October 23, 2013

Web UR:EFFECTIVE TECHNIQUES FOR WEB USAGE MINING AND RECOMMENDER SYSTEM

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

POORNALATHA G.

DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALORE - 575025

October, 2013

Dedicated to my dearest mother Smt.Sumithra

DECLARATION

By the Ph.D. Research Scholar

I hereby declare that the Research Thesis entitled **Web UR: EFFECTIVE TECHNIQUES FOR WEB USAGE MINING AND RECOMMENDER SYSTEM** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Information Technology** is a *bonafide report of the research work carried out by me.* The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

Poornalatha G. (092028IT09F03)
Department of Information Technology

Place: NITK, Surathkal.
Date: 25th October, 2013

CERTIFICATE

This is to *certify* that the Research Thesis entitled **Web UR: EFFECTIVE TECHNIQUES FOR WEB USAGE MINING AND RECOMMENDER SYSTEM** submitted by **POORNALATHA G**, (Register Number:092028IT09F03) as the record of the research work carried out by her, is *accepted as the Research Thesis submission* in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

Prof.Ananthanarayana V. S. Dr.Prakash S. Raghavendra

Research Guides

Prof.Ananthanarayana V. S.

Chairman - DRPC

Acknowledgements

This doctoral thesis would not have been possible without the guidance and the help of several individuals who contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, I would like to express my deepest and sincerest gratitude to my Ph.D. guide *Dr.Prakash S. Raghavendra*. He inspired me to explore new thoughts in my research work and supervised the work at all stages that enabled the successful completion of this work on time. I sincerely thank him for his consistent support, encouragement, motivation, valuable advice and excellent guidance.

I humbly place on record my sincere gratitude to *Dr.Ananthanarayana V. S.*, Professor and Head, Department of Information Technology, NITK, Surathkal, who guided me in the final stages of my research and provided all the required support. His valuable inputs proved of vital importance in the completion of my research work.

I am extremely thankful to *Dr.Ram Mohana Reddy*, Professor, Department of Information Technology, NITK, Surathkal, *Dr.Prasad Krishna*, Professor, Department of Mechanical Engineering, NITK, Surathkal, and *Dr. Ashvini Chaturvedi*, Department of Electrical and Electronics, NITK, Surathkal, who were part of my Research Progress Committee for their useful suggestions and valuable comments.

I thank *Mr.Praveen Shetty*, Assistant Professor, Engish Department, MIT, Manipal for his help in the language editing.

I cherish the help and support rendered to me by my fellow research scholars, Department of Information Technology, NITK, Surathkal.

I thank the administrators of MIT and Manipal University for permitting me to

carry out research work at NITK, Surathkal. I extend my thanks to all the faculty members and staff of Department of Information and Communication Technology, MIT, Manipal, for their support.

My special thanks to all faculty members and staf of Department of Information Technology, NITK, Surathkal for their support and help.

I am truly greatful and thankful to Dr.Janardhana Prabhu, my husband, for his support and encouragement without which it would not have been possible for me to pursue PhD. I am truly indebted and thankful to my mother Smt.Sumithra for her moral support and blessings. I cannot forget the sacrifice, love and co-operation of my son Prajwal and daughter Prapthi throughout my research work.

Last, but by no means least, I thank all my friends and others who have helped me directly or indirectly during my research work.

Place: NITK, Surathkal Poornalatha G.

Date: 25th October, 2013

ABSTRACT

The proliferation of internet along with the attractiveness of the web in recent years has made web mining as the research area of great magnitude. Web mining essentially has many advantages which make this technology attractive to researchers. The analysis of web users' navigational pattern within a web site can provide useful information for server performance enhancements, restructuring a web site, direct marketing in e-commerce etc.

This thesis discusses an effective clustering technique that groups user sessions, by modifying k-means algorithm. The proposed distance measures namely, the variable length vector distance, sequence alignment based distance measure, and hybrid sequence alignment measure are explained. The results obtained are validated.

The present work attempts to solve the problem of predicting the next page to be accessed by the user based on the mining of web server logs, that maintains the information of users who access the web site. The proposed model yields good prediction accuracy compared to the existing methods like Markov model, association rule, ANN etc.

A recommender system based on session collaborative filtering is proposed. The proposed recommender system is compared with a few other recommender systems by using precision and recall as metrics, and a better performance is observed. The outcome of prediction and recommender system could be used to suggest any structural modifications to the web site.

Keywords: Access Patterns, Clustering, Sequence Alignment, Web page prediction, Web page recommendation, Web session.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

INTRODUCTION

The present generation is living in an information era. The evolution of the internet along with the popularity of the web has made even an ordinary person to use the information available at his finger tips for various purposes. Web is a collection of inter-related files on one or more web servers. Web has been adopted as a critical communication and information medium by a majority of the population. However, the required information may not be obtained immediately from the web since there is an exponential growth in terms of amount of data available in all fields. Hence, there is a need to mine this huge web data to uncover the hidden knowledge.

1.1 Web Mining

Web mining has attracted a great attention among the researchers due to the wide popularity of the web in recent years. Web mining is the extraction of useful information from the huge amount of data available in the web logs. Web mining can be categorized into four types based on type of data used for mining: web content mining, web structure mining, web usage mining, and web user profile mining (Chu Hui et al. 2008) (Srivastava et al. 2000) as shown in Fig.1.1.

Web content mining focuses on useful knowledge which is extracted from web pages of a web site. It is used to discover what a web page is about and how to

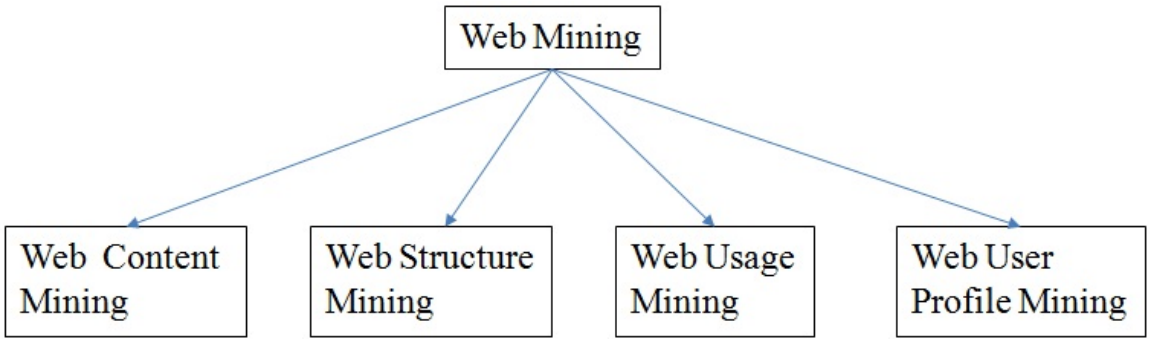

Figure 1.1: Types of web mining

uncover new knowledge from it. Web content mining is used to extract product features commented by consumers, determine whether the comments are positive or negative (semantic orientation), or produce a feature based summary instead of text summary etc.

Web structure mining is the process of applying the graph theory to analyze the node and connection structure of a web site. Web structure mining is used to analyze the links between web pages through the web structure to infer the knowledge. Extracting patterns from hyperlinks in the web and mining the document structure are the two kinds of structure mining.

Web usage mining is extracting the information from web log file which maintains the information of web pages accessed by users. From the technical point of view, web usage mining is the application of data mining techniques to usage logs of large data repositories maintained by web servers (Hu 2003) (Nina et al. 2009). Applications of web usage mining are personalization and collaboration, marketing, web site design, and evaluation etc.

Web user profile mining uses the demographic information (e.g.registration data) about users of the web site.

The proposed work mainly concentrates on web usage mining. Web usage mining is the discovery of meaningful patterns from data generated by the client-server transactions on one or more web localities (Umapathi et al. 2008). The data mining

techniques that are commonly used in web usage mining are clustering, association mining, and sequence mining. Clustering is natural grouping of users, pages, or sessions based on some similarity measures. Association finds the URLs that are accessed together frequently, whereas the sequential mining gives importance to the order in which URLs are accessed.

1.2 Motivation

The web poses great challenges for effective resource and knowledge discovery because of the following reasons (Han et al. 2006):

- the web seems to be too huge for effective data mining
- the complexity of web pages is far greater than that of any traditional text document collection
- the web is a highly dynamic information source
- the web serves a broad diversity of user communities
- due to the nested structure of HTML code, much of the web information is semi-structured

Due to the rapid growth in the use of web and also the challenges mentioned above, the task of analyzing, understanding and producing useful information manually from a vast quantity of data available on the web is a very complicated and time consuming task. These challenges have propelled researchers into finding, efficient use of resources and effective discovery of useful knowledge from the internet to discover potentially useful and previously unknown knowledge from the web data.

A user is defined as an individual, who accesses the web pages of a web site through a browser. A user session is the click stream of the pages viewed by a user in succession

within a time frame. A thirty minute time out is often used as the default method of breaking a user's click-streams into sessions (Catledge et al. 1995) (Srivastava et al. 2000). A session length can be defined as the number of pages viewed by a user in a session. Most of the research efforts use binary vectors to represent a session. The vector representation consumes more space, since the session length is not same for all sessions. Also, a web site may have large number of web pages that are hyperlinked, and a user may not visit all the pages of a web site in a session. Hence, there is a need to represent sessions effectively. This thesis represents session effectively, proposes three distance measures that are used to find the similarities between any two user sessions of variable length, and groups user sessions by applying clustering technique.

In general, a user may spend lot of time unnecessarily while navigating through a web site to find the relevant web page. Hence, predicting or forecasting the next page to be visited by the user may reduce user latency. Most of the researchers emphasize on the Markov model for prediction (Deshpande et al. 2004). Since the accuracy of lower order Markov models are less and higher order Markov models consume space for many states while improving the accuracy, researchers tried to integrate the Markov model with other data mining techniques like clustering, association rule etc. (Dutta et al. 2009) (Kim et al. 2004) to improve the prediction accuracy. However, the hybrid or integrated prediction models consume more time for prediction. Also, much improvement in the accuracy is not observed by most of the existing prediction models. Hence, there is a need to develop a prediction model that gives better prediction accuracy. In this thesis, we propose prediction models and measure the goodness by using accuracy as a parameter.

Web page recommendation, recommends web pages to the user which may be relevant or required to the user based on the past navigation history. This would help business and marketing applications (Adnan et al. 2011). Association rule and its variations are used in the literature to develop a recommender system (Forsati et

al. 2009). The association rules refer to sets of pages that are accessed together with a support value exceeding some specifed threshold. These pages may not be directly connected to one another via hyperlinks but it may reveal a correlation between pages (Srivastava et al. 2000). However, it is difficult to find suitable set of rules that finds accurate recommendations. Collaborative filtering (CF) is a popular technique used in web personalization for building recommender systems (Suryavanshi et al. 2005b). Hence, this thesis proposes a model for recommendation based on the CF technique. The efficiency of the recommender model is evaluated by considering precision and recall as parameters. The precision is the number of recommended pages that are relevant while, recall is, the number of pages that are correctly recommended (Kim et al. 2004).

1.3 Problem Statement

To design and develop effective techniques for extracting the similar kind of user access patterns from the huge web server log repository for web page prediction and recommendation. The objectives are as follows:

- to develop effective clustering technique for web usage mining based on user navigation patterns
- to predict users next request to improve access time or reduce access latency based on Most Recently Used/Accessed pages
- to develop a technique for web page recommendation system
- to suggest restructuring the web site based on prediction/recommendation

1.4 Outline of the Thesis

This thesis is organized as follows: Chapter 2 gives overview of the proposed model for web usage mining and details of data sets used. Chapter 3 discusses clustering technique and various methods used to measure similarity between pair of user sessions. Chapter 4 describes the proposed prediction models. Chapter 5 outlines the recommender system developed. Conclusions and future work are given in Chapter 6 followed by references and publications at the end.

Chapter 2

WEB USAGE MINING FRAMEWORK

The purpose of this chapter is to describe the general framework for web usage mining. The various stages of web usage mining are discussed. The standard data sets used in this work for experimental purpose are briefly discussed. Broadly speaking this chapter provides an overview of the framework used for web usage mining, and details of the components required for the same, that are essential to understand the remaining chapters.

2.1 Web Usage Mining

Web usage mining performs mining on web logs. A web log, also called click stream data, contains entries corresponding to a mouse click with respect to the page reference made by the user to the web server. By analyzing these logs, information about a user or a group of users may be detected. This valuable information could be used for various application, some of which are listed below:

- web page prefetching and caching (Srivastava et al. 2000)
- improve server performance (Srivastava et al. 2000)
- web site modification (Srivastava et al. 2000)

- business intelligence that improves sales by placing advertisements at appropriate location (Adnan et al. 2011)
- recommender system (Srivastava et al. 2000)
- social network analysis (Adnan et al. 2011)

Cooley et al. (1997a,b) proposed a definition of web mining, developed taxonomy of the various ongoing efforts related to it and presented a general model to identify transaction for web usage mining. Srivastava et al. (2000) discussed the three main tasks for performing web usage mining namely, preprocessing, pattern discovery and pattern analysis. The paper also provided a detailed taxonomy of the work in the area of web usage mining. Facca et al. (2005) presented a survey of developments in the area of web usage mining. They provided details for most of the commercial tools that perform analysis on web log data based on statistical analysis techniques and a few products that exploit data mining techniques.

2.2 Framework for Web Usage Minig

Fig.2.1 depicts various stages of web usage mining. The components with dashed outline, highlight the major contribution of the present work in terms of methods/techniques. The various components of the framework are as given below:

web server log file - It is a text file with one line for each user request. Each line has the information such as, host making the request, timestamp, requested URL, HTTP reply code, bytes in the reply etc. as shown below:
"unicomp6.unicomp.net - - [01/Jul/1995:00:00:06-0400] " GET / shuttle/ countdown/ HTTP/1.0" 200 3985"

parse log - The required fields need to be identified and extracted from the web server log file. The required fields could be IP address/hostname, date and time, requested URL, and response code. These fields are essential to create a user session. The same is stored in a database for easier and efficient handling of data. Table 2.1 shows the sample data after parsing the web server log file.

Table 2.1: Sample parsed data

Session number	Host name	Date time	Requested URL	Response code
1	unicomp6. unicomp.net	07/01/1995 0:00:06	/shuttle/countdown/	200

data cleaning - Generally, there are a variety of files accessed as a result of a request by a client to view a particular web page. These include image, sound, and video files; executable cgi files; and HTML files. Thus, the server logs contain many entries that are redundant or irrelevant for the data mining tasks. For example, all the image file entries are irrelevant or redundant. Since, as a URL, several image files is selected, the images are transferred to the client machine and these files are recorded in the log file as independent entries. The process of removing redundant or irrelevant entries from the web server log files is referred to as data cleaning. A very simple form of data cleaning is performed by checking the suffix of the URL name. For instance, all the log entries with filename suffixes such as, .gif, .jpeg, .GIF, .JPEG, .jpg, .JPG and map are removed from the log (Cooley et al. 1997a). Thus data cleaning removes unwanted data such as image files, script files, HTTP response code other than 200, like 400 series (400 - bad request, 401 - unauthorized, 404 - not found etc.), null request etc., as they do not contribute for session creation.

sessions - The task of user and session identification is to find out the different user sessions from the original web access log. The different IP addresses distinguish

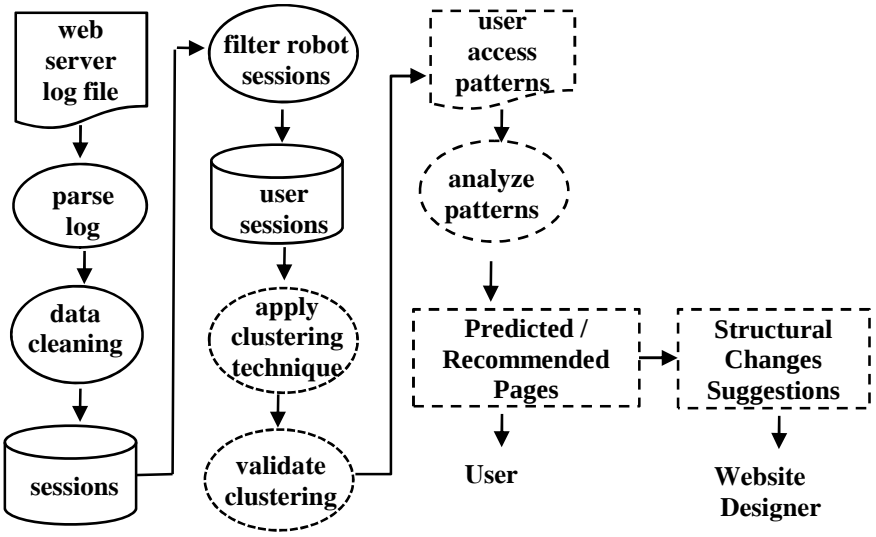

Figure 2.1: Framework for web usage mining

different users. The sessions are created based on IP address and timeouts (Dixit et al. 2010) (Pallis et al. 2005). These sessions are stored in a database. In general, a session consists of web pages visited by a user in succession. Suppose, a user visits pages P_1,P_2,P_7 of a web site in a sequence, then, the session 'S' is represented as S=(P_1,P_2,P_7).

filter robot sessions - Web robots are software programs or agents, that automatically traverse the hyperlink structure of the world wide web in order to locate and retrieve information (Tan et al. 2002). In order to cluster user sessions, it is required to remove these robot sessions. To remove these sessions, identify request with "robot.txt" in web log, find the corresponding IP address/host name and remove all sessions with these IP address/host name.

user sessions - The user sessions obtained after filtering are stored in a database for further analysis.

apply clustering technique - User sessions need to be grouped based on some similarity criteria, by using required clustering technique.

validate clustering - Clustering validation is a field where attempts have been made to find rules for quantifying the quality of a clustering result (Halkidi et al. 2001). This issue, however, is a difficult one and typically people evaluate clustering based on various quality measures such as, Dunn's validity index, Davies-Bouldin (DB) validity index, Silhouette validity index, Jaccard index, Rand index, Euclidean distance, Minkowski distance etc. (Han et al. 2006) (Zahid et al. 2011). The existing algorithms like, ROCK, CHAMELEON, TURN etc., that cluster the web sessions, have treated sessions as unordered sets of clicks. The similarity measures used to compare sessions were simply based on intersections between these sets, such as the Jaccard coefficient, which basically measures the degree of common visited pages in both sessions to be compared (Wang et al. 2002). The common pages between two sessions are computed during clustering and the same information could be used for Jaccard index that avoids re-computing the same information. Also, compare to other indices mentioned above, the Jaccard index is better in terms of complexity also. Hence, Jaccard index is used to validate the clusters.
The general criterion of a good partitioning is that, objects in the same cluster are close or related to each other, whereas objects of different clusters are far apart or very different (Han et al. 2006). The Davies-Bouldin (DB) index (Davies et al. 1979) attempts to minimize the average distance between each cluster and the one most similar to it. That means, the DB index considers within cluster scatter and separation between clusters also. Since, Jaccard index finds similarity within cluster, DB index is used to find separation between clusters.

user access patterns - The web page access patterns of user, obtained after applying the mining techniques onto the user sessions are stored in a file.

analyze patterns - The discovered access patterns are interpreted in order to identify the interesting patterns that represent knowledge.

predicted/recommended pages The next page to be visited by the user may be predicted, or relevant pages are recommended, based on analyzed patterns.

structural changes suggestions The web designer may use the outcome of prediction, or recommendation, for restructuring the web site.

The process of parsing the web server log, data cleaning and filtering are together referred as data preprocessing. Data preprocessing is the essential step that should be carried out before applying data mining techniques. This improves the overall quality of patterns mined and the time needed for mining is reduced. Therefore considerable amount of time is required to understand the format of data present in the log file. The preprocessing of data is done only once for a given data set in order to transfer the raw data to the required format. Once preprocessing is done, different data mining techniques could be applied on the data to extract useful information.

2.3 Data Sets

This section discusses about the data sets that are used to implement the proposed algorithms for clustering, prediction and recommendation techniques. Three standard data sets are considered that are available for free download. The first set is NASA log taken from NASA Kennedy space center www server in Florida (http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html). This consists of approximately 10, 00,000 entries. The log has the data collected from 00:00:00 July 1, 1995 through 23:59:59 July 31, 1995, a total of 31 days and the file size is 117,532 KB.

Each entry of the server log file is parsed to extract the required attributes such as

the IP address/user name, date/time, requested web page, HTTP response code and number of bytes transferred. These attributes are further cleaned to remove irrelevant entries and only valid entries are stored in a database table. For example, the entries of the log file with the HTTP response field containing error code or number of bytes transferred is zero, or the request field with a null value need not be considered and are removed. Thus after the data cleaning process 8,97,553 valid entries are obtained, and around 14.4% of data is eliminated by data cleaning. The data is ordered based on the IP address, date and time fields. Unique 860 entries are identified from the request field. The HTTP protocol establishes a separate connection for every file requested by the user. Consequently, this results in several entries in log file for graphics and script files against the request made by the user to access a single web page. Therefore, if multiple entries exist with the same IP address, date and time, only one entry is considered for constructing a user session. Thus, user sessions are identified based on IP address, date and time fields. Since date and time is considered to form the session, the web pages of a session will be automatically in sequential order. Hence, a session consists of the web pages viewed by a user in succession. The robot sessions are further filtered by searching for robots.txt files in the web log. After constructing distinct user requests, based on domain knowledge obtained, 30 categories of pages are formulated as given in Table 2.2 to analyze the clusters.

Table 2.2: Web page categories - NASA data set

P_1	/elv/	P_{11}	/icon/	P_{21}	/shuttle/countdown/
P_2	/facilities/	P_{12}	/images/	P_{22}	/image/movies/
P_3	/shuttle/mission/	P_{13}	/logistics/	P_{23}	/software/
P_4	/downs/	P_{14}	/mdss/	P_{24}	/statistics/
P_5	/base-ops/	P_{15}	/msfc/	P_{25}	/history/apollo/
P_6	/bio-med/	P_{16}	/news/	P_{26}	/history/gemini/
P_7	/facts/	P_{17}	/pao/	P_{27}	/history/mercury/
P_8	/finance/	P_{18}	/payloads/	P_{28}	/shuttle/
P_9	/history	P_{19}	/persons/	P_{29}	/shuttle/resources/
P_{10}	/htbin/	P_{20}	/procurement/	P_{30}	/shuttle/technology/

The second set is MSNBC data set taken from msnbc.com that gives the page visits of users who visited msnbc.com (http://kdd.ics.uci.edu/databases/msnbc/msnbc.html) on September 28, 1999 and the file size is 12,287 KB. Visits are recorded at the level of URL category and also in time order. Therefore, preprocessing was not required for this data set. The number of page categories for this data set is 17 and each category may have 10 to 5000 number of pages linked under each of these page categories. The description of page categories for this data set is as given in Table 2.3.

Table 2.3: Web page categories - MSNBC data set

P_1	Front page	P_7	Misc	P_{13}	Msn-sports
P_2	News	P_8	Weather	P_{14}	Sports
P_3	Tech	P_9	Msn-news	P_{15}	Summary
P_4	Local	P_{10}	Health	P_{16}	Bbs
P_5	Opinion	P_{11}	Living	P_{17}	Travel
P_6	On-air	P_{12}	Business		

The third set is Music Machine data set (MM data set) taken from http://www.cs.washington.edu/ai/adaptive-data/ that consists log of December 1997 and the file size is 4516 KB. The web site gives details of various musical instruments, samples, images, manufacturers of different types of instruments, details of dealers etc. The user sessions for the MM data set are created by preprocessing the log as described for the NASA data set. 4726 unique requests are identified from this data set.

Chapter 3

EFFECTIVE CLUSTERING TECHNIQUE

The purpose of this chapter is to describe the proposed clustering technique, that group web page sessions. To explore the navigation paths, some similarity criteria are required. Here, three approaches that find the distance between any two web page sessions are proposed, and discussed. Couple of well known statistical methods are used to determine the goodness of clusters formed by the proposed techniques.

3.1 Clustering

Clustering is a technique for grouping user sessions such that, within a single cluster the usage pattern is more similar while sessions in different groups are dissimilar. Clustering, in general, groups data objects based only on information found in data that describes the objects and their relationships. The goal of clustering is that, the objects within a group be similar to one another and different from the objects in other groups. In general, the major clustering methods can be classified into the following categories (Han et al. 2006).

- Hierarchical clustering - A set of nested clusters organized as a hierarchical tree
- Partitioning clustering - A division of data objects into non-overlapping subsets

(clusters) such that each data object is in exactly one subset

- Density based methods - Cluster data objects as long as the density (number of objects or data points) in the neighborhood exceeds some threshold
- Grid based methods - Grid based methods quantize the object space into a finite number of cells that form a grid structure
- Model based methods - Model based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model

The analysis of web users navigational pattern obtained by clustering provides useful information for applications like, server performance enhancements, restructuring a web site, direct marketing in e-commerce etc.

3.2 Clustering Background

A number of clustering approaches have been proposed in the literature. For example, Li (2008) pointed out that number of clusters, the initial point of the respective clusters and the defining of criterion function are the three key points that should be considered in web session clustering, and proposed an algorithm based on increase of similarities. Krol et al. (2008), investigated on internet system user behavior using cluster analysis. Here sessions are represented as vectors where each dimension represents a web page and stores the value of user interest in each page of a session. The sessions are clustered using hard c-means algorithm. Fu et al. (2000) proposed a generalization based clustering method which employs the attribute-oriented induction method to reduce the large dimensionality of data. Shi et al. (2009) considered the web pages visited by users and time spent at each of the web page to reveal the interests of web users while surfing. These approaches consider sessions as a vector of same length, but in general each user session may not be of equal length.

Various approaches are also discussed in the literature about the different types of distance measures used for clustering web user sessions. Hay et al. (2004) illustrated a new method for mining navigation patterns using a sequence alignment method. This method partitions navigation patterns according to the order in which web pages are requested and handles the problem of clustering sequences of different lengths. However, here the distance is measured based on the number of insertion/deletion/reordering operations and also the results are compared with a method based on Euclidean distance measure which does not consider the sequence information.

Khasawneh et al. (2007) introduced a multidimensional session comparison method using dynamic programming. Though more than one dimension is considered for comparison between pair of sessions, page list is the primary dimension for comparison. Chaofeng et al. (2007) introduced a method for measuring the similarities between web pages that takes into account viewing time of the visited web page along with the URL. Similarities between web sessions are measured using sequence alignment. Only 500 web sessions are considered for their experiment. Mojica et al. (2005) clustered web pages using a distance based algorithm by modifying gravitational algorithm which is similar to the basic k-means algorithm. Yilmaz et al. (2010) used ontology and sequence information for extracting behavior patterns from web navigation logs that merges web usage mining with web content mining. Seung-Joon (2007) proposed an extended concept of the measure of similarity for effective hierarchical clustering. Liu et al. (2010) focused on internal validation and presented some of widely used internal clustering validation measures. Xu et al. (2010) used cosine similarity as a distance measure that requires sessions to be of same length. Pallis et al. (2007) assessed the quality of user session clusters in order to make inferences regarding the users' navigation behavior. They used model based clustering algorithm to group web user sessions and the clusters are validated by using statistical chi-square test.

3.3 Modified K-Means Algorithm

The studies have shown that the most commonly used partitioning-based clustering algorithm is, the k-means algorithm, which is more suitable for large data sets. K-means clustering is a method of cluster analysis, which aims to partition 'n' observations into 'k' clusters, in which each observation belongs to the cluster with the nearest mean. Euclidean distance is generally used as a metric. The main advantages of this algorithm are, simplicity and speed, which allows it to run on large data sets. Its disadvantage is that, it does not yield the same result with each run since the resulting clusters depend on the initial random assignments.

The basic k-means algorithm initially selects the cluster centroids randomly and finds the new cluster centroid based on the average value obtained within each cluster, in each iteration. Though, initial random centroids are actual data points, the centroids obtained in further iterations, may be some points other than the actual data points. The k-means method, however, can be applied only when the mean of a cluster is defined. This may not be the case in some applications, such as when data with categorical attributes are involved (Han et al. 2006). The web user sessions could be considered as categorical data because, session consists of discrete set of pages. Hence, the basic k-means is modified. The variants of the basic k-means method like, PAM (partitioning around medoids), CLARA (clustering large applications) and CLARANS (clustering large applications based upon randomized search) etc. ensure that, one of the objects is the cluster centroid. However, PAM works efficiently for small data sets but does not scale well for large data sets. The complexity of each iteration is O(k$(n-k)^2$). The Efficiency of CLARA depends on the sample size. It draws multiple samples of the dataset and applies PAM on samples. A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased. The complexity of each iteration is O(ks^2 + k(n-k)), where, 's' is the size of the sample, 'k' is the number of clusters, and 'n' is the number of objects. CLARANS is the mixture of PAM and CLARA. The

clustering quality depends on the sampling method. The computational complexity of CLARANS is O(n^2), where 'n' is the number of objects (Han et al. 2006) (Pujari 2001). But, K-means is relatively scalable and efficient in processing large data sets compare to these medoid based algorithms. Hence, to have the centroids as actual data points, in each iterations, the basic k-means is modified suitably. In the modified k-means algorithm, the old cluster centroid is updated by the delta amount, where, delta is the average distance value of each cluster. Algorithm 1 shows major steps of the proposed modified k-means algorithm.

Algorithm 1 Modified k-means

Input: A set of data points $D = \{ d_1, d_2,..., d_n\}$, the desired number of k clusters, the maximum number of iterations itr;

Output: A set of clusters $C = \{C_1, C_2,..., C_k\}$ of D

1: $count \leftarrow 0$
2: select any k data points $\{d_1, d_2,..., d_k\}$ from D and set $m_i \leftarrow d_i$ to get the initial center points $M = \{m_1, m_2, ..., m_k\}$ where, $0 < i < k + 1$
3: $newC \leftarrow empty, newM \leftarrow empty$
4: **loop**
5: for each d_i, compute $Dist$=$\{dist_1$,$dist_2$,...,$dist_k\}$ where, $dist_i = |d_i - m_i|$ and $0 < i < n + 1$
6: assign d_i to C_j where $dist_j$=min($Dist$) and $0 < j < k + 1$
7: **for** each c_j **do**
8: $delta_j \leftarrow sum(distances\, of\, each\, d_i\, in\, C_j)/number\, of\, data\, points\, in\, C_j$
9: **end for**
10: $newM \leftarrow (m_1 + delta_1, m_2 + delta_2, ..., m_k + delta_k)$
11: **if** $(C = newC\, or\, M = newM\, or\, count > itr)$ **then**
12: break
13: **end if**
14: $newC \leftarrow C, newM \leftarrow M$
15: $count \leftarrow coun + 1$
16: **end loop**

The modified k-means selcts actual data points as new centroids, instead of having a random data point as a centroid. If a random point is selected as the cluster centroid, it is difficult to compare the sessions with the random points. Since, the requirement is to represent sessions as it is, and cluster them, comparison between

sessions is possible, if the centroid is also a session. Hence, the basic k-means is modified. The average distance of the sessions from the centroids is computed. The existing centroid is moved by delta amount based on the average value obtained. Therefore, in order to find the new centroid that represents one of the sessions in the cluster, the existing cluster centroid in moved by delta in each iteration. Also, the number of iterations required is less in the modified k-means because the centroid is one of the data point. Whereas, random point considered as a centroid in the basic k-means algorithm may require more iterations. To ensure that, the modified k-means groups sessions correctly, numerical example is taken. The numerical example is presented as given in Table 3.1 to compare the basic and modified k-means algorithm for the example data set D={11,22,18,15,25,36,27,8,39,10}. The results reveal that, the modified k-means algorithm is better than the basic k-means algorithm in terms of number of iterations taken to converge and the quality of clusters formed irrespective of the initial centroids selected. Though, the initial cluster centroids are changed randomly, the clusters formed by modified k-means are consistent and meaningfull compare to the clusters formed by the basic k-means method, as can be seen from the three different cases, shown in the Table 3.1. Thus, the empirical study shows that modified version of k-means is better than the basic k-means.

Table 3.1: Comparison of basic and modified k-means

No.	Initial centroids	Basic k-means clusters	iterations	Modified k-means clusters	iterations
1	m1=8	c1=11,15,8,10	5	c1=11,18,15,8,10	3
	m2=18	c2=22,18,25,27		c2=22,25,27	
	m3=36	c3=36,39		c3=36,39	
2	m1=11	c1=11,15,8,10	4	c1=11,15,8,10	5
	m2=22	c2=25,36,27,39		c2=36,39	
	m3=18	c3=22,18		c3=22,18,25,27	
3	m1=27	c1=22,18,25,36,27,39	20	c1=36,39	6
	m2=8	c2=8		c2=11,15,8,10	
	m3=10	c3=11,15,10		c3=22,18,25,27	

In general, the web user sessions are not simple data points, but n-dimensional vectors. Before clustering web user sessions, the algorithm, modified k-means is changed to suit the requirements of clustering sessions as given in Algorithm 2. Here, the set

of data points "D" is replaced by set of web user sessions "WS" as the input. To compute "Dist" given in line number 5 of the Algorithm 2 suitable distance measure, that finds distance between two sessions of variable lengths, should be used instead of general Euclidean distance.

Algorithm 2 Modified k-means for web sessions clustering

Input: A set of web user sessions $WS = \{ S_1, S_2,..., S_n\}$, the desired number of k clusters, the maximum number of iterations itr;

Output: A set of clusters $C = \{C_1, C_2,..., C_k\}$ of WS

1: $count \leftarrow 0$
2: select any k sessions $\{S_1, S_2,..., S_k\}$ from WS and set $m_i \leftarrow S_i$ to get the initial center points $M = \{m_1, m_2, ..., m_k\}$ where, $0 < i < k + 1$
3: $newC \leftarrow empty, newM \leftarrow empty$
4: **loop**
5: for each S_i, compute $Dist$=$\{dist_1$,$dist_2$,...,$dist_k\}$ where $0 < i < n + 1$
6: assign S_i to C_j where $dist_j$=min($Dist$) and $0 < j < k + 1$
7: **for** each C_j **do**
8: $delta_j \leftarrow sum(distance\, of\, each\, S_i\, in\, C_j)/number\, of\, sessions\, in\, C_j$
9: **end for**
10: $newM \leftarrow (m_1 + delta_1, m_2 + delta_2, ..., m_k + delta_k)$
11: **if** $C = newC\, or\, M = newM\, or\, count > itr$ **then**
12: break
13: **end if**
14: $newC \leftarrow C, newM \leftarrow M$
15: $count \leftarrow coun + 1$
16: **end loop**

Comment on completeness and correctness of the modified k-means algorithm: The completeness of an algorithm can be shown by considering all possible cases, and correctness can be inferred by verifying the output for all possible inputs (Shyamsunder 1998).

Input assertion: ASSUME, (A set of web user sessions $WS = \{S_1, S_2, ..., S_n\}$, length of each $S_i > 1$, $k > 1$, $itr > 0$).

Output assertion: ACHIEVE, (A set of clusters C = $\{C_1, C_2, ..., C_k\}$)

Assertion for line 1: *assert* count=0.

Assertion for line 2: *select 'k' sessions and assign* S_i to M_i, assert $0 < i < k+1$.
Assertion for line 3: assert newC and newM, empty.
Assertion for line 4: loop invariant newC, newM, count. **Pre-condition:** assert count=0, newC and newM are empty, follows from assertion for line 1 and 3; **post-condition:** Achieve newC=C or newM=M or $count > itr$.
Assertion for line 5-6: compute $Dist = \{dist_1, dist_2, ..., dist_k\}, assert\, S_i$ belongs to C_j where, $dist_j = min(Dist)$, this follows from the execution of this statement.
Assertion for line 7-9: compute average distance delta for each cluster, follows from the execution of this statement.
Assertion for line 10: assert newM=oldM+delta.
Assertion for line 11-13: on first entry, no change in loop invariants. on subsequent entries, achieve newC=C or newM=M or $count > itr$.
Assertion for line 14-15: assert newC=C, newM=M, $count = count + 1$.
Assertion for line 16: assert $C = \{C_1, C_2, ..., C_k\}$

Thus this algorithm satisfies intended specifications and all paths starting from the input point to the output point satisfy the output assertion. Hence, we can conclude that the algorithm is complete and correct since, it is not possible for the algorithm to get stuck or reach a dead-end at any point and the output assertion achieved.

The next section discusses the proposed VLVD method. The VLVD method measures the distance between any two user sessions, that are of uneven lengths, to form clusters of user sessions based on the similarity of pages accessed.

3.4 Variable Length Vector Distance (VLVD)

In general, a web site consists of large number of web pages and a user may not navigate through all pages of a web site. Also, the number of pages visited may vary among different users. Using 0 or 1, to represent presence or absence of a page, makes vector longer. Furthermore, less attention was paid on handling unequal length of

user sessions effectively. To mention a few, Yang et al. (2007) used triplet consisting of URL, time and frequency to represent navigation of a user for each page, Giannotti et al. (2002) used fixed length boolean attribute to represent a session, Xing et al. (2004) used user-access matrix, Xu et al. (2010) used url-user matrix, Mojica et al. (2005) created a distance matrix from a co-occurrence matrix, Lu et al. (2005) used the concept of sub abstraction. All these measures work, but they do occupy extra space for representing sessions and hence the time taken to compute distance may be more. Thus the issue of variable length of web user session vectors is not addressed effectively. Also, other measures such as, Hamming distance (Hamming et al. 1950), Euclidean distance etc., require, objects to be represented as n-dimensional space. The proposed VLVD (S_i, S_j) method, tries to deal with the variable length sessions. It finds the distance or dissimilarity between any two sessions as given in Algorithm 3.

Algorithm 3 VLVD distance between two user sessions

Input: two web user sessions S_i and S_j
Output: distance d_{VLVD} between S_i and S_j
1: $l_1 \leftarrow number\ of\ pages\ in\ session\ S_i$
2: $l_2 \leftarrow number\ of\ pages\ in\ session\ S_j$
3: $C \leftarrow number\ of\ common\ pages\ accessed\ by\ sessions\ S_i\ and\ S_j$
4: $dist \leftarrow l_1 + l_2 - 2C$
5: $len \leftarrow l_1 + l_2$
6: $d_{VLVD}(S_i, S_j) \leftarrow dist/len$

The *dist* (line number 4) determines the number of unique pages between the two web sessions. d_{VLVD} (line number 6) is determined by taking the ratio of number of unique pages to the total number of pages for both sessions. Hence, the value of d_{VLVD} always lies in the range of 0 and 1. The value 1 indicates that the two sessions are completely different, where as 0 indicates that the sessions are exactly similar. Consider an example data set with 5 sessions, to illustrate the VLVD method.

Example:

S_1=$(P_1, P_2, P_3, P_4, P_5)$

S_2=(P_4, P_5)
S_3=(P_1, P_2, P_5)
S_4=(P_6, P_7)
S_5=$(P_1, P_2, P_3, P_4, P_5)$

The distance between session S_1 and other sessions is computed by using Algorithm 3 and is given below:
VLVD $(S_1, S_2) = 0.42$
VLVD $(S_1, S_3) = 0.25$
VLVD $(S_1, S_4) = 1.0$
VLVD $(S_1, S_5) = 0.0$

The example clearly shows that, the sessions S_1 and S_5 are similar whereas, S_1 and S_4 are entirely different. Hence, distance between S_1 and S_5 is 0, whereas, the distance between S_1 and S_4 is 1. S_3 is closer to S_1 compare to S_2 with respect to the pages accessed. Therefore the distance between S_3 to S_1 is less compare to the distance between S_1 and S_2. Thus, it is possible to measure the distance between the sessions efficiently, though they are variable length sessions.

3.4.1 Results and discussions

To implement the modified k-means for web sessions clustering with VLVD method, two data sets are considered. Table 3.2 summarizes the details of these two data sets.

3.4.2 Analysis of clusters - NASA data set

Fig. 3.1 shows the frequency of access to various page categories in various clusters of NASA data. "/history/apollo/", and "/shuttle/missions/" categories are viewed more frequently in cluster 1 compared to other categories, while cluster 2 concentrates

Table 3.2: Data sets for VLVD

Data set	Time period	File size	Number of sessions considered	Number of page categories
NASA	1/7/1995 to 31/7/1995	117,532 KB	5000	30
MSNBC	28/9/1999	12,287 KB	5000	17

on "/shuttle/missions/" category most of the times. In cluster 3 the category "/elv/" is viewed majority of the times, and 50% of frequency is to "/shuttle/missions/" category. The users in cluster 4 are more interested in "/shuttle/missions/" and "/history/apollo" categories. Similar to cluster 4, the frequency is more for categories "/shuttle/missions/" and "/history/apollo" in cluster 5, along with the category "/shuttle/countdown/", whereas, cluster 4 users are not interested in "/shuttle/countdown" because the frequency is zero for this category in cluster 4. It may look like the categories of cluster 1 and 4 are similar, but, the usage patterns of these two clusters are different. i.e., in cluster 1, "/history/apollo" is viewed more than "/shuttle/missions/" whereas it is vice versa in cluster 4. In cluster 4, around 40% of frequency is to "/history/apollo/". Overall, it is observed that, the most frequently visited category is "/shuttle/missions/" in this web site. Thus, the clusters formed show different patterns of usage in combination with "/shuttle/missions/" category.

3.4.3 Analysis of clusters - MSNBC data set

Fig. 3.2 shows the frequency of access to various page categories in various clusters. More than 60% of times, request is to "misc", "on-air" while 40% of times, for "weather" and "sports" categories in cluster 1. It shows that, users of this cluster show more interest in these categories. In cluster 2, the users visit "front page" followed by "news" and "local" categories majority of the times, indicating their interest in local information and news. The users in cluster 3 do not belong to any specific

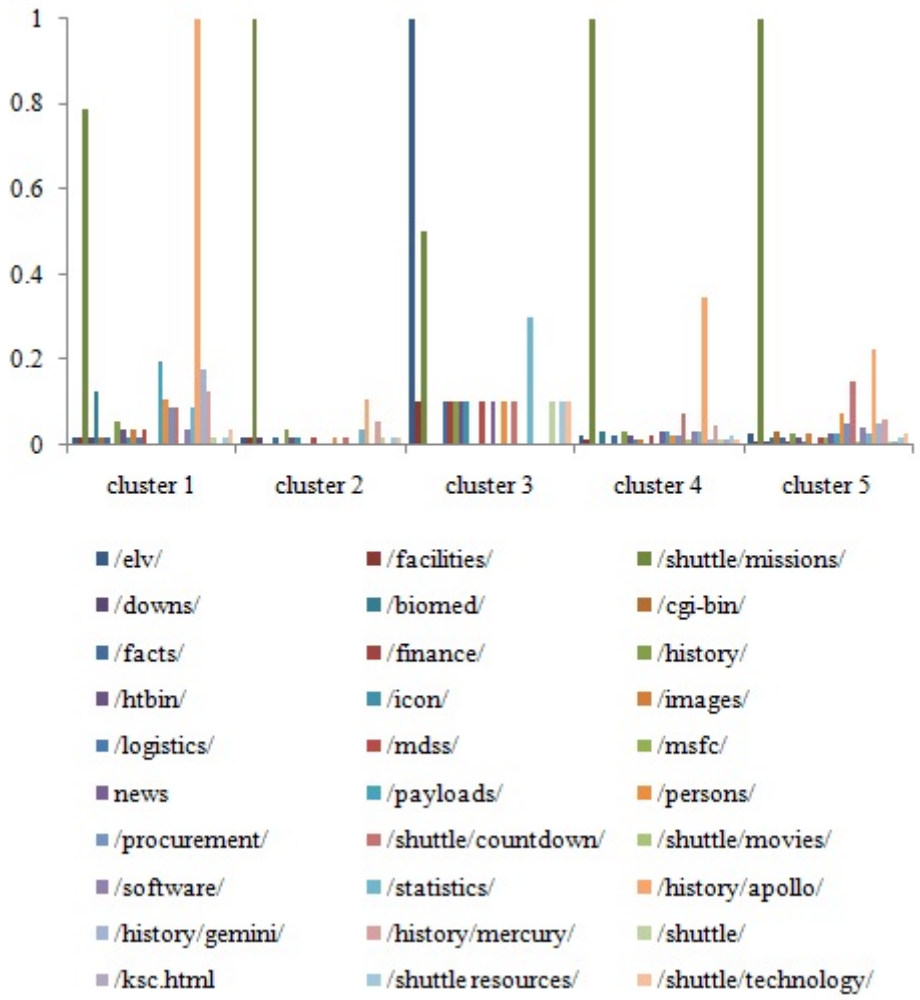

Figure 3.1: Normalised frequency of web page categories - NASA data set

categories. They visit "front page", and just visit other categories, while cluster 4 clearly shows more than 50% of times the visit is to "misc" category. In contrast, users in cluster 5 are more interested in "opinion" and subsequently in "on-air" and "summary" categories.

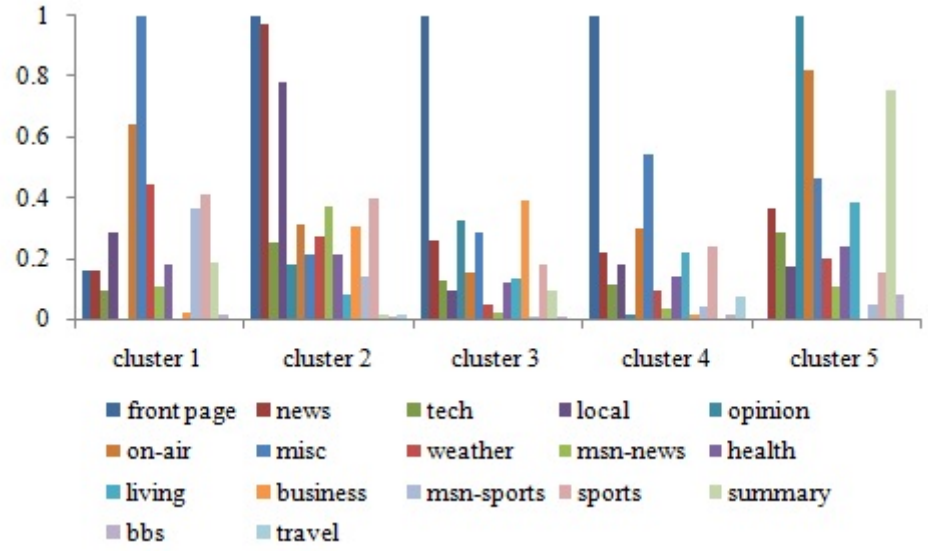

Figure 3.2: Normalised frequency of web page categories - MSNBC data set

3.4.4 Evaluation of clusters by using the VLVD as a distance measure

The graphs shown in Figs. 3.1 and 3.2, show the patterns obtained by the proposed method for the two data sets. The Jaccard index, also known as the Jaccard similarity coefficient is a statistic used for comparing the similarity and diversity of sample sets. In general a good clustering method should construct high quality clusters with low inter-cluster similarity. Here, Jaccard index is used to observe the similarity between clusters. The Jaccard index between any two clusters C_i and C_j is computed by using Eq. 3.4.1 and the major steps are given in Algorithm 4.

Algorithm 4 Jaccard index

Input: clusters of user sessions
Output: average jaccard index
1: find unique pages of each of the cluster
2: for each cluster find the Jaccard index with other clusters by using Eq.3.4.1
3: find the average Jaccard index

$$Jac(C_i, C_j) = \frac{\mid C_i \cap C_j \mid}{\mid C_i \cup C_j \mid} \tag{3.4.1}$$

If Jac (A, B) is equal to 1, it indicates that, the samples A and B are exactly similar. In our example, to compare the five clusters that were formed for the NASA data sets, Eq.3.4.1 is used and the average value for each cluster is less than 0.3 as shown in Table 3.3. This indicates that, the clusters obtained are not exactly the same and hence the distance between the clusters is more across all the clusters. Thus, it could be inferred that the clustering done is reasonably good. Similar analysis could be done on the clusters of MSNBC data set provided we get the data regarding the actual pages of the site in each category along with the main page categories. Due to the unavailability of details regarding the pages, the Jaccard index is not applied to the clusters obtained for the MSNBC data set. However, the analysis done on the NASA data set proves the goodness of the proposed clustering method and VLVD method that finds the distance between any two user sessions.

Table 3.3: Jaccard index by using VLVD as a distance measure to cluster sessions

	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
Average Jaccard Index	0.25	0.23	0.12	0.28	0.28

The modified k-means algorithm, clusters the web sessions based on VLVD method and considers the issue of variable length sessions. But, the order of page visits is not considered. For example, the sessions (P_1, P_2, P_3) and (P_1, P_3, P_2) are considered as same. Whether the page P_3 is visited before P_2 or after P_2 is not given importance and both sessions are considered as exactly similar based on the pages accessed, irrespective of the order in which these pages are accessed. But, for certain applications like web page prediction, caching and pre-fetching, retaining the order information become very essential. Based on the navigation pattern of usage, web page prediction/caching/pre-fetching can be done and hence the next section discusses the proposed sequence alignment based distance measure that tries to deal with the order in which web pages are accessed by the user.

3.5 Sequence Alignment Based Distance Measure (SABDM)

This section presents an algorithm that finds the distance between any two web sessions by taking sequence alignment as a similarity measure since, less attention was paid on handling unequal length of sessions effectively. The sequence information of navigation order of pages viewed by user is retained and the sessions need not be of same length. Further, retaining the information of sequence in which pages are accessed in a session is useful for web page prediction, caching, prefetching etc.

3.5.1 Sequence alignment

The sequence alignment method is used for aligning DNA and protein sequences. Each DNA sequence contains a sequence of amino acids. It is possible to determine whether significant homology exists between the proteins by finding similarities in the amino acid sequences of two proteins. Short sequences can be aligned manually but, the problems in general require the alignment of lengthy sequences that cannot be aligned by human effort. Therefore, computational approaches to sequence alignment are essential that fall into two categories: global alignments and local alignments. One sequence is transformed into the other in global alignment, whereas, in local alignment, all locations of similarity between the two sequences are returned (Brudno 2003).

Local alignments are more useful for dissimilar sequences that are suspected to contain regions of similarity within their larger sequence context. The Smith-Waterman algorithm (Smith 1981) is a general local alignment method and both global as well as local alignment methods are based on dynamic programming. Global alignments, which attempt to align every residue in every sequence, are most useful when the sequences in the query set are similar and of roughly equal size. The

Needleman-Wunsch algorithm (Needleman et al. 1970) is a general global alignment technique.

In general, a session consists of sequence of web pages accessed by user. Therefore, the problem of computing the similarity between web sessions, is converted to find the best matching between two web page access sessions, which consist of a sequence of web pages. Thus the techniques used in DNA/protein alignment, can be employed to find the alignments between any two users' web access sessions. Based on these alignments the distance between pair of web page session may be obtained. Further, clustering of sessions may be formed based on this distance.

3.5.2 Sequence alignment method (SAM)

This section deals with the sequence alignment method (Hay et al. 2004) in brief. The SAM method partitions the navigation patterns based on the order of web pages requested by the user. Here identical pages of a pair of session are reordered to make them align with each other and unique pages are either inserted or deleted to make both sessions exactly similar to each other. The number of insertion operations, deletion operations and reorder operations are used to compute the distance between pair of sessions. Because of reorder operation the sequence in which the user accessed web pages in a session is altered. The equation used in SAM method to find the distance between two sessions is given by Eq.3.5.1.

$$d_{SAM}(S_1, S_2) = (w_d D + w_i I) + \eta R \tag{3.5.1}$$

where:

- $d_{SAM}$is the distance between two sessions S_1 and S_2, based on SAM
- w_d is the weight value for the deletion operations, a positive constant not equal to zero, determined by the researcher

- w_i is the weight value for the insertion operations, a positive constant not equal to zero, determined by the researcher
- D is the number of deletion operations
- I is the number of insertion operations
- R is the number of reordering operations
- η is the reordering weight, a positive constant not equal to zero, determined by the researcher

Eq.3.5.1 indicates that, the distance between two sessions consists of the costs for deletion and insertion of unique elements and the costs for reordering common elements. If the number of common pages viewed in both sessions is more but the order of page views is different, considerable amount of computation work is needed to find the number of common elements and reorder them, and the reorder operation alters the original session.

To illustrate the SAM distance, consider the sessions S_1=($P_1, P_2, P_3, P_4, P_5, P_6$) and S_2=($P_1, P_3, P_4, P_2, P_5, P_2$). Assume w_d=w_i=1 and η=w_d+w_i. The common pages between S_1 and S_2 are P_1, P_2, P_3, P_4 and P_5. The unique page between them is P_6. The page P_2 should be reordered from 4th position to the 2nd position in S_2. After this reorder operation the common pages are P_1, P_2, P_3, P_4 and P_5. But, the number of unique pages is increased to two with pages P_6 and P_2. The page P_2 at the 6th position of session S_2 is considered unique after reordering since all other pages are identical to each other till 5th position of both sessions. The option available to make both sessions equal is by either insertion of P_6 in S_2 and deletion of P_2 from S_2, or deletion P_6 from S_1 and insertion of P_2 in S_1. Thus one insertion and one deletion operation are required to make both sessions exactly similar. Thus the SAM distance between S_1 and S_2 is computed as 4 by using the Eq.3.5.1.

3.5.3 SABDM algorithm

When any two sessions are considered, individual pair wise comparisons between two sessions are required in order to get the distance between them. Here, two pages are said to be a match, if they are same. The mismatch results, if the pages are different, indicating that either insertion/deletion/substitution operation is required to make both sessions exactly similar to each other. It is required to assign suitable scores for match and mismatch. Since, the intention here is to find the number of pair wise alignments between two sessions, the insertion/deletion/substitution operations is considered as same (mismatch). The score is assigned for both match and mismatch suitably say, match=2, mismatch=-1. By assigning score for match and mismatch, the comparison of each pair of pages is weighted into a matrix called 'Score' matrix. The distance between two web user sessions is computed based on the number of alignments obtained for pages in two sessions that are compared. The major steps of the proposed algorithm that finds the distance between any two users' web sessions are given by Algorithm 5.

Consider a simple example to find the distance between two sessions S_1 and S_2 by applying the sequence alignment based distance measure algorithm, to understand the proposed method of finding distance between any two web user sessions.

Let S_1= $(P_1, P_2, P_3, P_4, P_5)$ and S_2 = $(P_1, P_3, P_4, P_2, P_5)$ be two sessions to be aligned to compute the distance between them. Let m and n be the length of session S_1 and S_2 respectively. In this example the length of both S_1 and S_2 are same. Therefore, m=5 and n=5. The 'Score' matrix is constructed as given in Table 3.4 with size (m+1, n+1). The score values considered for match and mismatch are 2 and -1 respectively because, matches should be rewarded and mismatches should be penalized. Initialize the first row and first column of the score matrix with the value -1 as per the requirements to use the concept of the dynamic programming and also fill rest of the cells with the value of either match or mismatch according to the comparison made for pair of pages considered at a time. i.e., enter value of match

in the cells of 'Score' matrix whenever $S_1(i)$ is equal to $S_2(j)$ and set the cell value as mismatch if $S_1(i)$ is not equal to $S_2(j)$ for all i from 1 to m+1 and for all j from 1 to n+1. Construct a distance matrix, 'Dist' with size (m+1, n+1). Compute the distance matrix values as shown in Table 3.5 based on equations given in line number 22, 26 and 30 of Algorithm 5.

Table 3.4: Score matrix

	j	P_1	P_3	P_4	P_2	P_5
i	0	-1	-1	-1	-1	-1
P_1	-1	2	-1	-1	-1	-1
P_2	-1	-1	-1	-1	2	-1
P_3	-1	-1	2	-1	-1	-1
P_4	-1	-1	-1	2	-1	-1
P_5	-1	-1	-1	-1	-1	2

Table 3.5: Distance matrix

	j	P_1	P_3	P_4	P_2	P_5
i	0	-1	-2	-3	-4	-5
P_1	-1	2	1	0	0	0
P_2	-2	1	1	0	2	1
P_3	-3	0	3	2	1	1
P_4	-4	0	2	5	4	3
P_5	-5	0	1	4	4	6

Table 3.6: Pointer matrix

	j	P_1	P_3	P_4	P_2	P_5
i	0	0	0	0	0	0
P_1	0	3	2	2	0	0
P_2	0	1	3	23	3	2
P_3	0	1	3	2	12	3
P_4	0	0	1	3	2	2
P_5	0	0	1	-1	3	3

Algorithm 5 Sequence alignment based distance measure - SABDM

Input: sessions S_1=($pr_1, pr_2, ..., pr_m$) and S_2=($pc_1, pc_2, ..., pc_n$)
Output: distance d between S_1 and S_2

$match \leftarrow 2$
$mismatch \leftarrow -1$
$similaritycount \leftarrow 0$
for $i = 1$ to $m + 1$ **do**
  $Score(i, 0) \leftarrow mismatch$
end for
for $j = 1$ to $n + 1$ **do**
  $Score(0, j) \leftarrow mismatch$
end for
for $i = 0$ to m **do**
  for $j = 0$ to n **do**
    if $P_i = P_j$ **then**
      $Score(i, j) \leftarrow match$
    else
      $Score(i, j) \leftarrow mismatch$
    end if
  end for
end for
construct distance matrix $Dist$ and pointer matrix $Pointer$ of size (m+1, n+1)and compute entries as given below:
for $i = 1$ to $m + 1$ **do**
  $Pointer(0, i) \leftarrow 0$
  $Dist(0, i) \leftarrow Dist(0, i - 1) + mismatch$
end for
for $j = 1$ to $n + 1$ **do**
  $Pointer(j, 0) \leftarrow 0$
  $Dist(j, 0) = Dist(j - 1, 0) + mismatch$
end for
for $i = 1$ to m **do**
  for $j = 1$ to n **do**

$$Dist(i, j) = max \begin{cases} 0 \\ Dist(i - 1, j) + mismatch \\ Dist(i, j - 1) + mismatch \\ Dist(i - 1, j - 1) + Score(i, j) \end{cases}$$

  end for
end for

33: the value 0 is included to ignore possible negative alignment score. The second and third terms handle an extension of alignment by inserting a gap for insertion/deletion/substitution operation. Finally the fourth term considers an extension of the alignment by extending the two sequences of sessions compared by one page each. Store the pointer value as either top/left/lefttop/ combination of top, left and left-top in $Pointer(i, j)$ depending upon $Dist(i, j)$ where, $0 < i <= m + 1,\ \ 0 < i <= n + 1$

34: trace the distance matrix back, by finding the position of cell with maximum value, check for match or mismatch from Score matrix. Use the Pointer matrix to move to the next location. Whenever match is found increment the *similaritycount*

35: repeat the tracing process till a cell with value zero is encountered in *Dist* matrix

36: find the normalized distance between S_1 and S_2 as given below:

37: $d_{SABDM}(S_1, S_2) \leftarrow (max(m, n) - similaritycount)/max(m, n)$

Construct a 'Pointer' matrix of size (m+1, n+1) to store the positions of cells from which a value is obtained in 'Dist' matrix so that, these pointers can be used to trace back the distance matrix at a later stage. Table 3.6 shows the pointer matrix for the example considered. Here value 1 indicates link to top cell, value 2 indicates link to left cell and value 3 indicates link to left-top cell.

Trace the 'Dist' matrix back, by finding the position of cell with maximum value. In this example, the maximum value is 6, present at the cell with position (5, 5) in the 'Dist' matrix. Check for match or mismatch by referring to the 'Score' matrix. Increment the similarity Count value by 1 if match is found. Move to the next location according to the links stored in the 'Pointer' matrix and repeat the process till the value 0 is obtained in 'Dist' matrix or (0,0) cell is reached or first row/first column of 'Dist' matrix is reached. Finally, compute the normalized distance value between S_1 and S_2 by using equation given in line number 37 of Algorithm 5 which gives the distance as 0.2.

The distance value always lies in the range of 0 and 1. The value 1 indicates that the two sequences are entirely different and the value 0 indicates that the two sequences are exactly similar to each other. As the distance becomes closer to 0, it means that, the sequences are closer and as the sequence becomes different the

distance value tends to be closer to 1 than 0. Since the distance value 1 indicates that the two sessions are completely different, such session can be considered as outlier and need not be included to the cluster while performing the clustering process. Thus the proposed SABDM distance measure can also be useful in finding the outliers, if any, when forming clusters.

3.5.4 Results and comparison

The SABDM algorithm uses dynamic programming. In general, dynamic programming is employed to search efficiently and also to optimize the problems with overlapping sub problems. The SABDM uses the dynamic programming to find the alignments between any two sessions without reordering the common pages. Also, SABDM clearly differentiates between two sessions that are exactly similar and sessions that are completely different. The SAM reorders the common pages to align them which in turn alters the order in which the pages are viewed in the actual session. The SAM distance just gives distance value for pair of sessions that are compared but, fails to indicate the difference between the sessions that are completely different and the sessions that have some similarity with respect to the order in which the pages are visited. For example, consider the sessions S_1=(P_1, P_2, P_3, P_4) and S_2=(P_4, P_5, P_1, P_6). The SABDM distance between S_1 and S_2 is 1. This indicates, S_1 and S_2 are completely different because the order in which the pages are viewed are different though the pages P_1 and P_4 are common among them. The SAM reorders P_1 and P_4 and considers number of reorder, insertion and deletion operation to give distance value between S_1 and S_2 and it fails to convey the information that the session S_1 and S_2 are completely different in terms of the pages viewed in succession. Thus by looking at the distance value of SAM, it is not possible to differentiate between the sessions that are completely different and sessions that have some common pages with respect to sequence information between them.

To illustrate the proposed SABDM measure and compare with the SAM (Hay et al. 2004) method, consider the following cases:

Case 1: two sessions of same length
Session S_1: $(P_1, P_2, P_{45}, P_{27}, P_{28}, P_{112})$
Session S_2: $(P_1, P_{45}, P_{27}, P_2, P_{28}, P_2)$
The distance between session 1 and 2 is 4 in SAM since 4 operations (insertion, deletion, reordering) are performed whereas, the distance is 0.33 by using the SABDM method because, here out of six pages of session S_1, four pages are aligned with sequence of session S_2. The pages P_2 and P_{112} are not aligned and importance is given to the number of alignments made. The SAM method considers number of operations to be carried out to make both sessions equal as the distance, whereas the SABDM method considers the number of alignments obtained determining the distance. If Needleman Wunsch (Needleman et al. 1970) global alignment is used, the number of alignments made will be only two and the normalized distance will be 0.66.

Case 2: two sessions of dissimilar lengths
Session S_1: (P_1, P_3, P_4, P_2)
Session S_2: (P_1, P_2, P_3)
The global alignment finds only one alignment for the first position, where as the proposed method finds two alignments for the pages P_1 and P_2 by considering P_3 and P_4 as mismatch. Therefore the distance between these two sessions are 0.75 in global alignment and 0.5 in the SABDM method and the SAM method considers three operations (reordering of P_2 in session S_1 and insertion of P_4 in session S_2) and the distance is 3.

Thus it is very clear that, the proposed method of distance measure is better than global alignment as well as the SAM method in finding the number of alignments between any two sessions. The results obtained by considering various sessions are

compared with the Needleman-Wunsch (NW) global sequence alignment method and the SAM method. Table 3.7 illustrates the outcome of the proposed SABDM method and also gives the comparison with global alignment and SAM methods.

Table 3.7: Comparison of SABDM distance with NW and SAM method

No	Session S_1	Session S_2	NW method	SAM method	SABDM method
1	$(P_1, P_2, P_3, P_4, P_5)$	$(P_1, P_3, P_4, P_2, P_5)$	0.6	2	0.2
2	$(P_1, P_2, P_3, P_6, P_7, P_8)$	$(P_1, P_6, P_7, P_4, P_5, P_9)$	0.83	6	0.67
3	(P_1, P_2, P_3, P_4)	(P_1, P_5, P_4)	0.5	5	0.5
4	(P_1, P_3, P_4, P_2)	(P_1, P_2, P_3)	0.74	3	0.5
5	$(P_1, P_2, P_5, P_6, P_3, P_1, P_2, P_5, P_4, P_8, P_7, P_2)$	$(P_1, P_2, P_7, P_7, P_5, P_4, P_1, P_2, P_6, P_5, P_8, P_7, P_3)$	0.53	19	0.38
6	(P_1, P_3, P_4, P_2)	(P_1, P_3, P_2)	0.25	3	0.25

Compared to NW method, the proposed method finds more alignments between sequences as it tries to find regions of similarities within sequences instead of aligning every residue of sequence. Therefore, the distance value obtained will be lesser than NW method. For example, if we look at the case of second row of Table 3.7, NW method aligns only page P_1. If both sessions S_1 and S_2 are observed, it is clear that after visiting page P_1, user visits pages P_6 and P_7 in both sessions. Only difference is that, in S_1, pages P_2 and P_3 are visited before going to P_6, P_7 whereas in S_2, immediately after visiting P_1, user accesses P_6 and P_7. The reason could be, there may be some correlation between pages P_1, P_6 and P_7. This information is not captured by NW method, but the proposed algorithm incorporates such information as well. Therefore, the number of alignments found is more in the SABDM method compared to NW method. Since the SAM method finds the distance between pair of sessions based on the number of insertion, deletion and reordering operations, the distance obtained will be more compared to the proposed SABDM method. Thus Table 3.7 illustrates the goodness of the proposed technique compared to the NW technique of sequence alignment as well as the SAM method.

From Table 3.7, it is very clear that, the proposed method yields good measure of

distance values compared to others. As the number of alignments found will be more in the proposed SABDM method, the distance between two sessions will also become less whenever more correlations are present between pair of sessions considered, compared to other two methods. Thus by retaining the information of navigation order along with considering the issue of unequal length of sessions, the proposed SABDM method finds the distance between two sessions effectively.

Thus, the proposed SABDM method finds the distance between user sessions that considers the issue of the uneven lengths of sessions and also retains the order of pages visited by the user. Since order of navigation path is considered while clustering, it is further useful for applications like web page prediction, caching and pre-fetching. The next section discusses the impact of SABDM method for clustering user sessions for various number of clusters and also analyzes the goodness of the clustering done by using SABDM as a distance measure that compares two user sessions.

3.5.5 SABDM clustering

The clusters are formed by using the Modified k-means algorithm given in (2011). This algorithm partitions the web sessions into disjoint set of clusters. The SABDM method is used to find the distance between pair of sessions which is based on the order in which web pages is viewed by user. So, the clusters are disjoint with respect to sessions and not web pages. There is a possibility of presence of some web pages like front page or any other frequently used pages by the users, in more than one cluster. For example, the pattern $(P_1, P_2, P_3, P_4, P_5, P_6)$ and $(P_1, P_2, P_7, P_8, P_9, P_{10})$ is different and hence they may be in different clusters. Therefore pages P_1 and P_2 will be present in both clusters whereas the sessions belong to any one of the clusters. So, the cluster contains disjoint set of user sessions but not the pages. The modified k-means algorithm (2011) that uses the SABDM is given in Algorithm 6.

Algorithm 6 Modified k-means for web session clustering - SABDM

Input: A set of web user sessions $WS = \{ S_1, S_2,..., S_n\}$, the desired number of k clusters, the maximum number of iterations itr;

Output: A set of clusters $C = \{C_1, C_2,..., C_k\}$ of WS

1: $count \leftarrow 0$
2: select any k sessions $\{S_1, S_2,..., S_k\}$ from WS and set $m_i \leftarrow S_i$ to get the initial center points $M = \{m_1, m_2, ..., m_k\}$ where, $0 < i < k + 1$
3: $newC \leftarrow empty, newM \leftarrow empty$
4: **loop**
5: for each S_i, compute D=$\{d_1,d_2,...,d_k\}$ where, $d_i = d_{SABDM}$ (S_i,m_j) and $0 < j < k + 1$, $0 < i < n + 1$
6: assign S_i to C_j where d_j=min(D) and $0 < j < k + 1$
7: **for** each C_j **do**
8: $delta_j \leftarrow sum(distances\, of\, each\, S_i\, in\, C_j)/number\, of\, sessions\, in\, C_j$
9: **end for**
10: $newM \leftarrow (m_1 + delta_1, m_2 + delta_2, ..., m_k + delta_k)$
11: **if** $C = newC$ or $M = newM$ or $count > itr$ **then**
12: break
13: **end if**
14: $newC \leftarrow C, newM \leftarrow M$
15: $count \leftarrow coun + 1$
16: **end loop**

3.5.6 Cluster validation

Since any data can be clustered, it is important to know that the clusters formed are meaningful. In general, clustering validation can be categorized into two classes, external clustering validation and internal clustering validation. External validation measure uses external information not present in the data whereas, internal validation measures rely on information in the data and measures the goodness of clustering without respect to external information (Liu et al. 2010). Since web log does not provide any external information, internal validation of clustering is carried out for the clustering by the proposed method of clustering based on sequence alignment. 5000 sessions are considered from the NASA and MSNBC data set for validation purpose.

The clusters formed by this distance measure are validated by using two different

statistical methods namely, R^2 and χ^2 measure. The R^2 is used to decide the number of clusters to be formed and the χ^2 is used to measure the goodness of the clustering done.

R^2 Validation R^2 is one of the widely used internal validation measures. R^2 is a statistical measure of how well a regression line approximates real data points. R^2 is the ratio of sum of squares between clusters to the total sum of squares of the whole data set and is given by the equation 3.5.2. It is a descriptive measure between zero and one. Given a model, the closer its R^2 value is to one, the greater the reliability of a relationship identified by regression analysis.

$$R^2 = 1 - \frac{ESS}{TSS} \tag{3.5.2}$$

where,

- ESS is the expected sum of squares
- TSS is the total sum of squares

The steps used to calculate R^2 is as given in Algorithm 7.

R^2 value depends on how it is measured as well as the kind of data and application considered. Even R^2 of 10% or 5% may be statistically significant in some applications (Fuqua School of Business year 2005). If application is to predict human behavior which is not the working of physical systems, R^2 of 40% is also excellent (Kevinmacdonell 2010). Since the clusters of web page sessions is used to analyze the navigation pattern of users in a web site, 50% and above is considered as good to decide the number of clusters to be formed in the proposed method. The number of clusters to be formed for NASA data set is 26 and MSNBC data set is 29. Thus using R^2 measure, the number of clusters to be formed is decided for two data sets considered for clustering and the graph is different for different data sets indicating that, R^2 varies for different data sets.

Algorithm 7 R^2 for deciding number of clusters

Input: user sessions
Output: R-squared values

1: consider the whole data set as one cluster and make each session as the cluster center
2: for each such center, find the sum of squared distance of all sessions to the center session
3: find the maximum sum of squared distance to get the TSS
4: for each cluster i from 1 to 100
   - find the sum of squared distances from the sessions to their respective cluster center
   - add all these sums to get ESS
   - calculate the R^2 by using the equation 3.5.2
5: decide the number of clusters to be formed when R^2 reaches 0.6 or higher

Fig.3.3 shows the graph of R^2 values for various numbers of clusters ranging from 1 to 100 for the NASA data set. The figure shows that, as the number of clusters increases, the R^2 increases up to some threshold. In this case at cluster number 26 the R^2 value reaches 0.6, and beyond this point, the increase in R^2 is only in small quantities compared to the R^2 values before this critical point. Therefore, it can be inferred that, the number of clusters to be formed for this data set is 26. Similarly the number of clusters for MSNBC data set is 29 as can be seen from the graph given in Fig.3.4. Thus using R^2 measure, the number of clusters to be formed is decided for both data sets considered for clustering and the graph is different for different data sets indicating that, R^2 varies for different data sets.

Chi-squared (χ^2) Validation In general, the χ^2 is a hypothesis test carried out to see the association between categorical variables. The χ^2 test is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. The χ^2 value is calculated by using Eq.3.5.3.

$$\chi^2 = \Sigma \frac{(O - E)^2}{E} \tag{3.5.3}$$

Figure 3.3: R^2 values for various number of clusters for NASA data set

χ^2 hypothesis test steps:

1. State null hypothesis H_0 and alternate hypothesis H_1
2. Create a contingency table of clusters and frequencies of different page categories for each cluster and record the observed frequencies (O) in each cell
3. Calculate row, column and grand total
4. Calculate expected frequency (E) for each cell, by using Eq.3.5.4

$$E = \frac{row\_total \times column\_total}{grand\_total} \tag{3.5.4}$$

5. Find critical value from χ^2 table, as appended with (row-1)(column-1) degrees of freedom
6. Use Eq.3.5.3 to calculate χ^2 value
7. If the χ^2 value obtained in step 6 is equal or greater than the value noted in step 5, reject the null hypothesis and accept the alternate hypothesis

Figure 3.4: R^2 values for various number of clusters for MSNBC data set

Table 3.8 shows the results obtained by applying χ^2 test for NASA and MSNBC data sets. The null and alternate hypothesis is:

H_0: there is no association between the page categories and clusters

H_1: there is an association between the page categories and clusters

Table 3.8: Chi-squared validation

Data set	Level of significance / Degree of freedom			χ^2 obtained
	0.05	0.01	0.001	
NASA	788.8	816.5	848.4	1443
MSNBC	498.3	520.5	546.2	20290

The chi-square value obtained is much higher than the critical value. Therefore, the null hypothesis H_0 is rejected which says that, there is no correlation between the page categories and the clusters. The hypothesis H_1 is accepted, which indicates the goodness of the clustering done by the proposed method of sequence based alignment distance measure by using the modified k-means algorithm.

3.5.7 Navigation patterns

By observing the frequency of page categories in each cluster, the user navigation patterns are observed. Some of the patterns for the NASA data set are given below:
{/shuttle/missions/, /history/apollo/, /shuttle/countdown/}
{/shuttle/missions/,/history/apollo/,/shuttle/countdown/, /biomed/}
{/elv/, /shuttle/missions/, /history/apollo/}
{/shuttle/missions/, /payloads/, /software/}
{/history/apollo/, /shuttle/missions/,/history/mercury/}
{/shuttle/missions/, /history/mercury/, /history/gemini/}

Similarly few of the patterns obtained from the clusters of the MSNBC data set observed are as follows:
{front page, news, sports}
{front page, misc, on-air}
{on-air, tech, msn-news}
{local, misc, front page}
{front page, business}

In Hay et al. (2004) the number of clusters formed is three, and in Pallis et al. (2007) the number of clusters is five. Though the data set and the clustering algorithm used are different, compared to Hay et al. (2004) and Pallis et al. (2007), when the number of sessions is more, it may not be possible to capture all the patterns of user navigation with only three or five clusters. Since human behavior pattern is dynamic in nature and because of the availability of various page categories with many pages in each categories, a wide variety of patterns is expected. Chances of obtaining majority of the sequential patterns is more, when the number of clusters are also reasonably more. In the present work, by using R^2 measure, the number of clusters is decided and hence possible to get a variety of navigational patterns as depicted by some of

the patterns given above, that are based on frequently accessed navigation paths in different clusters.

3.6 Hybrid Sequence Alignment Measure (HSAM)

The SABDM algorithm considers only the number of alignments between pair of sessions. For example, consider S_1=(P_1,P_2,P_3,P_4) and S_2= (P_1,P_7,P_2,P_3,P_5). The SABDM distance between S_1 and S_2 is 0.4. Suppose if one more page say 'P_8' is added to the end of the first session, the SABDM distance will be again 0.4 between S_1=(P_1,P_2,P_3,P_4,P_8) and S_2=(P_1,P_7,P_2,P_3,P_5). The distance is same for both the cases because, the pages that are not aligned or the gaps inserted while finding the alignments are not taken into consideration for the SABDM distance measure. Though S_2 is same in both cases and S_1 is slightly different, the distance value obtained is same for both cases because only the number of alignments is given importance in SABDM distance. Also, the direct alignment that exists between two sessions and the alignment obtained by inserting gaps are not differentiated in the earlier work. To find the direct alignments between pair of sessions the dynamic algorithm may not be required. They can be obtained by simple comparison of each page of two sessions. However, the alignments found by inserting gaps are due to the alignment method used. Hence, direct alignments and alignments found by the dynamic method are differentiated in the proposed HSAM algorithm.

In the SAM distance, the original user sessions are changed by the reorder operation while measuring the distance. As a result, considerable amount of computation is required for reorder operation, if a pair of sessions has more number of common pages, but the order in which they are viewed by the user differs. In addition, the reorder operation modifies the actual session which is not desirable. So, both SABDM and SAM are combined to get a better distance measure.

The reorder operation of SAM is not considered in HSAM but the concept of

identifying the unique pages is used. Thus, unique pages are found without reorder operation HSAM. The method of finding the alignments by SABDM is retained. Further, the direct alignments that exist between pair of sessions are differentiated from the alignments found by inserting gaps in HSAM while both alignments are considered the same in SABDM. Thus the equation given in line number 36 of Algorithm 5 of SABDM and Eq.3.5.1 of SAM are combined to get a better measure to determine the distance between pair of sessions as given in Eq.3.6.1. As a result, the hybrid approach finds the number of unique elements as in SAM and finds the number of alignments as in SABDM without changing the order in which the pages are viewed by the user.

$$HSAM_{dist}(S_1, S_2) = \frac{NUP + [2 \times (|NAP - NDA|)]}{|S_1| + |S_2|} \tag{3.6.1}$$

where:

- $HSAM_{dist}(S_1, S_2)$ is the distance between user sessions S_1 and S_2
- NUP (Number of Unaligned Pages) refers to the unique web pages of a pair of sessions that cannot be aligned by inserting gaps. It is calculated by adding the number of pages present only in the first session that are not present in the second session and the number of pages present only in the second session that are not present in the first session. For example, if S_1=(P_1, P_2, P_3, P_4) and S_2= (P_1, P_2, P_3, P_5) then P_4 and P_5 are unique and hence NUP is two; NUP is same as $(w_d D + w_i I)$ of SAM; w_d, w_i are weights for deletion and insertion operations indicated by D and I respectively. Since deletion is one operation and insertion is one operation, both are assigned a value 1.
- NAP (Number of Aligned Pages) is the number of alignments found by employing local alignment method used in SABDM. NAP refers to the pages that could be aligned by inserting gaps. For example, if S_1=$(P_1, P_2, P_3, P_6, P_7, P_8)$

and S_2=(P_1, P_6, P_7, P_5, P_4) then, by inserting two gaps after P_1 in S_2, P_6 and P_7 could be aligned to P_6 and P_7 of S_1. Thus NAP is two.

- NDA (Number of Direct Alignments) refers to the pages that are aligned in the original pair of sessions. For example, if S_1=($P_1, P_2, P_3, P_6, P_7, P_8$) and S_2=(P_1, P_5, P_7, P_6, P_4) then, NDA is two. Here, pages P_1 and P_6 are aligned between S_1 and S_2. The page P_1 is present in the first position where as the page P_6 is at the fourth position of both the sessions.
- $|S_1|$ is the length of session S_1
- $|S_2|$ is the length of session S_2
- $|NAP - NDA|$ gives the actual number of pages that could be aligned by inserting gaps. This is multiplied by a constant 2 because two operations are performed while inserting a gap. i.e., inserting a gap is one operation and moving the existing page/s where gap is inserted is another operation.

Consider the example given below, to demonstrate the HSAM. Let S_1 and S_2 be two user sessions that represent the navigation path followed:

$S_1 = (P_1, P_2, P_5, P_6, P_3, P_1, P_2, P_5, P_4, P_8, P_7, P_2)$
$S_2 = (P_1, P_2, P_7, P_7, P_5, P_4, P_1, P_2, P_6, P_5, P_8, P_7, P_3)$

The alignments found between S_1 and S_2 are :

P_1 P_2 - - P_5 P_6 P_3 P_1 P_2 - P_5 P_4 P_8 P_7 P_2
| | | | | | | |
P_1 P_2 P_7 P_7 P_5 P_4 - P_1 P_2 P_6 P_5 - P_8 P_7 P_3

Algorithm 8 Hybrid sequence alignment measure - HSAM

Input: any two user sessions S_1 and S_2 of length m and n respectively
Output: distance between the sessions S_1 and S_2

assign scores for variables *match* and *mismatch* suitably. assume $match \leftarrow 2$ and $mismatch \leftarrow -1$. Initialize variables NAP and NDA to zero
determine NUP by counting the number of unique web pages between S_1 and S_2
construct $Score$ matrix of size $(m+1, n+1)$ and initialize as follows:
for $i = 1$ to $m+1$ **do**
  $Score(i, 0) \leftarrow mismatch$
end for
for $j = 1$ to $n+1$ **do**
  $Score(0, j) \leftarrow mismatch$
end for
for $i = 0$ to m **do**
  for $j = 0$ to n **do**
    if $P_i = P_j$ **then**
      $Score(i, j) \leftarrow match$
    else
      $Score(i, j) \leftarrow mismatch$
    end if
    if $i = j$ and $P_i = P_j$ **then**
      $NDA \leftarrow NDA + 1$
    end if
  end for
end for
construct distance matrix $Dist$ and pointer matrix $Pointer$ of size (m+1, n+1)and compute entries as given below:
for $i = 1$ to $m+1$ **do**
  $Pointer(0, i) \leftarrow 0$
  $Dist(0, i) \leftarrow Dist(0, i-1) + mismatch$
end for
for $j = 1$ to $n+1$ **do**
  $Pointer(j, 0) \leftarrow 0$
  $Dist(j, 0) = Dist(j-1, 0) + mismatch$
end for
for $i = 1$ to m **do**
  for $j = 1$ to n **do**

$$Dist(i, j) = max \begin{cases} 0 \\ Dist(i-1, j) + mismatch \\ Dist(i, j-1) + mismatch \\ Dist(i-1, j-1) + Score(i, j) \end{cases}$$

  end for
end for

36: store the pointer value as either top or left or left-top or combination of top, left and left-top in $Pointer(i, j)$ depending upon $Dist(i, j)$ where, $0 < i \leq m + 1$, $0 < i \leq n + 1$
37: trace the distance matrix back, by finding the position of cell with maximum value, check for match or mismatch from *Score* matrix. Use the *Pointer* matrix to move to the next location. Whenever match is found increment NAP if $i \neq j$.
38: repeat the tracing process till a cell with value zero is encountered in *Dist* matrix
39: find the normalized distance between S_1 and S_2 as given below:
40: **if** $m = n$ and $NDA = m$ **then**
41: $HSAM_{dist}(S_1, S_2) \leftarrow 0$
42: **else if** $NAP = 0$ and $NDA > 0$ **then**
43: $HSAM_{dist}(S_1, S_2) \leftarrow NUP/(|S_1| + |S_2|)$
44: **else**
45: $HSAM_{dist}(S_1, S_2) \leftarrow NUP + 2 \times (|NAP - NDA|)/(|S_1| + |S_2|)$
46: **end if**

Here '-' indicates a gap and '|' indicates alignment. Since number of alignment obtained by inserting gaps are 6, NAP is 6. But, 1 and 2 are directly aligned and therefore NDA takes value 2. The unique pages left are 7, 7 and 2 according to SAM and thus NUP is 3. Length of sessions S_1 and S_2 are 12 and 13 respectively. Therefore the HSAM distance, $HSAM_dist$ (S_1, S_2) is 0.44 by Eq.3.6.1. Thus, the proposed HSAM algorithm finds the distance between any two sessions that may be of uneven lengths by using the sequence information and without modifying the sequence. The major steps of the proposed HSAM algorithm are given in Algorithm 8.

Table 3.9: HSAM distance for different types of sessions

No.	Session S_1	Session S_2	m	n	NDA	NAP	NUP	$HSAM_{dist}$
1	P_1, P_2, P_3, P_4, P_5	P_1, P_2, P_3, P_4, P_5	5	5	5	0	0	0
2	P_1, P_2, P_3, P_4, P_5	$P_{11}, P_{12}, P_{13}, P_{14}, P_{15}$	5	5	0	0	10	1
3	P_1, P_2, P_3, P_4	P_1, P_2, P_8, P_4	4	4	3	0	2	0.25
4	$P_1, P_{16}, P_3, P_4, P_{18}$	$P_{24}, P_1, P_3, P_4, P_{18}, P_{25}$	5	6	3	0	3	0.27
5	$P_{24}, P_{26}, P_1, P_2, P_{20}, P_{21}$	$P_{24}, P_2, P_{20}, P_{18}, P_9, P_2, P_{12}$	6	7	1	2	7	0.69

Table 3.9 illustrates the behavior of the proposed HSAM distance method for different types of sessions. Consider the first entry of the table. Both sessions are of equal length and are exactly similar. Therefore, the distance between them is zero.

Similarly the sessions of second entry are completely different and the distance between them is one. The sessions given in the third row are more similar to each other and therefore the distance between them is less. Here, NAP is equal to zero, NDA is equal to 3, NUP is equal to 2. Since NAP is zero and NDA is greater than zero, the distance between them is determined as, $HSAM_{dist}(S_1, S_2) = NUP/(m+n) = 2/8 = 0.25$. The fourth row of the table shows the distance value as 0.27 which indicates that both the sessions S_1 and S_2 considered are more similar in sequence. The entries in fifth row show the case where sessions are not much similar in sequence. Here, length of session S_1=m=6, length of session S_2=n=7, NUP=7, NDA=1, NAP=2. So the $HSAM_{dist}(S_1, S_2) = [NUP + [2(\mid NAP - NDA \mid)]/(m+n) = (7+2(2-1))/(6+7) = 0.69$. The distance value for more similar sessions will be towards zero whereas the distance value between more dissimilar sessions will be towards one. Thus, the proposed HSAM algorithm differentiates between exactly similar sessions, completely different sessions and the sessions with varying degrees of similarities.

3.6.1 Cluster validation

Clusters of user sessions are formed by using the modified k-means for web sessions algorithm discussed in section 3.3 and HSAM is used to find the distance between any two user sessions. R^2 statistic measure is used to decide the number of clusters to be formed. Fig. 3.5 shows the R^2 values plotted for SAM, SABDM and the HSAM methods. It can be seen from the graph that the overall R^2 value is higher for HSAM compared to the other two methods. Initially the R^2 is 0.89 for the SAM method but if we observe the average or overall R^2, HSAM is definitely better. Since, HSAM is preferred over SAM and SABDM, R^2 value of HSAM is considered to decide minimum number of clusters to be formed. The R^2 reaches 0.8 at cluster number 4 for the HSAM and hence 'k' (number of clusters) value is taken as 4. Instead of deciding the 'k' value randomly or asking the user to enter some value for 'k', R^2 statistical

measure is used here.

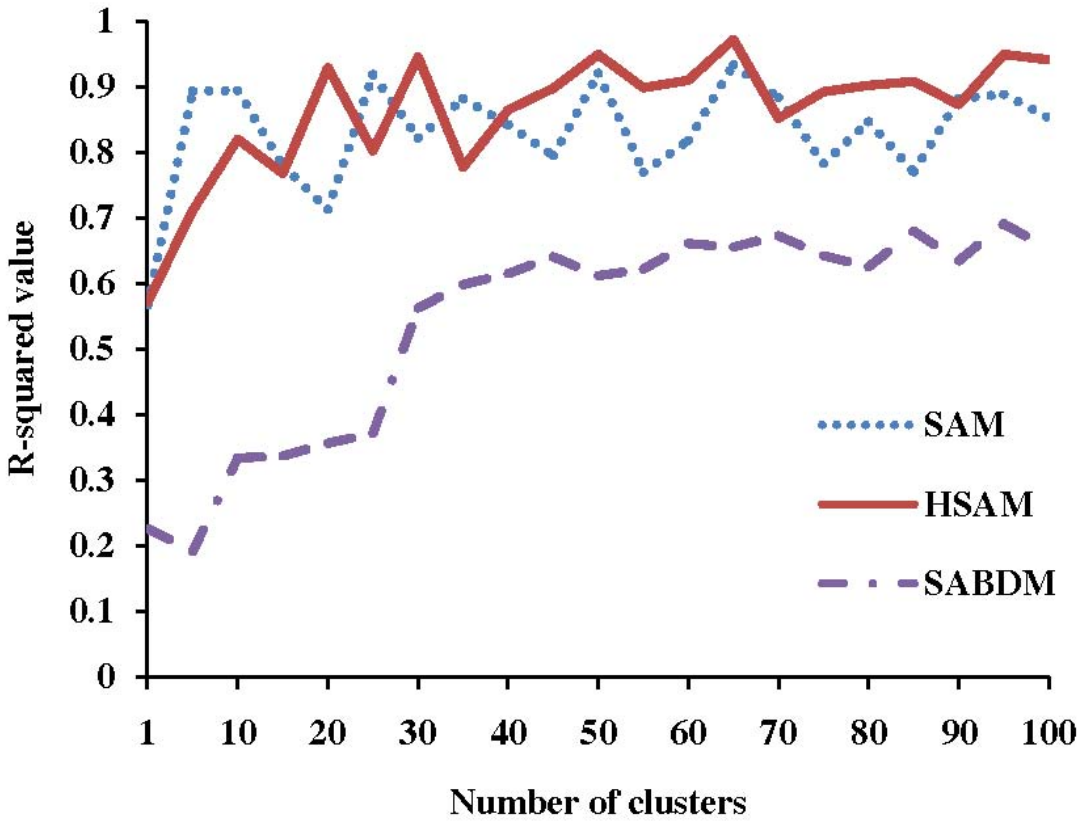

Figure 3.5: Figure showing R^2 value against number of clusters for SAM, HSAM, SABDM

Jaccard index and Davies-Bouldin validity index are employed for cluster validation. Fig.3.6 shows the average Jaccard index values for various numbers of clusters. The Horizontal axis represents the number of clusters and the vertical axis shows the Jaccard index values for both SAM and HSAM methods. The graph shows that the average Jaccard value for HSAM clusters is less than the SAM method. Given two clustering methods, the method with lesser Jaccard index is preferred for good clustering. Thus, based on the overall Jaccard index value it could be inferred that the clustering done by HSAM method as a distance measure is better compared to the SAM method.

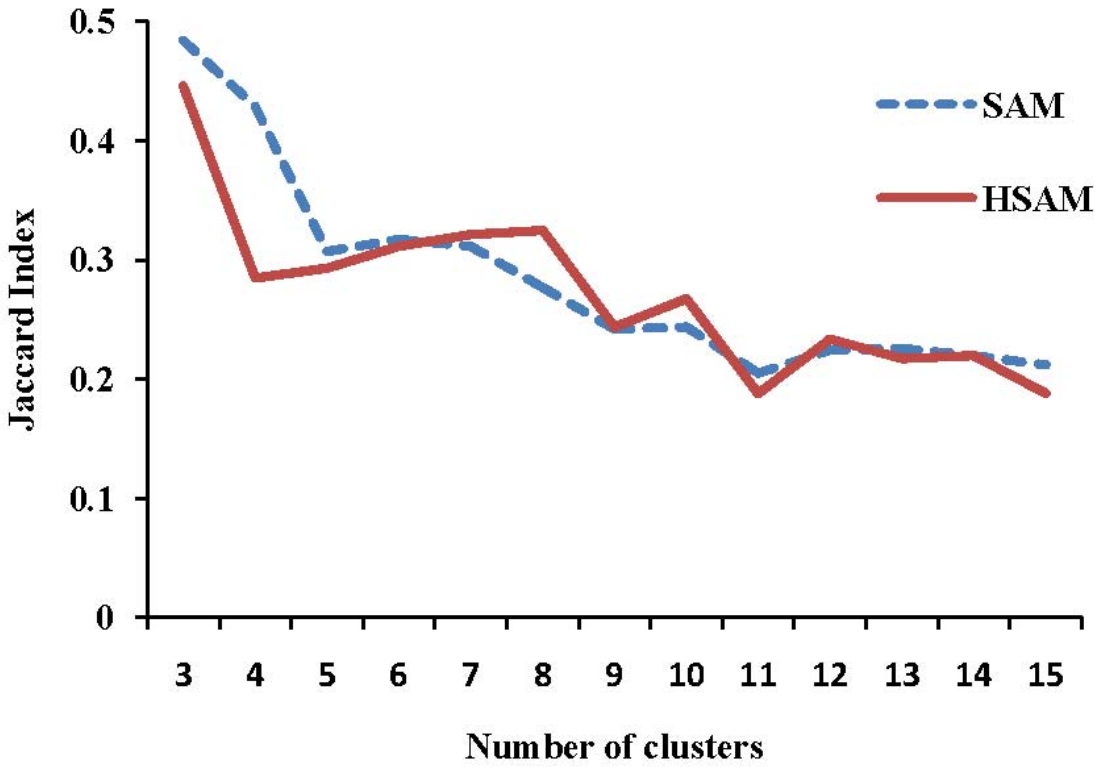

Figure 3.6: Figure showing the Jaccard index against number of clusters for SAM and HSAM

Davies-Bouldin (DB) Index The DB index is a well known function defined as, the ratio of the sum of within-cluster scatter to between-cluster separation and is computed by using equation 3.6.2.

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j} \left\{ \frac{S_k(C_i) + S_k(C_j)}{S(C_i, C_j)} \right\} \tag{3.6.2}$$

where:

- k is the number of clusters
- $S_k(C_i)$ is the average distance of all sessions from the cluster to their cluster center
- S(C_i,C_j) is the distance between cluster centers C_i and C_j

The major steps used to calculate the DB index are given in Algorithm 9. The

Algorithm 9 Davies-Bouldin index

Input: clusters of user sessions
Output: average DB index

1: for each cluster, find the average distance of all sessions of that cluster to their cluster center
2: for each cluster
   - find the DB index with other clusters by using equation 3.6.2
   - find the largest DB index for the cluster
3: add the largest DB index of each cluster together and divide this sum by the number of clusters to get the average DB index

clustering algorithm that produces a collection of clusters with the smallest DB index is considered the best algorithm based on this criterion. Hence the ratio is small if the clusters are compact and far from each other. Consequently, DB index will have a small value for a good clustering. Fig. 3.7 shows the DB Index for various numbers of clusters for SAM and HSAM methods. It can be seen from the graph that, the average DB index is lesser for HSAM method compared to the SAM method.

Thus Figs.3.6 and 3.7 show, the average or overall values of both Jaccard and DB index. Since the value of 'k' is decided as 4 based on the R^2, four clusters are formed. Table 3 clearly shows that the HSAM values for both validity indices are much lesser than the SAM. Table 3.10 shows the Jaccard and DB index values for the SAM and proposed HSAM algorithms for k=4 respectively. The table entries clearly show that, the Jaccard and DB index values are less for the proposed HSAM algorithm than the SAM method. Thus both Jaccard index and DB validity index confirm the goodness of clustering done by the proposed HSAM method as a distance measure compared to the SAM as a distance measure.

Table 3.10: Jaccard and DB indices for SAM and HSAM

Method	Number of clusters	Jaccard index	DB index
SAM	4	0.43	2.61
HSAM	4	0.28	1.37

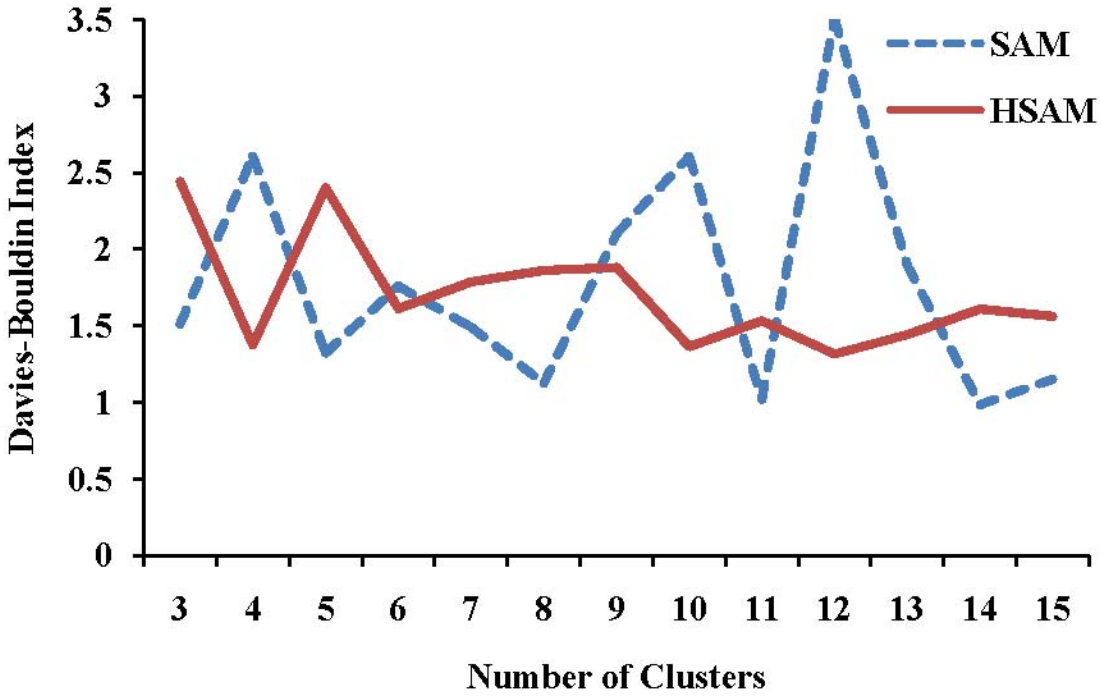

Figure 3.7: Figure showing DB index against number of clusters for SAM and HSAM

3.6.2 Navigation Patterns

R^2 is used to decide the optimal number of clusters to be created. Four clusters are formed by the proposed HSAM method. The navigation patterns of users are discovered by considering the frequency of page categories in each cluster.

Fig.3.8 shows the frequency of page categories for each of the four clusters formed by the proposed HSAM method. The users in cluster 1 are interested in "/shuttle/ missions/" category. In cluster 2, equal interest is shown for categories "shuttle/ missions" and "/history/apollo/" followed by the categories "/history/gemini/" and "/history/mercury/". Clusters 3 and 4 access other pages in combination with "/shuttle/missions" and "/history/apollo/". The major difference between these two clusters is, "/finance/" and "/procurement/" categories are accessed in 4th cluster

whereas, they are not at all accessed by the sessions of the 3rd cluster. The frequencies for "/payloads/" and "/persons/" are more in 4th cluster compared to the 3rd cluster. Clusters 3 and 4 may appear similar, but they are not exactly same. Because the graph is plotted based on frequency of category in each cluster, same category may be present in both clusters but, the pages accessed may be different. For example, a few of the pages accessed in cluster 3 are {"/biomed/ text", "/history/apollo/apollo-5/sounds", "/history/apollo/ apollo-8/ videos/", "/pao/factsheets/", "/shuttle/ missions/status/"}. These paths are not explored by sessions in 4th cluster. Similarly, some of the paths not accessed in 3rd cluster but traversed by sessions of the 4th cluster are {"/facilities/", "/biomed/climate/", "/biomed/soils/", "/biomed/wetlands/", "/finance/"}. The reason could be that, group of scientists may work together to achieve a certain task. Naturally, they will be more interested to access pages related to their work compared to other web pages. As a consequence, every cluster shows its own interest through the navigation paths accessed. However, the categories "/shuttle/ missions/" and "/history/apollo/" seem to be viewed by majority of the sessions. In this way the clusters formed by the proposed HSAM distance measure group the sessions based on the order in which the pages are visited. Frequently accessed navigation patterns in each of the clusters are also obtained.

Thus the HSAM method uses the navigation order of pages in sessions which may be of uneven lengths, to find the distance between any two sessions without modifying the navigation path of sessions. 5000 sessions are considered from the NASA data set for the experimental purpose. The results are compared with the SAM method. R^2 statistic measure is used to decide the optimal number of clusters to be formed. Jaccard index and DB validity index are used to validate the goodness of the clustering done and also compared the proposed HSAM method with the SAM distance measure. The graphs clearly indicate the goodness of HSAM over SAM. The results obtained are encouraging in terms of the various navigation patterns obtained. Thus the proposed approach is capable of capturing the interests of user

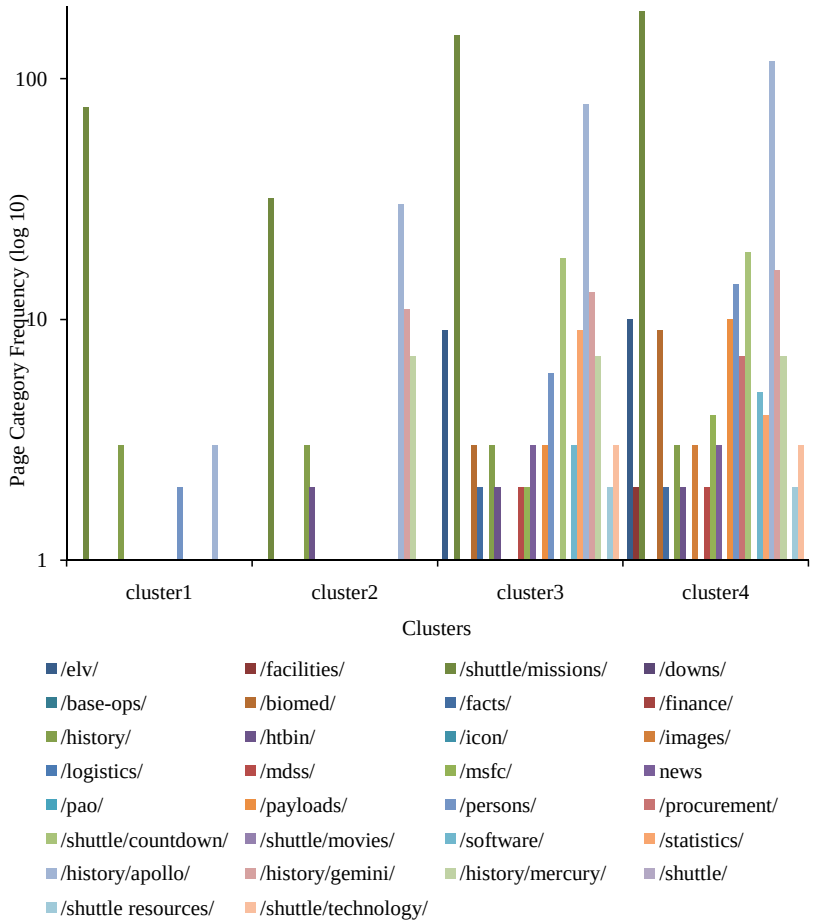

Figure 3.8: Navigation patterns of various clusters for HSAM

while navigating through a web site.

Comment on completeness and correctness of proposed algorithms Consider the proposed VLVD method as given in Algorithm 3.

Input assertion: ASSUME, S_i and S_j are any 2 web user sessions and (length(S_i) > 0, length(S_j) > 0; i.e. $l_1 > 0, l_2 > 0$).

Output assertion: ACHIEVE, *distance* d_{VLVD} between S_i and S_j, $0 <= d_{VLVD} <= 1$.

Assertion for line 1: assert $l_1 > 0$, follows from the input assertion.

Assertion for line 2: assert $l_2 > 0$, follows from the input assertion.

Assertion for line 3: $C >= 0$, follows from the execution of intersection of 2 sessions. If sessions are completely different, C will be 0 otherwise, C will be equal to number of common pages between S_i and S_j, which will be greater than 0.
Assertion for line 4: $dist = l_1 + l_2$, follows from the assertion of line 3. If, $C = 0$ (no common pages between sessions, sessions are completely different), $dist = 0$ (session length is same and all the pages are common, $l_1 + l_2 = 2_C$), otherwise, $dist < l_1 + l_2$ (if $C > 0$ and $C! = l_1 + l_2$).
Assertion for line 5: $len = l_1 + l_2$, follows from the execution of this statement, ($len >= 2$, follows from input assertion).
Assertion for line 6: $d_{VLVD} = 0$ for exactly similar sessions, (from assertion for line 4, if, $l_1 + l_2 = 2C$), $d_{VLVD} = 1$ for completely different sessions, (from assertion for line 4, if C=0), $0 <= d_{VLVD} <= 1$ (from assertion for line 4, if $C > 0$). $len >= 2$, follows from the input assertion and assertion for line 5. Therefore, "divide by zero" error will never occur and thus the value of d_{VLVD} will be greater than or equal to zero and less than or equal to 1. Hence, output assertion is correct.

Thus this algorithm satisfies intended specifications and all paths starting from the input point to the output points satisfy the output assertion. Hence, it can concluded that, the algorithm is complete and correct since, it is not possible for the algorithm to get stuck or reach a dead-end at any point of execution.

Similarly, the SABDM (see. Algorithm 5) and HSAM(see. Algorithm 8) algorithms find the distance between any two web sessions, and are based on the concept of sequence alignment. The input and out specification for these two algorithms are same as VLVD algorithm. The possible case of inputs could be: exactly similar sessions, completely different sessions, sessions that have some common pages, navigated in the same order. Various cases of sessions are considered in Table 3.7 and Table 3.9 indicating that the algorithms are complete. Also, the correctness of these algorithms follows from the outputs obtained for various possible cases of inputs. The distance is

0 for exactly similar sessions and the distance is 1 for entirely different sessions. The values that lie between 0 and 1 indicate the quantity by which sessions are similar or dissimilar.

To summarize, we discussed how to represent variable length user sessions effectively, proposed methods to find the distance between user sessions by using VLVD, SABDM, and the HSAM methods, explained modified k-means algorithm, and also commented on the completeness and correctness of these algorithms. Validated results by using a couple of standard statistical measures.

Chapter 4

PREDICTION

Web page prediction is the problem of forecasting the next page that might be visited by the user from the current active page or most recently visited previous pages. Various applications like web page recommendation, web site restructure, web caching and pre-fetching, determining most appropriate place for advertisements, search engines etc. would benefit from the good prediction model. Consequently the web page prediction has gained more importance in recent years among research community. The current work proposes a prediction model that could be employed for above mentioned applications in general and web page pre-fetching in specific.

4.1 Background

There are many architectures and related algorithms for developing web page predictor. Markov model is a mathematical tool for statistical modeling. The basic concept of the Markov model is to predict the next action, depending on the result of previous actions. Researchers have adopted this technique successfully in literature for training and testing the user actions and thus predicting their behavior in future. Deshpande et al. (2004) discussed different techniques for selecting parts of different order Markov models to get a model with high predictive accuracy and less state complexity. The main idea was to eliminate some of the states of different order

Markov models based on frequency, confidence and error. They predicted last page of the test session for evaluation purpose. Kim et al. (2004) proposed a hybrid model by using Markov model, sequential association rule, association rule and a default model to improve the performance, specifically the recall. However, it did not improve the prediction accuracy. Khalil et al. (2008) tried to improve the web page prediction accuracy by integrating clustering, association rules and Markov models. Dutta et al. (2009) integrated first order Markov model with web page rank based on the link structure to get better prediction accuracy. Awad et al. (2007) combined Markov model with artificial neural network (Awad et al. 2007), Support Vector Machines (Awad et al. 2008) and the final prediction is based on the Dempsters Rule. LRS is used to reduce the model complexity. Frequency matrix is used to represent first order Markov model.

Pitkow et al. (1999) made an effort to reduce the model size compared to Markov models. But the static LRS model used by them may not be suitable for real time prediction model. Jalali et al. (2008a,b) proposed LCS based algorithm for predicting users future requests. They used graph partitioning algorithm for clustering and used LCS for classification. A new session is classified into one of the clusters and prediction list is generated based on the navigation patterns of corresponding cluster. Gunduz et al. (2003) proposed a prediction model by considering order information of pages and time spent on them in a session. User sessions are clustered by using graph partitioning algorithm and each cluster is represented by click stream tree. Tseng et al. (2008) proposed Temporal N-Gram algorithm that considers the temporality property in web usage evolution. Hofgesang (2006) investigated the relevance of time spent on a page by the user. The paper also defined similarity measure based on the combination of frequency and time spent on page. Xing et al. (2004) proposed a concept called preference which represents the navigation interest and intention of a user based on the viewing time. Mukhopadhyay et al. (2006) proposed a clustering method to group related web pages based on access patterns. They used page ranking

to build the prediction model in the initial stages. However, the average hit percentage was around 50% for prediction window size of three and 35% for the prediction window with size two. Anitha (2010) clustered the web log based on pair-wise nearest neighbor method, determined the next page accessed by sequential pattern mining. The sequence is not taken into consideration for clustering and Markov model is used for sequential mining. Lu et al. (2005) generated significant usage patterns from the abstracted web session clusters by using Needleman-Wunsch global alignment algorithm. First order Markov model was built for each cluster. Since the accuracy of lower order Markov models are less and higher order Markov models consume space for many states while improving the accuracy, researchers tried to integrate the Markov model with other data mining techniques like clustering, association rule etc.

The next section discusses the general concept of hashing along with the requirements of hashing and the consquences of using hashing.

4.2 Hashing

Hashing provides a means for accessing data without the use of an index structure. The hash function is used to map the search key to a value, to quickly locate an object. The values returned by a hash function are called hash value or hash code. Typically, the domain of a hash function (the set of possible keys) is larger than its range (the number of different table indexes), and so it will map several different keys to the same index. Therefore, each slot of a hash table is associated with (implicitly or explicitly) a set of objects, rather than a single object. For this reason, each slot of a hash table is often called a bucket, and hash values are also called bucket indices. The expected time to search for an element in a hash table is O(1). Hash tables are particularly efficient when the maximum number of entries can be predicted in advance, so that the bucket array can be allocated once with the optimum size and never resized. In cases where, ranges are infrequent, hashing provides faster insertion,

deletion, and lookup than ordered indexing.(Cormen et al. 2009).

Requirements:

- The hash function, h, is a function from the set of all search-keys, K, to the set of all bucket addresses, B.
- Insertion, deletion, and lookup are done in constant time.

Consequences:

- Two different keys may be sent to the same address generating a collision.
- Hash tables become quite inefficient when there are many collisions.
- Hashing is less efficient if queries to the database include ranges as opposed to specific values.

The next section discusses a prediction model based on web user sessions clustering. Maintaining the information of order in which a user accesses web page of a web site is essential for predicting the next page to be accessed by the user. Hence, hybrid distance measure based on the sequence alignment technique is used to compute the similarities between any two sessions.

4.3 Hash Based Prediction Model

The proposed hash based prediction model works in two phases namely offline and online phase as shown in Fig.4.1. The offline phase is carried out at the server whereas, the online phase includes both the server and the client. The various components depicted in this figure are discussed below:

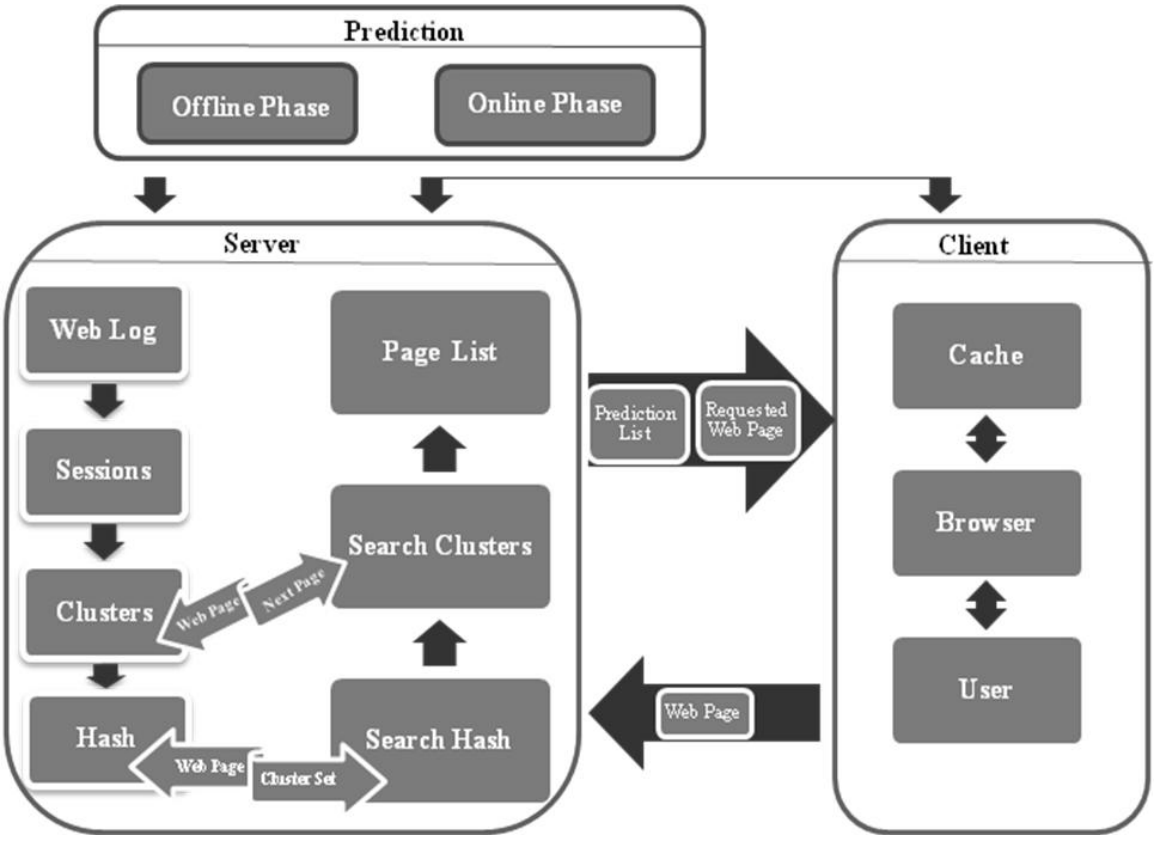

Figure 4.1: Hash based prediction model

Web log and Sessions The offline phase considers server logs first. The log contains an entry for each of the access to the server by client. Each entry has information like, client IP address, date and time at which the server is accessed, the HTTP method such as get, the requested URL, the response code, number of bytes transferred from server to client etc. Each entry is parsed to extract IP address, date and time, HTTP method and the URL requested. User sessions are created based on IP address, date and time. Further the sessions are filtered to remove image files, robot navigations etc. Unique requests are identified from these filtered sessions and unique identifier is given to each of them. For example if there are 10 unique pages, they are identified uniquely as P_1, P_2, P_3 and so on.

Clusters User sessions are divided into training and testing sets. 60% of sessions are considered for training and the remaining 40% sessions are taken as the test data set. Clusters of training sessions are formed by using HSAM and unique identification is given to each cluster. For example if four clusters are formed, they are uniquely identified as C_0, C_1, C_2 and C_3 respectively.

Hash A hash is created for each page by storing the cluster identification as value and the page identifier as the key. Suppose a page appears in more than one cluster, the cluster identifiers are concatenated and stored. Therefore, each entry in hash is a pair with a key and corresponding value. Thus the hash table contains entry for each of the unique URLs of a given web site. For example, if page P_1 is available in clusters C_1 and C_3, and page P_2 is present in the cluster C_2, the hash entry appears as shown in the Table 4.1. When a user accesses a page, it may be difficult to classify his session into any one of the clusters, based on only one page. So it is better to search in more than one cluster and the search space may be minimized as the user navigates further towards other pages.

Table 4.1: Sample hash table

Key	Value
P_1	C_1 C_3
P_2	C_2

Client Whenever user requests for a web page, the browser first looks into the cache for the presence of the requested web page. If found, the page is immediately retrieved and displayed to the user otherwise request is forwarded to the actual web server. The server responds by sending the requested page followed by a prediction list. When the browser is idle, it tries to download the web pages listed in prediction list which are not presently available in the cache. When the user requests for the next page, the user immediately gets the page from local cache and the user gets the

response immediately. Thus by pre-fetching the web pages, the user latency could be minimized.

Search Hash When the server receives request from the client, the hash table is searched by giving the key as the requested web page. The corresponding value with cluster identifier or the cluster set, is returned by the 'Hash' component to the 'Search Hash' component.

Search Clusters This component initiates the search process in the clusters formed during the offline phase. If the requested web page is present in the cluster, it returns the next page to the 'Search Clusters' component. Thus this component searches for the presence of the requested web page based on the cluster set obtained from the 'Search Hash' component. Then it collects the next page and maintains the count for each of the next page depending upon the number of sessions in which the next page is present followed by the current requested page.

Page List This component prepares a unique list of pages from the input received by the 'Search Clusters' component. Further, the list can be pruned based on certain criteria. For example, the list may be pruned depending on the count value of each page based on certain threshold and retaining the pages whose count exceeds this threshold. For example, if the count of some page is one, indicates that only one user has visited this page after viewing the current page. Therefore, the possibility of accessing this page could be less and hence it could be removed from the page list. The other criterion for pruning is list size based on frequency. That means top 'n' web pages based on highest frequency could be prepared. For example, top 10 pages could be retained from page list. Thus after pruning the page list a prediction list is formed and sent to the client along with the response to the requested page.

Thus the proposed hash based model predicts the next page possibly to be viewed by the user so that the pages could be pre-fetched. This results in reducing the delay

at the user end. Otherwise the user has to wait till the actual page is brought to him from the remote web server. The next section discusses about the experimental analysis done for the proposed hash based prediction model.

4.3.1 Prediction details

This section discusses the prediction done using the proposed model in detail. The proposed hash based prediction model consists of two phases as depicted in Fig.4.1 namely offline and online. The Modified k-means algorithm for web sessions clustering is used with HSAM as a distance measure. User sessions are divided into train set and test set with 60:40 ratio. The train set is considered for clustering and the test set for the evaluation purpose. The main advantage of clustering is homogeneity. Also, searching is faster because of fewer amounts of data in each clusters compared to the entire data. 5000, 10000, 15000 and 20000 sessions are considered for the experiment from the NASA data set.

We define few terms to understand the various activities of online phase. The list of symbols used is given in Table 4.2 to provide better readability and understanding. The proposed prediction model consists of two phases as depicted in figure namely offline and online. In the first phase, sessions are created after preprocessing the entries of the server log file. The 60% of sessions, referred as "train set" are clustered by HSAM method. In general a cluster contains some unique pages but there is a possibility that some pages like front page may be present in more than one cluster. Also, clusters are formed based on the sequence in which pages are accessed by the user and not based on the kind of pages accessed. For example, the page P_i may be accessed after P_j by some users, and before P_j by some other users. So, the sessions containing P_iP_j and P_jP_i may be present in different clusters. Therefore, hash is used to store the information of clusters in which each web page is present by having the page identifier as the key and cluster identifiers as the value.

Table 4.2: List of symbols used

Symbol	Description
WP	Set of web pages, WP=$\{P_1, P_2,...,P_n\}$
n	Cardinality of set WP i.e. total number of web pages
US	Set of user sessions obtained after pre processing the web log, US=$\{S_1, S_2, ..., S_u\}$
u	Cardinality of the set US
S_i	ith user session containing web pages visited by a user in succession represented as S=$\{P_a, P_b, ..., P_k\}$ and $S_i \subset$ WP
m	Session length
PL	Prediction list, set of URLs sent by the server to the client
PLS	Prediction list size, requirement:PLS $<< n$
CP	Number of correct predictions of a session
TS	Set of test sessions with cardinality ts
w	Window size; assume w=3

The online phase starts when the user requests a web page. The browser first checks the local cache for the required page. If the page is not available in the cache, the request is forwarded to the remote web server. The server searches in the hash for the presence of the requested web page to retrieve the cluster information. The clusters are explored for the presence of the requested page to find the next page accessed by various sessions and maintain a count for each of the next page. Thus given a web page P_i, the server prepares a unique list of URLs called Prediction List denoted as PL and sends to the client along with the response.

The size of prediction list denoted by PLS can be defined as the number of URLs present in the prediction list PL and PLS<<n. For example if PL(P_i)=$\{P_a, P_b, P_c\}$, PLS(P_i)=3.

The PLS for a session S_i is obtained taking average of PLS for each of the pages of a session. Thus PLS of a session S_i denoted by PLS(S_i) is given by Eq.4.3.1.

$$PLS(S_i) = \frac{\sum_{j=1}^{m-w} PLS(P_j)}{m-w} \tag{4.3.1}$$

The browser pre-fetches the URLs in the prediction list during its idle time. The pages already present in the cache need not be pre-fetched. Therefore, the browser first checks the local cache for the presence of URLs of the prediction list and fetches those web pages that are not currently in cache. As a consequence, the delay at the user is reduced considerably because the page to be accessed by the user is now available locally if the prediction is correct.

If the user visits a web page P_j immediately after viewing P_i and P_j is in PL, it is said to be a hit. If the length of a session S_i is 'm' and 'h' is the number of hits obtained for that session then the correct prediction CP(S_i) is defined as the number of hits divided by the session length minus 'w' because, the first 'w' page is not predicted. Thus the number of correct prediction CP is obtained by using Eq.4.3.2.

$$CP(S_i) = \frac{h}{m - w} \tag{4.3.2}$$

The overall percentage of correct predictions achieved for the test set is determined by taking the average of CP for all sessions. Thus the percentage of correct predictions is determined by using Eq.4.3.3.

$$\%CP(TS) = \frac{\sum_{j=1}^{ts} CP(S_j)}{ts} \times 100 \tag{4.3.3}$$

The percent of number of correct predictions made should be higher if the model used for prediction is good.

Similarly, the average size of the prediction list for test sessions is determined by Eq.4.3.4 and the list size should be smaller to make the prediction model effective and efficient.

$$PLS(TS) = \frac{\sum_{j=1}^{ts} PLS(S_j)}{ts} \times 100 \tag{4.3.4}$$

4.3.2 Prediction validation

This section discusses about the validation to be carried out which is essential to ensure that, the proposed model is useful for web page prediction. The user sessions are divided into train and test sets with 60:40 ratio and the test set is considered for the validation purpose. The probability or expected chance of a correct prediction depends on the prediction list size. The expected value of a page P_i is computed as given in Eq.4.3.5.

$$E(P_i) = \frac{PLS(P_i)}{n} \tag{4.3.5}$$

Hence, the expected value of a session S_i is computed by adding together the ratio of PLS by total number of web pages 'n' for each of the pages except the last page in that session. The expected value of a session S_i is denoted by E(S_i) and is given by Eq.4.3.6.

$$E(S_i) = \sum_{j=1}^{m-w} E(P_j) \tag{4.3.6}$$

Similarly, the average of expected values for all test sessions is given by Eq.4.3.7.

$$AE(TS) = \frac{\sum_{i=1}^{ts} E(S_i)}{ts} \tag{4.3.7}$$

Lemma 4.1 The best case and the worst case values for E(S_i) are (m-w)/n and (m-w) respectively.

Proof. If the prediction list size is one, for all the pages of a given session S_i, and suppose prediction is true for all the pages of S_i, then CP(S_i) is equal to one by Eq.4.3.2. This means that, the prediction list contains only one page and it results in a hit always. Thus, if 'm' is the session length and 'n' is the total number of web pages and we are giving prediction list for (m-w) number of pages, E(S_i) will be (m-w)/n by Eq.4.3.6 for the above scenario. As the prediction list size increases, chances of obtaining more number of hits will also increase. If the prediction list contains all

the pages of web site, definitely the prediction will always be correct. Though the number of hits will be more in this case, the client side browser have to pre-fetch all the pages given in the prediction list which may consume more time as well as space at the client. This is an indication of poor or no intelligence in the prediction model or algorithm. Thus in the worst case, prediction model may give all the pages as prediction list to the client which results in the value of E(S_i) as (m-w).

Thus, it is difficult to achieve best case, and the worst case is not desirable. So, the ideal prediction model should yield the value of E(S_i) such that it is nearer to the best case. For example, consider a session $S_i = (P_1, P_2, P_7, P_9, P_5)$ and assume that, total number of pages 'n' is 10. The session length m is 5 and we provide a list to 4 pages assuming that first page cannot be predicted. Also, assume that, the prediction list size is 1 for each of the four pages and CP(S_i)=1. By using equation E(S_i) is 4/10 i.e.,(m-w)/n. Similarly, if we assume that the prediction list size is 10 for each of the four pages and CP(S_i)=4 then, E(S_i)=4 i.e., (m-w) by Eq.4.3.6.

Observation : Let D denote the difference between the actual number of correct prediction CP and the expected value E computed as D=CP-E. The prediction model is good only if the value of D lies in the range of zero and one. i.e., the value of D should be greater than or equal to zero and less than one and is represented as $0 \leq D < 1$.

Remarks The maximum possible value of CP for any given session is one when the prediction is correct for all the pages of a session and the minimum is zero if none of the predictions are correct for a session. The value of E can range from (m-w)/n to (m-w) by Lemma 4.1. That means, as the PLS of a page increases the value of E also increases. This results in higher value of E after computing the expected value of the session and exceeds one, indicating that the PLS is larger. Because of this, the E value will be greater than the CP value of a session and D will be less than zero. If the average E =1/m, and if CP is also one indicating all the predictions are correct,

only then the D value will be zero. Thus, the D value would be always greater than or equal to zero and less than one. This shows that more number of correct predictions could be obtained with less number of PLS only if $0 \leq D < 1\square$.

The expected value gives the probability of the correct predictions for the given prediction list. For example, if the total number of web pages is 100 and a prediction list for a web page has 5 pages, then there is only 5% chance of a correct prediction. If the prediction list length is 100, then the prediction will be always correct and indicates that there is no intelligence in the prediction algorithm. Therefore the requirement is to correctly predict the next page with less number of pages in the prediction list. The browser has to pre-fetch them before user request for the next page. However, as the number of pages in prediction list increases the time taken to pre-fetch will also increase, and the space required to store these web pages will also be more. Hence the prediction algorithm is good only if the actual number of correct prediction CP is greater than the expected value E. The expected value will be greater than the actual value only if the prediction list size is larger, which is not desirable.

4.3.3 Results

This section analyses the results obtained by the proposed hash based prediction model. Figs.4.2 and 4.3 show the prediction accuracy for various numbers of sessions. The actual value i.e. number of correct predictions is more than the expected value in all the four cases. Thus these two graphs clearly demonstrate the goodness of the proposed model.

Table 4.3 and 4.4 show the best case, worst case and actual expected value based on the prediction list size for window sizes 1 and 2 respectively. By Lemma 4.1, the expected value should be closer to the best case for the ideal prediction model. The values of these two tables clearly indicate that, the expected value is nearer to the

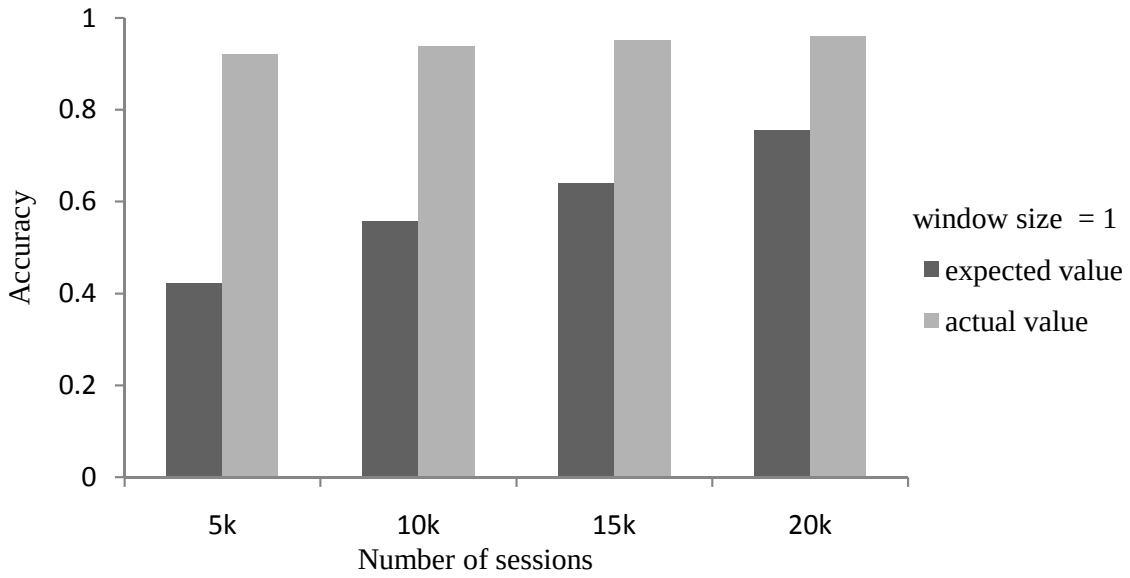

Figure 4.2: Prediction accuracy for window size 1

best case. That means, the prediction list size is much lesser than the total number of available pages. Also, high accuracy is obtained with the prediction list size being smaller.

Table 4.3: Expected values based on list size with window size 1

Sessions	Best case	Worst case	Expected value
5k	0.009763	8.3965	0.4237462
10k	0.010404	8.94775	0.5568659
15k	0.010029	8.6255	0.64038247
20k	0.010453	8.990125	0.7563575

The prediction list may be further pruned based on frequency and by limiting the list size. For example, top ten pages in the prediction list may be retained based on their frequency value. Figs.4.4 and 4.5 show the results obtained for frequency prunned prediction list for window sizes 1 and 2 respectively. The x- axis represents sessions of different numbers and the y-axis represent the accuracy i.e. the number of correct predictions obtained. In both cases the accuracy is more than 70% as can be seen from these two figures. The actual value is much more than the expected value

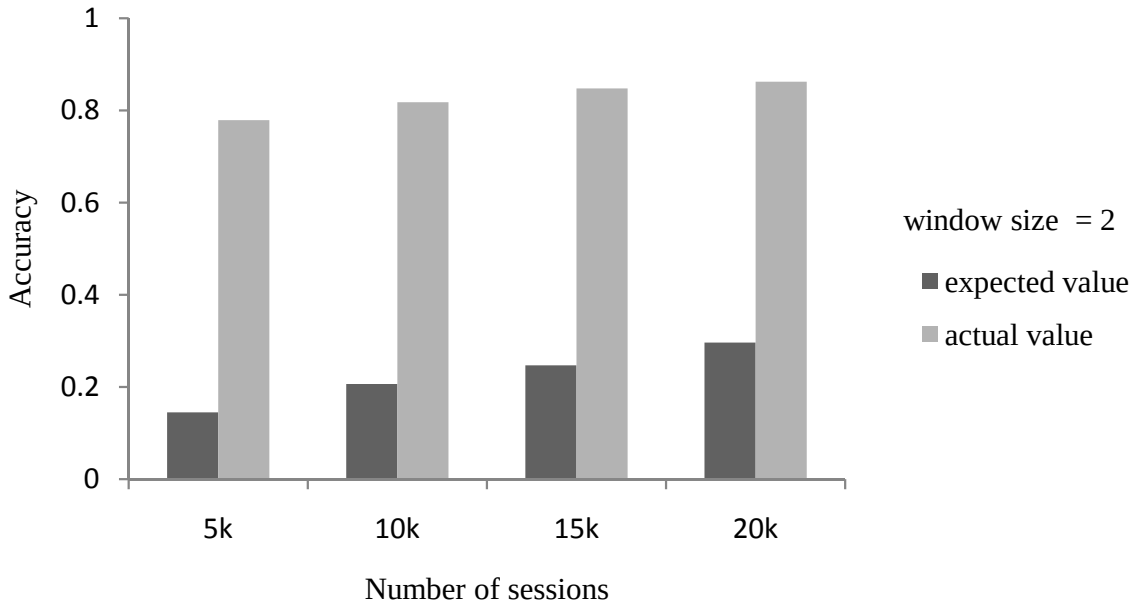

Figure 4.3: Prediction accuracy for window size 2

Table 4.4: Expected values based on list size with window size 2

Sessions	Best case	Worst case	Expected value
5k	0.008601	7.3965	0.14531234
10k	0.009242	7.94775	0.20657706
15k	0.008867	7.6255	0.2470867
20k	0.009291	7.990125	0.2960891

indicating the goodness of the model. Thus reasonably good accuarcy is obtained after prunning the prediction list based on frequency values.

4.4 Modified Prediction Model

The prediction model depicted in Fig.4.1 is independent of the clustering algorithm with respect to the accuracy, and is dependent on the clustering algorithm as well as the distance measure used for clustering, with respect to the time taken for generating the prediction list by the server. The Hash component of the model given in Fig.4.1 gives cluster set to the Search Hash component, which in turn is used to search in

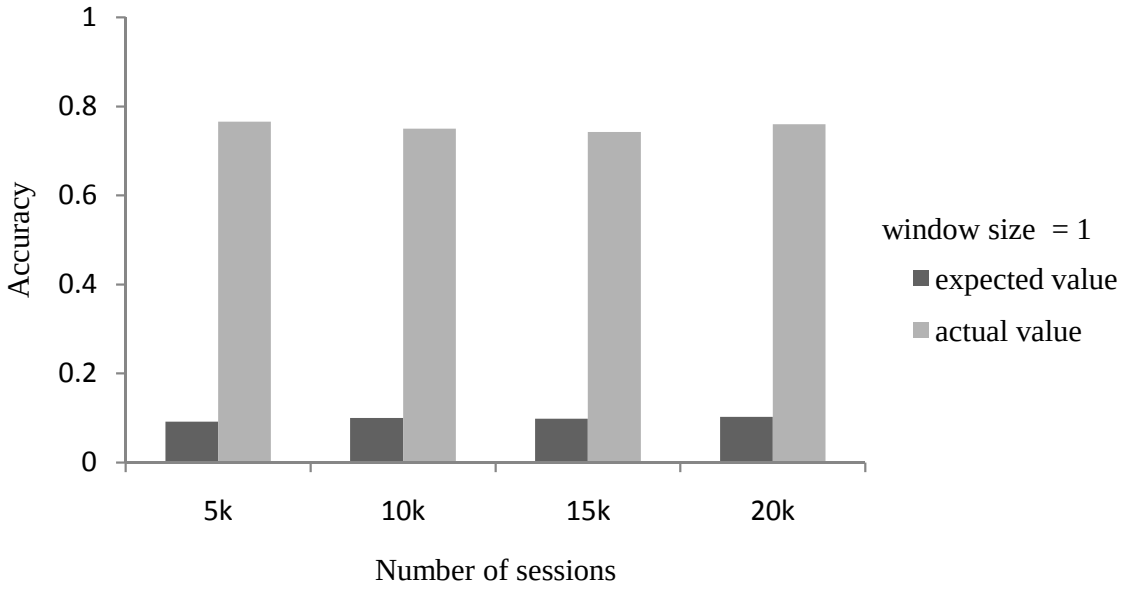

Figure 4.4: Prediction accuracy for top 10 pages with window size 1

various clusters for predicting the next page based on the given previous page. If the clustering algorithm is good, number of clusters in the cluster set will be less and hence searching for next page is limited to a few clusters. On the other hand, if the cluster algorithm is not good, each page may be present in every cluster. This may take more time to produce prediction list since, each of the cluster has to be searched to produce list that contains next page to be visited by the user. Thus in the worst case, search becomes linear and there is no use of the clustering done. Hence the model shown in Fig.4.1 is modified, so that the distance measure used and the clustering done could be used to predict the next page. Fig.4.6 shows the modified prediction model.

The first phase of this model is same as the model given in Fig.4.1, except that the hash is not used to maintain the cluster set to which a page belongs. The online phase starts when the user requests a web page. The browser first checks the local cache for the required page. If the page is not available in the cache then the request along with the previous pages viewed by the user is forwarded to the remote web server.

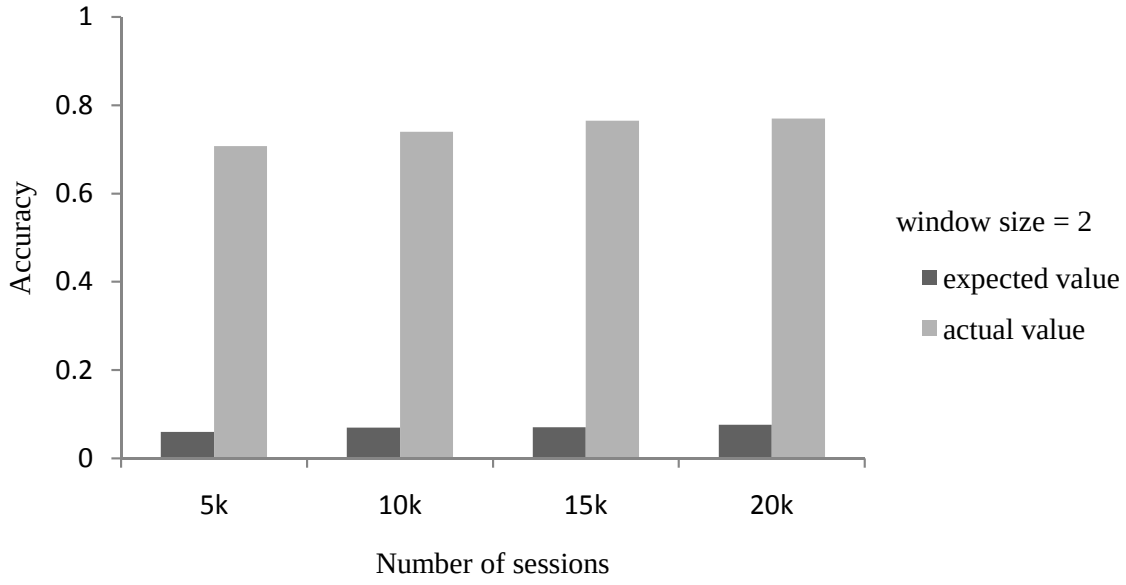

Figure 4.5: Prediction accuracy for top 10 pages with window size 2

The server finds the cluster to which the request is nearest, by applying the HSAM distance as a similarity measure. The nearest cluster is explored for the presence of the last page of the request to find the next page accessed by various sessions and maintain a count for each of the next page. Thus given a request, the server prepares a unique list of URLs called "Prediction List" and sends to the client along with the response. The browser pre-fetches the URLs in the prediction list during its idle time. The pages already present in the cache need not be pre-fetched. Therefore the browser first checks the local cache for the presence of URLs of the prediction list and fetches those web pages that are not currently in cache. As a consequence, the delay at the user is reduced considerably because the page to be accessed by the user is now available locally if the prediction is correct.

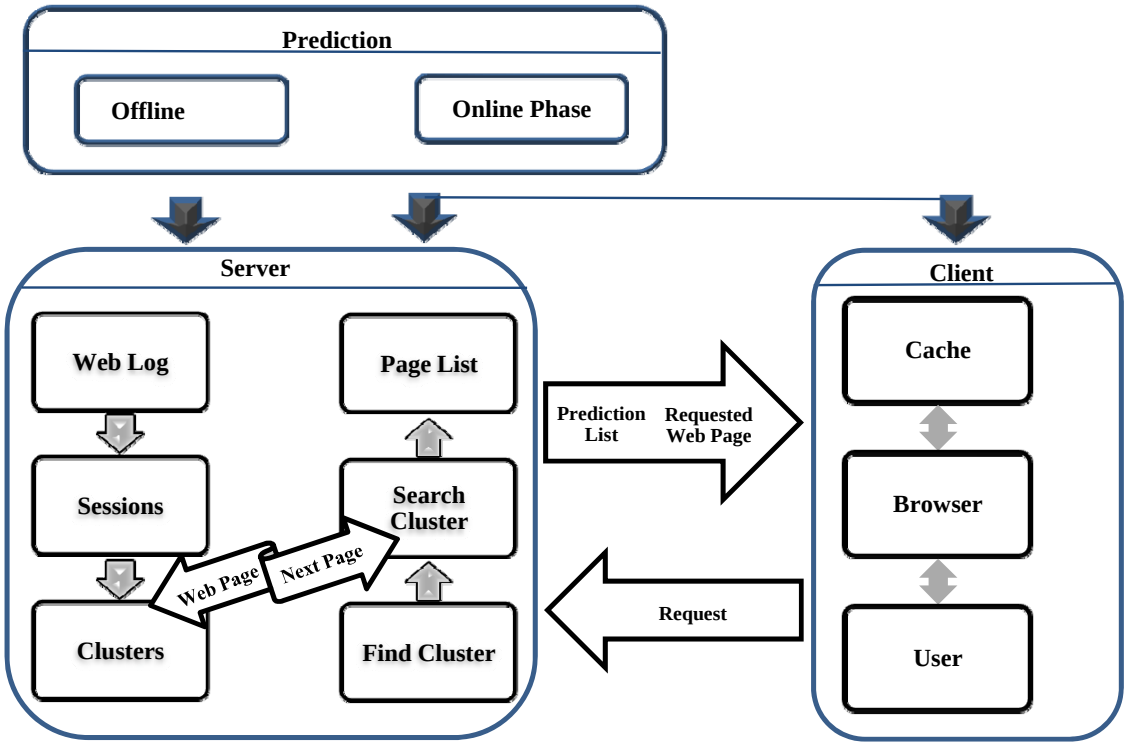

Figure 4.6: Modified prediction model

4.4.1 Results

Couple of experiments are conducted to evaluate the proposed prediction model by using the NASA and MM data sets. 5K, 10K and 15K number of sessions are considered for the experiment. To evaluate the accuracy of the predictions obtained by the proposed model, we use some definitions as discussed below:

Hit The predicted next page of a test session is present in the prediction list. Suppose $(P_{n1}, P_{n2}, P_{n3}, P_{n4})$ is a test session and (P_{n1}, P_{n2}, P_{n3}) is considered to find the nearest cluster C. The cluster C is searched for the presence of the page P_{n3} and a prediction list is prepeared that contains pages that are viewed immediately after viewing P_{n3} in C. A hit results if P_{n4} is present in the prediction list.

Miss The predicted next page of a test session is not present in the prediction list. Suppose $(P_{n1}, P_{n2}, P_{n3}, P_{n4})$ is a test session and (P_{n1}, P_{n2}, P_{n3}) is considered to find the nearest cluster C. The cluster C is searched for the presence of the page P_{n3} and a prediction list is prepeared that contains pages that are viewed immediately after viewing P_{n3} in C. A miss occurs if P_{n4} is not present in the prediction list.

Match The page or the path of a test session, is found in a given cluster or the prediction list is not empty. Suppose $(P_{n1}, P_{n2}, P_{n3}, P_{n4})$ is a test session and (P_{n1}, P_{n2}, P_{n3}) is considered to find the nearest cluster C. The cluster C is searched for the presence of the page P_{n3}. If P_{n3} is present in C it is said to be a match.

Mismatch The page or the path of a test session is not found in a given cluster, or prediction list is empty. Suppose $(P_{n1}, P_{n2}, P_{n3}, P_{n4})$ is a test session and (P_{n1}, P_{n2}, P_{n3}) is considered to find the nearest cluster C. The cluster C is searched for the presence of the page P_{n3}. If P_{n3} is not present in C it is said to be a mismatch.

Accuracy The ratio of number of correct predictions or hits to the number of matches. The number of mismatches are not considered for accuracy since the proposed model cannot predict for mismatches.

In the first experiment, the last page is predicted for the purpose of evaluating the proposed prediction model. Each session from the test data is considered one at a time. The last page of the test session is removed and the remaining pages are considered to find the nearest cluster center. For example, if a session consists of n number of pages $(P_1, P_2, ..., P_n)$, the last page P_n is removed. The remaining pages $(P_1, P_2, ..., P_{n-1})$ are considered to find the nearest cluster. The nearest cluster obtained is searched for the presence of the page P_{n-1}. A prediction list is prepared that contains the next page visited followed by P_{n-1} in the cluster. The presence of actual last page of the session 'P_n' is checked in the prediction list. A hit results

if the prediction list contains P_n and the number of hits is incremented. Note that, the pages from P_1 to P_{n-1} are considered to find the nearest cluster. But, once the nearest cluster is found, only the last page i.e., P_{n-1} is used to predict the next page. Fig.4.7 depicts the accuracy for both NASA and MM data set.

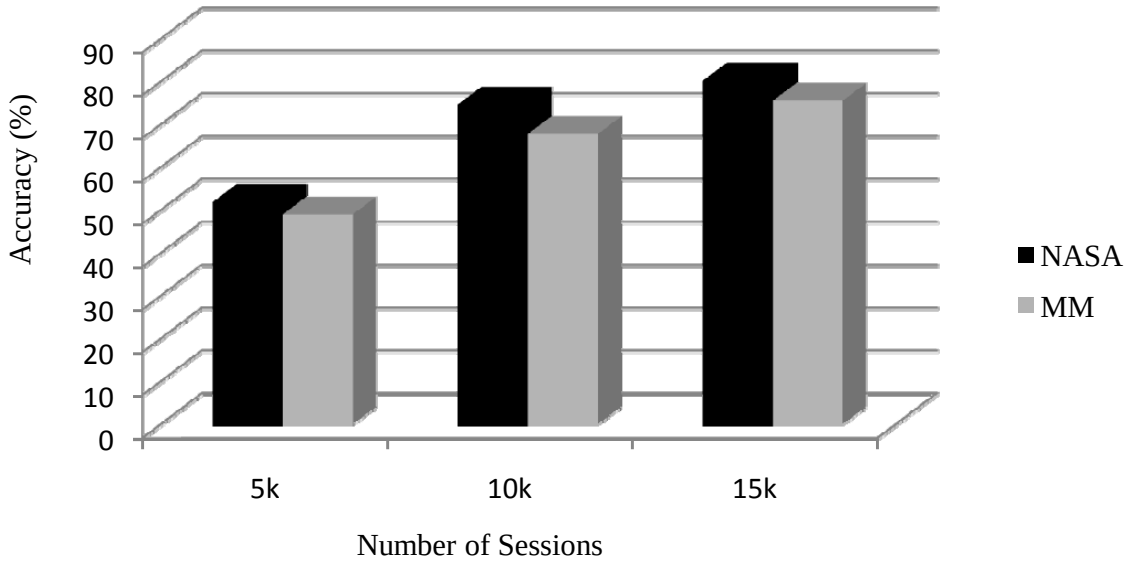

Figure 4.7: Prediction accuracy for NASA and MM data sets

Table 4.5 shows the number of hits, misses, matches, mismatches, and average session length for both data sets, by considering different number of sessions. It can be observed that, the overall number of matches is more than the mismatches. The numbers of hits are more than the misses as the size of training data increases. The better % of matches indicates the goodness of the clustering algorithm as well as the HSAM distance measure which is used to find the distance from the test session to the nearest cluster.

The second experiment is conducted to limit the list size and analyze the accuracy of predictions. The page list is sorted in descending order based on the frequency and the top n pages from this sorted page list are selected to form the prediction list. The test data set is evaluated by assuming the 'n' value in the range from 1 to 10 for both the data sets. Figs.4.8 and 4.9 depict the accuracy for these top 'n' values for NASA

Table 4.5: Results of predicting the last page

Data Set	NASA			MM		
Number of sessions	5k	10k	15K	5K	10K	15K
hit %	57.91	71.58	80.29	56.64	69.87	75.05
miss %	42.09	28.42	19.71	43.36	30.13	24.95
match %	87.90	97.20	98.17	82.45	92.60	93.13
mismatch %	12.10	2.80	1.83	17.55	7.40	6.87
average session length	8.39	8.94	8.62	9.83	10.01	9.7

and MM data set respectively. These graphs illustrate that, as the number of pages in the prediction list increases, the accuracy also improves. Instead of providing a prediction list with all the pages, it would be better to have 'n' number of pages in the prediction list as this list is used for pre-fetching pages from the client browser. Depending on the bandwidth available or the server load, suitable value for 'n' could be determined.

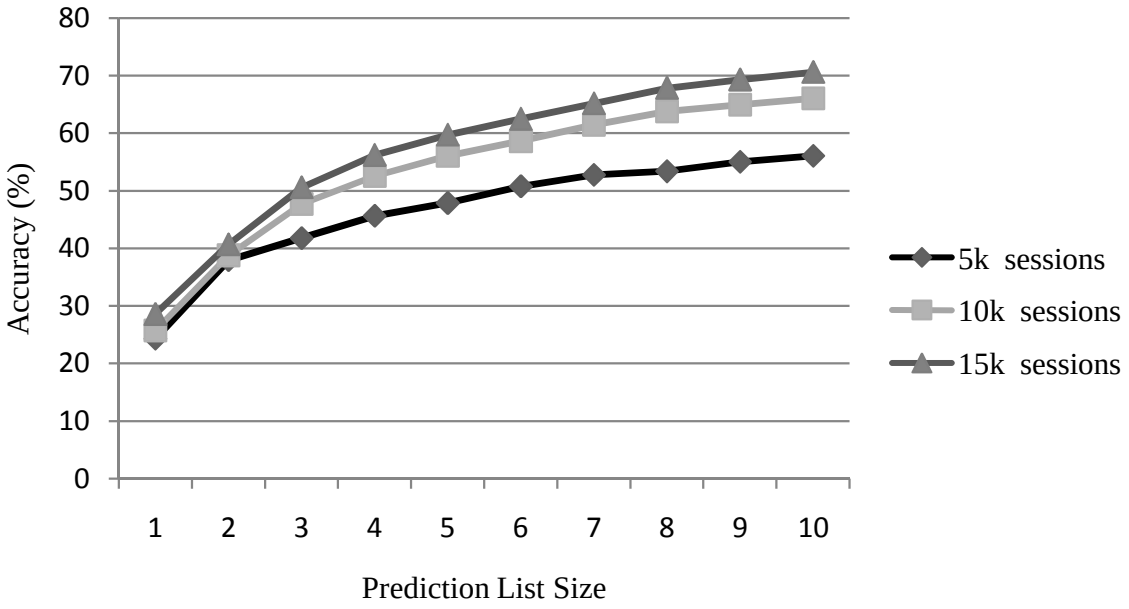

Figure 4.8: Prediction accuracy for NASA data set based on list size

Awad et al. (2007) compared prediction results of various methods like Markov, ANN, ARM, All-kth-Markov, All-kth-Dempsters rule etc. by considering ranks for web pages based on highest confidence. The probability of hit by match is not more

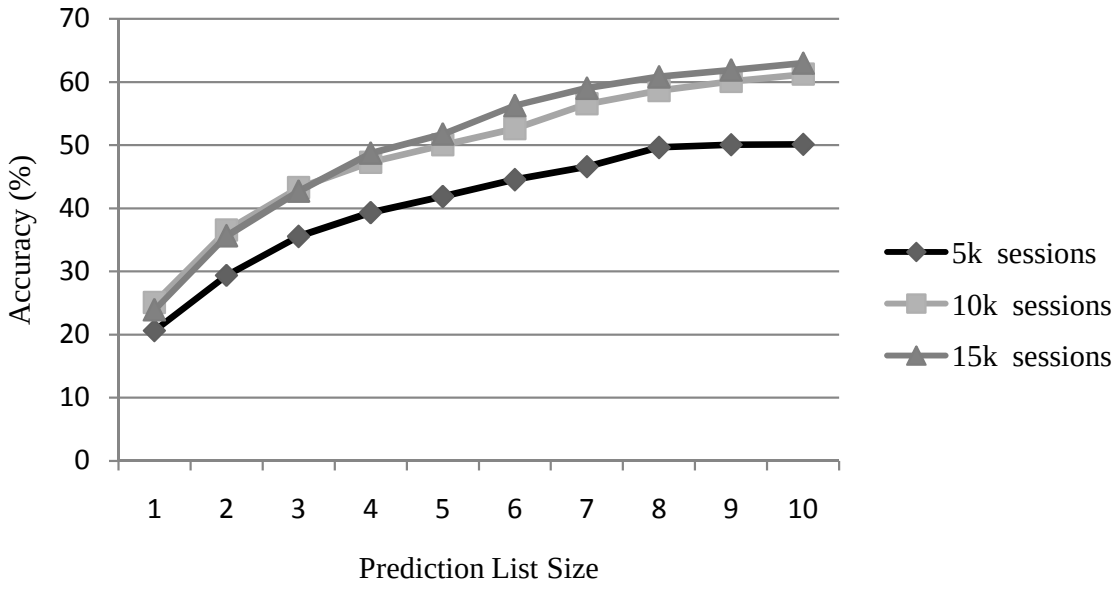

Figure 4.9: Prediction accuracy for MM data set based on list size

than 54% for ranks 1-8 for various techniques. The Figs.4.8 and 4.9 clearly shows that, the accuracy achieved by the proposed model is much better than the accuracy obtained by All-kth-Dempsters rule. The proposed model ranks only the predicted web pages based on highest frequency whereas Awad et al. (2007) constructed frequency matrix. The entries of this matrix represent frequency of two consecutive pages for each of the web page. As the number of pages in a web site will be more, the frequency matrix consumes more space. The accuracy of prediction increases as the number of hops is increased. But, the proposed method achieves better accuracy with only one hop. Hop is the number of pages considered to predict the next page (Awad et al. 2007).

Khalil et al. (2009) combined Markov model, association rules and clustering to predict the next page to be accessed by user. They used 4 different data sets to evaluate their model. However, the accuracy obtained was 45%, 55%, 65% and 35% respectively for four data sets used. Pitkow et al. (1999) tested their prediction model by comparing one-hop LRS with one-hop Markov model. The probability of hit by

match was 25% and 31% for k-th order Markov model. Guo et al. (2007) ranked web pages based on access time, length and frequency. The prediction accuracy obtained was around 50% for top 3 predictions and less than 60% for top 5 predictions. Thus, the other methods in literature mentioned above do not accomplish good accuracy compared to the proposed prediction model.

Predicting last page of a session is useful only for the purpose of evaluating the proposed model. In reality it is not possible to know which page would be the last page to be viewed by a user and also the exact session length is not known in advance. Therefore, it would be better to predict and pre-fetch web pages as soon as a user starts navigation based on the window size. Here, window size refers to minimum number of pages that may be required to find the cluster to which a session may belong to. Hence the third experiment was conducted to predict the next page from the beginning of session. Here, window size is assumed as 3 and hence first three pages are not predicted. The first three pages are considered to find the distance with all cluster centers by using the HSAM distance measure. Prediction list is prepared based on the nearest cluster as done in the first two experiments. If the actual 4th page visited by the user is present in the prediction list, CP is incremented. Next, first four pages are considered and again distance to all cluster centers is found to predict the next page i.e. 5th page and so on. Suppose $(P_i, P_j, P_k, P_l, P_m, P_n)$ is test session, (P_i, P_j, P_k) is considered first and the nearest cluster is found to predict P_l. It may result in either hit or miss or mismatch. Next, (P_i, P_j, P_k, P_l) is taken into consideration to predict P_m and finally $(P_i, P_j, P_k, P_l, P_m)$ is taken to predict P_n. At the end numbers of hits/misses/mismatches are determined. The Fig.4.10 shows the percent of correct predictions (accuracy) for the NASA and Music Machine data sets respectively.

In the above experiment, as the user navigates further, all the earlier pages starting from the first page are considered to predict next page. But, the next page to be viewed by the user may not depend on all earlier pages. The next page may depend

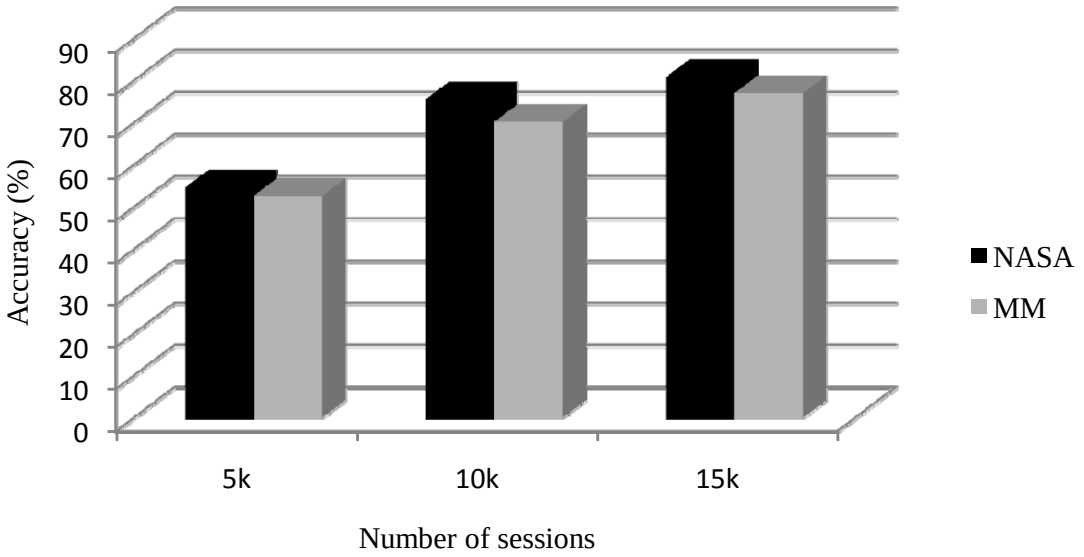

Figure 4.10: Prediction accuracy without sliding window

on most recently used 'w' pages rather than all pages. So, as the variation of the second experiment, only the recent 'w' pages are considered for predicting the next page. Fig.4.11 shows the result of the accuracy obtained with the sliding window. Suppose $(P_i, P_j, P_k, P_l, P_m, P_n)$ is a test session, (P_i, P_j, P_k) is considered first and the nearest cluster is found to predict P_l. Next, (P_j, P_k, P_l) is considered to predict P_m. Finally (P_k, P_l, P_m) is considered to predict P_n. Thus only the most recently viewed 'w' pages are considered each time as the user navigates further through a web site. By comparing Fig. 4.10 and 4.11, it can be observed that the accuracy is almost same for both the data sets considered. Therefore, instead of considering all the pages for prediction, only the most recently viewed pages could be considered to find the nearest cluster.

Table 4.6 presents the expected values for various numbers of sessions for both the data sets. The actual expected value obtained is nearer to the best case compared to the worst case value. This proves the goodness of the proposed prediction model by Lemma 4.1.

The actual correct predictions are more than the corresponding expected values

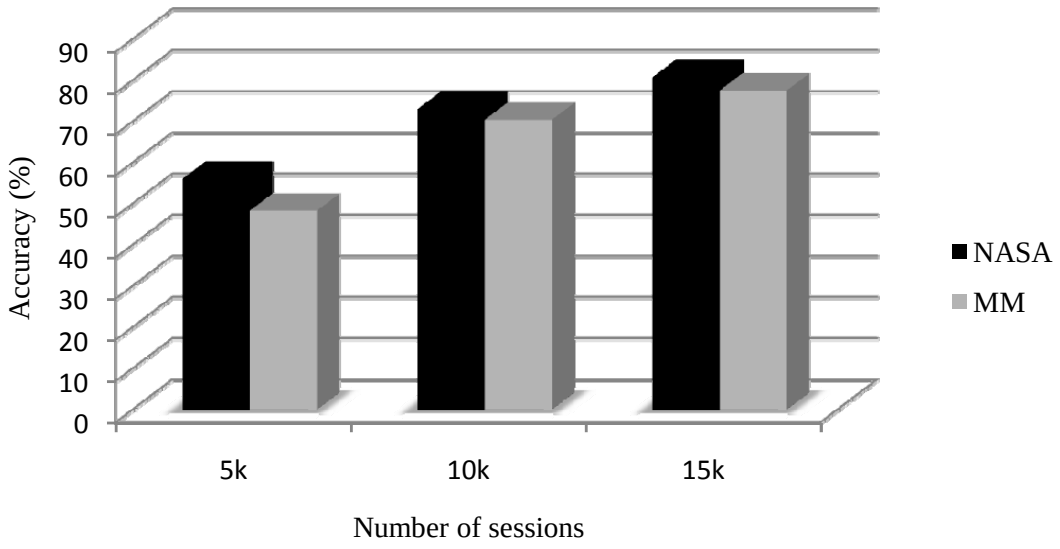

Figure 4.11: Prediction accuracy with sliding window, w=3

Table 4.6: Expected values

Data Set	Number of sessions		5k	10k	15k
NASA	Expected value (best case)		0.007	0.008	0.008
	Expected value (worst case)		6.397	6.948	6.626
	Actual expected value	Without sliding window	0.044	0.118	0.162
		With sliding window	0.046	0.105	0.161
MM	Expected value (best case)		0.002	0.002	0.002
	Expected value (worst case)		7.838	8.014	7.701
	Actual expected value	Without sliding window	0.015	0.029	0.045
		With sliding window	0.011	0.030	0.043

for all the experiments carried out as can be observed from various graphs shown in this section. Hence, the difference between correct predictions (CP) and the expected value (E) is always positive. Thus, it can be inferred that the proposed model is good for web page prediction.

To summerize, we discussed and elaborated on hashing, explained hash based prediction model with details and validated the results obtained by prediction in this chapter. The prediction model was dependent on the clustering technique. Hence, the proposed prediction model is modified, to improve the accuracy. The same model

could be used for web page recommendation system, as well as for web site restructure, as both these applications would benefit from the good prediction model. For example, suppose the prediction model predicts that, the page P_j is visited immediately after P_i; whenever a user accesses page P_i , P_j could be recommended to the user indicating that majority of users have accessed P_i and P_j in succession. Similarly, if there is no direct link to page P_j from page P_i , the web site designer may change the site structure accordingly. Thus, the knowledge discovered by analyzing the user access pattern is useful for prediction, recommender system as well as to reorganize the structure of a given web site.

Chapter 5

WEB PAGE RECOMMENDER MODEL

Web page recommender system assists user by providing recommendations to ease their navigation through a web site. Recently, many recommender systems have been developed to discover web pages that are useful to the user. This chapter discusses a novel recommender system which adopts the concept of collaborative filtering. The performance of proposed recommender system is evaluated based on precision and recall metrics. Also, results obtained are encouraging in terms of precision and recall compared to couple of other results in the literature.

5.1 Introduction

The number of people who use internet is increasing at a rapid pace. The users are enforced to spend more time in looking for appropriate and desired information from huge quantity of web pages available over internet. Web page recommender system addresses this problem by providing suitable suggestions to user. The goal of recommender system is to determine the web pages relevant to a user, based on user's present action and the actions performed by other users earlier.

A recommender system plays an important role in various applications like e-commerce. Adnan et al. (2011) (Schafer 1999) integrated data mining techniques

and social network techniques to analyze web server logs to assist web site owners to understand the visitors behavior. This may hep the site owner to decide placement of advertisements for new products, announce discounts, give special offers, provide link to popular products etc. so that, the customers can take advantage of new schemes or offers, which may not be known otherwise. The owner will be benefitted from extra sales made through recommendations and also there is a chance of converting a mere browser into a potential customer. Thus recommendation helps both customers as well as web site owners.

As people participate actively in social networking and peer-production sites, implicit relations may emerge from various activities (Li et al. 2012). Discovering such relations, by mining the users' activities, leads to better recommendations. Also, people depend on information available on the web for varius activities like financial, health, career, education etc. regularly. Even search engines play a passive role by trying to provide relevant information that is explicitly asked or requested by the user. Many times a user may not know existence or availability of some useful information but, the recommender system may suggest such useful information to the user implicitly based on past activities. Thus, recommender systems provide relevant and previously unknown information to the user, and hence play a significant role in many web based applications.

Yan et al. (1996) represented user session as a vector, clustered sessions by using leader algorithm and dynamically suggested links to user. Mobasher et al. (2002) presented techniques to discover aggregate profiles that can be used by recommender systems for real time web personalization. Gunduz et al. (2003) proposed a new model that considers both the order information of pages in a session and the time spent on them. The click-stream tree of the cluster is used to generate the recommendation set. Mobasher et al. (2001) proposed effective and scalable techniques for web personalization based on association rule discovery from usage data. Forsati et al. (2009) proposed a web recommendation system based on the weighted association

rule (WAR) model. They extended the association rule mining by assigning a significant weight to the pages based on time spent by each user on each page and visiting frequency of each page. Kazienko (2009) mined indirect association rules for web recommendation. Suryavanshi et al. (2005a) used relational fuzzy subtractive clustering as the first level modeling and then mined association rules within individual clusters. They proposed a two level model-based technique, which is scalable and is an enhancement over association rule based recommender systems. Kumar et al. (2010) proposed a recommender system based on fuzzy association rule mining. Yong et al. (2005) proposed algorithm for pruning sequence association. Thus association rules is used by many to develop recommender systems. Using association rules for web page recommendation involves too many rules and difficult to find a suitable subset of rules to make accurate and reliable recommendations (Ping et al. 2010) and hence, there is a need to look at other alternative methods.

The other common methods used for recommender system to improve its performance are clustering (Mobasher et al. 2002) (Peng 2006), graph based (Wang et al. 2008), web content based (Salin 2009) (Li 2004). Various hybrid models are also proposed to overcome the limitations of these individual methods. For example, Goksedef et al. (2010) proposed a hybrid recommender system that combined results of several recommender techniques based on web usage mining. Kim et al. (2004) proposed a hybrid model that includes Markov model, sequential association rule, association rule and a default model that recommends based on frequency from a whole data set. Sobecki (2007) proposed a hybrid recommendation method based on the ant colony metaphor. Golovin et al. (2004) combined different algorithms (e.g. top N, sequence patterns, collaborative filtering) in a single recommendation database. Though the intention of hybrid models is to overcome drawback of each of the individual model, they require more time for both off line as well as on line phase becaue of applying individual models sequentially.

Collaborative filtering is a well known technology that uses past navigation behavior to generate web pages as recommendations to the user (Bell 2007). Collaborative filtering uses the known preferences of a group of users to make recommendations of the unknown preferences for other users (Su 2009). The conventional collaborative filtering method finds recommendations from the complete data base of user sessions that lead to scalability problem. The proposed method uses clustering as well as collaborative filtering technique to achieve better performance in terms of precision and recall. The proposed SCFR system finds recommendations within the nearest cluster and not the complete data base of user sessions. As a result, SCFR system resolves scalability issue associated with the conventional collaborative filtering method. Also, the pages that are visited more than once in a session are removed since they do not contribute for recommendations.

The next section explains the proposed recommender model based on session collaborative filtering to improve performance in terms of recall and precision.

5.2 Proposed Recommender System

The goal of the present work is to develop a simple recommender system based on user sessions clustering and collaborative filtering to improve the performance over couple of existing recommender models. This section explains the activities of off line and on line components used in the proposed recommender system.

5.2.1 Session based collaborative filtering recommender (SCFR) system

Fig.5.1 depicts the session based collaborative filtering recommender system that recommends web pages to an active user, based on the previous similar sessions.

The system has offline and online components. The details of activities of these two components are discussed in the following two sections.

Off line component

The majority activities of off line component are similar to that of any other web usage mining system. The web server log maintains IP address, date, time, HTTP method, requested URL, response code, number of bytes transferred, referrer etc. for each request. The log file is parsed to extract essential fields like IP address, date, time and the requested URL. The data is cleaned by removing image files, response code with error and empty requested URL field. Unique URLs are identified from cleaned data and unique identities are assigned for each unique URL. The user sessions are

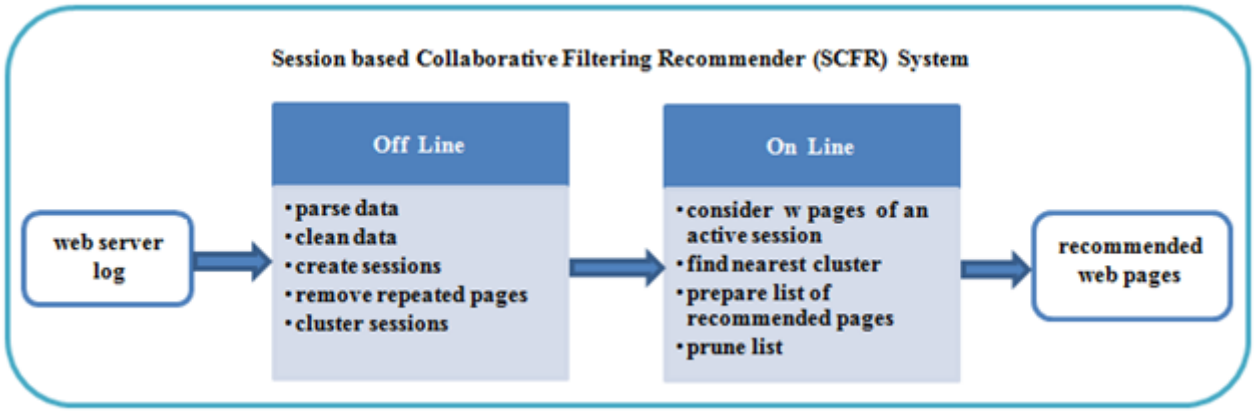

Figure 5.1: Session based collaborative filtering recommender system

created based on IP address, date and time fields. A session represents the sequence of web pages viewed by a user within certain time duration - 30 minutes (Catledge et al. 1995) (Srivastava et al. 2000). If a user views page 1, page 3, page 1, page 5 and page 6 in succession, the session is represented as $(P_1, P_3, P_1, P_5, P_6)$. Once sessions are created, repeated pages are removed since they do not contribute for recommendations. For example, in the above session page P_1 occurs twice and after removing duplicate pages the session becomes (P_1, P_3, P_5, P_6). Possible reason for

existence of repeated pages in a session is, click on back button of browser, referring to the earlier visited page.

Once sessions are created and duplicate pages are removed, they are clustered by using k-means algorithm as given in Algorithm 10.

Algorithm 10 K-means for recommender system

Input: user sessions, number of clusters n
Output: set of clusters containing user sessions
1: Select n user sessions randomly as initial centroids
2: **loop**
3: Assign each session to the nearest cluster
4: Count frequency of each page for respective clusters
5: Select pages that are accessed more frequently ($> 50\%$) and concatenate them to form new cluster centroids
6: Exit if there is no change in centroids or sessions remain in same cluster with new centroid
7: **end loop**

Algorithm 10 gives major steps of k-means, used to cluster web user sessions. Depending on the number of clusters, the initial cluster centroids are randomly selected from user sessions. Each user session is assigned to the nearest cluster, after measuring its distance with cluster centroids. For measuring the distance between any two sessions, VLVD method (see Algorithm 3) and cosine distance method (see section 5.3)are used. Once the initial clusters are formed, the new cluster centroids are computed. To get new cluster centroids, the count of each page in the respective cluster is used. The page count is determined during assigning the session to the cluster. Therefore, at the end of clustering the count for each page is available. The frequency of each page is determined, by dividing the page count by the total number of sessions in the respective cluster. The frequency value greater than 0.5, indicates that the page is accessed by more than 50% of sessions in the cluster, and hence the page is considered to be one of the frequently viewed pages of the cluster. Similarly, the other pages that are accessed more frequently ($> 50\%$) are determined and are concatenated to obtain new cluster centroids. Thus, the new centroids represent the

web pages that are accessed by more than 50% sessions in respective clusters.

On line component

During the on line phase, first 'w' pages of an active user session is taken and is assigned to the nearest cluster. Once the nearest cluster is found, the distance between active session and other sessions in the cluster are computed. The sessions that are more similar to the active session are considered, and a list of pages that are not present in the active session is prepared. This page list is further pruned based on the frequency. Thus, top 'n' pages are recommended from the cluster based on the frequency of co-occurrence in comparison with the active user session. The model is evaluated by using precision and recall as metrics and is discussed in section 5.4.

5.3 Cosine Distance Method

Cosine similarity is the most common method used to find similarities between vectors in information retrieval. The cosine similarity between two sessions is given by Eq.5.3.1 that uses dot product and magnitude to compute similarity between them. The resulting value ranges from 0 to 1, where 0 indicates the sessions are completely different and 1 results if sessions are exactly similar to each other. The steps used to find distance between two sessions is given in Algorithm 11.

$$d_{COS}(S_i, S_j) = \frac{S_i.S_j}{||S_i||\ ||S_j||} \tag{5.3.1}$$

Algorithm 11 Cosine similarity between two user sessions

Input: two web user sessions S_i and S_j
Output: distance d between S_i and S_j
$l_1 \leftarrow number\ of\ pages\ in\ session\ S_i$
$l_2 \leftarrow number\ of\ pages\ in\ session\ S_j$
$c \leftarrow number\ of\ common\ pages\ accessed\ by\ sessions\ S_i\ and\ S_j$
$d_{COS}(S_i, S_j) \leftarrow c/(sqrt(l_1) \times sqrt(l_2))$

5.4 Evaluation Metrics

The proposed SCFR system is evaluated by using precision and recall as metrics (Wei 2009). The precision is number of recommended pages that are relevant and the recall is number of pages that are correctly recommended. Let 'RP' be the total number of pages recommended and 'n' denote the length of an active session. First 'w' number of pages of an active session is considered and assigned to the nearest cluster. The remaining part of an active session, after first 'w' pages is denoted by (n-w). Using the above given notations, precision and recall of a session are defined as given in Eq.5.4.1 and 5.4.2 respectively.

$$precision(s_i) = \frac{RP \cap (n-w)}{RP} \tag{5.4.1}$$

$$recall(s_i) = \frac{RP \cap (n-w)}{(n-w)} \tag{5.4.2}$$

The 40% of sessions are considered as test set to evaluate the proposed recommender system. If 'TS' is number of test sessions considered for evaluation, overall or average precision and recall are given by Eqs.5.4.3 and 5.4.4 respectively.

$$precision(TS) = \frac{\sum_{i=1}^{TS} precision(s_i)}{TS} \tag{5.4.3}$$

$$recall(TS) = \frac{\sum_{i=1}^{TS} recall(s_i)}{TS} \tag{5.4.4}$$

5.5 Experimental Results

The MM data set is considered first for evaluating the proposed system with 5000 and 10000 sessions. In both the cases the total sessions are divided into 60:40 ratios. 60% of sessions are considered for building the recommender system as shown in the off line component of Fig.5.1 and remaining 40% sessions are taken as test data. The average session length is 7.91 and 7.96 for 5000 and 10000 sessions respectively.

For each test session, nearest cluster is found based on first 'w' pages and 'w' is assumed as 3. A list of recommended pages is generated by comparing the 'w' pages of test session with other sessions of cluster based on cosine and VLVD similarity. The list with recommended pages is pruned based on frequency. The experiment is conducted for various size of recommended list ranging from top 1 to top 10. The precision and recall are calculated by using Eqs.5.4.1 and 5.4.2 respectively. The average or overall performance is determined by using Eqs.5.4.3 and 5.4.4 respectively.

Figs.5.2, 5.3, 5.4 and 5.5 show precision and recall values for 5000 and 10000 sessions by using cosine and VLVD distance measures respectively. The recall increases as the number of recommended pages increases while, the precision decreases as the number of recommended pages increases. This is because the average session length is 8 and first 3 pages are not considered for recommendation. The recommendation is given to the last 5 pages of session and maximum of 5 pages could be accessed by user though the recommendation list contains more than 5 pages. Therefore, the precision reduces as the recommendation list size increases. The results are almost consistent for both cases of cosine and VLVD distance measures as can be seen from Figs. 5.2, 5.3, 5.4 and 5.5. Though the results are almost same, it could be inferred that cosine similarity is not efficient than VLVD because of square root operation.

Similarly NASA data set for the month of July is considered for experiment purpose. The proposed system is evaluated by using 5000, 10000 and 15000 sessions. The average session lengths of these sessions are 6.68, 6.77 and 6.82 respectively. Figs. 5.6, 5.7, 5.8 and 5.9 show recall and precision values for 5000, 10000 and 15000

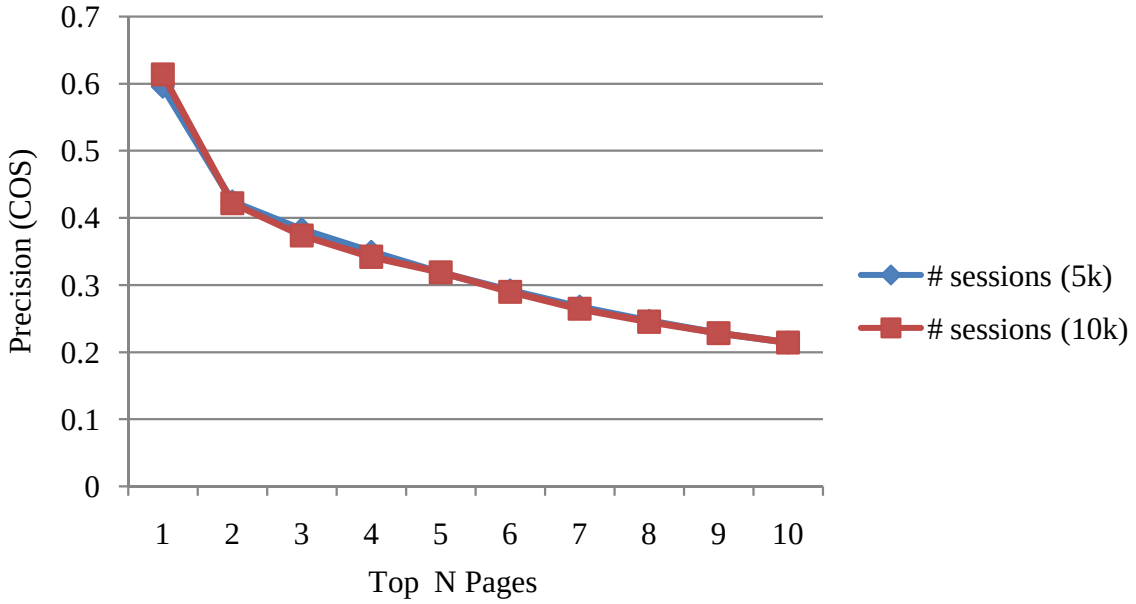

Figure 5.2: Precision for 5k and 10k sessions with cosine similarity as distance measure

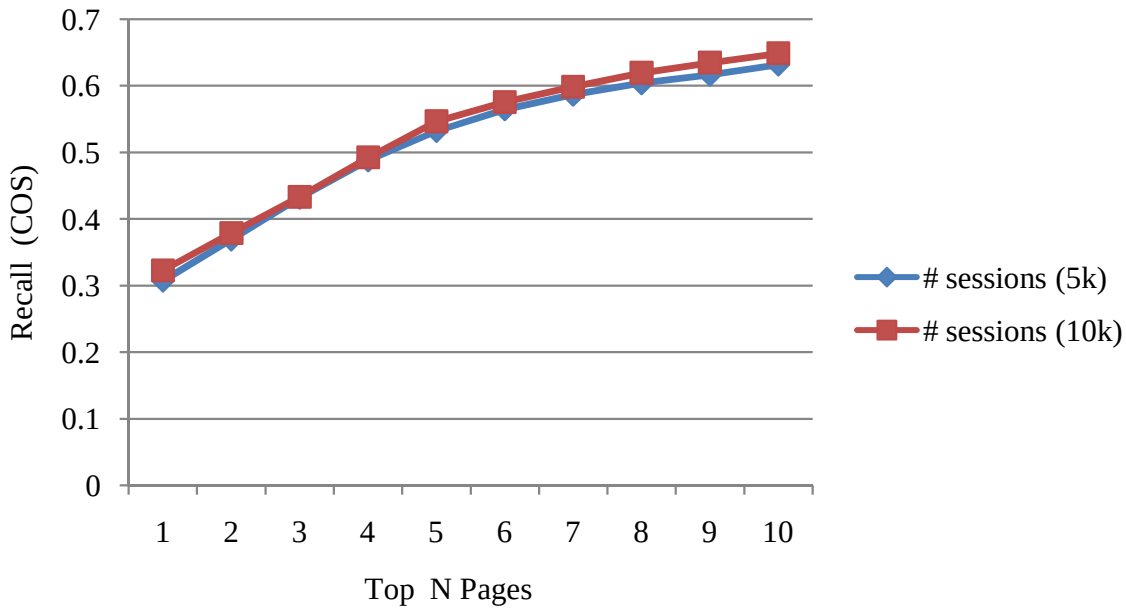

Figure 5.3: Recall for 5k and 10k sessions with cosine similarity as distance measure

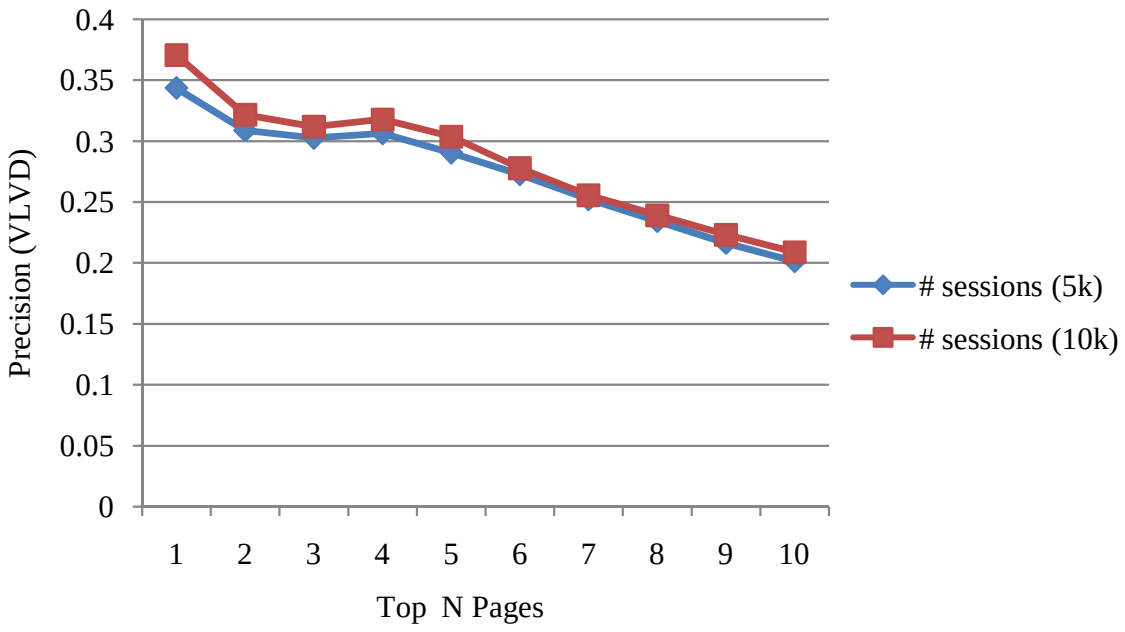

Figure 5.4: Precision for 5k and 10k sessions with VLVD similarity as distance measure

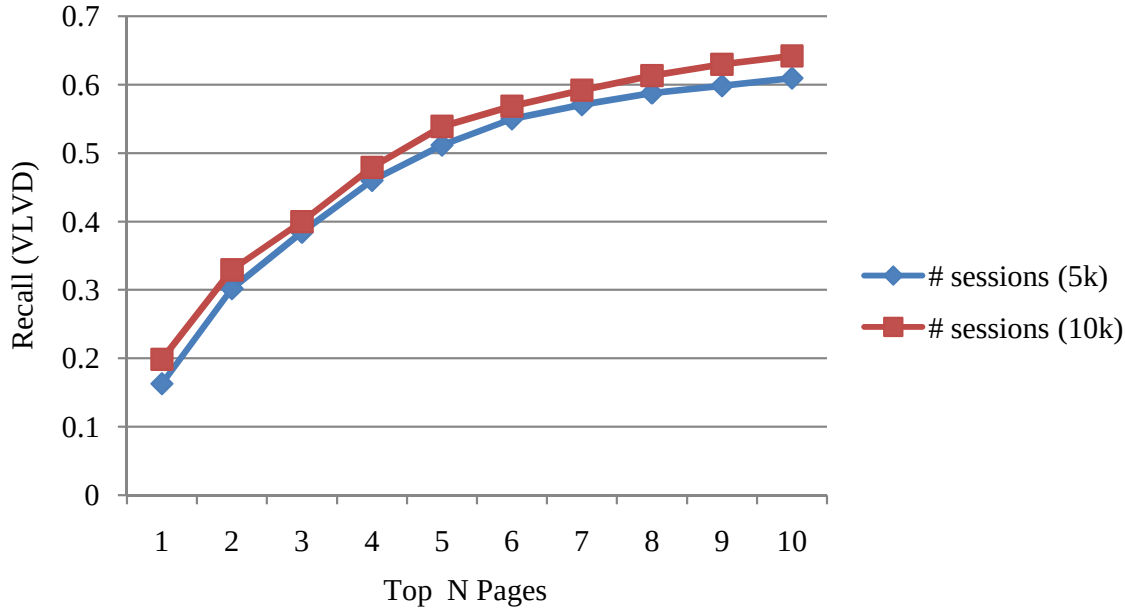

Figure 5.5: Recall for 5k and 10k sessions with VLVD similarity as distance measure

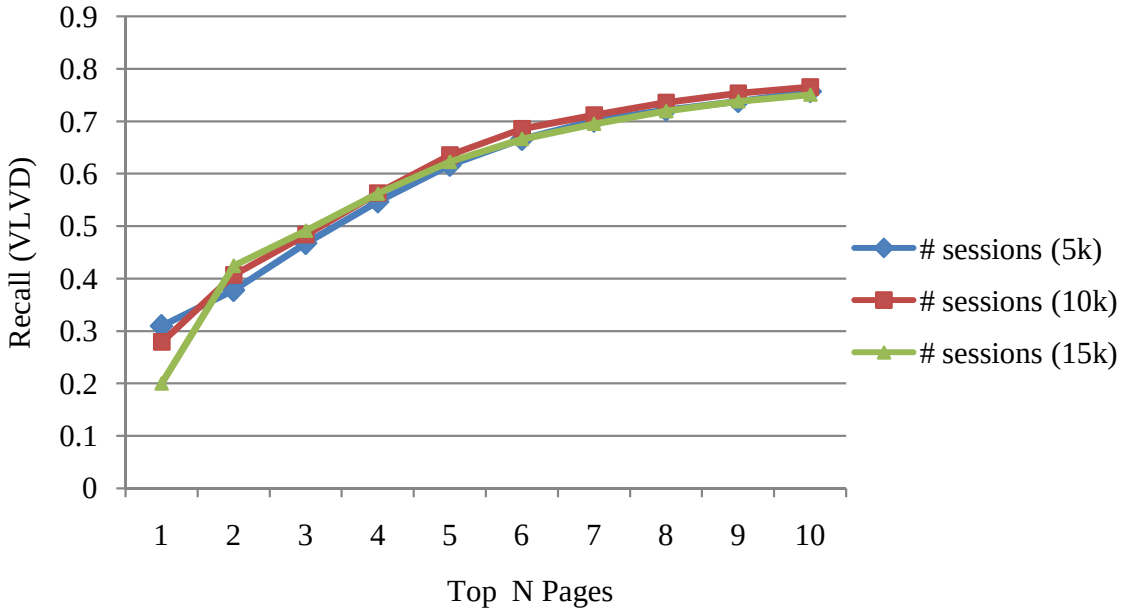

Figure 5.6: Recall for 5k, 10k and 15k sessions with VLVD as distance measure

sessions with VLVD and cosine similarity as a distance measure. Again the results are consistent for various session sizes for NASA data set also. The recall is more than 60% for top 5 pages that indicates goodness of the proposed SCFR system.

Kim et al. (2004) proposed a hybrid click-stream based collaborative model. They used four different existing models namely, Markov model, sequential association rule, association rule and default model. The default model recommends pages based on frequency from whole data set. The experiment is done by applying these models in sequence with different combinations. For example, SMAD hybrid model applies sequential association rule first to the active session. If sequential association rule does not cover the active session, Markov model is used. If Markov model fails, association rule is used. If association rule also fails, finally default model is used. They evaluated performance of the model over varying top number of pages recommended by the model. They used web server log of NASA data set for the period from July 1, 1995 to July 5, 1995 as training data set and July 6, 1995 as test data set. To compare the proposed SCFR system with hybrid model, the same data set is used. The recall

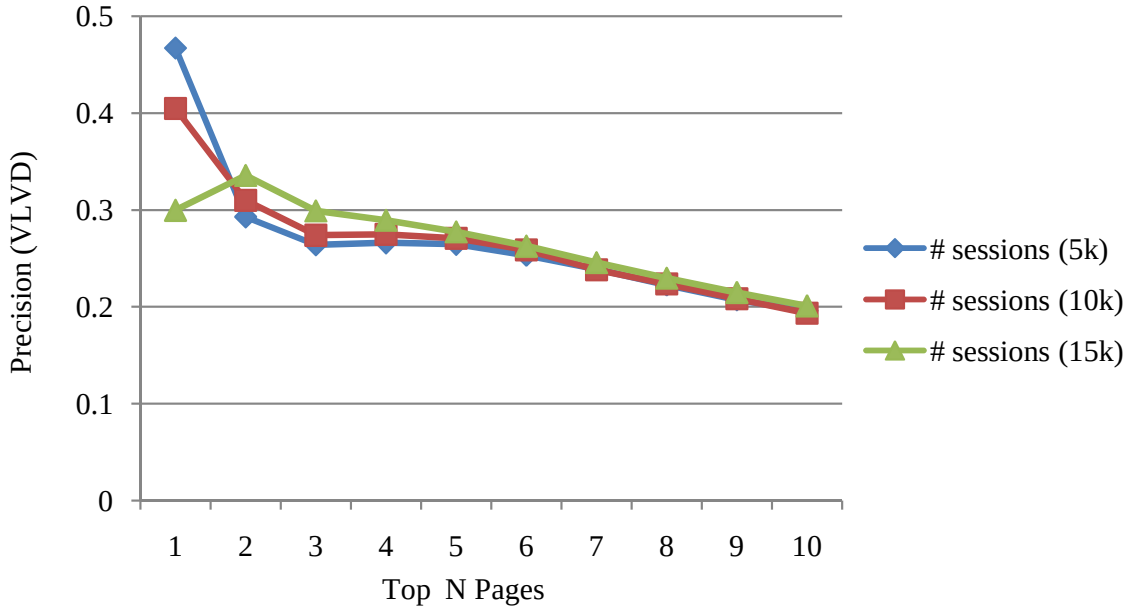

Figure 5.7: Precision for 5k, 10k and 15k sessions with VLVD as distance measure

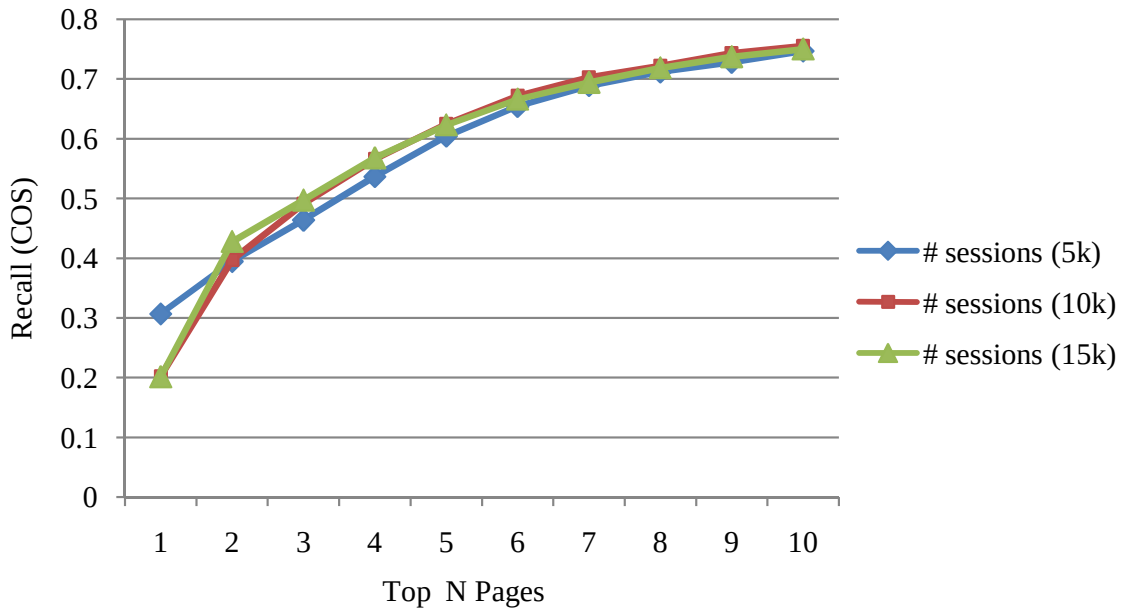

Figure 5.8: Recall for 5k, 10k and 15k sessions with cosine similarity as distance measure

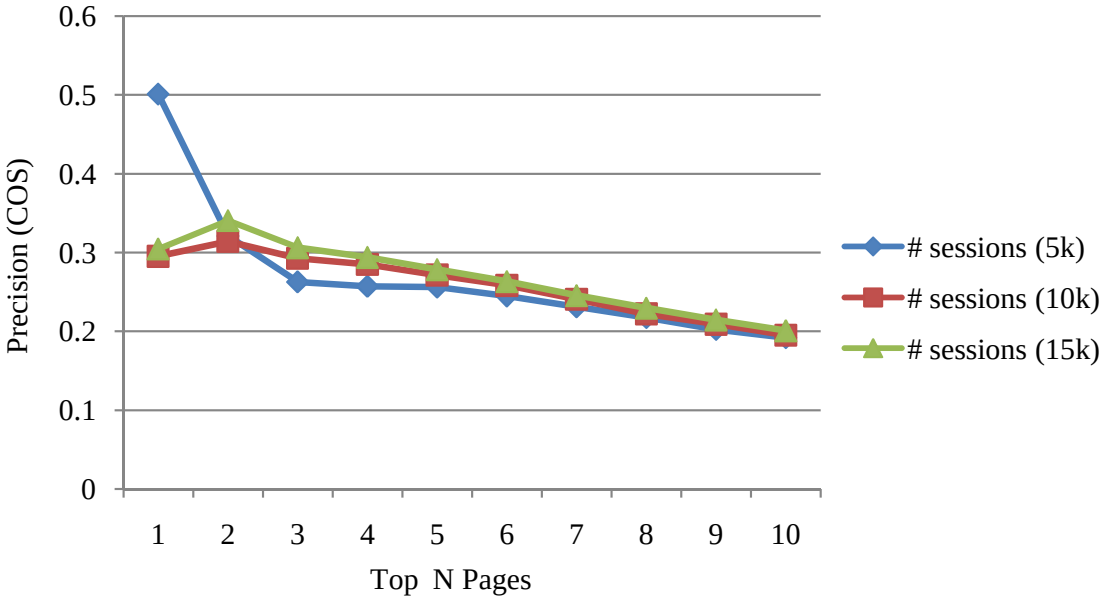

Figure 5.9: Precision for 5k, 10k and 15k sessions with cosine similarity as distance measure

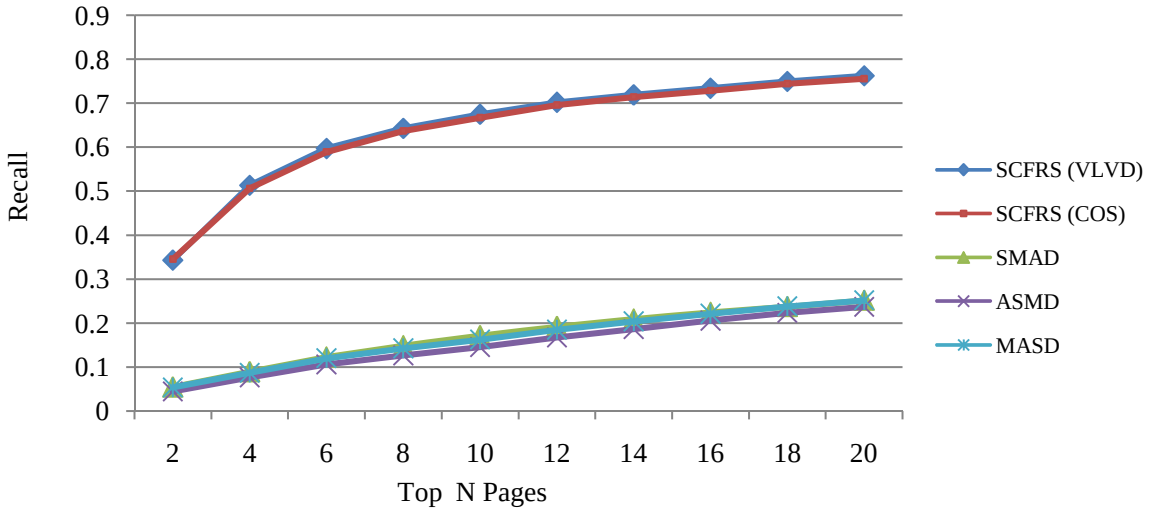

Figure 5.10: Comparison of proposed SCFR system with the hybrid model (recall)

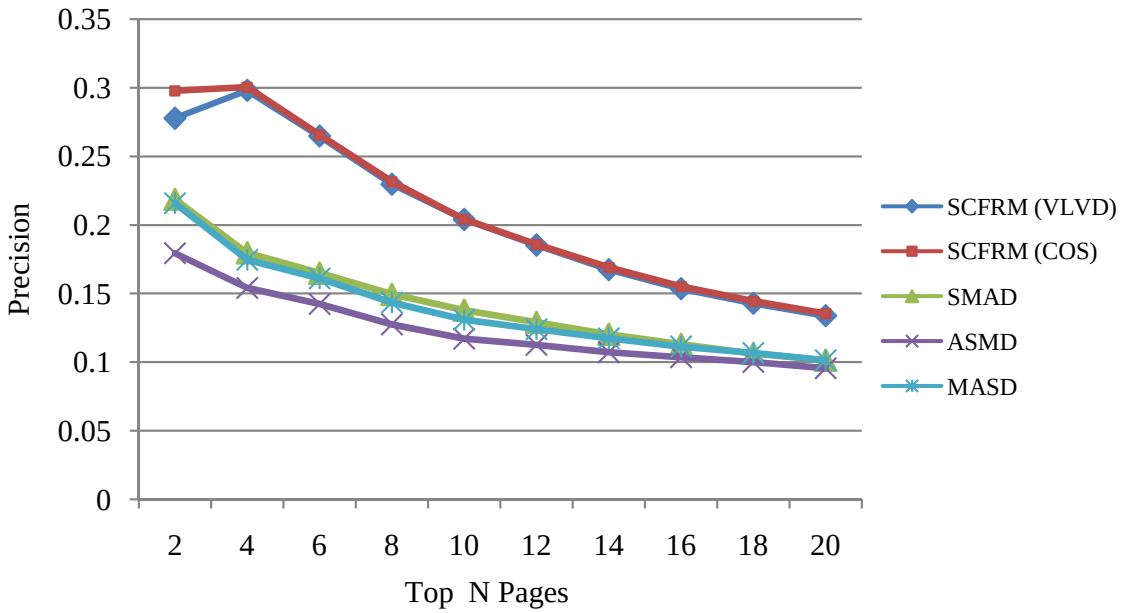

Figure 5.11: Comparison of proposed SCFR system with the hybrid model (precision)

and precision of all these models are depicted in Figs.5.10 and 5.11 respectively. The performance of the VLVD and cosine similarity measures used in SCFR system, are almost same and are better than the various combination of models used in the hybrid model for both recall and precision as can be seen from Table 5.1 and 5.2. The graphs of Figs.5.10 and 5.11 clearly show the goodness of the proposed SCFR system compared to other hybrid models in terms of both recall and precision. Also, for online recommendation the hybrid model may take more time because, it tries to cover the part of active session by various models in sequence and in the worst case all the four models may be required to give recommendations. In the proposed SCFR system the active session is assigned to the nearest cluster and recommendations are suggested based on the cluster to which active session is assigned. Hence, proposed SCFR system is efficient compared to the hybrid click-stream based collaborative model.

To summarize, an efficient recommender system based on collaborative filtering and user session clustering is discussed in this chapter. The proposed SCFR system

Table 5.1: Recall of the various models for top n pages

	Top 2	Top 4	Top6	Top 8	Top 10	Top 12	Top 14	Top 16	Top 18	Top 20
$SCFRS_{VLVD}$	0.3432	0.5131	0.5969	0.6424	0.6745	0.7018	0.7188	0.7338	0.7493	0.7619
$SCFRS_{COS}$	0.3454	0.5064	0.5894	0.6368	0.6673	0.6960	0.7143	0.7287	0.7440	0.7557
SMAD	0.0542	0.0892	0.1227	0.1483	0.1710	0.1920	0.2089	0.2244	0.2373	0.2516
ASMD	0.0445	0.0764	0.1059	0.1266	0.1453	0.1675	0.1861	0.2056	0.2234	0.2370
MASD	0.0536	0.0867	0.1199	0.1423	0.1623	0.1847	0.2037	0.2211	0.2381	0.2516

Table 5.2: Precision of the various models for top n pages

	Top 2	Top 4	Top6	Top 8	Top 10	Top 12	Top 14	Top 16	Top 18	Top 20
$SCFRS_{VLVD}$	0.2777	0.2979	0.2647	0.2296	0.2039	0.1853	0.1673	0.1534	0.1428	0.1338
$SCFRS_{COS}$	0.2978	0.3003	0.2656	0.2315	0.2041	0.1858	0.1690	0.1552	0.1444	0.1355
SMAD	0.2186	0.1797	0.1648	0.1494	0.1378	0.1290	0.1203	0.1131	0.1063	0.1014
ASMD	0.1794	0.1540	0.1423	0.1276	0.1171	0.1125	0.1072	0.1036	0.1000	0.0955
MASD	0.2159	0.1748	0.1610	0.1434	0.1309	0.1240	0.1173	0.1114	0.1066	0.1014

is evaluated using the standard metrics, recall and precision. The results obtained are compared with the hybrid recommender system that is based on Markov model, sequential association rule, association rule and the default mode. The results clearly illustrate the goodness of proposed recommender system compared to the hybrid recommender model with respect to precision and recall.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

With the exponential growth of the web-based applications, there is a significant interest in analyzing the web usage data for the task of understanding the users web page navigation and apply the outcome knowledge to better serve the needs of user. Here, a modified k-means algorithm to cluster web user sessions is proposed. The VLVD function that computes the distance between user sessions and considers the uneven lengths of sessions is discussed. The SABDM method, which pays attention to the dissimilar length of web page sessions as well as uses the information of order in which the web pages are visited by the user in a session, is explained. Further, HSAM distance measure, which is an extended version of SABDM, by integrating with the SAM method, is described. Two prediction models that achieve good prediction accuracy are explained. Finally, a recommender model based on session collaborative filtering is discussed. The efficiency of the proposed recommender model is evaluated, based on precision and recall. The outcome of the prediction as well as recommendation model could be used to suggest structural changes to the web site. All the methods are evaluated by using standard statistical measures and

also compared with few other results available in the literature, to demonstrate the goodness of these proposed methods.

6.2 Future Work

The models proposed for prediction and recommendation, compare the part of active user session with the nearest cluster to give prediction or recommendation. In the proposed model, the active user session is compared with all the cluster centers and sessions of the nearest cluster are compared with the active session. This could be further improved by finding the frequent patterns of various clusters during the off line phase. Instead of comparing the active session with each session of a cluster, comparison could be limited to only frequent patterns which are less in number. This may avoid the time taken to compare active session with other sessions of cluster.

The future work may consider implementing incremental based clustering technique to cluster user sessions whenever server log is updated. The present work considers the existing server web logs. The web log keeps on growing in size as the number of users who use the web increases. Incremental based clustering technique could be developed to address this issue. That means, the existing clusters created by the static web log, could be reconstructed based on, only the new log records added in the server log. The new sessions could be assigned to a cluster by considering the cluster representatives stored in memory, without looking at the previously seen patterns. This makes clustering algorithm more scalable and avoids re-clustering the entire data set.

REFERENCES

Adnan, M., Nagi, M., Kianmehr, K., Tahboub, R., Ridley, M., and Rokne, J. (2011). "Promoting where, when and what? An analysis of web logs by integrating data mining and social network techniques to guide ecommerce business promotions." *J. Soc. Netw. Anal. Min.*, Springer, 1(3), 173-185.

Anitha. (2010). "A new web usage mining approach for next page access prediction." *Int. J. Computer Applications*, 8(11), 7-10.

Awad, M. A., and Khan, L. R. (2007). "Web navigation prediction using multiple evidence combination and domain knowledge." *IEEE Trans. Systems, Man and Cybernetics-part A:Systems and Humans*, 37(6), 1054-1062.

Awad, M. A., and Khan, L. R. (2008). "Predicting www surfing using multiple evidence combination." *The VLDB Journal*, 17(3), 401-417.

Bell, R. M., Koren, Y., and Volinsky, C. (2007). "Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems." *Proc. The 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, ACM, 95-104.

Brudno, M., Malde, S., Poloakov, A., Do, C. B., Courancne, O., Dubchak, I., and Batzogiou, S. (2003). "Glocal alignment:finding rearrangements during alignment." *J. Bioinformatics*, 19(1), 54-62.

Catledge, L., and Pitkow, J. (1995). "Characterizing browsing behaviors on the world wide web." *J. Computer Networks and ISDN Systems*, 27(6), 1065-1073.

Chaofeng, L., and Yansheng, L. (2007). "Similarity measurement of web sessions based on sequence alignment." *J. Natural Science*, 12(5), 814-818.

Chu Hui, l., and Yu Hsiang, F. (2008). "Web usage mining based on clustering of browsing features." *Proc. The 8th Int. Conf. Intelligent Systems Design and Application*, IEEE, 281-286.

Cooley, R., Mobasher, B., and Srivastava, J. (1997a). "Grouping web page references into transactions for mining world wide web browsing patterns." *Proc. The IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97)*, 2-9.

Cooley R, Mobasher, B., and Srivastava, J. (1997b). "Web mining:information and pattern discovery on the world wide web." *Proc. The 9th IEEE Int. Conf. Tools with Artificial Intelligence (ICTAI'97)*, 558-567.

Cormen, T. H., Leiserson, C. E., Rivest, R. I., and Stein, C. (2009). "Introduction to algorithms." *PHI Learning Private Limited, New Delhi*, 2nd Edition.

Davies, D. L., and Bouldin, D. W. (1979). "A cluster separation measure." *IEEE Trans. Pattern Anal. Machine Intell.*, 1(4), 224-227.

Deshpande, G., and Karypis. (2004). "Selective markov models for predicting web page accesses." *ACM Trans. Internet Technology*, 4(2), 163-184.

Dixit, D., and Gadge, J. (2010). "A new approach for clustering of navigation patterns of online users. *Int. J. Engineering Science and Technology*, 2(6), 1670-1676.

Dutta, R., Kundu, A., Dattagupta, R., and Mukhopadhyay, D. (2009). "An approach to web page prediction using markov model and web page ranking." *J. Convergence Information Technology*, 4(4), 61-67.

Facca, F.M., and Lanzi, P.L. (2005). "Mining interesting knowledge from weblogs: a survey." *J. Data and knowledge Engg.*, Elsevier, 53(3), 225-241.

Forsati, R., Meybodi, M. R., and Rahbar, A. (2009). "An efficient algorithm for web recommendation systems." *Proc. IEE/ACM Int. Conf. Computer Systems and*

Applications, 579-586.

Fu, Y., Sandhu, K., and Shih, M. (1999). "Clustering of web users based on access patterns." *Lecture Notes in Artificial Intelligence* Springer, 1836, 21-38.

Fuqua School of Business year. (2005). "Whats a good value for R-squared?" http://www. duke.edu / rnau/rsquared.html, (June, 2011).

Giannotti, F., Gozzi, C., and Manco, G.(2002). "Charaterizing web user accesses: a transactional approach to web log clustering." *Proc. Int. Conf. Information Technology:Coding and Computing (ITCC'02)*, IEEE, 312-317.

Goksedef M., and Gunduz, S. (2010). "Combination of web page recommender systems." *J. Expert Systems with Applications*, 37(4), 2911-2922.

Golovin, N., and Rahm, E. (2004). "Reinforcement learning architecture for web recommendations." *Proc. Int. Conf. Information Technology: Coding and Computing (ITCC-2004)*, IEEE, 398-402.

Gunduz, S., and Tamer, O. M. (2003). "A web page prediction model based on click-stream tree representation of user behavior." *Proc. The 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 535-540.

Guo, Y. J., Ramamohanarao, K., and Park, L. A. F. (2007). "Personalized page rank for web page prediction based on access time-length and frequency." *IEEE/WIC/ACM Int. Conf. Web Intelligence*, 687-690.

Halkidi, M. V. M., and Batistakis, Y. (2001). "On clustering validation techniques." *J. Intelligent Information Systems*, 17(2-3), 107-145.

Hamming., and Richard, W. (1950). "Error detecting and error correcting codes." *Bell Systems Technical Journal*, 29(2), 147-160.

Han, J., and Kamber, M. (2006). "Data Mininng Concepts and Techniques." *Morgan Kaufmann Publishers, Elsevier Inc.*, 2nd Edition.

Hay, B., Wets, G., and Vanhoof, K. (2004). "Mining navigation patterns using a sequence alignment method." *J. Knowledge and Information Systems*, Springer, 6(2),150-163.

Hofgesang, P. I. (2006). "Methodology for preprocessing and evaluating the time spent on web pages." *Proc. IEEE/WIC/ACM Int. Conf. Web Intelligence*, IEEE, 218-225.

Hu, W., Zong, X., Lee, C., and Yeh, J. (2003) World wide web usage mining systems and technologies. *J. Systemics, Cybernetics and Informatics*, 1(4),53-59.

Jalali, M., Mustapha, N., Sulaiman, N. B., and Mamat, A. (2008a). "A web usage mining approach based on LCS algorithm in online predicting recommendation systems." *The 12th Int. Conf. Information Visualisation*, IEEE, 302-307.

Jalali, M., Mustapha, N., Mamat, A., and Sulaiman, N. B. (2008b). "A new classification model for online predicting users' future movements." *Int. Symp. Information Technology*, IEEE, 4, 1-7.

Kazienko, P. (2009). "Mining indirect association rules for web recommendation." *Int. J. Appl. Math. Comput. Sci.*, 19(1), 165-186.

Kevinmacdonell. (2010). "How high, R-squared? " http://cooldata.wordpress.com/ 2010/ 04/19/how-high-r-squared/, (June, 2011).

Khalil, F., Li, J., and Wang, H. (2008). "Integrating recommendation models for improved web page prediction accuracy." *Proc. The 31st Australian Conf. Computer Science (ACSC'08)*, 74, 91-100.

Khalil, F., Li, J., and Wang, H. (2009). "An integrated model for next page access prediction." *Inderscience Enterprises Ltd.*, 1-18.

Khasawneh, N., and Chan, C. (2007). "Multidimensional sessions comparison method using dynamic programming." *Proc. The 4th Int. Conf. Innovations in Information Technology*, IEEE, 581-585.

Kim, D., Lm, L., Adam, N., Atluri, V., Bieber, M., and Yesha, Y. (2004). "A click stream-based collaborative filtering personalization model: towards a better performance." *Proc. The 6th Annual Int. Workshop on Web Information and Data Management*, ACM, 88-95.

Krol, D., Scigajlo, M., and Trawinski, B. (2008). "Investigation of internet system user behavior using cluster analysis." *Proc. The 7th Int. Conf. Machine Learning and Cybernetics*, IEEE, 3408-3412.

Kumar, A., and Tambidurai, P. (2010). "Collaborative web recommendation systems based on an effective fuzzy association rule mining algorithm." *Indian J. Computer Science and Engg.*, 1(3), 184-191.

Li, C. (2008). "Algorithm of web session clustering based on increase of similarities." *Proc. Int. Conf. Information Management, Innovation Management and Industrial Engg.*, IEEE, 316-319.

Li, C., Datta, A., and Sun, A. (2012). "Minig latent relations in peer-production environments: a case study with Wikipedia article similarity and controversy." *J. Soc. Netw. Anal. Min.*, Springer, 2(3), 265-278.

Li, J., and Zaiance, O. R. (2004). "Combining usage, content and structure data to improve web site recommendation." *Lecture Notes in Computer Science : E- commerce and Web Technologies*, Springer, 3182, 305-315.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). "Understanding of internal clustering validation measures." *Proc. The IEEE 10th Int. Conf. Data Mining (ICDM 2010)*, 911-916.

Lu, L., Dunham, M., and Meng, Y. (2005). "Discovery of significant usage patterns from clusters of click stream data." *WebKDD '05*, ACM, 139-142.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2001). "Effective personalization based on association rule discovery from web usage data." *Proc. The 3rd Int. Workshop on Web Information and Data Management*, ACM, 9-15.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2002). "Discovery and evaluation of aggregate usage profiles for web personalization." *J. Data Mining and Knowledge Discovery*, Springer, 6(1), 61-82.

Mojica, J, A., Rojas, D, A., Gomez, J., and Gonzalez, F. (2005). "Page clustering using distance based algorithm." *The 3rd Latin American Web congress (LA-WEB05)*, IEEE.

Mukhopadhyay, D., Mishra, P., Saha, D., and Kim, Y. (2006). "A dynamic web page prediction model based on access patterns to offer better user latency." *Proc. The 6th Int. Workshop MSPT*, Youngil Publication, 59-64.

Needleman,S, B., and Wunsch, C, D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *J. Mol. Bio.*, 48(3), 443-453.

Nina, S.P., Rahman, M. Bhuiyan, K.I., and Ahmed, K. (2009). "Pattern discovery of web usage mining." *Proc. Int. Conf. Computer Technology and Development*, 499-503.

Pallis, G., Angelis, L., and Vakali, A. (2007). "Validation and interpretation of web users' sessions clusters." *J. Information Processing and Management*, Elsevier, 43(5), 1348-1367.

Pallis, G., Angelis, L. and Vakali, A. (2005). "Model based cluster analysis for web user sessions." *ISMIS 2005*, LNAI 3488, 219-227.

Peng, Y., Xiao, G., and Lin, T.(2006). "Prediction of user's behavior based on matrix clustering." *Proc. The 5th Int. Conf. Machine Learning and cybernetics*, IEEE, 1343-1346.

Ping, W. (2010). "Web page recommendation based on Markov logic network." *Proc. The 3rd IEEE Int. Conf. Computer Science and Information Technology (ICCSIT)*, 254-257.

Pitkow, J., and Pirolli, P. (1999). "Mining longest repeating subsequences to predict world wide web surfing." *Proc. The 2nd USENIX Symp. Internet Technologies and Systems*, 139-150.

Poornalatha, G., and Prakash, S. R. (2011). "Web user session clustering using modified k-means algorithm." *Proc. ACC-2011*, CCIS 191, Springer, 243-252.

Pujari, A. K. (2001). "Data Mining Techniques." *Universities Press India*, 1st Edition.

Salin, S., and Senkul, P.(2009). "Using semantic information for web usage mining based recommendation." *Proc. The 24th Int. Symposium on Computer and Information Sciences*, IEEE, 236-241.

Schafer, J. B., Konstan, J., and Riedl, J. (1999). "Recommender systems in e-commerce." *Proc. The 1st ACM Conference on Electronic Commerce*, ACM, 158-166.

Seung-Joon, O. (2007). "Mining clusters of sequences using extended sequence element-based similarity measure." *Proc. The 2nd Int. Conf. Innovative Computing, Information and Control (ICICIC07)*, IEEE, 232-235.

Shi, P. (2009). "An efficient approach for clustering web access patterns from web logs." *Int. J. Advanced Science and Technology*, 5, 1-13.

Shyamsunder, R. K.(1998). "Algorithms: correctness of programs." *J. Resonance*, Springer India, 3(4), 15-29.

Smith, T, F., and Waterman, M, S. (1981). "Identification of common molecular subsequences." *J. Mol. Bio.*, 147(1), 195-197.

Sobecki, J. (2007). "Web-based system user interface hybrid recommendation using ant colony metaphor." *Knowledge-Based Intelligent Information and Engineering Systems*, LNCS 4694, Springer, 1033-1040.

Soniya, T. (2011). "Contingency Tables and Chi-squared Tests." www.palgrave.com/ business/taylor/ taylor1/lecturers/.../hChap10.doc, (June, 2011).

Srivastava, J., Cooley, R., Deshpande, M., and Tan, P. N. (2000). "Web usage mining: discovery and applications of usage patterns from web data." *ACM SIGKDD Explorations Newsletter*, 1(2), 12-23.

Su, X., and Khoshgoftaar, T. M. (2009). "A survey of collaborative filtering techniques." *J. Advances in Artificial Intelligence*, Hindawi, 1-19.

Sumathi, C. P., Valli, R. P., and Santhanam, T.(2010). "An application of session based clustering to analyze web pages of user interest from web log files." *J. Computer Science*, 6(7), 785-793.

Suryavanshi, B. S., Shiri, N., and Mudur, S. P. (2005a). "Improving the effectiveness of model based recommender systems for highly sparse and noisy web usage data." *Proc. The 2005 IEEE/WIC/ACM Int. Conf. Web Intelligence*, 618-621.

Suryavanshi, B. S., Shiri, N., and Mudur, S. P. (2005b). "A fuzzy hybrid collaborative filtering technique for web personalization." *Proc. The 3rd Workshop, Intelligent Techniques for Web Personalization (ITWP'05)*, Edinburgh, Scotland.

Tan, P. N., and Kumar, V. (2002). "Discovery of web robot sessions based on their navigational patterns." *J. Data Mining Know. Disc.*, Springer, 6(1), 9-35.

Tseng, V.S., Lin, K. W., and Chang, J. (2008). "Prediction of user navigation patterns by mining the temporal web usage evolution." *J. Soft. Comput.*, 12(2), 157-163.

Umapathi, C., and Raja, J. (2008). "Discovering frequent patterns and trends by applying web mining technology in web log data." *Int. J. Soft. Comput.*, 3(2), 99-105.

Wang, W., and Zaiane, O. R. (2002). "Clustering web sessions by sequence alignment." *Proc. The 13th Int. Workshop on Database and Expert Systems Applications, DEXA '02.*, IEEE, 394-398.

Wang, Y., Dai, W., and Yuan, Y. (2008). "Website browsing aid: a navigation graph-based recommendation system." *J. Decision Support Systems*, 45(3), Elsevier, 387-400.

Wei, L., and Shu-hai, Z. (2009). "A hybrid recommender system combining web page clustering with web usage mining." *Proc. Int. Conf. Computational Intelligence and Software Engineering*, IEEE, 1-4.

Xing, D., and Shen, J. (2004). "Efficient data mining for web navigation patterns." *J. Inform. Softw. Tech.*, 46(1), 55-63.

Xu, J., and Liu, H. (2010). "Web user clustering analysis based on k means algorithm." *Proc. The Int. Conf. Information, Networking and Automation (ICINA)*, IEEE, 26-29.

Yilmaz, H., and Senkul, P. (2010). "Using ontology and sequence information for extracting behavior patterns from web navigation logs." *Proc. IEEE Int. Conf. Data Mining Workshop*, 549-556.

Yan, T. W., Jacobsen, M., Garcis-Molina, H., and Umeshwar, D. (1996). "From user access patterns to dynamic hypertext linking." *Proc. The 5th Int. World Wide Web Conference*, Paris, 28, 1007-1014.

Yang, Q., Kou, J., Chen, F., and Li, M. (2007). "A new similarity measure for generalized web session clustering." *Proc. The 4th Int. Conf. Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, IEEE, 278-282.

Yong, W., Zhanhuai, L., and Zhang, Y. (2005). "Mining sequential association-rule for improving web document prediction." *Proc. The 6th Int. Conf. Computational Intelligence and Multimedia Applications (ICCIMA)*, IEEE, 146-151.

Zahid, A., Azeem, M. F., Ahmed, W., and Babu, A. V. (2011). "Quantitative evaluation of performance and validity indices for clustering the web navigation sessions." *J. World of Computer Science and Information Technology*, 1(5), 217-226.

PUBLICATIONS

List of Publications / Communications Based on Thesis:

1. Poornalatha, G., Prakash, S. R. (2011). "Web User Session Clustering Using Modified K-means Algorithm." *Proc. The 1st Int. Conf. Advances in Computing and Communications - ACC 2011*, Part II, CCIS(191), 243-252.

2. Poornalatha, G., Prakash, S. R. (2011). "Alignment Based Similarity Distance Measure for Better Web Sessions Clustering." *J. Procedia CS*, Elsevier, 5, 450-457.

3. Poornalatha, G., Prakash, S. R. (2011). "Clustering Web Page Sessions Using Sequence Alignment Method." *Proc. The 1st Int. Conf. Computational Intelligence and Information Technology (CIIT-2011)*, CCIS(250), 479-483.

4. Poornalatha, G., Prakash, S. R. (2013). "Web Sessions Clustering Using Hybrid Sequence Alignment Measure (HSAM)." *J. Social Netw. Analys. Mining*, Springer, 3(2), 257-268.

5. Poornalatha, G., Prakash, S. R. (2012). "Web Page Prediction by Clustering and Integrated Distance Measure." *The IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, ASONAM 2012*, 1349-1354.

6. Poornalatha, G., Prakash, S. R. (2012). "Session based Collaborative Filtering for web page Recommender (SCFR) system." *J. Computer Information*

Systems. (under review)

7. Poornalatha, G., Prakash, S. R. (2012). "Prediction Model for Prefetching Web Pages Based on the Usage Pattern." *J. Intelligent Information Systems*(under review)

Brief Bio-Data

Poornalatha G.
Research Scholar
Department of Information Technology
National Institute of Technology Karnataka, Surathkal
P.O.Srinivasanagar
Mangalore 575025

Phone: 9480483821
Email: poornalathag@gmail.com

Permanent address

Poornalatha G. *w/o* Dr.Janardhana Prabhu

"Pooja Sannidhi"

No.4-142A, Amba Road, Ambalpady

Udupi-576103

Karnataka.

Qualification

M.Tech. Computer Science, Manipal University, 2007.