted the running time against the number of sessions for each particular distance measure. We observe from the graph (Figure 9) that Spearman's Rank correlation method takes the longest time to cluster sessions. Since the Spearman's Rank correlation distance measure takes a very long time when compared to other distance measures, it has been removed from the graph of Figure 9 to give a clearer picture about the other distance measures. Of the remaining distance measures, we see from Figure 11 that the Manhattan distance measure takes the least amount of time to cluster. We observe that the website administrators may use this information in

selecting the best clustering algorithm if they have the running time as one of their criteria.

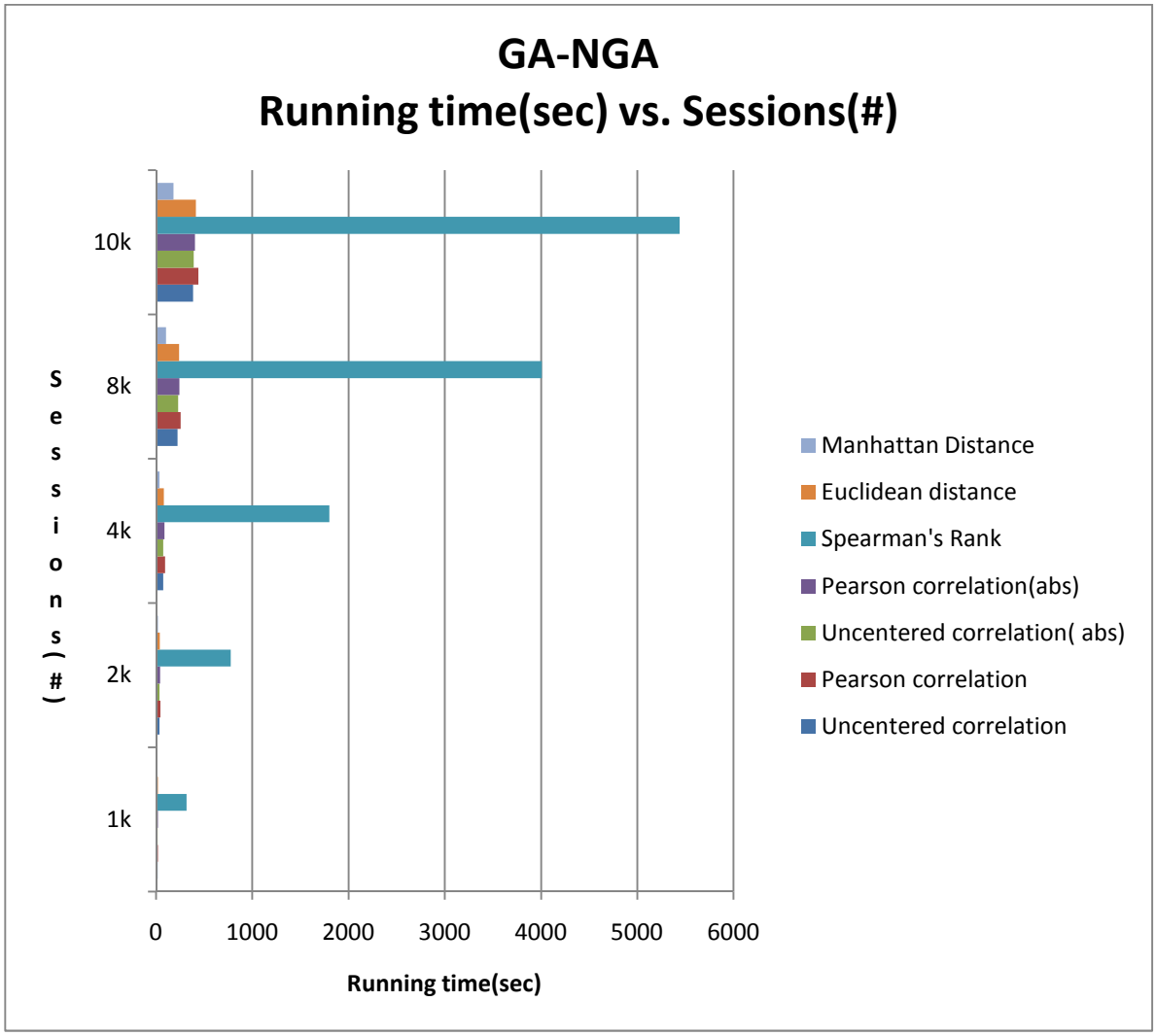


Figure 9-GA-NGA, Running time (sec) vs. Sessions (#) (all distance measures)

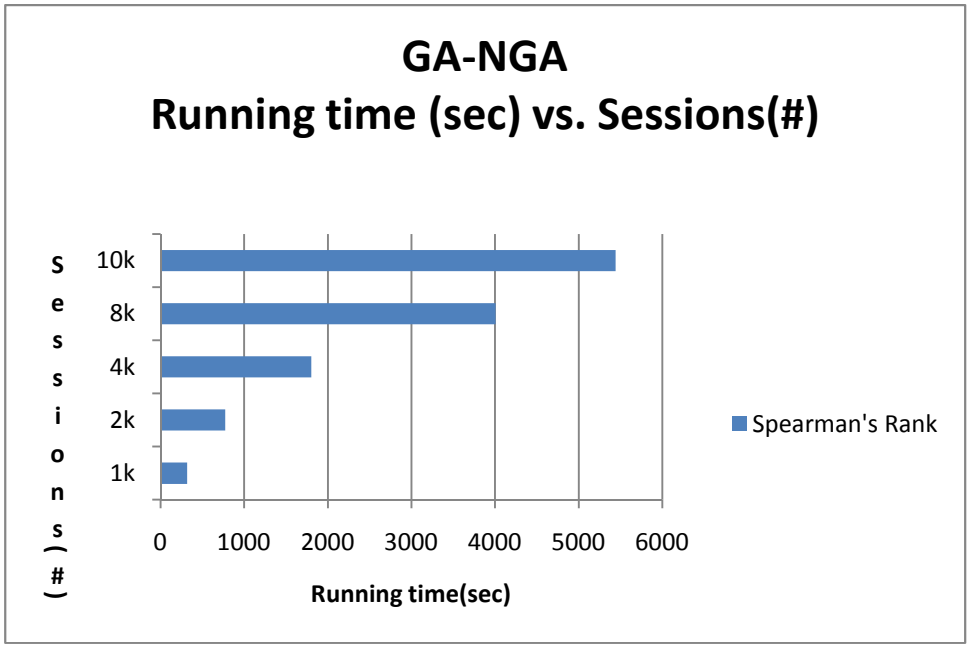


Figure 10 - GA –NGA, Running time (sec) vs. Sessions (#) (Spearman's Rank)

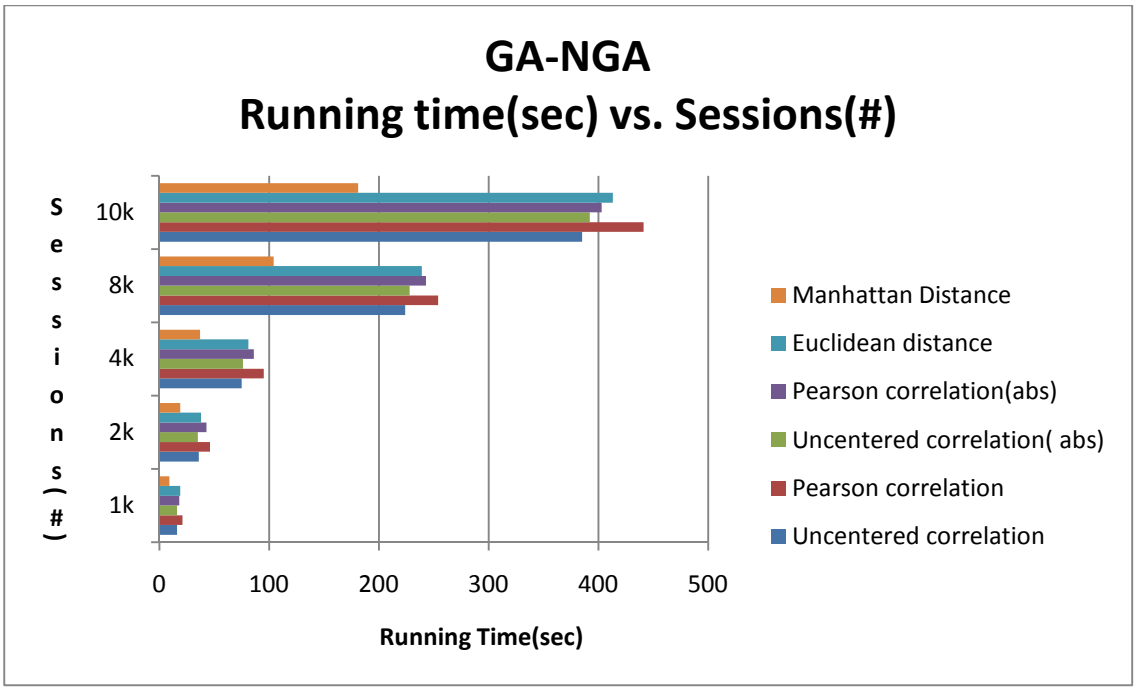


Figure 11- GA –NGA, Running Time (sec) vs. Sessions (#) (without Spearman's Rank)

b) Sensitivity (S_n) vs. Sessions (#): Next we consider sensitivity versus number of sessions graph (Figure 12). We observe that, on average, Euclidean distance measure has the highest sensitivity, while Uncentered correlation and the Manhattan distance also performing well on sensitivity.

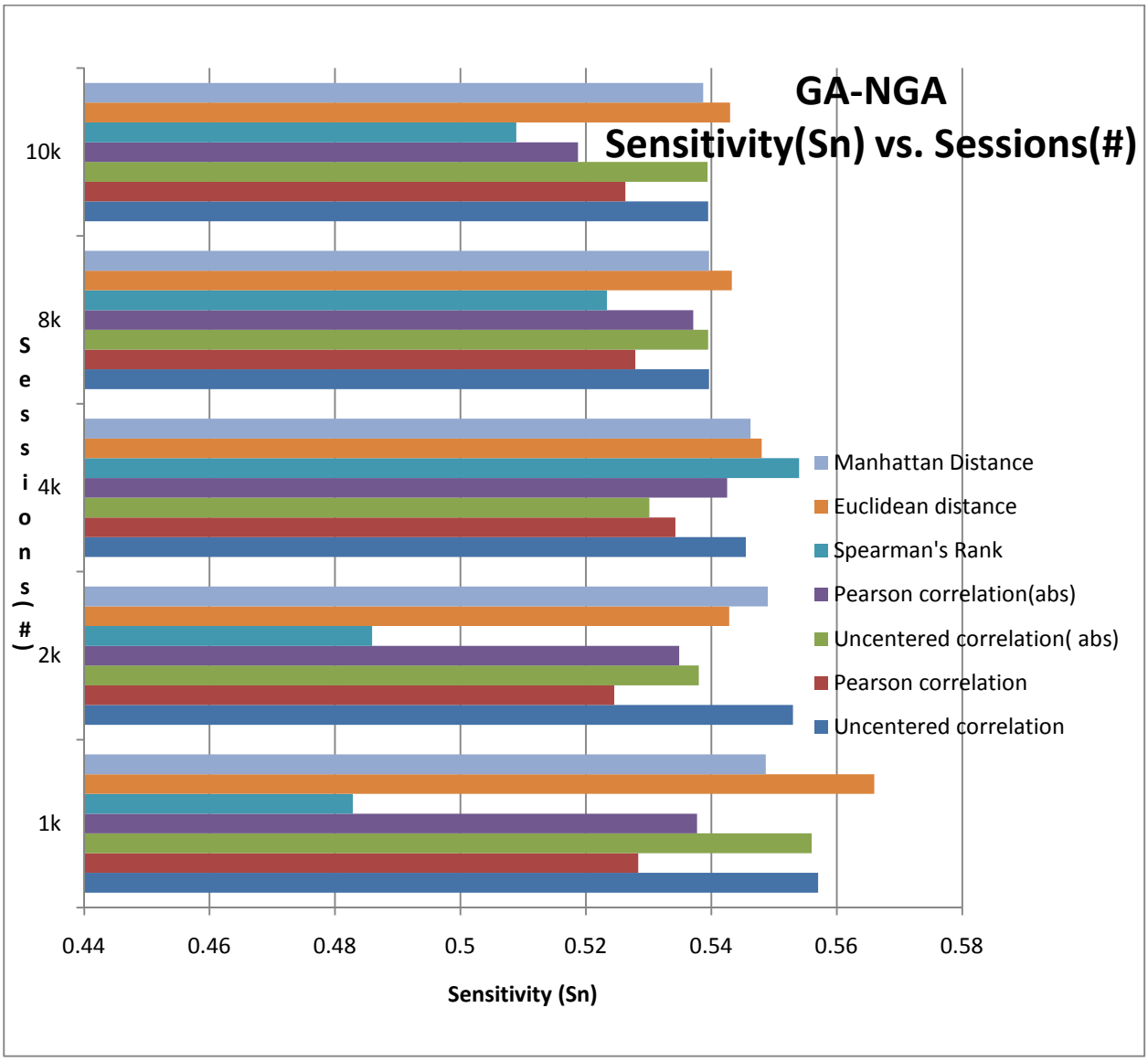


Figure 12- GA-NGA, Sensitivity (S_n) vs. Sessions (#)

c) Specificity (S_p) vs. Sessions (#): The graph in Figure 13 plots the specificity values against the number of sessions for each distance measure. For this dataset, Euclidean distance measure gives us the best specificity values on an average, but Uncentered correlation, Uncentered correlation (absolute) and Manhattan distance measures also give us higher values of specificity.

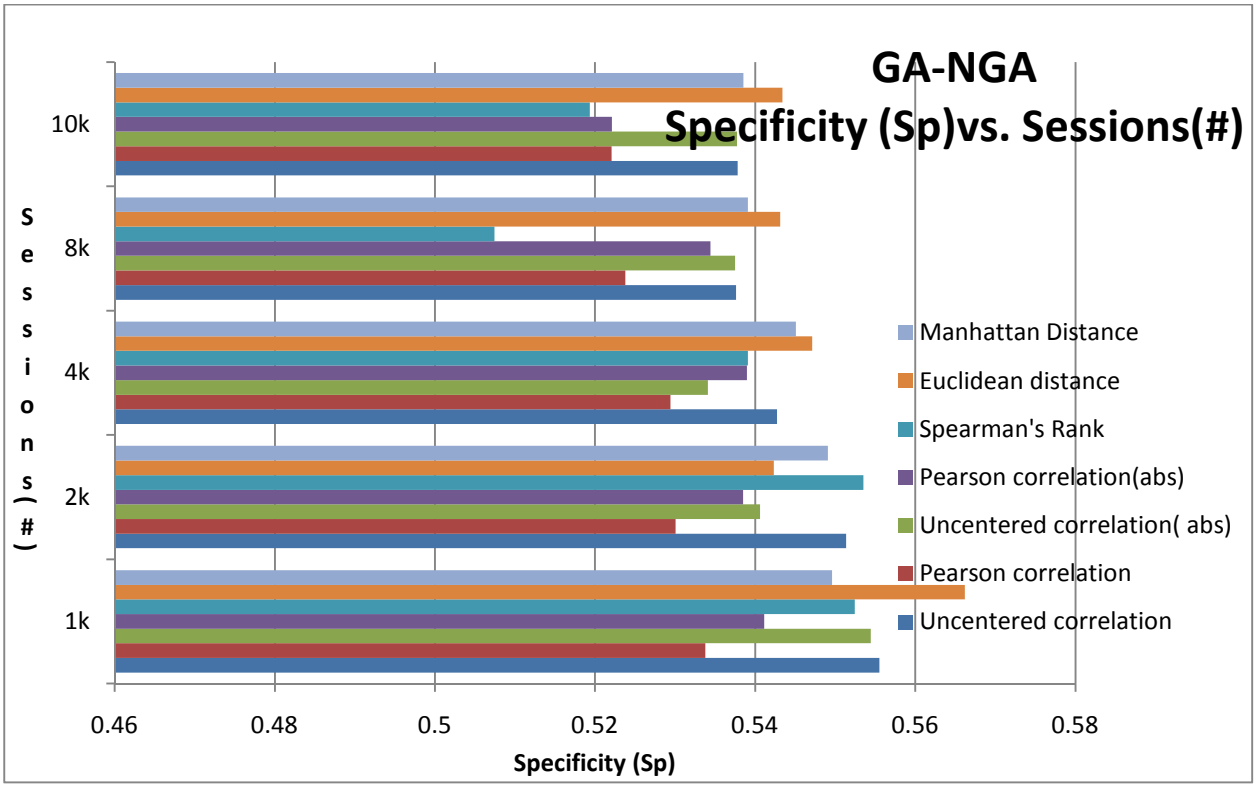


Figure 13 - GA-NGA Specificity (S_p) vs. Sessions (#)

d) Weighted Average of specificity and sensitivity (S_n, S_p) vs. Sessions (#): Instead of just considering the specificity or sensitivity values in isolation, a website administrator may want to consider the average of these values in order to select the best distance measure. In the graph shown in Figure 14, Euclidean and Manhattan distance and Uncentered correlation distance measure give us the best values on, an average. If we consider higher number of sessions (more than 8000 sessions), Uncentered correlation (absolute) distance measure also give us higher values.

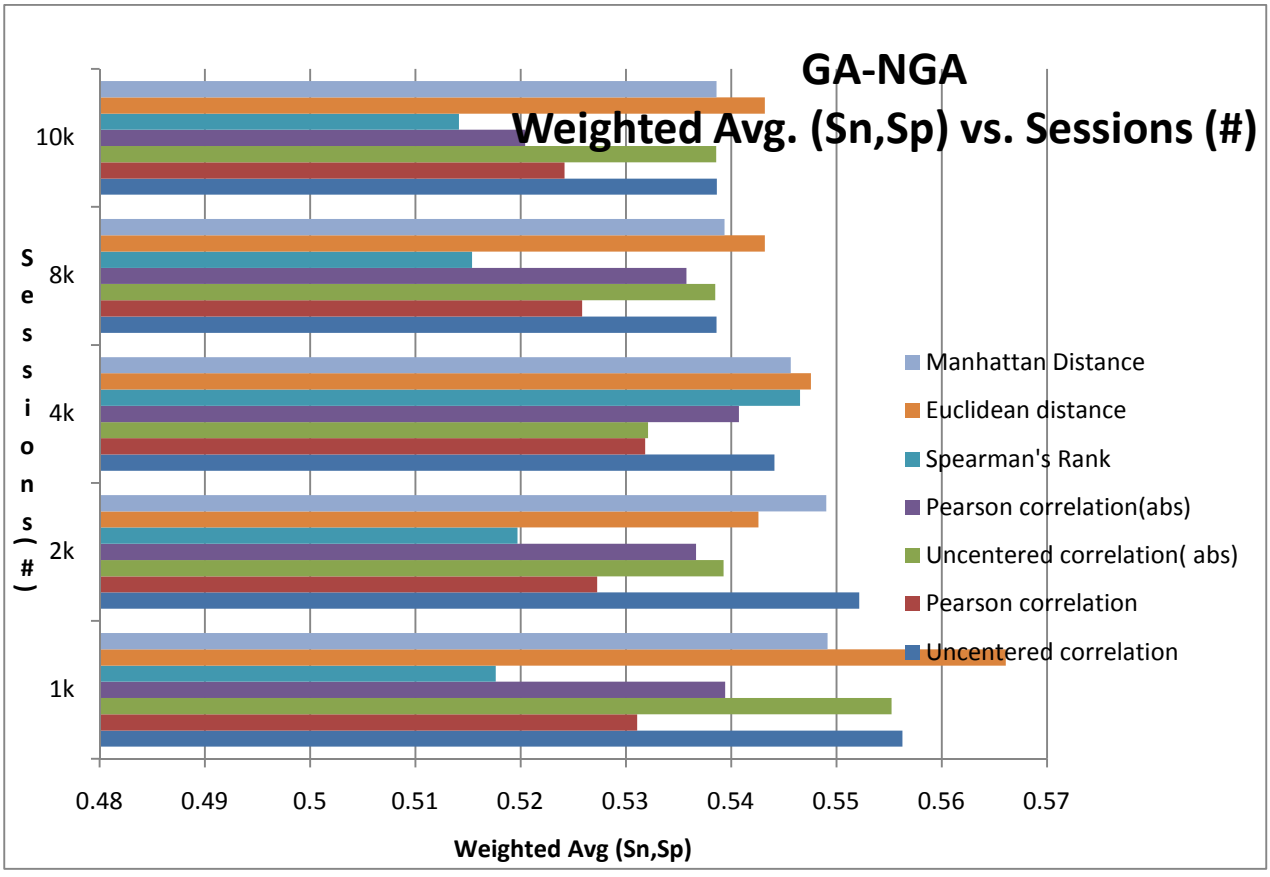


Figure 14 - GA-NGA, Weighted Average (S_n, S_p) vs. Sessions (#)

The above graphs were for the category Inside Georgia (GA) versus Outside GA (NGA).

If the site administrators were interested in an Inside Athens (ATH) versus Outside Athens (NATH) categorization, we would have the following results:

a) Running time (sec) vs. Sessions (#): In the graph in Figure 15, we observe that Spearman's rank correlation takes the longest time to cluster the sessions as we had observed earlier in the earlier categorization.

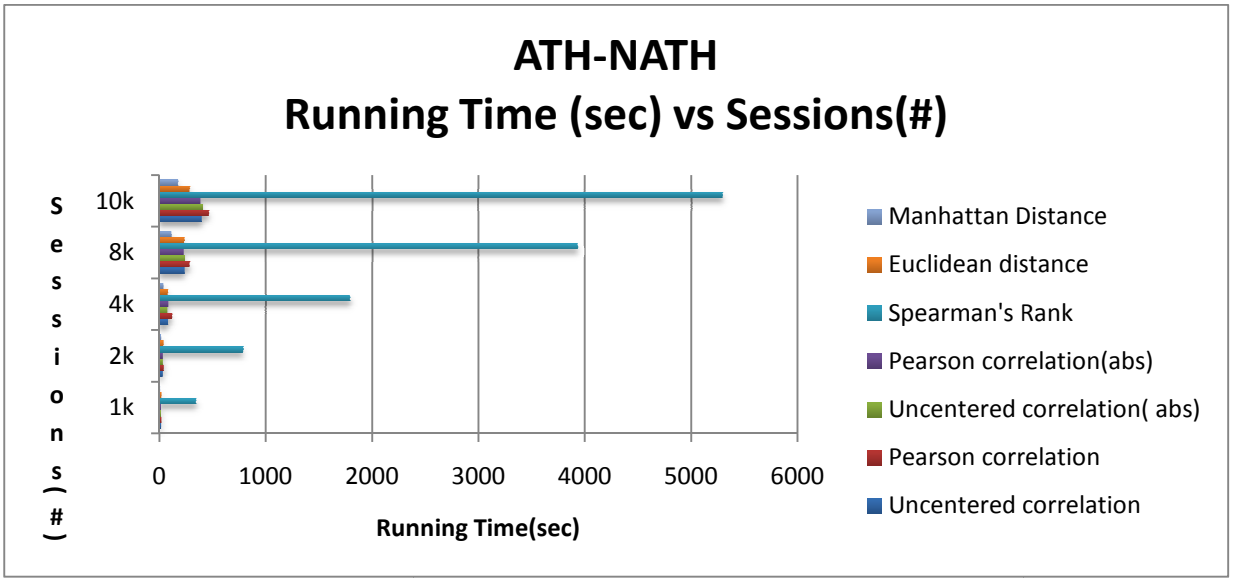


Figure 15- ATH-NATH, Running Time (sec) vs. Sessions (#) (all distance measures)

Taking out the Spearman's correlation from the Figure 14 and observing the other distance measures, we get the graph in Figure 16. We observe that in Figure 16, that the Manhattan distance measures take the least amount of time on average for this particular dataset.

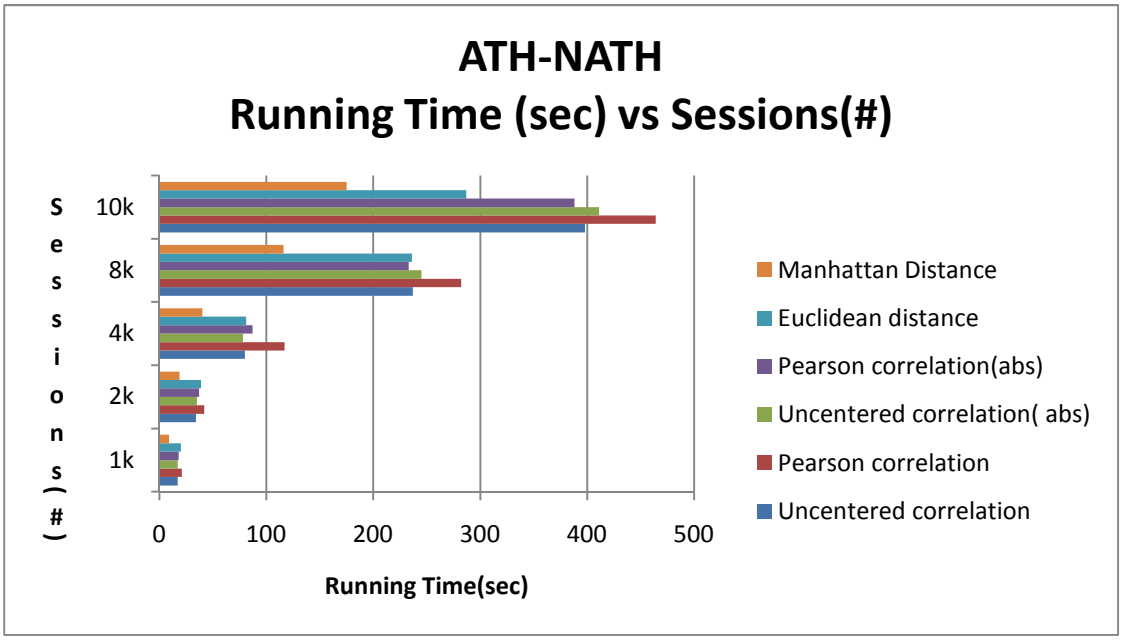


Figure 16- ATH-NATH, Running Time (sec) vs. Sessions (#) (without Spearman' Rank)

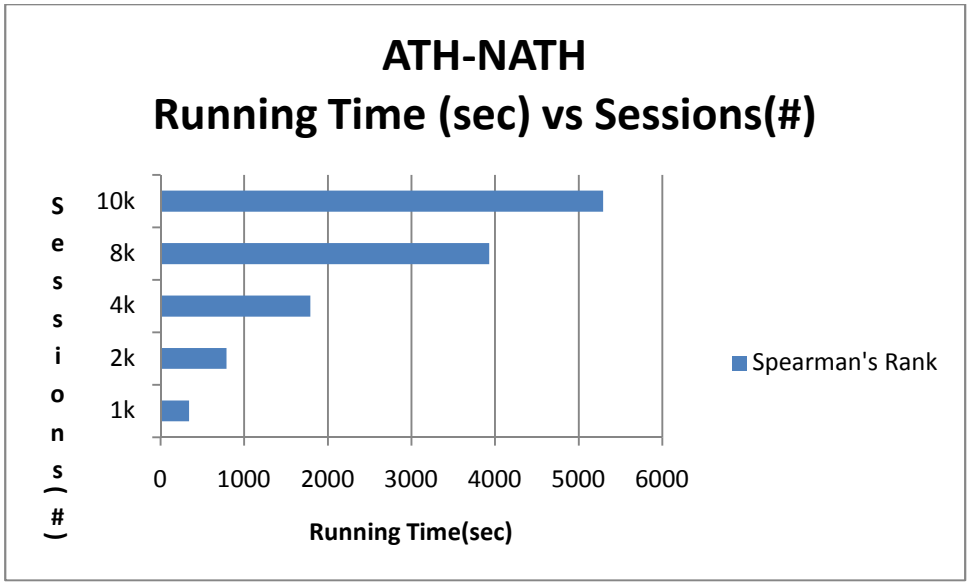


Figure 17- ATH-NATH Running Time (sec) vs. Sessions (#) (Spearman's Rank)

b) Sensitivity (S_n) vs. Sessions (#): In the graph (Figure 18), we infer that the Manhattan distance gives us higher sensitivity values for small number of sessions (less than 2000). But for session sizes greater than 4000, Uncentered correlation, Uncentered correlation (absolute), Pearson correlation and Pearson correlation (absolute) distance measures also give us good high average sensitivity values.

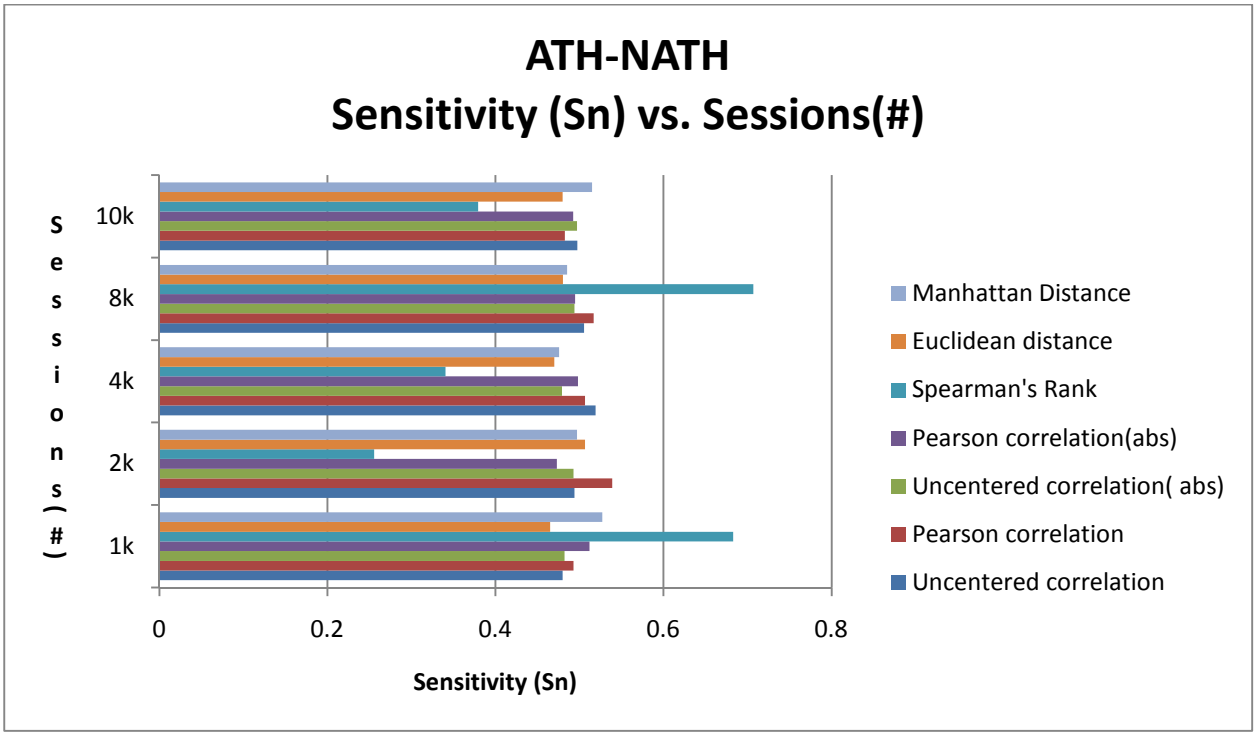


Figure 18 - ATH-NATH, Sensitivity (S_n) vs. Sessions (#)

c) Specificity (S_p) vs. Sessions (#): From the graph in Figure 18, we come to the conclusion that Pearson correlation (absolute) distance measure gives us the best result on an average. Also for session sizes ranging from 2k-8k, Uncentered correlation and Uncentered correlation (absolute), give us high values of specificity on an average.

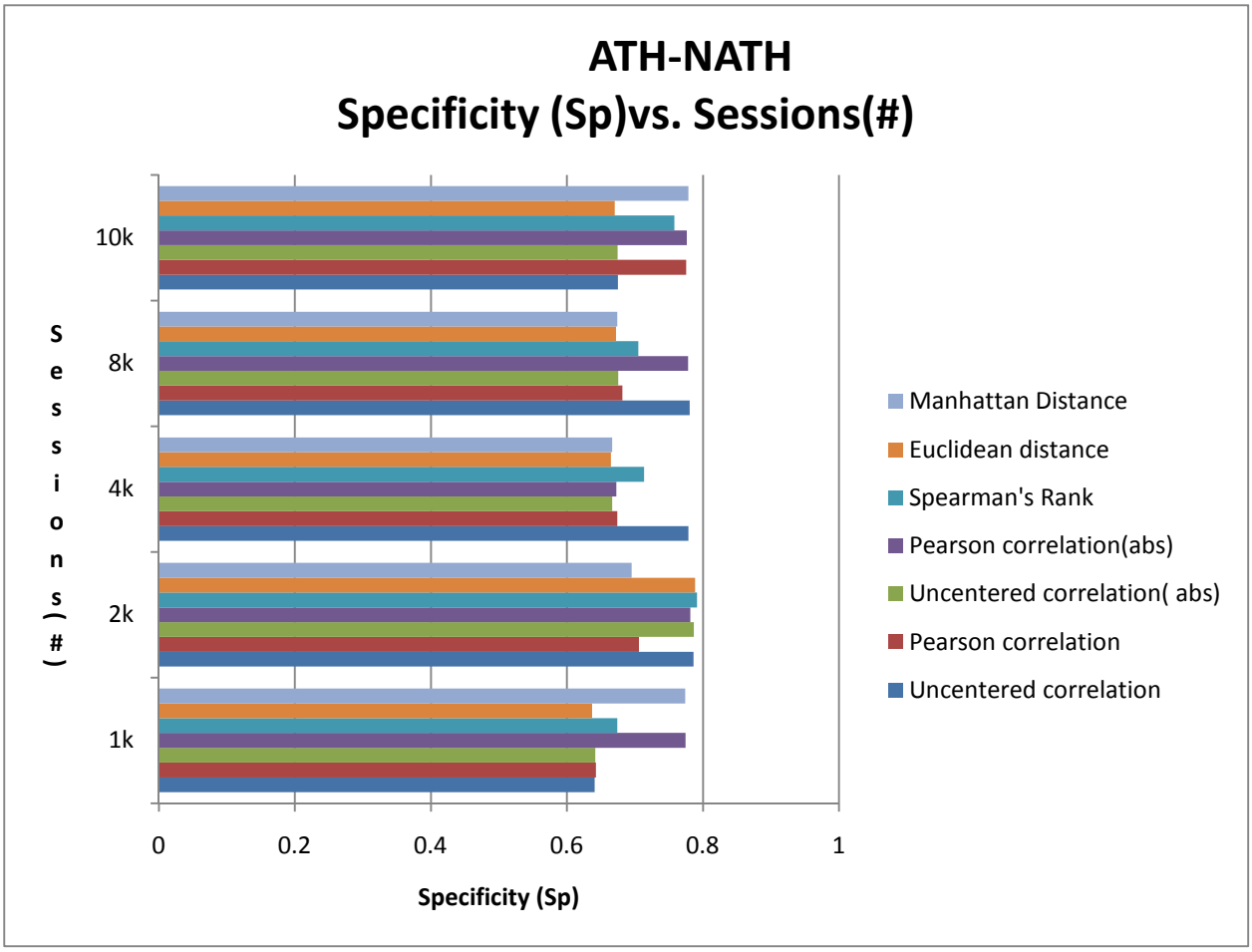


Figure 19 - ATH-NATH, Specificity (S_p) vs. Sessions (#)

d) Weighted Average (S_n, S_p) vs. Sessions (#): In the graph in Figure 20, we observe that Pearson correlation and Pearson correlation (absolute value) gives us the highest values on average. Also Uncentered correlation gives us good weighted average values for session sizes ranging from 2k-8k.

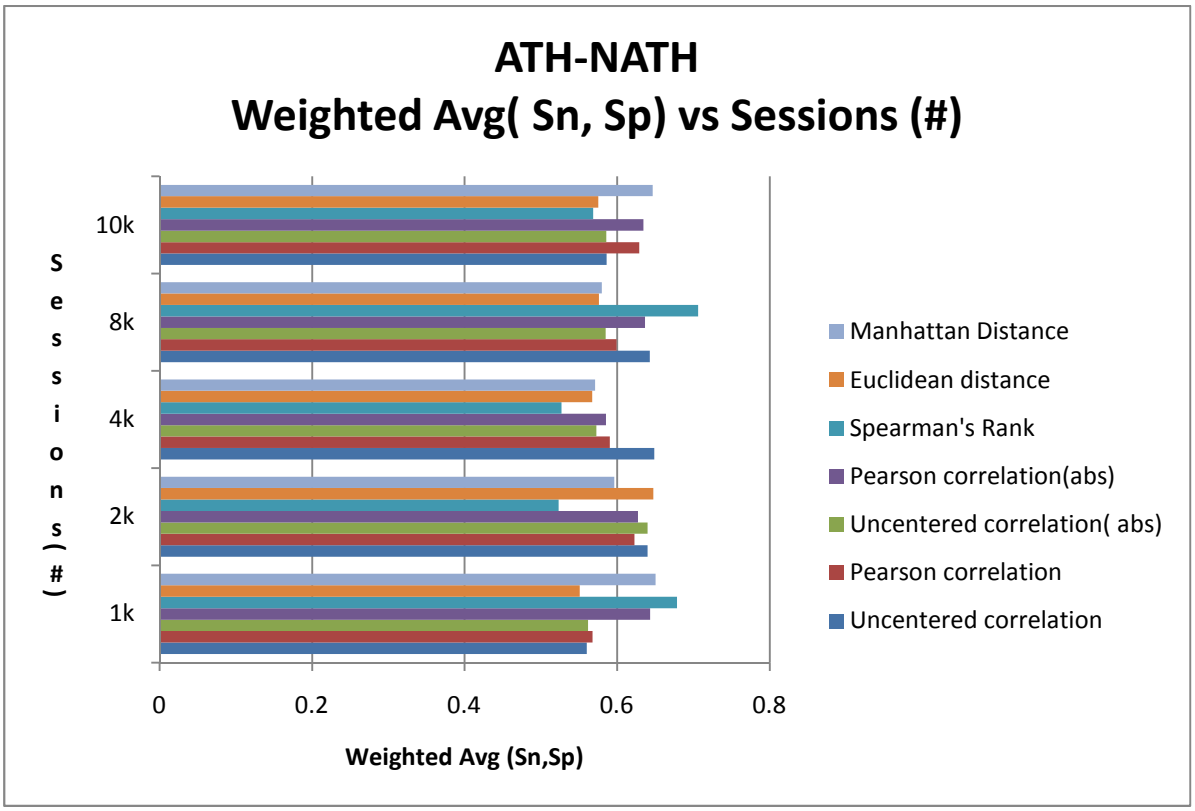


Figure 20- ATH-NATH, Weighted Average (S_n, S_p) vs. Sessions (#)

Chapter 6

SUMMARY AND FUTURE WORK

6.1 Summary

We can summarize from our work that Spearman's rank correlation distance measure performs the worst in terms of the running time and Manhattan distance performs the best. While, Euclidean distance gives us the best values for the weighted average of sensitivity and specificity for GA-NGA categorization, Uncentered correlation gives us the highest values for the weighted average for ATH-NATH categorization. Hence website administrators might look at this data and choose the appropriate parameters for their dataset.

6.2 Future work

In future**,** we plan to extend this tool to include other algorithms like *k*-medoids. In our experiments we have chosen a categorization based on the location. In future experiments, we might look at other parameters in which a website administrator might be interested in like browsers vs. donors. Also this tool can be further refined to look at the inter cluster distance and the intra cluster distance as a possible means to effectively evaluated the quality of a particular clustering result. Clustering results having a low inter cluster distance and a high inter cluster distance will definitely point to good clustering parameters.

References

1) T.B. Lee, R. Cailliau, J.F Groff, B. Pollermann, "World-wide web: the information universe", Internet Research, Vol. 20 Iss: 4, 2010, pp.461 – 471
2) L. Martin, “Usability analysis and visualization of web 2.0 applications”, In “10th Intl. Symp. on Web site evolution”, Beijing, China, 2008.
3) J. Nielsen, “Usability inspection methods”, Conference companion on Human factors in computing systems, Boston, Massachusetts, United States, 1994.
4) M.Y. Ivory and M. A. Hearst, “The state of the art in automating usability evaluation of user interfaces”. ACM Comput. Surv. , 2001.
5) A. Joshi, K. Joshi and R. Krishnapuram, “On mining web access logs”, In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000.
6) M. Eirnaki and M. Vazirgiannis, “Web Mining for web personalization”, ACM Trans. Internet Techno. , 2003.
7) M.S. Chen, J.S. Park, and P. S. Yu., “Efficient data mining for path traversal patterns by mining web logs” , IEEE Trans. On Knowl. And Data Engg., Vol. 10, No.2, March/April 1998.
8) *k*-means Algorithm – Wikipedia - http://en.wikipedia.org/wiki/Kmeans
9) Hierarchical clustering – Wikipedia - http://en.wikipedia.org/wiki/Hierarchical_clustering
10) Fuzzy clustering – Wikipedia - http://en.wikipedia.org/wiki/Fuzzy_clustering

11) J. Srivastava, R. Cooley, M. Deshpande, and P.N. Tan "Web usage mining: Discovery and applications of usage patterns from web data", SIGKDD Explor. News1. , 2000.

12) K. Storm, E.T Kraemer, C. Aurrecoeche, M. Heiges, C. Penington, J.C. Kissinger, "Web site Evolution: Usability Evaluation using time series analysis of selected episode graph". In WSE '09: Proc. of 9th IEEE Symp. On Web site evolution, 2009.

13) M. Eisen and M. de Hoon , Cluster 3.0 Manual (pdf), 1998. http://bonsai.hgc.jp/~mdehoon/software/cluster/cluster3.pdf

14) H. Kagdi and J. I.Maletic, "Mining for Co-Changes in the context of web localization", In 8th IEEE Intl. Symp. Of Website Evolution, (WSE'06), 2006.

15) Pearson correlation - Wikipedia http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

16) Spearman's Rank correlation–Wikipedia- http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficienthttp://en.wikipedia.org/wiki/Euclidean_distance

17) Euclidean distance – Wikipedia - http://en.wikipedia.org/wiki/Euclidean_distance.

18) Manhattan distance – Wikipedia - http://en.wikipedia.org/wiki/Manhattan_distance.

19) *k*-medoids algorithm – Wikipedia - http://en.wikipedia.org/wiki/K-medoids.

20) Lloyd's Algorithm – Wikipedia - http://en.wikipedia.org/wiki/Lloyd's_algorithm

21) O. Nasraoui, H.Frigui, R. Krishnapurma, A.Joshi , "Extracting web user profiles using relational competitive fuzzy clustering" , in *Intl J. Artif. Intell. Tools* , 2000.

22) F.E. Ritter, A.R. Freed and O. L.M. Haskett, "Discovering User Information Needs: The Case of University Department Web sites", Interactions, v.12 n.5, September-October 2005.

23) J. Pei, J. Han, B. Mortazvi-asl, and H Zhu, “Mining Access Patterns Efficiently from Web logs “, 2000

24) O. Nasraoui and R. Krishnapuram, “An evolutionary approach to mining robust multi-resolution web profiles and context sensitive URL associations”, *Intl. J. Comput. Intell. And Appl.*, 2002.

25) R. Kosala and H. Brockeel, “Web mining research: A survey”, ACM SIGKDD Explorations, 2, 2000.

26) Tec-ed Inc, “Assessing web site usability from Server Log Files”, “White paper”, Dec., 1999.

27) T. Draier, P. Gallinari, “Characterizing sequences of user actions for access logs analysis”, In Proc. of the 8th Intl. conf. on user modeling, London, UK, 2001.

28) P. Berkhin, “Survey of Clustering Data Mining Techniques”, Accrue Inc. 2002.

29) C. Kurz, H. Hlavacs, and G. Kotsis.”Workload generation by modeling user behavior in an isp subnet”, 2001.

30) E. Adar, D. S. Weld, B. N. Bersha, S. D. Gribble, “Why we search: Visualizing and predicting User behavior “, In WWW 2007, Alberta, Canada, 2007.

31) K. Storm, “Phd. Thesis” (unpublished), University of Georgia, GA, USA, 2011.

32) M. Eirinaki and M. Vazirgiannis, “Web Mining for Web Personalization”, ACM Trans. Internet Techno. , 2003.

33) Q. Yang, H. Zhang, T. Li, “Mining web logs for prediction models in www caching and prefetching”. In KDD’01, 2001.

34) Z. Yang, Y. Wang, and M. Kitsuregawa, “An effective system for mining web log”, “*Lecture Notes in Computer Science”,* 2006, pp 40-52.

35) S. K. Madria, S. S. Bhowmick, W. K. Ng, E.P. Lim, "Research Issues in Web Data Mining," Proc. First Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '99), 1999.

36) Z. Li, M. Sun, M. Dunham, and Y. Xiao, "Improving the Web site's effectiveness by considering each page's temporal information", In "*Lecture notes on Computer Science*", Springer Berlin / Heidelberg, 2003.

37) A.K Jain, M.N Murty and P.J.Flynn, "Data Clustering: A Review", ACM Computing Surveys, Vol. 31, No.3, Sept. 1999.

38) O. Etzioni, "The World Wide Web: Quagmire or Gold mine", Communications of the ACM, 1996.

39) J. Borges and M. Levene , "Data mining and User Navigational Patterns", In Proc. of the WEBKDD' 99 Workshop on Web Usage Analysis and User Profiling., August, 1999, San Diego, USA, pages 31-36,1999.

40) S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins ,"Mining the link structure of the World Wide Web", IEEE Computer, 32(8): 60-67, 1999.

41) J.M Kleinberg, "Authorative sources in a hyper-linked environment", In Proc. Of ACM-SIAM Symposium on Discrete Algorithms, 1998, pages 668-677, 1998.

42) S. Brin and L. Page. "The Anatomy of a large-scale hypertextual Web search engine". In 7th Intl. World Wide Web Conf.", Brisbane, Australia, 1998.

43) M. Basheer, "Master's Thesis", Univ. of Louisville, USA, 2001.

44) Log Files – Apache HTTP Server. http://httpd.apache.org/docs/2.1/logs.html#accesslog.

45) J.E Pitkow and K.A. Bharat, “WebViz: A tool for world-wide web access log analysis”, In Proc. of WWW1, Geneva, Switzerland, May 1994.

46) T. W. Yan, M. Jacobsen, H. G. Mollina, U. Dayal, “From User Access patterns to Dynamic HyperText linking”, In 5th Intl. World Wide Web Conference, (WWW5), Paris, France, 1996.

47) O. Nasraoui, H. Frigui, A. Joshi, R. Krishnapuram, “Mining web access logs using relational competitive fuzzy clustering”, In 8th Int. World Wide Web Conf. , Toronto, Canada, 1999.

48) L. Becker, “Is your Website Effective?”, Online video, http://videos.webpronews.com/2008/06/15/larry-becker/

49) Session,“Definition”, https://www.adobe.com/livedocs/coldfusion/6.1/htmldocs/shared28.htm

50) Sensitivity and Specificity, Wikipedia, http://en.wikipedia.org/wiki/Sensitivity_and_Specificity.

51) M. Spiliopoulou, C. Pohle, L. C. Faulstich, “Improving the effectiveness of a website with web usage mining”, In Proc. of WEBKDD’99, 1999.

52) M. Gillenland, “Permutation Generator”, (Permutation generator Software), http://www.merriampark.com/perm.htm.

53) M. J. L. de Hoon, S. Imoto, J. Nolan, and S. Miyano, “Open Source Clustering Software “, In Bioinformatics, (Cluster 3.0 Software), 20 (9): 1453--1454, 2004.

APPENDIX A

ABBREVIATIONS USED

Table 1: List of Abbreviations used

HTTP	HYPER TEXT TRANSFER PROTOCOL
HTML	HYPER TEXT MARKUP LANGUAGE
URL	UNIFORM RESOURCE LOCATOR
OLAP	ON-LINE ANALYTICAL PROCESSING
OLAP	ON-LINE ANALYTICAL PROCESSING