2009

Improving Web Recommendations Using Web Usage Mining and Web Semantics

Neha Saxena
San Jose State University

IMPROVING WEB RECOMMENDATIONS USING WEB USAGE MINING AND WEB SEMANTICS

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

Neha Sushil Saxena

May 2009

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Teng Moh,Department of Computer Science Date

Dr. Agustin Araya, Department of Computer Science Date

Dr. Mark Stamp, Department of Computer Science Date

APPROVED FOR THE UNIVERSITY

San Jose State University

ABSTRACT

IMPROVING WEB RECOMMENDATIONS USING WEB USAGE MINING AND WEB SEMANTICS

by Neha S. Saxena

This project addresses the topic of improving web recommendations. With the immense increase in the number of websites and web pages on the internet, the issue of suggesting users with the web pages in the area of their interest needs to be addressed as best as possible. Various approaches have been proposed over the years by many researchers and each of them has taken the solution of creating personalized web recommendations a step ahead. Yet, owing to the large possibilities of further improvement, the system proposed in this report takes generating web recommendations one more step ahead. The proposed system uses the information from web usage mining, web semantics and time spent on web pages to improve the recommendations.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Teng Moh for being my advisor and mentor. His guidance and motivation proved invaluable without which the completion of this project would have not been possible.

I would also like to thank my committee members, Dr. Agustin Araya and Dr. Mark Stamp, for their time and inputs provided for this work.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

In recent years popularity of the internet has grown to a great extent with nearly every person, young or old, using it for a variety of purposes. People use the internet to get information in areas of interest, do research related to work or study, get good deals for commodities or travel, increase awareness about their surroundings and the world, get latest news, etc. With each passing day, large amounts of informative web sites, web pages or web documents get added to the already huge collection. Any popular search engine returns thousands of related links to a search query. It has become difficult for users to get the most relevant information from this plethora of related information readily available. Users often spend considerable time browsing the web pages for getting the right information. If the users' intention and interest for browsing a web page can be identified, it will be easier to make available that area of information with higher priority.

Identifying user intentions not only helps personalizing their web browsing experience but also has other advantages. Intuitive web pages that a user is likely to browse can be cached and thus the retrieval time and load on network can be reduced considerably. Another advantage of knowing the users' intentions is for the e-commerce purpose. Only related and targeted marketing advertisements can be shown to the user thus resulting in increased number of customers.

This report proposes a system which evaluates user's interest based on combination of other users' browsing pattern, the content of web pages and the time spent by users on web pages.

1.1 Overview of the Project

Web mining is a process that enables finding and predicting users' interests and thus personalizes the web. Web mining deals with analyzing the user's web browsing pattern or the structure of a website or the content of web pages. A lot of research has been directed to constantly improve the process of personalizing the web, mostly by just using the browsing patterns of other users. However, with the number of web pages growing every second, personalization based only on web usage mining has the shortcoming of not taking the context of the web page into account. Thus web semantics, which defines the context of a web page, is an equally important concept to be considered. Though some researches have explored this area, there is still scope of improvement.

The system proposed in this report combines web semantics and web usage mining while further improving web recommendations by taking into consideration the time spent on each browsed web page. A user will spend more time on a web page generally when s/he finds the content of the web page interesting. Thus, the time spent on a web page while browsing is an important metric to judge the user's interest in that page and thus the importance of the page.

1.2 Organization of the Report

This report gives a detailed description about the research work and the experiments carried out. Section 2 explains the foundation concepts of the proposed system which are web personalization, web mining and web recommendations. The next section gives an

overview of the work that is done till now in this area to improve web recommendations. Section 4 describes what the current systems lack and why and how they can be taken a step further improve web recommendations. This forms the motivation of the proposed system. Next section describes the software architecture of the proposed system in detail. It gives details of the various components that make up the proposed system and gives the algorithms for them. This is followed by the explanation of the experimental setup which gives details of the technology, datasets and the metrics used to evaluate the proposed system's recommendations. Section 6 contains the evaluation results and the graphs obtained from these results. These help in comparing the proposed system to the existing systems and showing the improvement of the proposed system. The last section gives the conclusions and future work that take the area of web recommendations up one more level.

2. What is Web Personalization?

The process of providing information that is related to user's current page is known as web personalization. This information is usually displayed on the current page in the form of web page links. The idea behind web personalization is that the web page currently being browsed by a user indicates his/her interest in that topic and it is likely that the user would be interested in more similar information. For example, in case of e-commerce the related information could be about other similar products to those that the user is viewing or about products that other users who bought or viewed this product also bought. This example would also work for a research or target oriented web browsing.

The key information that is required for suggesting these similar web pages comes from the knowledge of other users who have also visited the current page as well as other pages before and after this current page. In addition to other users' browsing information, web personalization can also take advantage of the web page content, the structure of the web page or the user's profile information. All these help in creating a focused and personalized web browsing experience for the user.

Web mining is the process that helps to collect and analyze information such as users' browsing pattern, content of the web pages, structure of the web site, etc. and aid in creating a personalized web browsing experience. There are four categories into which web data are classified [1]. They are:

- Usage data: This consists of a website's usage data such as the IP address of the users that visit this website, the date and time of their visit, the complete path of the web page within this website that they visit, the outcome of their visit i.e., whether the web page was viewed successfully or whether it resulted in some error, etc. The collection of this data is store in files called a web access log. This log contains the above details for each and every visit to each and every web page.
- Content data: This represents the actual data in the web page. This could be text, image, audio or video. Text is the most widely used type of content that is mined. Content of a web page helps in identifying what the web page is about.
- Structure data: This represents the way in which the data in a web page or the web pages of a web site are organized. The links to other pages included in a web page

is one of the things that constitutes a web page structure and is of importance as the linked web pages are often those that are related to the page being considered.

- User profile data: This type of data constitutes information about the user like name, age, country, gender, etc. This information is usually acquired via registration forms or questionnaires. Such information helps in analyzing the user's likes and dislikes based on the demographic knowledge.

A general web personalization process follows the high level steps shown in the following diagram:

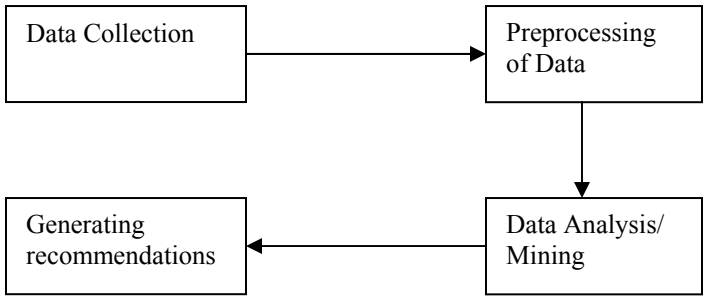

Figure 1: High Level Steps of Web Personalization

- Collection of web data: The web logs which contain details of the user visits to a website's web pages, the structure of the web site, content of the web pages, and information of the user profile constitute web data. Depending on the method of data mining employed, this information is collected to be processed and analyzed further.
- Pre processing of the collected data: During this stage, relevant and meaningful data is separated from the data collected in previous step. As an example, web

logs of a particular website contain information about all the web pages visited by all the users. These visits could have been successful or not i.e. the user may have been able to view the web page successful or it may have resulted in some error. For the next stage of analyzing the collected information only those pages should to be considered that were successful displayed to the user.

- Analyzing the pre-processed data/ web data mining: Here, the pre processed data is analyzed to gather knowledge about the user's behavior of browsing pages, the similarity of the web pages based on their content and/ or structure. This analysis helps in determining common patterns which later help in predicting web pages to current users.
- Personalizing the page: In this step, with the knowledge of browsing patterns gathered from the previous stage, current user can be recommended links to pages that are similar to the one s/he is currently viewing. Current user's browsing pattern or web page content is matched with the analyzed data of previous users and based on this comparison, current user is suggested similar web pages of interest not yet visited.

3. Previous Work

This section gives an overview of the work that has been done in the area of web personalization. Research to improve web personalization has been going on for several years. Majority of this research focuses on web usage mining techniques. Web usage mining focuses on determining patterns from users' browsing behavior and use this

pattern to personalize the browsing of the current user. Recently, web semantic mining has also been included in the efforts to improve web personalization. Below are the details of a few papers that have explored the area of web personalization.

The basic of web personalization are explained in [2]. The approach described in this paper is comprised of two components: an offline component for data preparation and usage mining and an online component which is a recommendation engine. In this paper the authors have focused on using only web usage mining and user profiles for web personalization.

The web usage logs contain information about the user's web page visits along with a lot of other attributes. The data preparation stage is responsible for extracting only that information which is necessary for the next stages of web personalization. For the system described in this paper, the data preparation stage is responsible for identifying a set of user sessions from the raw data retrieved from web servers. Only information about the user's page visits is taken into consideration. The usage mining stage determines the frequent itemsets which define the relationships between the URLs references based on user navigational patterns. These rules are formed based on how frequently a web page or a set of web pages are accessed by the users. The system proposed by authors in this paper does not generate the association rules from the frequent itemsets all at once prior to generating recommendations. Instead, the usage mining stops at finding the frequent itemsets and finds the recommendations from these itemsets. For generating recommendations, only those itemsets are considered which have items (web pages) one more than that in the navigational path for which recommendation is being generated.

The advantage of using frequent itemsets as opposed to association rules is that generating all the association rules takes a lot of memory and time. Generating frequent itemsets can be done offline and only the recommendation generation is done online i.e. as and when a new user visits the web pages the recommendations are generated for each web page visited.

Researches that focus only on web usage mining for creating personalization have the short coming of not considering the content of web pages. Web page content adds to the knowledge of the user's interest area and thus is an important factor for improving web personalization. Few of the other papers that also focus only on web usage mining are [3], [4], [5]. In [3], the authors explore the relation between the queries that the users use for searching web pages and their navigational patterns. A site-keyword graph is formed based on these two attributes based on which recommendations are generated for the new users. In [4], the authors propose to improve web personalization by considering the web usage data along with the user preferences that the users specify via registration to websites. [5], too, aims at improving web personalization using only web usage mining.

Recent papers have explored the possibility of including web semantics to improve web personalization. Papers [6], [7], [8], [9], [10], [11], [12] have targeted this area. While [6], [7], [8] have used both web usage and web semantics mining the others have solely concentrated on semantic mining where the documents are clustered together based on their content and the user is recommended pages form the cluster to which the

current document belongs. This method does not take into account the activities of previous users found in the web logs.

The approach in [8] is of much interest as it explores the ways in which the semantics of a web document along with the navigational pattern obtained from the web log can be combined and used to give better recommendations. Here, the authors first download the web documents and find its keywords using TFIDF. Each of the keywords is mapped to concept taxonomy (category) created prior to this step. The weights of the categories assigned are updated based on the similarity of the keyword to the category. Each document is then represented as a set of pairs containing the categories and the relevance (weight of categories). Document clusters of similar categories are formed. These clusters will thus contain semantically similar documents.

The web log is also modified by adding the categories to the web links thus creating concept-logs (c-logs). Data mining techniques are applied to this modified web log to get the association rule based recommendations. The recommendations, in general, could be original (recommend RHS of association rules), semantic (consider one most similar rule, find the cluster of the RHS document of rule, recommend all documents in that cluster) or category-based (represent documents as categories and recommend all documents in the RHS category). In this paper the authors have focused on semantic recommendations while in other papers [6] and [7] written by the same authors, they have focused category based recommendations. Based on experiments carried out by the authors, semantic recommendations gave the best results.

The research done in this paper can be taken a step further by considering one more attribute, the time spent on the web pages. The time spent on web pages indicates the interest level of the users. There has been some research on considering the time attribute in [13] but it has not considered web semantics for personalizing the web. Thus, considering time attribute in addition to web usage and web semantics mining is a new step in improving web personalization.

4. Motivation for Proposed System:

A lot of research has been done to improve the quality of web personalization but none of them have experimented with considering the time spent on the web pages in addition to considering web usage data and web semantics. While web usage data gives insights to the users' browsing patterns and web semantics gives insights to the web page's content, the time spent on a web page indicates the interest level of a user for the area that the web page covers.

To better understand this, consider web pages 1, 3, 4, 7, 8 which are not semantically similar to each other. Consider a user who has navigated pages 1, 3, 4, 7 and spent less than a minute, 3 minutes, 5 minutes and 10 minutes on these pages respectively and another user who has visited pages 1, 3, 4, 8 and spent less than a minute, 10 minutes, 3 minutes and 6 minutes on these pages respectively. Now suppose we have a new user who visits pages 1, 3, 4 for duration less than a minute, 10 minutes and 3 minutes. Which page recommendation would be more relevant page 7 or page 8? Recommending page 8 would make more sense and be closer to the current user’s interest.

The web personalization approaches that rely solely on web usage data would recommend both, pages 7 and 8. Those approaches that rely on web usage data and web semantics would recommend pages 7, 8 and other pages semantically similar to both these pages. But now that we have considered the time spent on the web pages, we know that the new user's interest will be more similar to those users who have spent approximately the same time on web pages browsed prior to web page 8. Thus, now when we combine web usage data and web semantics along with time spent on the web pages we get the new recommendation as web page 8 and all other pages that are semantically similar only to web page 8.

Also, instead of considering the time spent as it is, it makes more sense to have time slots and associate the web pages with the corresponding time slot. For example, one could have three time slots where time spent on a web page is: less than 2 minutes, between 2 minutes to 5 minutes and greater than 5 minutes. The range for time slots can be decided based on the average time spent on web pages of the web site being considered. The advantage of using time slots over individual time duration is that it is more flexible and does not differentiate between pages that have similar but not the same amount of time spent on them by different users.

For huge websites or in case of search engine results, every single browsed page or a single search query returns large sets of similar pages as recommendations. It is impossible to list all these suggestions as recommendations but important that the recommendations provided to the current user are extremely focused. Considering time spent narrows the suggestions list while making it even more focused to the current user's

interest.

5. Software Architecture of the Proposed System:

The software architecture of the proposed system is shown below:

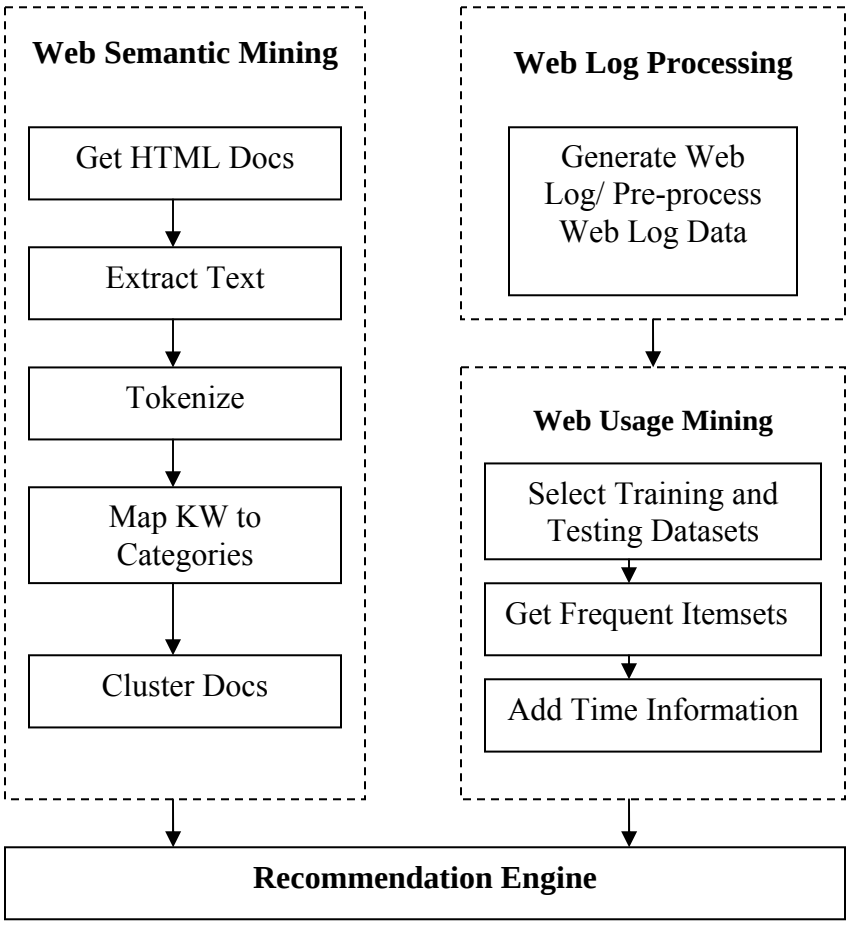

Figure 2: Architecture of Proposed System

As shown in the diagram, the proposed system consists of four components: first that generates web logs to generate the synthetic data (in case of actual web logs this component will be replaced by web log pre-processing component), second that

processes the web usage data to get frequent itemsets which help in determining users' browsing patterns, third processes the web semantics and clusters the documents based on semantic similarity and fourth that uses the second and third components to generate recommendations.

5.1 Web Logs Processing:

A web log can be considered as a very large database of transactions, where the transactions are web pages visited by a user during a single browsing session. An example of a web log entry is shown below:

123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0" 200 6248 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)" [14]

- 123.123.123.123 – represents the IP address of the user visiting the web page
- '- -' is for username/ password if user authentication is involved
- [26/Apr/2000:00:23:48 -0400] – is the timestamp which gives the details of when and at what time the web page was accessed
- "GET /pics/wpaper.gif HTTP/1.0" – mentions the type of request made by the user
- 200 – represents the result status code where 200 stands of success and 404 for error
- 6248 – is the number of bytes transferred
- "http://www.jafsoft.com/asctortf/" – is the URL of the web page accessed by the user

- "Mozilla/4.05 (Macintosh; I; PPC)" – gives details of the browser/ software used by the user to access the web page

Processing Web Logs to Clean Unwanted Data:

A web log contains a large set of above shown entries and it needs to be processed to remove unnecessary information. All the failed requests (those with result status other than 200) and those that have the request type other than "GET" are removed. A GET request represents that the web page was requested to be viewed. Other types could be POST where the user has submitted some information or HEAD which fetches only the document header. It is not possible to know different users from the web log since the only identification of the user that is present in a web log is the IP address. It could be the case that different users used the same computer (IP address) for their browsing needs or the same user used different computers at different times. Thus, in a web log a unique IP address is considered to be a unique user. Also, the time spent on a web page can be calculated by taking the difference of the time between two web page accesses. For further processing of web logs to get the usage patterns, the fields required are web page URLs visited and the time spent on them for every unique user. A single user will usually have multiple web page accesses that make up a single transaction.

Since there was no access to any real web logs this work is based on synthetic data. To generate synthetic web logs, first a tree of the web pages to be included in the web log was created. A sample web page tree used in this research is shown below:

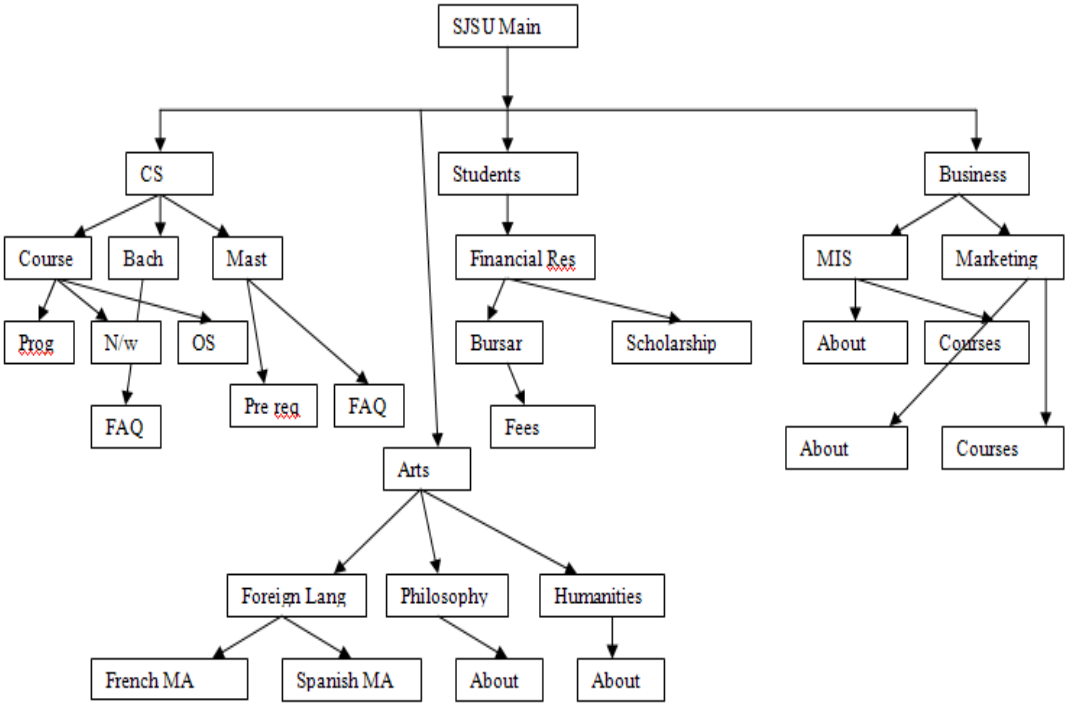

Figure 3: Sample of the Web Pages Selected

Once the tree was created, all the paths and sub paths (minimum length of three web pages) of the tree were found. These paths and sub paths will represent the navigation pattern of the users. The source code used to create a tree and generate all paths from it was got from [15]. The logic to generate sub paths was added later. For this research, three time slots indicated by ‘a’, ‘b’ and ‘c’ were used. The time slot represents little, average and large time spent on the web pages. For example, one could consider ‘a’ as 1 sec to 1 min, ‘b’ as 1 min to 5 min and ‘c’ as 5 time spent greater than 5 min. For every web page a random time slot is appended to the web page. Thus, the synthetic web log now contains web page browsing transactions with access time information appended to

each page. In an actual web log any single web page browsing pattern is duplicated by many other users. Thus, to get a web log that is similar to an actual web log, these paths were duplicated randomly. The synthetic web log now resembles an actual processed web log which contains a large dataset of repeated transactions and where each web page has the time spent on it appended to it. The algorithm to generate synthetic web log is given as follows:

1. Create a tree of the web pages
    a. For every web page create a node
    b. Add its children node to the new node created
2. Get all paths and sub paths (minimum length of three paths) of the tree starting from the second level to avoid all paths starting from the same root node
    a. Visit a node and get its children
    b. Recurse the above step for each of the child
    c. Once all paths have been found, for each path remove the last node and save the path as a new path
    d. Iterate above step until the size of path is three
3. Append random time slots to each web page of every transaction (web page browsing path)
4. Randomly duplicate the transactions to get a large transaction dataset

This large dataset was broken into a ratio of 8:2 where 80% of the transactions were used as training set to generate the frequent itemsets for generating recommendations and the other 20% of the transactions were used as testing set to test the recommendations.

The training dataset is used in the web usage mining step while the testing dataset is used by the recommendation engine.

5.2 Web Usage Mining:

In this step, the large dataset of transactions, which are previous users' web browsing information, are searched for those web pages that occur frequently. These frequent itemsets help to find interesting patterns that occur in large databases. It is using these frequent itemsets that recommendations for current users can be generated. For the research, the training dataset is used to generate frequent itemsets.

Apriori algorithm is the most popular algorithm that is used to find frequent itemsets from a large database of transactions. The algorithm is given as follows [16]:

Pass 1

1. Generate the candidate itemsets in C_1
2. Save the frequent itemsets in L_1

Pass *k*

1. Generate the candidate itemsets in C_k from the frequent itemsets in L_{k-1}
    1. Join L_{k-1} p with L_{k-1}q, as follows:

        insert into C_k

        select $p.\text{item}_1, q.\text{item}_1, \ldots, p.\text{item}_{k-1}, q.\text{item}_{k-1}$

        from L_{k-1} p, L_{k-1}q

        where $p.\text{item}_1 = q.\text{item}_1, \ldots p.\text{item}_{k-2} = q.\text{item}_{k-2}, p.\text{item}_{k-1} < q.\text{item}_{k-1}$

2. Generate all (k-1)-subsets from the candidate itemsets in C_k
3. Prune all candidate itemsets from C_k where some (k-1)-subset of the candidate itemset is not in the frequent itemset L_{k-1}

2. Scan the transaction database to determine the support for each candidate itemset in C_k
3. Save the frequent itemsets in L_k

The source code for finding frequent itemsets was used from [16]. This accepts the input in form of a binary table where the rows are transactions and the columns are web pages (item). The presence of 1 in cell [i, j] indicates that the $item_j$ exists in $transaction_i$ while a zero represents the item's non existence in that transaction. The output of this component is the set of frequent items based on the user defined support threshold.

The frequent itemsets are generated without using the time information since using the time slots for a web page would result in different web pages based on time. For example, consider a web page represented by number 9. If this page was accessed by three different users for three different time slots then there would be 9a, 9b and 9c items. For frequent itemset generation these three would be considered as three different items even though they are one and the same, just with different time slots appended to them.

Once the frequent itemsets are generated using items without time information, the time information is later appended to each page of the frequent itemset. For each item in every frequent itemset, select the transaction that has the all or most of the items of the frequent itemset and append the time for the items from the selected transaction to the items in the frequent itemset. The frequent itemsets now have the time information

included. The frequent itemsets will be later used to create recommendations for the testing dataset.

There were more than 300 frequent itemsets generated when support threshold was 2.0. Some of the frequent itemsets that were generated are shown in below. The numbers represent the items which in this case are web pages and ‘a’, ‘b’, and ‘c’ represents the time slots for that page.

	Frequent Itemsets with Time	**Frequent Itemsets without Time**
Itemsets of size 3	0b 2b 16b, 0b 2c 17a	0 2 16, 0 2 17
Itemsets of size 4	2b 4a 11c 16c, 2b 4b 11a 25c	2 4 11 16, 2 4 11 25

Figure 4: Sample of Frequent Itemset

5.3 Web Semantic Mining:

Web semantic mining aims at mining the content of web pages and finding similarity between web pages based on the content. A web page can have text, images, video and/or audio as its content. For the purpose of mining, considering only the text of the web pages helps the most. One of the most popular ways to determine what the text of a document is about is by using term frequency where term is the keyword (word of relevance) in the document.

For web semantic mining, list of categories is formed based on the area of browsed web pages. The text from web pages is extracted and processed to get all the keywords and their frequencies. These keywords are mapped to the categories using a similarity

measure. This mapping enables determining which categories a web page belongs to. Once the similarity measure of each document to all the categories has been evaluated, the documents are then clustered to bring together pages with similar content. This process helps to cluster together those pages which are contextually similar but not structurally connected to each other. Thus, even though previous users may not have visited similar pages not connected via links, semantic clustering brings together such pages and recommends them to the current user. Each step of the web semantic mining is described in detail below.

Extracting Text from HTML Documents and Tokenizing:

Text mining is based on the fact that any document that contains text addressing a certain topic will have words (terms) that describe the topic. For example, a document that talks about computer science degree would have the words computer, science, degree, courses, etc. occurring more than other general words. Thus, if the frequency of these words is high in a document one can conclude that the document is related to computer science degree. Another point to keep in mind is to ignore stop words. Stop words are those words that do not add any information to know what the document text is about. Examples of these words are article, prepositions, conjunctions, etc. In addition, one can specify other words as stop words too which are too general to add to information retrieval process. Term frequency-inverse document frequency (tf-idf) is a weight that is extensively used for text mining. This weight gives the importance of a term with respect to a collection of documents containing similar context content. The importance of a term proportionally increases with the increase in its occurrence in a

single document. The inverse document frequency offsets this by considering the term's frequency in the collection of documents [17].

In case of web semantics for web pages, it is very likely that the collection of pages being considered is not contextually similar. Thus, it makes more sense to consider only the term's frequency as weight and not the term's frequency with respect to a collection of documents (inverse document frequency). For this project only the term frequency has been used to determine what a document (web page) is about. The text from web pages needs to be extracted before term frequencies can be determined.

Once the text is extracted from web pages, each text file is read and the term frequency is calculated. A list of stop words, previously created, is used to remove all the stop words from the final list of terms and their frequencies. The algorithm to calculate term frequency is given below:

1. For every document read one word at a time
2. For every word check if it is a stop word
    a. If it is stop word, ignore it
    b. If it is not a stop word, check if it has already been read
        i. If it has already been read, increase its frequency by 1
        ii. If it has not yet been read, add it to the list of terms with frequency 1
3. Sort the terms based on frequency and return the list

The source code this component was got from [18]. The list of stop words was got from [19] which were later extended to include more area related non important words.

Mapping Keywords to Categories:

Any two web pages may have multiple keywords that belong to the same area but are not the same. For example, keywords like science, math, courses, etc are all different yet related to each other to some extent. If the keywords are used as is there is very little possibility of having overlapping keywords between two pages to be able to mark them as similar. Also, there is no end to having similar keywords. Instead, if a finite set of categories are identified based on the area of the web pages, these keywords can be compared to the categories. The degree of match will help in identifying how much this category represents the document whose keywords are being matched. Higher the degree of matching to a particular category, higher is the possibility that this document talks about the area related to the category. The frequency of the keywords also plays an important role in determining to what extent the document matches to a particular category. For example, if in a document the keyword management is repeated 4 times while the keyword marketing is repeated 10 times, the document is about management but more specifically about marketing. Thus the frequency of the keywords adds a weight to them increases the influence of category matching.

Here, the first seven most repeated keywords were matched with a list of 23 categories. The thesaurus from Wordnet [20] and the JCn (Jiang and Conrath) similarity measure is used to evaluate the similarity between the keywords and the categories. The source code for the similarity measure was used from [21]. Every keyword of a document is compared with all the categories. The weight of a document for a particular category was

calculated as ∑(similarity * frequency of keyword) / total frequency. The algorithm for mapping keywords to categories is given as follows:

1. For each document read the first seven terms tokenized and sorted by frequency
    a. For each term calculate the similarity between the term and all the categories
        i. Calculate document weight as (similarity * frequency) + previous weight
    b. Update document weight as weight / total frequency
2. Return the weight of document.

The documents were then labeled with those categories that produced the high weights. After this process each document was represented by a set of categories.

Cluster Similar Documents:

Clustering documents helps in putting together documents with similar context. A modified density based clustering algorithm, DBSCAN, was used to cluster the documents. The important factor in this algorithm is the calculation of distance between two elements. The algorithm is as good or as bad as the method used for calculating this distance. In this project the distance or similarity between documents was calculated by calculating the similarity between the set of categories. The clustering algorithm is as follows [22]:

1. For each unvisited document in dataset get its neighbors by calculating the similarity between set of categories for each document

a. If the neighbors are less than minimum threshold mark document as noise
b. If the neighbors are greater than minimum threshold mark document as visited and add it to the current cluster

2. Recurse the above for all the documents

5.4 Recommendation Engine:

This is the final component of the system. It combines the analysis of the usage mining and semantic mining components and produces recommendations for current users. The current user's navigation path is compared with the frequent itemsets generated by the usage mining component. If the user's current path is three items long then only frequent itemsets of length four items are considered. Thus each selected frequent itemset will produced one recommendation. For each of the recommended page, it is checked which cluster does it belong to and all the documents within that cluster are recommended to the user. The algorithm for the recommendation engine is as follows:

1. For each transaction in testing dataset, divide it into two halves
    a. For the first half of size 'x' get all the frequent itemsets of size x + 1
    b. Get the page from the itemset that is not yet visited by user
    c. Get all the pages, not yet visited by user, that belong to the cluster to which this page belongs
2. Recommend the retrieved pages

6. Experimental Setup:

The proposed system was built using JAVA and Eclipse IDE. Extensive experiments were carried out using a synthetic dataset of 10,000 transactions containing of 120 URLs. The URLs were selected from four different California State Universities: San Jose State University, CSU East Bay – Hayward, CSU Long Beach and CSU Chico. From each of these universities web pages were selected belonging to four different areas: science (computer science), arts, business and fees/ scholarship. About 10 web pages were selected from each of the above mentioned areas for each of the four universities.

To conduct experiments the original dataset was divided into two parts where one part, called the training dataset, was used to train the system and the second part, called the testing dataset, was used to test the system. The ratio of training dataset to testing dataset was 8:2 and thus the training dataset consisted of 8000 transactions while the testing dataset consisted of 2000 transactions.

Precision and coverage (recall) are the two most popular metrics that were used to evaluate and compare the performance of the proposed system. Precision measures the preciseness of a system to predict correct or relevant recommendations and thus the accuracy of the system. On the other hand, coverage measures the how well the system covers the pages of the area under consideration [23].

For the experiments, each transaction in the testing dataset was divided into half. The first half of the transaction represents current user's navigation path and the second half of the transaction represents the web pages that the user will visit in future. The correctness of the recommendations produced by the system is compared with this second

half of the transaction. For every first half of the transaction of size w, all frequent itemsets generated by the training dataset which have size w + 1 are selected. All web pages that have not yet been visited by the user i.e. that are not in the first half of the transaction are candidate recommendations to the user. Since web semantic mining is also being applied, all web pages that belong to the cluster to which the candidate recommendation page belongs to and that are not yet visited by the current user, are also included in the recommendations. This procedure was repeated for different values of support. Support narrows down the number of frequent itemsets generated from the training dataset. A higher support value means more frequent itemsets.

Let $U = \{u_1, u_2, \ldots u_n\}$ be the set of web pages that the user will visit where n is the number of pages visited and let $V = \{v_1, v_2, \ldots v_m\}$ be the set of web pages recommended by the system where m is the number of pages recommended. The precision of the system is given as number of web pages correctly predicted divided by the total number of web pages predicted i.e. $|U \cap V| / m$. Coverage of the system is given as number of web pages correctly predicted divided by the number of web pages visited by the user i.e. $|U \cap V| / n$ [23]. The precision and coverage was evaluated for all the transactions in the testing dataset and their average was calculated. The average precision and average coverage help in evaluation of the system.

A confusion matrix was also constructed for the said dataset. Confusion matrix is used to evaluate the performance of a classification model based on the counts of the test records correctly and incorrectly predicted by the model [24]. A confusion matrix is shown below:

		Predicted Cases	
		True	False
Actual Cases	True	A: True Positive (TP)	B: False Negative (FP)
	False	C: False Positive (FP)	D: True Negative (TN)

Figure 5: Confusion Matrix

For the recommendation systems, we require only A, B and C. A gives the number of cases that were correctly predicted, B gives the cases that should have been predicted but were not and C gives the cases that were predicted but should not have been. For the recommendation systems, C will always give a high number since the systems predicts/ recommends even those pages that the users have not visited but they are related to the users' interest area.

7. Experimental Results:

For the evaluation of the proposed system experiments and comparisons were carried out between proposed system (system considering time spent on web pages) and vanilla system (system without considering time spent on web pages). Two types of experiments were done where one was between the two systems with semantic clustering and other without semantic clustering. Thus the experiments carried out are as follows:

- Comparison of proposed and vanilla systems with semantic clustering
- Comparison of proposed and vanilla systems without semantic clustering

For each of these average precision and coverage were calculated at different support values. Confusion matrix was also evaluated for the same comparisons. The results that were obtained are shown in graphs below.

Precision Comparison with Clustering:

The graph shows that the proposed system gives results with better precision than the vanilla system. The proposed system generated recommendations that take the time spent on web pages into consideration and thus the recommendations are more focused and precise. The vanilla system produces a large number of recommendations thus reducing the preciseness of the system.

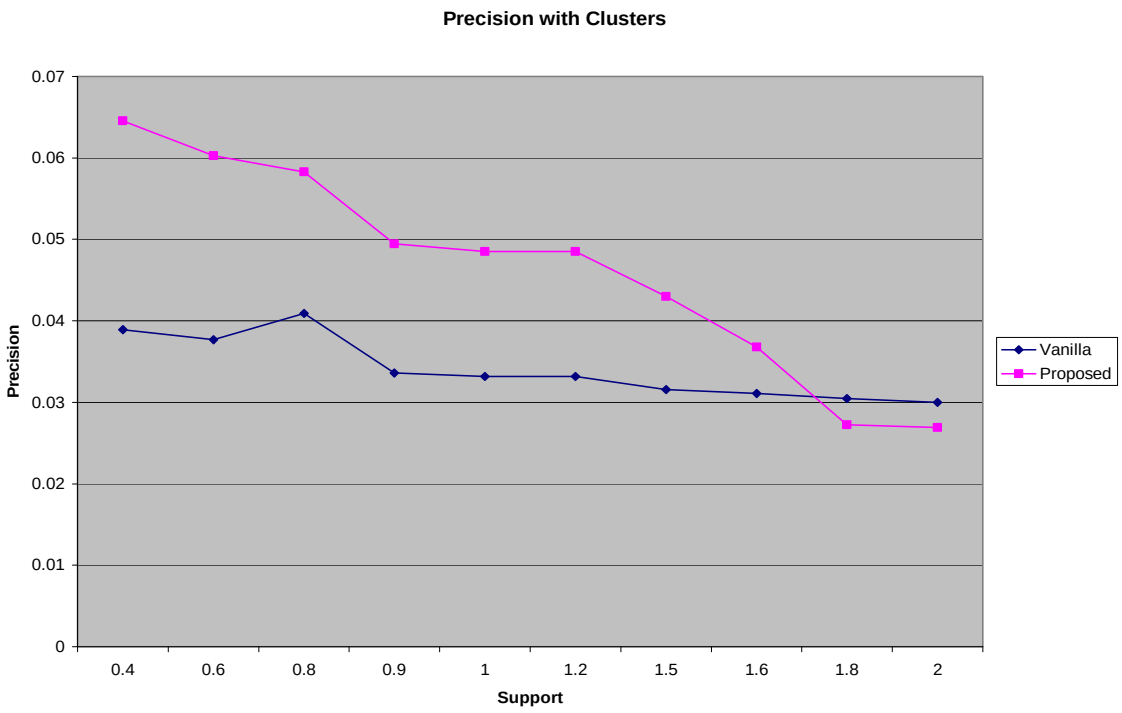

Figure 6: Precision with clustering

Precision Comparison without Clustering:

The graph below shows the precision comparison of proposed and vanilla systems without semantic clustering. This graph also clearly shows that the proposed system is better than the vanilla system. In this case too, the proposed system produces results with better precision as compared to the vanilla system.

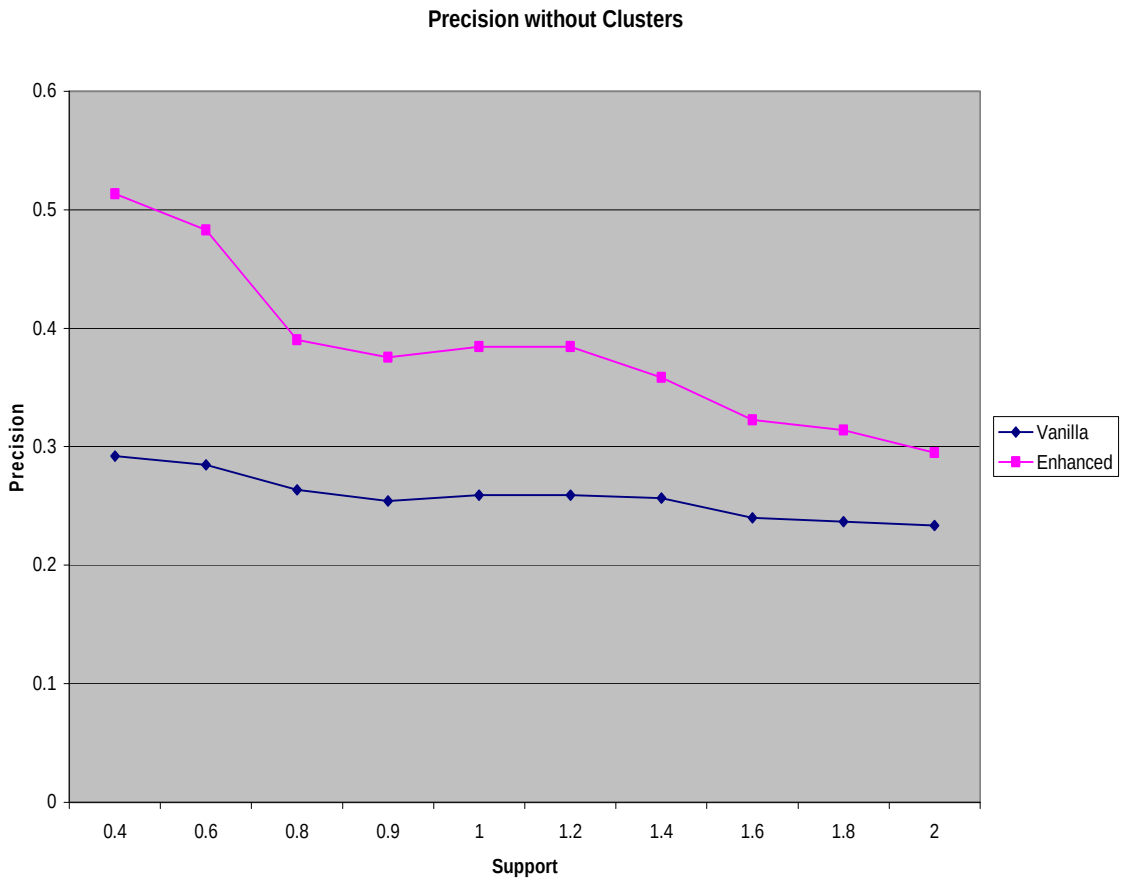

Figure 7: Precision without clustering

Precision Comparison of All Systems:

The graph shows the comparison of all four systems: vanilla with and without clustering and proposed with and without clustering. In both cases, with and without clustering, the proposed system gives higher precision. The reason why the precision of system without clusters is better than that of with clusters is since the system with clusters considers and recommends even those pages which may have not been visited by other users due to lack of link connectivity. Thus, the recommendations of systems with clusters are larger than that of without clusters leading to reduced precision.

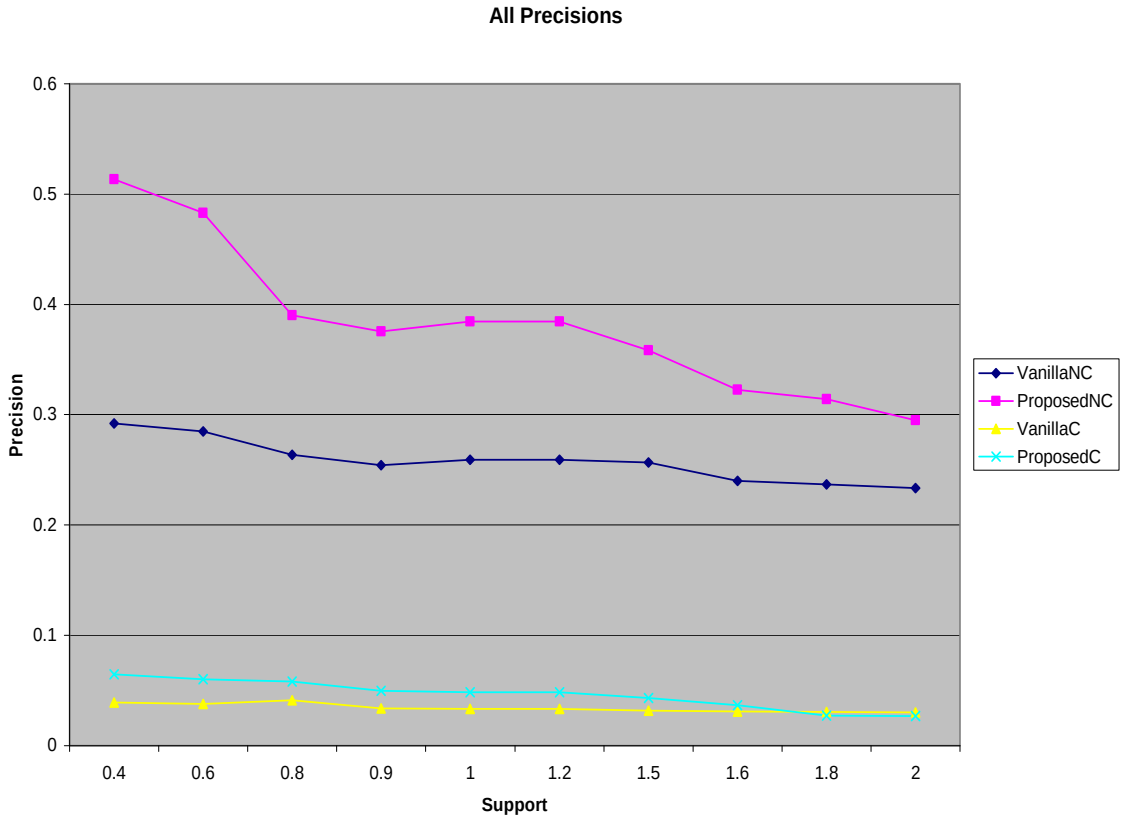

Figure 8: All Precisions

Coverage Comparison with Clusters:

The coverage reduces with increase in support threshold. With the increase in support threshold, the frequent itemsets reduce resulting in fewer possibilities for recommendations. Also, the proposed system gives more focused recommendations which are less in number as compared to the vanilla system resulting in lower coverage for both cases, with and without clustering.

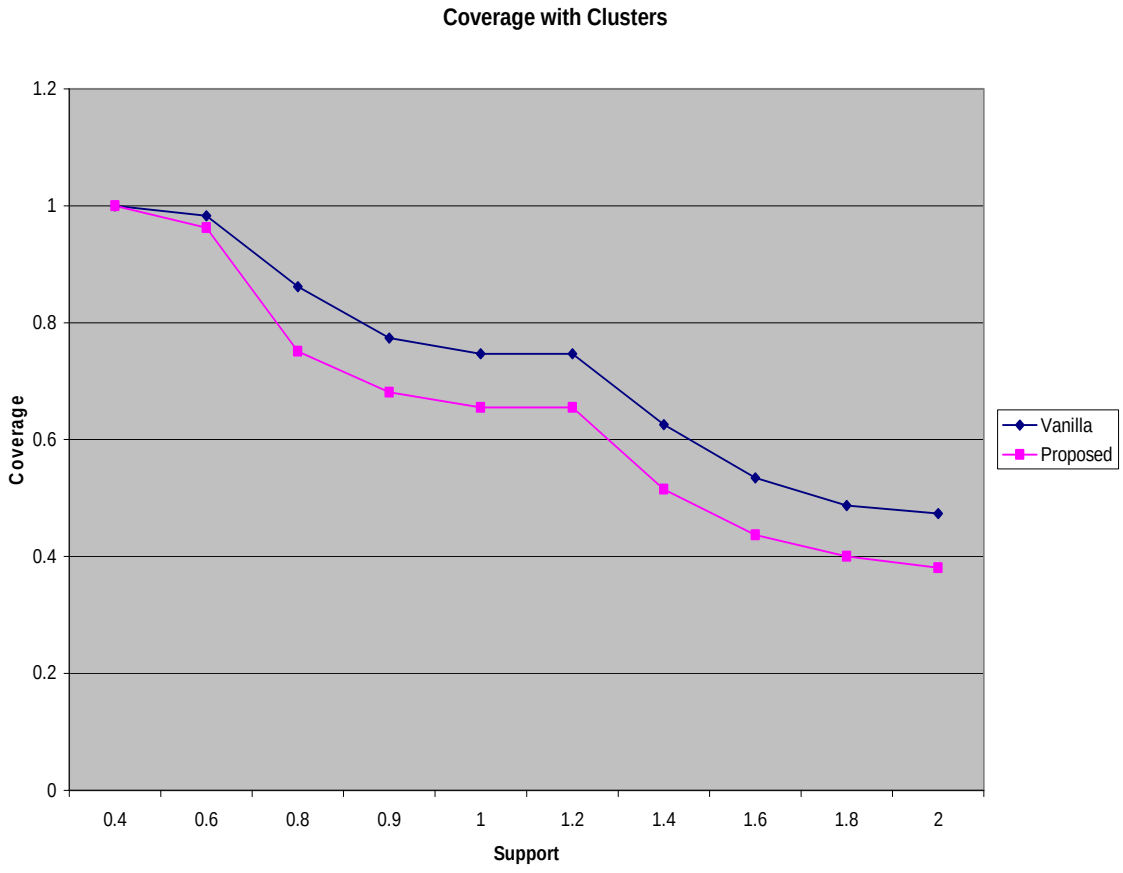

Figure 9: Coverage with Clusters

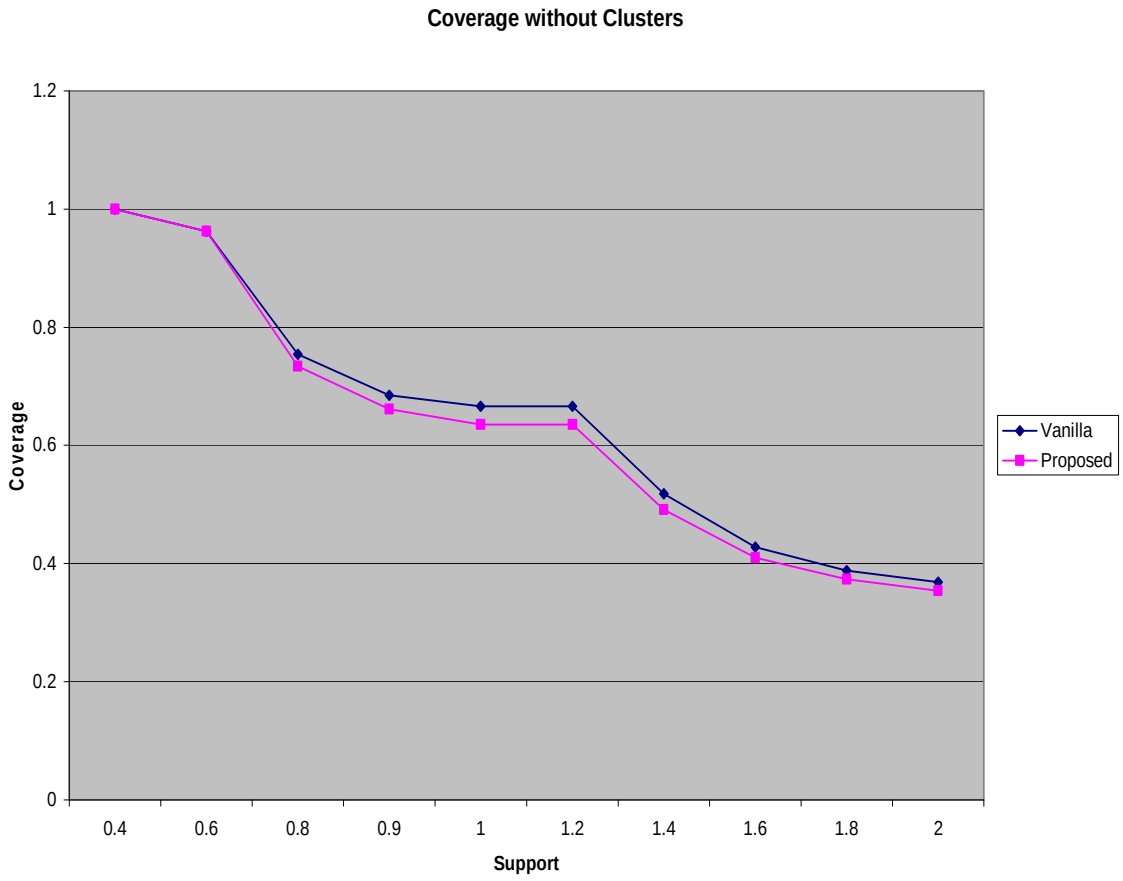

Figure 10: Coverage without Clusters

Coverage Comparison of All Systems:

As shown in the graph, the coverage of systems with clustering is better than that of those without clustering. This is because with clustering even those pages are considered which are not structurally connected thus giving the users a wider but targeted recommendation.

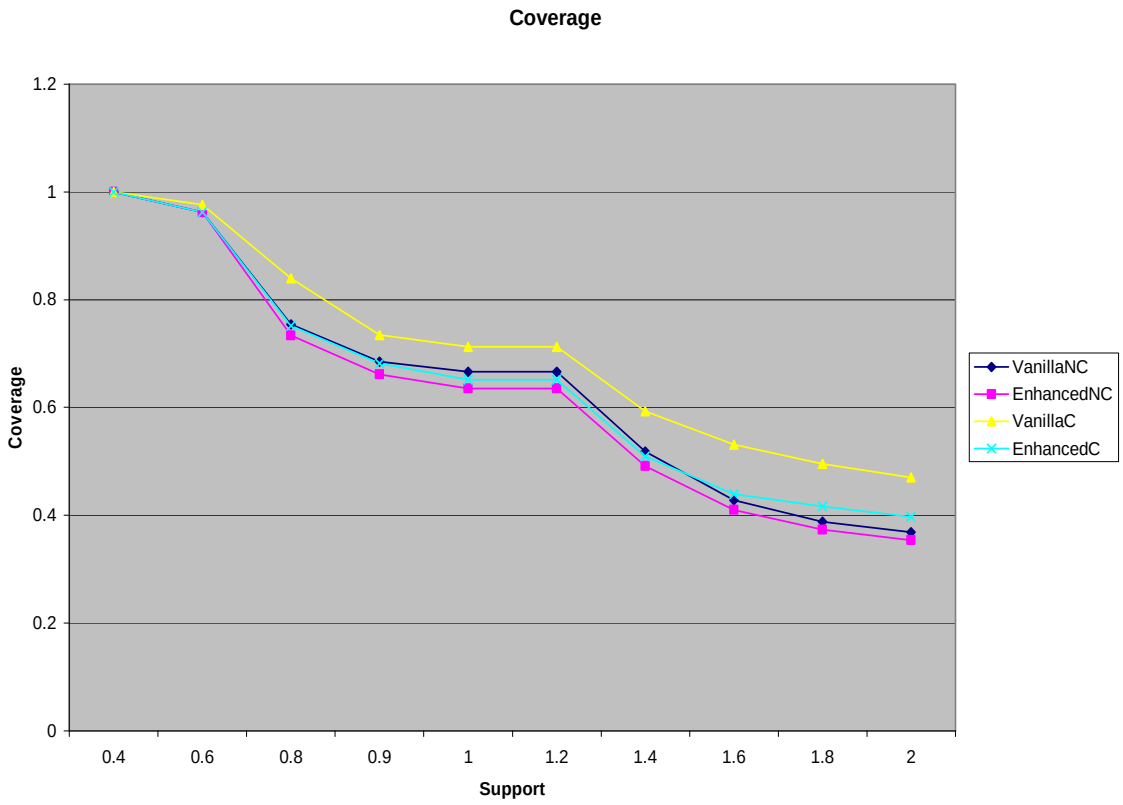

Figure 11: All Coverage

Confusion Matrix:

As expected, the graphs of the confusion matrix of system with and without clusters shows that C (false positive) component is higher than A (true positive) while B (false negative) component is lower than A. This is because a recommendation system will recommend pages that are related to the users' interest and navigation pattern but not visited by the user.

The graph also shows that the component C is high for vanilla system as compared to the proposed system. This means that the proposed system produces false negatives but they are not very huge in number. Thus the recommendations of the proposed system are more focused and targeted as compared to the vanilla system.

The C component of the systems with clusters is much higher than that of the system without clusters. This is because the systems with clusters give a larger recommendation by considering semantic clusters of the pages and recommending semantically similar pages too.

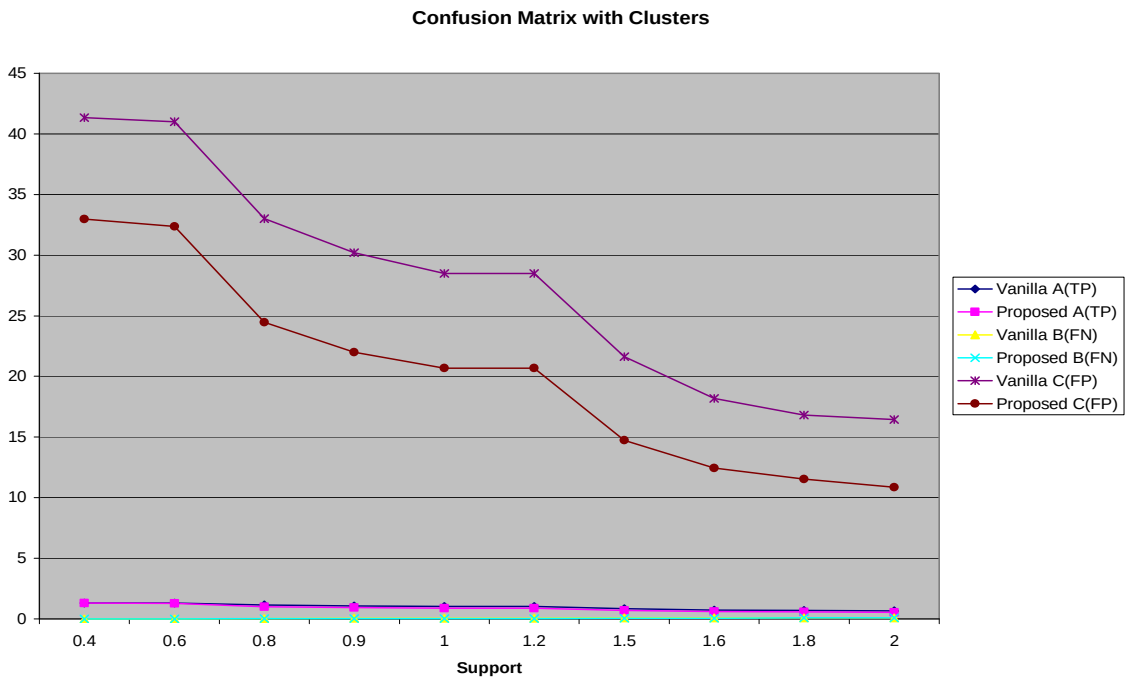

Figure 12: Confusion Matrix with Clusters

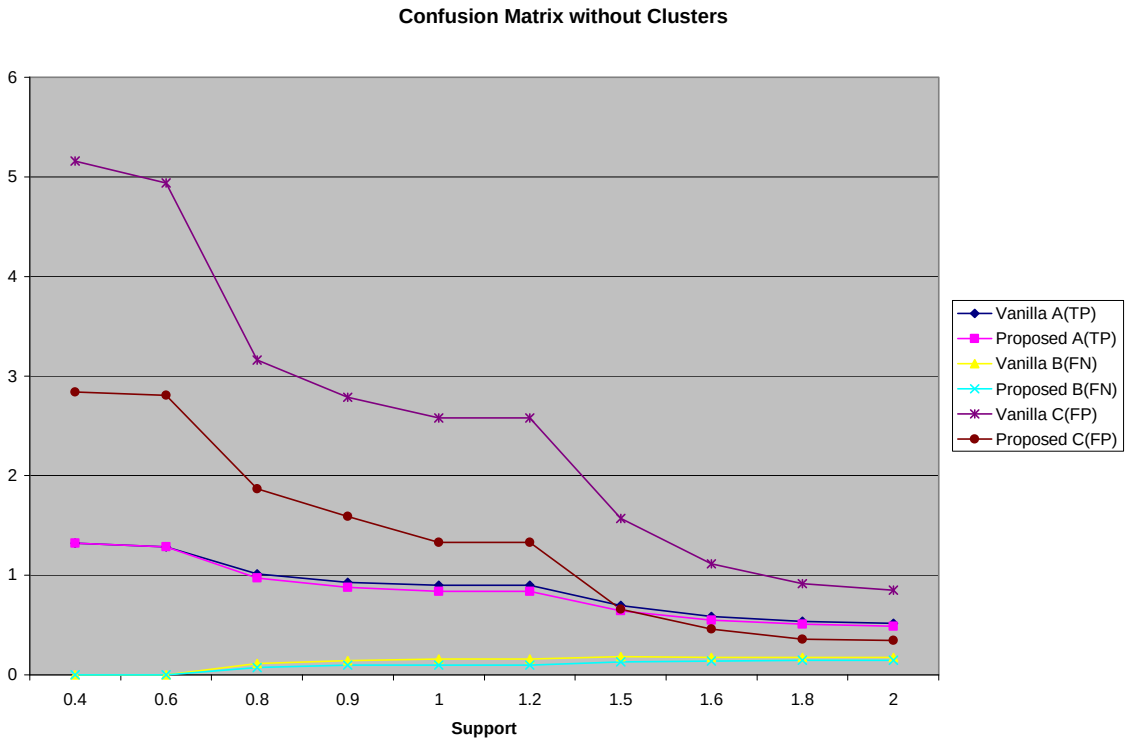

Figure 13: Confusion Matrix without Clusters

8. Conclusion and Future Work:

This report proposes a system which aims to improve the web recommendations by taking the time spent on web pages into consideration in addition to using web usage and web semantics mining. During web usage mining, after getting frequent itemsets based on the browsing patterns of previous users, the time that users have spent on web pages is also accounted for and appended to the frequent itemsets. Taking the time spent on web pages adds more information about the users' interest and thus makes recommendations more focused and personalized. Web semantics mining finds similar web pages based on the content of the web pages and clusters them together. This helps in the recommendation of those pages that were not browsed by previous users because they are not structurally linked to the browsed pages. Experimental results support the above argument and prove that taking the time spent on web pages into consideration improves the recommendations of the system.

This work can be further extended by taking more attributes into consideration that increase the knowledge about a user. Attributes like user's gender, location, age, etc. tell us more about the user whereas considering the time at which the web page was browsed or the number of times that the web page was browsed help in increasing the knowledge about a web page. Taking these attributes while generating web page recommendations will make the recommendation more personalized and useful for the users.

List of References:

[1] Erinaki, M. and Vazirgiannis, M.: Web Mining for Web Personalization [Electronic version]. ACM Transaction on Internet Technology, Volume 3, Issue 1, Pages 1 – 27, 2003

[2] Bamshad M., Cooley R. and Srivastava J.: Automatic Personalization Based on Web Usage Mining[Electronic version]. Communications of the ACM Vol 43. No. 8, 2000.

[3] Murata, T. and Saito, K.: Extracting Users' Interest from Web Log Data [Electronic version]. Proceedings of the 2006 IEEE/ WIC/ ACM International Conference on Web Intelligence.

[4] Albanese, M., Picariello, A., Sansone, C., & Sansone, L. A Web Personalization System Based on Web Usage Mining Techniques [Electronic version].WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters

[5] Mobasher, B., Dai, H., Luo, T., & Nakagawa, M.: Effective Personalization Based on Association Rule Discovery from Web Usage Data [Electronic version]. November 2001 WIDM '01: Proceedings of the 3rd International Workshop on Web Information and Data Management.

[6] M., Eirinaki, C., Lampos, S., Paulakis, M., Vazirgiannis: Web Personalization Integrating Content Semantics and Navigational Patterns [Electronic version]. WIDM'04, November 12-13, 2004, Washington, DC, USA.

[7] M., Eirinaki, G., Tsatsaronis, D., Mavroeidis, M., Vazirgiannis: Introducing Semantics in Web Personalization: The Role of Ontologies [Electronic version]. Semantics and Web Mining 2005, LNAI 4289, 2006

[8] M., Eirinaki, I., Varlamis, M., Vazirgiannis: SEWeP: Using Site Semantics and Taxonomy to Enhance the Web Personalization Process [Electronic version]. SIGKDD '03, August 24-27, 2003, Washington, DC, USA.

[9] Yu, J., Luo, X., Xu, Z., Liu, F., & Li, X; Representation and Evolution of User Profile in Web Activity [Electronic version]. Semantic Computing and Systems, 2008. WSCS '08. IEEE International Workshop on 14-15 July 2008

[10] Luo, X., Xu, Z., Yu, J., & Liu, F.: Discovery Of Associated Topics For The Intelligent Browsing [Electronic version]. Ubi-Media Computing, 2008 First IEEE International Conference.

[11] X.F Luo, N.Fang, et.al.: Semantic Representation of Scientific Documents for the e-Science Knowledge Grid [Electronic version]. Concurrency and Computation: Practice and Experience. Published online in Wiley InterScience (www.interscience.wiley.com).DOI: 10.1002/cpe.1271.

[12] Xiangfeng Luo, ZhengXu, Jie Yu. Discovery of Associated Topics for the Intelligent Browsing [Electronic version]. The First IEEE International Conference on Ubi-media Computing (UMedia 2008).Lanzhou University, China, July 15-16, 2008

[13] Ahmad, A.M., Hijazi, M.H.A., & Abdullah, A.H.: Using Normalize Time Spent on a Web Page for Web Personalization [Electronic version]. 2004 IEEE Region 10 Conference Volume B, 21-24 Nov. 2004 Page(s):270 - 273 Vol. 2
[14] http://www.jafsoft.com/searchengines/log_sample.html
[15] http://it-essence.xs4all.nl/roller/technology/entry/three_tree_traversals_in_java
[16] http://www2.cs.uregina.ca/~hamilton/courses/831/notes/itemsets/itemset_prog1.html
[17] http://en.wikipedia.org/wiki/Tf-idf
[18] http://www.faqs.org/docs/javap/source/WordCount.java
[19] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
[20] http://wordnet.princeton.edu/
[21] http://nlp.shef.ac.uk/result/index.html
[22] http://en.wikipedia.org/wiki/DBSCAN
[23] Shyu, M., Haruechaiyasak, C., Chen, S., and Zhao, N.: Collaborative Filtering by Mining Association Rules from User Access Sequences. Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration, Pages 128 – 135, 2005.
[24] Tan, P., Steinbach, M., and Kumar, V.: Introduction to Data Mining, Chapter 4.