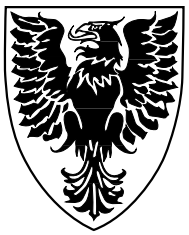


Combined Mining of Web Server Logs and Web Contents for Classifying User Navigation Patterns and Predicting Users' Future Requests

Haibin Liu

Technical Report CS-2005-14

July 19, 2005

Faculty of Computer Science
6050 University Ave., Halifax, Nova Scotia, B3H 1W5, Canada

Combined Mining of Web Server Logs and Web Contents for Classifying User Navigation Patterns and Predicting Users' Future Requests

by
Haibin Liu

Submitted in partial fulfillment of the
requirements for the degree of
Master of Electronic Commerce

at

Dalhousie University
Halifax, Nova Scotia
July, 2005

© Copyright by Haibin Liu, 2005

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**Combined Mining of Web Server Logs and Web Contents for Classifying User Navigation Patterns and Predicting Users' Future Requests**" by **Haibin Liu** in partial fulfillment of the requirements for the degree of **Master of Electronic Commerce**.

Dated: July 19, 2005

Supervisor: _____
Dr. Vlado Kešelj

Readers: _____
Dr. Qigang Gao

_____
Dr. Christian Blouin

DALHOUSIE UNIVERSITY

Date: **July 19, 2005**

Author: **Haibin Liu**

Title: **Combined Mining of Web Server Logs and Web Contents for Classifying User Navigation Patterns and Predicting Users' Future Requests**

Department: **Computer Science**

Degree: **M.E.C.** Convocation: **October** Year: **2005**

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis (other than brief excerpts requiring only proper acknowledgement in scholarly writing) and that all such use is clearly acknowledged.

Table of Contents

List of Tables

List of Figures

Abstract

With the explosive growth of knowledge available on the World Wide Web, it becomes much more difficult for users to access relevant information efficiently and it also presents a challenging task for web designers to organize site contents to meet the needs of users. Automatic classification of user navigation patterns provides a solution to these problems. In this thesis, we propose a novel approach to classifying user navigation patterns and predicting users' future requests using the N-gram-based user navigation profiles extracted by combined mining of Web server logs and web contents. We have applied the approach to build an experimental system. The performance of the system is evaluated based on both classification and prediction accuracy. Our system achieves the classification accuracy of nearly 70% and the prediction accuracy of about 65%, which is comparable to the state-of-the-art systems. This approach may be used to facilitate better web personalization and website organization.

Chapter 1

Introduction

With the explosive growth of knowledge available on the World Wide Web, which lacks an integrated structure or schema, it becomes much more difficult for users to access relevant information efficiently. Meanwhile, the substantial increase in the number of websites presents a challenging task for webmasters to organize the contents of the websites to cater to the needs of users. Automatic classification of user navigation patterns provides a solution to these problems. With the classification, profiles on navigation behaviour of site users can be extracted. On one hand, the profiles can be used for predicting the navigation behaviour of current users, thus aiding in web personalization. On the other hand, webmasters can improve the design and organization of websites based on the acquired profiles.

From the user perspective, the classification of navigation patterns can enhance the quality of personalized web recommendations that aim to predict which web pages are more likely to be accessed next by current users. Through the classification, the desires of a current user can be estimated based on the acquired profiles. Therefore, the related unrequested web pages have great potential to be the next pages that the user wants to see. As the recommendations, the links of these pages will then be inserted into the currently requested page dynamically for display. This will help users access their favourite information efficiently. From the perspective of websites, the classification of navigation patterns can guide webmasters to organize the contents of sites. Instead of being arranged purely according to topics of the contents, the sites will be adjusted in terms of the desires of users. For instance, necessary links will be added between the web pages which seemingly do not share the same topic, but were visited one after another by plenty of users. Also, pages which drew lots of clicks will be highlighted from their categories of topics, while pages which were not visited for a period of time will be moved or discarded. In fact, organizing websites by topics is both static and reactive. Since users' navigation patterns will be learned

periodically, the change of their navigation interest can be captured regularly and then the site organization can be adjusted accordingly. This is a dynamic and proactive way of managing websites. As a result, the passing visitors will be enticed to become consumers or users of the site while current users are willing to remain loyal to the site.

Web usage mining techniques have been widely applied for discovering interesting and frequent user navigation patterns from Web server logs. Sequential pattern mining [1], association rule mining [2, 3] and clustering [4, 5] discover different access patterns from web logs that can be modeled and used to offer a personalized and proactive view of the web services to users. At the same time, web content mining approaches have also been investigated and implemented for extracting knowledge from the contents of websites. For example, the classification of web pages is a typical application of content mining techniques [6].

A project aiming at extracting navigation behaviour models of a site's visitors was introduced in [7]. In the project, two classification-type experiments were implemented to predict the sex of a visitor and to predict if a visitor would be interested in some section of web pages. The results of both experiments were not very good with all classification accuracy under 56%. One reason for such results discussed by the authors was they did not exploit content mining techniques; they only considered the algorithm for classifying access patterns from logs. However, the contents of accessed pages may reveal topics related to visitors' profiles which can improve the classification accuracy.

While many results were reported in the web usage and content mining separately, few efforts were made to integrate these two aspects for a more effective classification of user navigation patterns. Inspired by the work of Baglioni *et al.* [7], we propose an experimental system to investigate whether associating a content mining approach with regular web usage mining could result in a more accurate classification of user navigation patterns and consequently lead to a more accurate prediction of users' future requests. In this system, we will attempt to apply web usage mining to extract the navigation patterns of users. We will then integrate the content features of web pages into usage mining results to build N-gram-based user profiles of navigation patterns. Next, we will apply the k Nearest neighbours method to web users

classification and further prediction of their future requests based on the constructed profiles. Finally, we will evaluate the proposed experimental system using two defined measures: *classification accuracy* and *prediction accuracy*. We will determine if our system improves the classification accuracy and achieves the good prediction accuracy.

In summary, this thesis discusses how to combine the content features of web pages with the usage mining output to construct integrated user profiles, how to classify user navigation patterns and further predict users' future requests based on the obtained profiles, and in which way we can build profiles which lead to the best experimental results both in classification and prediction. The two main objectives of this thesis are:

- To achieve both more accurate classification of web user navigation patterns and prediction of their future requests by combining the information extracted from Web server log with the application of natural language processing techniques on web page contents; and
- To provide an efficient way to facilitate better web personalization and website organization.

The rest of the thesis is organized as follows: Chapter 2 gives a review of web mining, the evaluation of clustering quality, the character N-gram representation, the kNN classification method and other important knowledge used in this thesis. Chapter 3 describes the architecture of the experimental system proposed for classifying user navigation patterns and predicting users' future requests. It also explicates the approaches and algorithms applied in each module of the system, and discusses some issues in the implementation. Chapter 4 presents the experimental results and the evaluation of the proposed system. Finally, Chapter 5 summarizes the thesis and introduces future work.

Chapter 2

Related Work

In this Chapter, some necessary background knowledge will be introduced. First, we will define what is web mining and introduce its three categories. We will mainly describe web content mining and usage mining since they are the research areas we mostly investigate in this thesis. Second, we will describe some existing measures for evaluating the quality of clustering results. Third, we will illustrate the character N-gram representation and its applications in current literature. Fourth, we will present the k Nearest neighbours method, the base classification method used in the thesis. At last, we will introduce several proposed measures for computing the dissimilarity between profiles.

2.1 Web Mining

Web mining is the mining of data related to the World Wide Web. In the area of web mining, we rely on data mining techniques to automatically discover and extract information from the data which may be present in web pages or related to web activities [8]. The web data can be generally classified into several categories [8]:

- Contents of web pages
- Intra-page structure (HTML or XML code of pages) and inter-page structure (linkage structure between pages)
- Recorded user access usage data
- Registration information obtained from users (demographic information, etc)

Based on the categories of web data, web mining is further divided into three active research areas: content mining, structure mining and usage mining [8].

Content mining is the process of extracting knowledge from the content of web-sites, for example contents of documents or their descriptions. Structure mining, on

the other hand, uses links and references within web pages to obtain the underlying topology of the interconnections between web objects. Usage mining, also known as web-log mining, studies user access information from logged server data in order to extract interesting usage patterns [8]. Since our work is mostly related to the web usage mining and content mining, we will focus on introducing these two areas in the following sections.

2.1.1 Web content mining

Web content mining can be treated as the extended work of a basic search engine [8]. Most search engines are keyword-based, while web content mining extends this traditional information retrieval (IR) technology by constructing concept hierarchies, extracting user profiles and analyzing the links between web pages. In general, traditional search engines require crawlers to search the web and gather information. It also needs indexing techniques to store the information, and query processing support to provide fast and accurate information to users [9]. By using data mining techniques, web content mining helps search engines improve their efficiency and scalability.

Web content mining has been further divided into both agent-based and database approaches based on a taxonomy of web mining [10]. Agent-based approaches use software agents, for example intelligent search agents, to perform the content mining, while database approaches view the web as a multilevel database and use query languages to extract information.

Basic web content mining can be treated as a type of text mining [8]. However, the objects of content mining not only include the plain text and structured documents, which are the traditional objects for text mining, but also comprise a large volume of different types of data, including semi-structured documents, dynamic documents, and multimedia documents [11]. These make the task of web content mining more difficult.

2.1.2 Web usage mining

Web usage mining performs mining on web usage data, for example Web server logs. As the primary source of data in usage mining, Web server logs record the activities of web users when accessing the websites. These logs can be examined from either

a client perspective or a server perspective. From a server perspective, usage mining discovers information about where the service is located in the sites. This information will help web designers to improve the design of the sites and thus attract and better serve visitors. From a client perspective, more information about users' behaviour is detected, which will be used to cache web pages and provide adaptive services.

Web usage mining can be used for many different purposes. The profiles of site users can be developed by studying their patterns on accessing web pages, and can be further used to provide users with more personalized services. In the meantime, usage mining can be used to evaluate the overall quality and effectiveness of the web pages within a website. Furthermore, user access patterns discovered from usage mining can be used to direct business intelligence for improving targeted sales and advertising [12].

Web usage mining consists of three major types of activities [8]: preprocessing, pattern discovery and pattern analysis. Preprocessing activities focus on reformatting the log data of Web server before further processing. Pattern discovery activities look to find hidden user access patterns within the log data. Once the patterns have been identified, they must be analyzed to determine how the information can be used. Therefore, pattern analysis is the process of analyzing and interpreting the discovered patterns.

2.1.3 Recent research advances in both web content and usage mining

Web content mining: Most research activities in web content mining have centered around techniques for extracting information from web contents. Based on textual contents of recently requested web pages, Davison [13] proposed an approach for pre-loading web pages into the local cache for a visitor. The requests for the preloaded pages are the visitor's predicted further requests that even have never been taken. They focus on the appropriate ranking measurement of textual similarity between recently requested pages by a visitor and the links within a page. However, we consider this a reactive method. Intentions of visitors might change during their browsing, therefore, the new prediction has to be made frequently in terms of the current request. This results in heavy server computational load in calculating textual similarity, ranking web pages, and caching new pages. In addition, the predicted

pages are also to some extent limited by the contents of recently requested pages. It is more desirable for websites to have a proactive method to predict a group of related future requests of visitors based on their recently requested pages. This requires that the navigation patterns of site visitors have to be determined before the prediction is made.

Some researchers also have devoted themselves to improving the performance of web page classification [6, 14]. Based on the plain text and the meta-data of web pages, they devised different learning algorithms to build classifiers which are able to accurately assign unseen web pages into the corresponding labeled classes. However, these algorithms purely process web pages from the text mining perspective and only concentrate on the contents of web pages. We conjecture that if the web pages can be classified according to the page contents as well as site users' navigation patterns, it will be more instructive for business to understand site users' needs and provide personalized services. It also will be desirable for web designers to organize the site contents in a more attractive way.

Web usage mining: Web usage mining has also drawn intensive attentions of research community because of its great potential for adaptive websites and user profiling. Many existing tools can be used to generate fixed reports from Web server logs, such as, AWStats [15] or Webalizer [16]. These tools help the day-to-day operation of websites and identify basic trends and patterns of user navigation. However, deeper analysis involving discovery of hidden access patterns embedded in the logs cannot rely only on them [12]. Many approaches have been proposed toward this direction. In [17], a so called WAP-tree was developed for fast and scalable mining of access patterns from log records. Another approach based on an indexing method for larger log files was presented in [18]. In addition, web usage mining is also used to deal with the practical problems in the field of commerce. Batista *et al.* [19] made effort to extract navigation patterns of users from the logs of an on-line newspaper site. They experimented with different clustering operations and association analysis on newspaper sections using some commercial software. Although the results were not good enough, they highlight the benefits of web usage mining.

A project aiming at extracting navigation behaviour models of a site's visitors was

introduced in [7]. The extracted knowledge from the logged data was deployed to offer a personalized and proactive view of the web services to users. Two classification-type experiments were implemented: one was to predict the sex of a visitor and the other was to predict if a visitor would be interested in some section of web pages. The result of the first experiment was not very good with classification accuracy only 54.8%, while the result of the second experiment was a little better. Several reasons for such results were discussed by the authors. One was they did not exploit content techniques, i.e. they only considered the algorithm for classifying access patterns from logs. However, the contents of accessed pages may reveal topics related to visitors' profiles which can improve the classification accuracy.

2.1.4 Combining web usage and content mining

From research activities independently in both web usage and content mining, we realize that both of them have limitations. We believe that to combine web usage and content mining is an applicable approach to dealing with more complicated problems. In fact, web mining activities are sometimes correlated and the distinction between usage mining and content mining is not clear-cut.

Being an active research domain, *personalization* is a suitable application area for combining web content mining and usage mining. With personalization, the contents of web pages are modified according to a user's desires as the recommendations to the user for better meeting his needs. To obtain users' desires requires not only examining web log data to uncover access patterns of users, but also analyzing the contents of web pages which were visited during their navigation. Some systems have been developed based on web mining for automatic personalization [1, 20, 21]. They generally consist of two major processes: off-line mining and on-line recommendation. Figure 2.1 shows the general architecture of a personalization system.

In the off-line mining process, all the access activities of users in a website are recorded into the log files by the Web server. Then, some web mining processes are applied to the server logs to mine the hidden navigation models of users. In this thesis, we will combine usage and content mining to process web logs for building user navigation profiles, and then use these profiles to classify the site users. We will discuss the details of our mining approaches in the Chapter 3. In the on-line recommendation

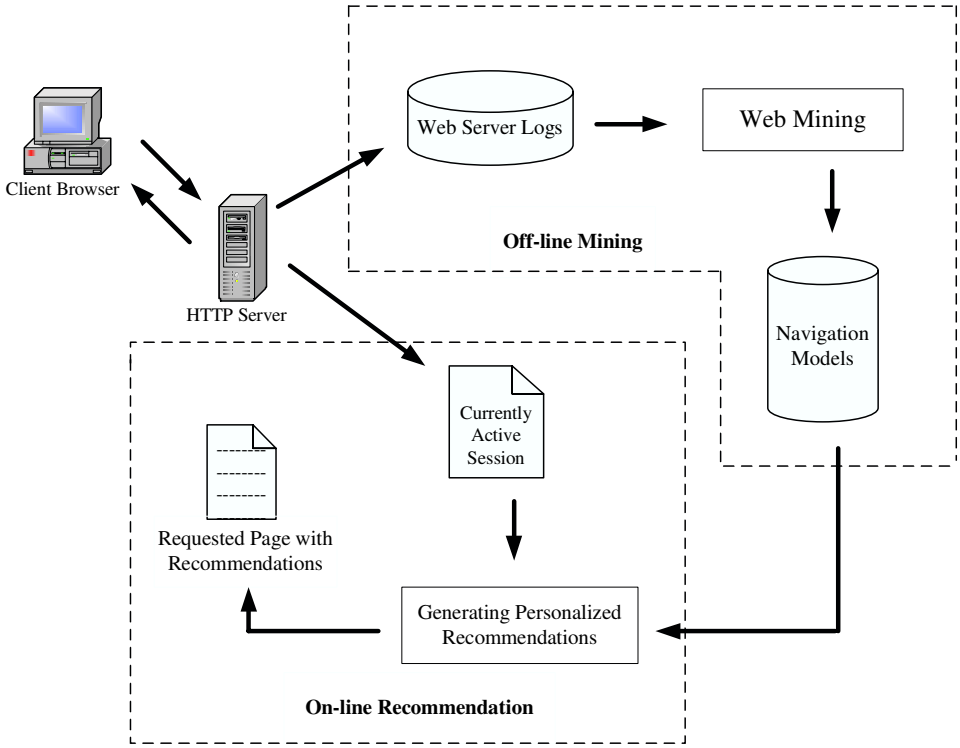


Figure 2.1: Architecture of Personalization System

process, user's requests in his current active session are recorded. By comparing these requests with the models obtained from the off-line mining, appropriate personalized recommendations are generated. These recommendations will then be inserted into the current requested page dynamically for display. In this thesis, we will use the requests in the active sessions to construct the current navigation profile of a user. By matching the current profile with the navigation profiles built in the off-line process, we are able to predict users' future requests. Based on the prediction, corresponding recommendations will be generated by web recommendation systems.

There are at least two ways to integrate content features of web pages into usage mining results during the off-line process: pre-mining integration and post-mining integration [22]. The pre-mining integration involves the transformation of normal user access sessions into "content-enhanced" sessions containing the semantic features of the web page contents. The post-mining integration denotes performing usage mining and content mining independently and then combining their mining results. Strictly, the approach that we propose in the thesis during the off-line process belongs

to the pre-mining integration.

Compared to the activities reported in the web usage and content mining separately, only a few efforts were made to combine these two aspects. Mobasher *et al.* [23] made an attempt to integrate both usage and content attributes of a site into a web mining framework for personalization, but the techniques proposed were limited to the use of clustering to separately build site usage and content profiles. Moreover, Guo *et al.* [24] designed a system to integrates web page clustering into log file association mining. The methods of document clustering are first performed on the website contents to group web pages into a certain number of clusters. Then, the meaningful information of page clusters is integrated into original web log files as the content indicator. Finally, the association rule mining algorithm is applied to the "content-enhanced" data source. The system demonstrates that novel and interesting association rules can be discovered from the integrated log data, while they could not be mined using only the standard log files. In this thesis, we will explore the application of some of the newer successful techniques in text clustering and classification [25–27] in web usage and content mining to build integrated profiles for the more effective classification and prediction.

2.2 Heuristic Methods of Session Identification

Web usage mining is essential for discovering users' navigation patterns. However, the reliability of web usage mining results depends heavily on the proper preparation of the input dataset. As a critical component of data preparation, session identification attracts intensive research attention. A session can be described as the group of activities performed by a user from the moment he entered the site to the moment he left it. Therefore, session identification is the process of segmenting the access log of each user into sessions [28].

Mobasher *et al* did a series of research projects on the data preparation of web usage mining. He specifically proposed three heuristic methods for session identification [28–30]. A sessionizing heuristic is a method for performing the session identification on the basis of assumptions about users' behaviour or the site characteristics [29]. In the proposed three heuristics, two of them are time-oriented and the

other is navigation-oriented. The time-oriented heuristic method considers boundaries on the time spent in the entire site or on a web page during a single visit. It is a reasonable implication that if a long time elapses between two requests, the latter request is the start of a new visit. The navigation-oriented heuristic method, however, takes the linkage between web pages into account, specifically the referrer information recorded in the Web server log. The "referrer" of a URL request denotes the web page from which the request was issued [28]. Since users usually follow hyper-links to reach a web page, rather than typing URLs, it is rational to rely on the exploitation of the referrer information to segment sessions [28]. These heuristic methods are defined as follows [30]:

Definition of the session-duration-based method: The duration of a session must not exceed a threshold θ. Let t_0 be the time stamp of the first URL request in a session. A URL request from the same user with time stamp t is assigned to this session if $t - t_0 \leq \theta$. Otherwise, this URL request becomes the first request of the next new session.

Definition of the page-stay-time-based method: The time spent on a web page must not exceed a threshold θ. Let t_0 be the time stamp of the URL most recently assigned to a session. The next URL request from the same user with time stamp t is assigned to this session if $t - t_0 \leq \theta$. Otherwise, this URL request becomes the first request of the next new session.

Definition of the referrer-based method: Let p and q be two consecutive URL requests from the same user with time stamps t_p and t_q, and p belongs to a session S. The q will be assigned to S if the referrer for q was previously requested within S, or if the referrer is undefined and $(t_q - t_p) \leq \theta$. θ is a specified time delay. Otherwise, the q becomes the first request of the next new session.

These sessionizing heuristic methods were respectively evaluated by a set of proposed performance measures, and the results showed that different methods worked well in different cases [29, 30]. In this thesis, we will concentrate on performing two time-oriented sessionizing heuristic methods on our experimental dataset.

2.3 Evaluation of Clustering Quality

Clustering is the task of grouping a set of data objects into clusters in an unsupervised way so that data objects in a cluster are more similar to each other than to the data objects in different clusters [31]. Specifically, objects in a cluster share similar characteristics and the distance between objects in the cluster is less than the distance between a object in the cluster and any object outside it.

There exist many types of clustering algorithms, which can be roughly classified into two categories [31]: hierarchical clustering algorithm and partitioning clustering algorithm. With hierarchical clustering, a nested set of clusters is created. Each level in the hierarchy has a separate set of clusters. At the lowest level, each object is in its own unique cluster. At the highest level, all objects belong to the same cluster. Noticeably, the desired number of clusters is not necessary to be designated as the input. The AGNES (AGglomative NESting) and DIANA (Division ANAlysis) are examples of hierarchical clustering algorithm. With partitioning clustering, the algorithm creates only a set of clusters. It uses the desired number of clusters to drive how the final set is created. A typical example of partitioning algorithm is K-means, which is also the base clustering algorithm in this thesis.

In order to achieve an optimal clustering result, it is desirable to evaluate the clustering quality of a clustering system. Human inspection of the clustering result may be the most intuitive clustering evaluation method, however, it lacks the scalability to large and complicated problem domains [32]. Meanwhile, the exhausting manual work is always not desirable and feasible. Therefore, quantitative assessment of clustering quality is of great importance for various clustering applications.

Although a multitude of quantitative clustering evaluation measures have been studied in the literature, two kinds of them are widely used: internal quality measure and external quality measure [33]. The internal quality measure compares different sets of obtained clusters without references to external knowledge, while the external quality measure evaluates clusters by comparing them to the known labeled classes. These two measures are of high scalability and able to evaluate a wide range of cluster systems. However, if the class labels are not prepared before performing the process of clustering, the internal quality measure always becomes the choice of evaluating a clustering result.

As we know, the objective of clustering is to minimize the distances among the data objects in each individual cluster and to maximize the distances between clusters. Therefore, it is a natural way to evaluate both the intra-cluster homogeneity and the inter-cluster separation of the clustering result. Based on this idea, *cluster compactness (Cmp)* and *cluster separation (Sep)* for the output clusters $c_1, c_2, \cdots, c_c$ were proposed in [32] to respectively measure the intra-cluster homogeneity and the inter-cluster separation. The definitions of them are given below:

$$Cmp = \frac{1}{C} \sum_{i}^{C} \frac{v\left(c_i\right)}{v(X)} \tag{2.1}$$

where C is the number of clusters generated on the data set X, $v\left(c_i\right)$ is the variance of the cluster c_i, and $v\left(X\right)$ is the variance of the data set X.

$$v\left(X\right) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d^2\left(x_i, \overline{x}\right)} \tag{2.2}$$

where $d\left(x_i, x_j\right)$, for example the *Euclidean distance*, is a distance measure between two vectors x_i and x_j, N is the number of members in X, and $\overline{x}$ is the mean of X. The smaller the Cmp value, the higher the average compactness in the output clusters.

$$Sep = \frac{1}{C\left(C-1\right)} \sum_{i=1}^{C} \sum_{j=1, j \neq i}^{C} \exp(-\frac{d^2(x_{c_i}, x_{c_j})}{2\sigma^2}) \tag{2.3}$$

where σ is the standard deviation of the data set X, C is the number of clusters, x_{c_i} is the centroid of the cluster c_i, $d\left(\right)$ is the distance measure used by the clustering system, and $d(x_{c_i}, x_{c_j})$ is the distance between the centroid of c_i and the centroid of c_j. Similar to Cmp, the smaller the Sep value, the larger the overall dissimilarity among the output clusters.

In addition, the combined measure of both Cmp and Sep, namely *overall cluster quality (Ocq)*, was also proposed in [32] to overcome the deficiency of each measure and assess the overall performance of a clustering system.

$$Ocq\left(\beta\right) = \beta \cdot Cmp + \left(1-\beta\right) \cdot Sep \tag{2.4}$$

where $\beta \in [0, 1]$ is the weight that balances the measures Cmp and Sep. $Ocq\left(0.5\right)$ is often used to give equal weights to the two measures.

2.4 Character N-gram Representations

A character N-gram is an N-character substring of a longer string [25]. The character N-gram representation of a document can be obtained by orderly extracting contiguous n characters across the whole document. In the process, a non-letter character is replaced by a space, while two or more consecutive spaces are only treated as a single one. Furthermore, an underscore character is also adopted to represent the space as well as the beginning and ending of a string. For instance, "Fox is quick." can be represented with the following character N-grams, shown in the Table 2.1.

	N-gram samples
Bi-grams	_F Fo ox x_ _i is s_ _q qu ui ic ck k_
Tri-grams	_Fo Fox ox_ x_i _is is_ s_q _qu qui uic ick ck_
Quad-grams	_Fox Fox_ ox_i x_is _is_ is_q s_qu _qui quic uick ick_

Table 2.1: Example of Character N-grams

The character N-gram representations have been successfully used in many research applications. For instance, a character N-gram-based information retrieval system was implemented by combining N-gram representations of documents with the vector processing models [25]. Instead of traditional term frequencies, the frequency of N-gram occurrence in queries and documents was used as the basis for the element value of vectors. The system provided substantially good retrieval performance on the various TREC datasets. The character N-gram representation was also used in the Authorship attribution tasks [26]. An optimal set of N-grams was chosen from the training data to be included in the author profiles. By comparing the distance between author profiles and a document profile, the author of the document could be automatically identified. The experiments in [26] performed on the data of different languages demonstrated that the N-gram-based approach was very effective. In addition, the classification and hierarchical clustering of biological genome sequences were also performed based on the N-gram representation of genome sequences [27]. The approach was tested on two corpora of genomes and the results suggested that N-gram representation could be successfully applied to a variety of related problems.

Three distinct advantages of the character N-gram representation can be generalized. Firstly, it provides a robust representation because it is relatively insensitive

to the spelling variations or errors. Secondly, the character N-gram representation is language and topic independent and requires no special preparations. Thirdly, it is a simple, but efficient approach for implementation.

2.5 *k* Nearest neighbours

The k Nearest neighbours (kNN) is a common classification method based on the use of distance measures [31]. In this technique, the entire training set includes not only the data in the set but also the desired classification for each training sample. Therefore, the training samples are used to build the classification models. When a classification is to be made for an unknown sample, kNN searches the training samples in terms of the distance between the unknown sample and training samples. Only the k closest training samples are considered the k nearest neighbours of the unknown sample. Then, the unknown sample is assigned the most common class among its k nearest neighbours. Especially, when $k = 1$, the unknown sample is assigned the class of the training sample that is closest to it. The k Nearest neighbours method has been widely used for various tasks, for instance, Text categorization [34], Authorship attribution [26] and Information retrieval [25]. It is seen that the measure for calculating the distance between the unknown sample and training samples is an important factor of the success of the kNN classification method.

2.6 Dissimilarity Measures

As a function of two profiles, the dissimilarity measure reflects the dissimilarity between profiles. It always returns a positive number as the result, and for two identical profiles, the dissimilarity is 0. Several different dissimilarity measures have been applied to the experiments in literature.

Eq.(2.5) was used in the Authorship attribution task [26] to calculate the dissimilarity between the author profile and the document profile. The author profile is composed of a set of L pairs $\{(x_1, f_1), (x_2, f_2), \cdots, (x_L, f_L)\}$, where x_j is the one of the L most frequent N-grams and f_j is the normalized frequency of x_j $(j = 1, 2, \cdots, L)$. In the equation, $f_1(n)$ and $f_2(n)$ are frequencies of an N-gram n in the author and the document profile, and the N-gram frequency is normalized by dividing by the

average frequency of $f_1(n)$ and $f_2(n)$.

$$\sum_{n\in profile}\left(\frac{f_1(n)-f_2(n)}{\frac{f_1(n)+f_2(n)}{2}}\right)^2=\sum_{n\in profile}\left(\frac{2\times(f_1(n)-f_2(n))}{f_1(n)+f_2(n)}\right)^2 \tag{2.5}$$

Miao *et al.* [35] attempted to revise the Eq.(2.5) into Eq.(2.6) to explore a wider range of dissimilarity measures:

$$\sum_{n\in profile}\frac{(f_1(n)-f_2(n))^2}{\left(\frac{(f_1(n)+f_2(n))}{2}\right)^{\alpha}} \tag{2.6}$$

When $\alpha=2$, Eq.(2.6) is same as Eq.(2.5) and when $\alpha=0$, Eq.(2.6) is the squared *Euclidean distance*. However, Miao found that $\alpha=1$ produced the best experimental results in clustering. Therefore, the dissimilarity measure they chose in the experiments is:

$$\sum_{n\in profile}\frac{(f_1(n)-f_2(n))^2}{\frac{(f_1(n)+f_2(n))}{2}}=\sum_{n\in profile}\frac{2\times(f_1(n)-f_2(n))^2}{(f_1(n)+f_2(n))} \tag{2.7}$$

In [27], Tomovic and his colleagues continued to follow the idea of Eq.(2.5) and tried to extend its applicability in a wider domain by adapting its normalization scheme. Instead of using the average (arithmetic mean value $-\ (f_1(n)+f_2(n))/2$) frequency for a given N-gram as the normalization scheme, they attempted other mean values, including geometric mean, harmonic mean, quadratic mean, etc. By comparing the experimental results, the dissimilarity measure normalized by geometric mean value, Eq.(2.8), became their final choice of dissimilarity measures.

$$\sum_{n\in profile}\frac{|f_1(n)-f_2(n)|}{\sqrt{f_1(n)\times f_2(n)}+1} \tag{2.8}$$

We will apply these three proved dissimilarity measures, Eq.(2.5), Eq.(2.7) and Eq.(2.8), to the analogous problems in our experiments.

Chapter 3

Methodology and Implementation

In this chapter, we will introduce our designed experimental system for classifying user navigation patterns and predicting users' future requests. We will explicate the approaches and algorithms applied in each module of the system, and discusses important issues in the implementation.

3.1 System Description

We have designed an experimental system to assist our investigation on whether associating a content mining approach with regular web usage mining could result in a more accurate classification of user navigation patterns and consequently lead to a more accurate prediction of users' future requests. Figure 3.1 illustrates the overall data flow of the system, which consists of five major modules:

1. Web-log Preprocessing:
   (1) Data cleaning,
   (2) User differentiation, and
   (3) Session identification.
2. Web Usage Mining:
   (1) Session vectorization,
   (2) Session clustering, and
   (3) Identification of the optimal number of clusters.
3. Building Navigation Pattern Profiles:
   (1) Web content cleaning,
   (2) Using N-grams to combine web usage mining with content mining, and
   (3) User navigation pattern profiling.

4. Classification and Prediction:

   (1) User navigation pattern classification, and

   (2) Users' future request prediction.

5. System Performance Evaluation

In this scheme, we start with the primary web-log preprocessing to extract user navigation sessions from dataset. From these, we would apply web usage mining techniques to the training set of sessions to mine the representatives of user navigation patterns. When the patterns are obtained, we associate them with corresponding web page contents to build navigation pattern profiles. They are then used on the testing set of sessions to classify user navigation patterns and predict users' future requests. At the end, the system evaluates the results to demonstrate its performance. In the following sections, we will describe the algorithms and implementations for each component of the system in detail.

3.2 Web-log Preprocessing

As the first module of the system, web-log preprocessing aims to reformat the original web logs to identify all web access sessions. The Web server usually registers all users' access activities of the website as Web server logs. Due to different server setting parameters, there are many types of web logs, but typically the log files share the same basic information. Figure 3.2 shows a sample user access entry of the Apache server log which contains the client IP address, request time, requested URL, HTTP status code, etc. Some descriptions of the log fields are given in the Table 3.1 below.

Generally, several preprocessing tasks need to be done before performing web mining algorithms on the Web server logs. For our work, these include data cleaning, user differentiation and session identification. These preprocessing tasks are the same for any web usage mining problem and are discussed by Cooley *et al.* [28]. The original server logs are cleansed, formatted, and then grouped into meaningful sessions before being utilized by web usage mining.

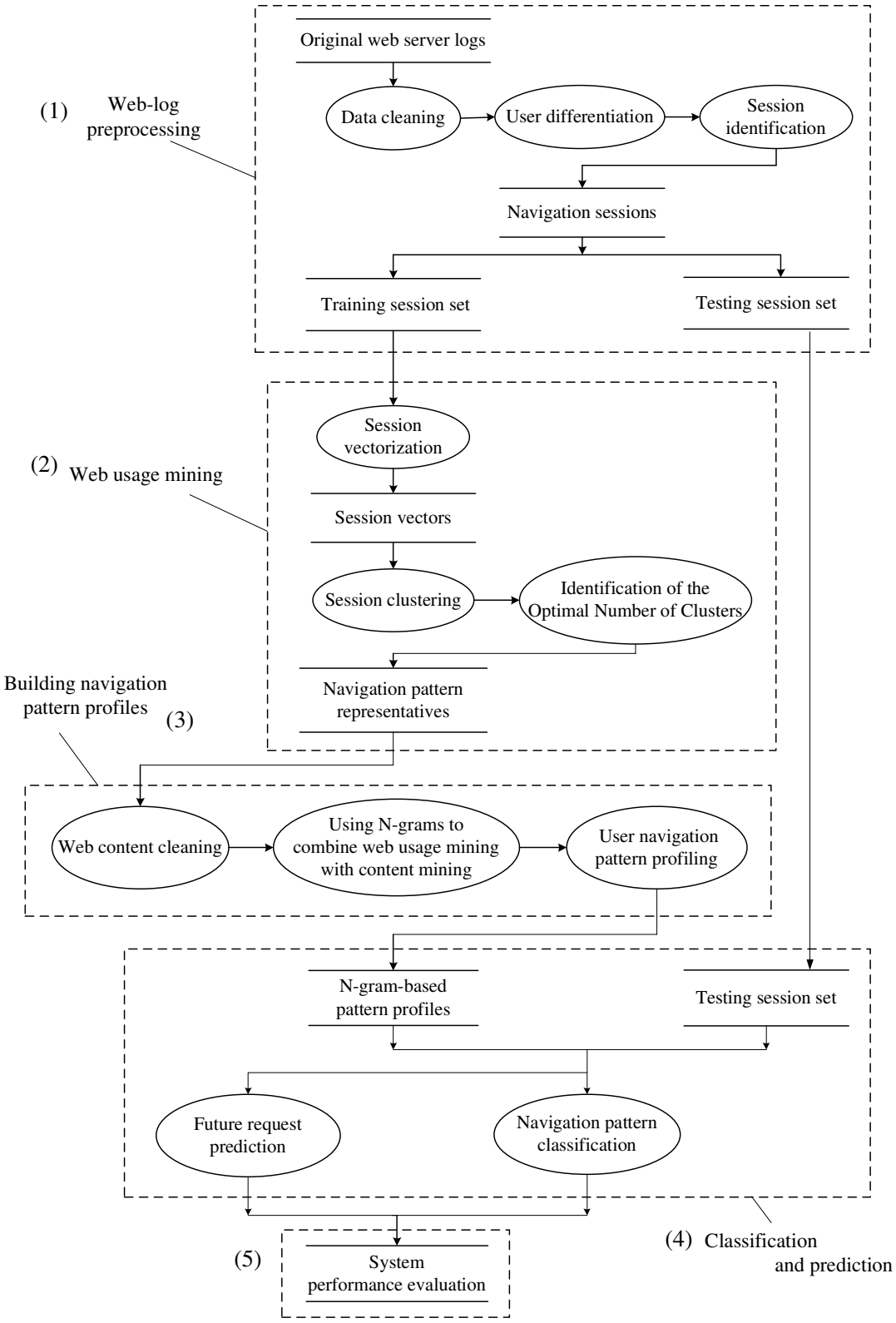


Figure 3.1: System Dataflow Diagram

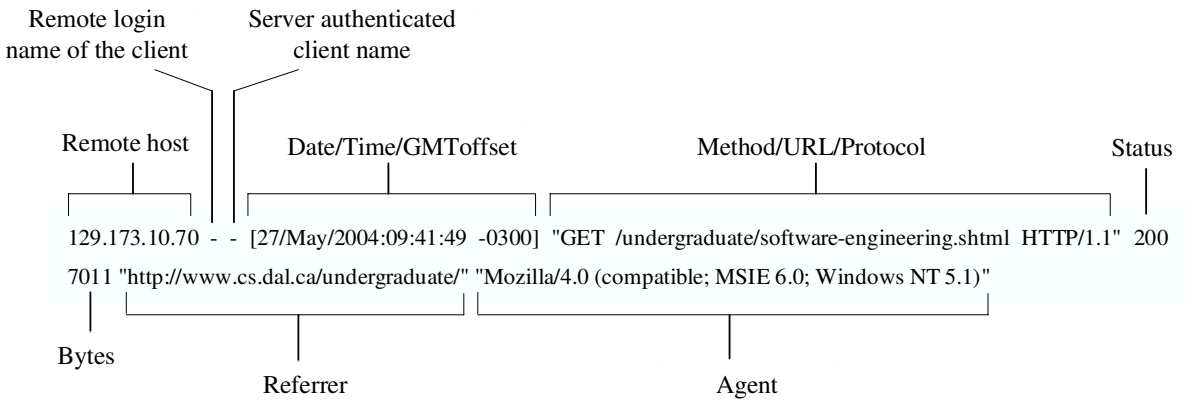


Figure 3.2: A Sample Entry of Apache Server Log

Term	Description
Remote host	remote host name or IP address
Date/Time	date and time of request
GMToffset	local time offset from Greenwich Mean Time
Method	method of request (Get, Post, Head, etc.)
URL	path and filename of the requested file
Protocol	type of protocol used for the request
Status	HTTP status code generated by the request
Bytes	length of requested file
Referrer	URL that request originated from
Agent	OS and browser software at the client

Table 3.1: Web Log Field Description

3.2.1 Data cleaning

In the original web logs, not all the log entries are valid for web usage mining. We only want to keep the entries that carry relevant information. Therefore, data cleaning is used to eliminate the irrelevant entries from the log file. The irrelevant entries includes:

- Requests executed by automated programs, such as web robots, spiders and crawlers need to be discarded because the traffic to websites that these programs generate can dramatically bias site statistics [12] and also do not belong to the category which web usage mining investigates. The robot list that we used in the experiments can be found on [36].

- Requests for picture files associated with requests for particular pages are eliminated. A user's request to view particular page often results in several log entries because that page includes other graphics. We are only interested in what the users explicitly request, which are usually textual files.

- HTTP status codes are used to indicate the success or failure of the requested event. We are only interested in successful entries with codes between 200 and 299, and others are therefore deleted from the web logs.

- Log entries with request methods except "Get" and "Post" are also filtered out.

3.2.2 User differentiation

For web usage mining, to get knowledge about each user's identity is not necessary. However, a mechanism to distinguish different users is still required for analyzing user access behaviour [30]. Since users are treated as anonymous in most Web servers, two heuristic strategies, the proactive strategy and the reactive strategy, have been proposed to help differentiate users [30]. The proactive strategy tries to unambiguously associate each request to a site visitor before or during the visitor's interaction with the site, while the reactive strategy attempts to associate requests with the visitor after the interaction with the site, based on the existing, incomplete records. As a rule, a cookie-based identifier is a must for applications of proactive strategy [30]. However, the use of cookie needs to comply with existing laws [37], at least that users must be clearly made aware of its presence. Therefore, the proactive strategy is not always a feasible option.

In this thesis, we apply the reactive strategy to web logs, and approximate users in terms of IP address, type of operating system and browsing software. In other words, requests are treated as from the same user and put into the same group under that user only if these requests possess the same IP address, operating system and browsing software.

3.2.3 Session identification

For log entries from one user that span long periods of time, it is very likely that the user has visited the website more than once. As the final preprocessing step

for web usage mining, the session identification aims at dividing web logs of each user into individual access sessions. A simple method, called *session-duration-based* method, is to set a session duration threshold. If the duration of a session exceeds a certain limit, it could be considered that there is another access session of the user. Discovered from empirical findings, a 30-minute threshold for total session duration has been recommended [29, 30]. The session-duration-based method is defined as follows:

Definition 3.1: The duration of a session must not exceed a threshold θ. Let t_0 be the time stamp of the first URL request in a session. A URL request from the same user with time stamp t is assigned to this session if $t - t_0 \leq \theta$. Otherwise, this URL request becomes the first request of the next new session.

Likewise, another commonly used approach for the session identification is the *page-stay-time-based* method. The time spent on a page must not exceed a threshold. If the difference of the requested time between the request most recently assigned to a session and the next request from the user is greater than the threshold, it can be assumed that a new access session has started. A conservative threshold for page-stay time, 10 minutes, has been proposed to capture the time for loading and studying the contents of a page [29, 30]. The definition of the page-stay-time-based method is given below:

Definition 3.2: The time spent on a web page must not exceed a threshold θ. Let t_0 be the time stamp of the URL most recently assigned to a session. The next URL request from the same user with time stamp t is assigned to this session if $t - t_0 \leq \theta$. Otherwise, this URL request becomes the first request of the next new session.

In fact, these approaches are the two time-oriented heuristic methods introduced in the Section 2.2. We decide to apply these heuristic methods to the task of identifying sessions. It is more advisable for us to choose one method leading to better experimental results in the later steps. Once the sessions are identified, we split them up into two sets based on the date of the data entries: the training and the testing sets of sessions. The training set will be used by the next module of web usage mining to build profiles of users, while the testing set is prepared for the succeeding experiments of classification and prediction, which are two objectives of the system.

3.3 Web Usage Mining

The task of the second system module is to perform web usage mining on the user access sessions derived from web-log preprocessing. Our way to accomplish this task is to cluster the user access sessions. Clustering is an important operation of web usage mining, which aims to group sessions into clusters based on their common properties. Since access sessions are the images of browsing activities of users, the representative user navigation patterns can be obtained by clustering them. Then, these patterns will be used to help the process of user profiling, the main task of the third system module.

In this section, we will introduce how we perform the session clustering and how we identify the optimal number of clusters from clustering results. First, we start with the session vectorization.

3.3.1 Session vectorization

Let P be a set of web pages accessed by users in Web server logs, $P = \{p_1, p_2, \cdots, p_m\}$, each of which is uniquely represented by its associated URL. Let S be a set of user access sessions. Hence, $S = \{s_1, s_2, \cdots, s_n\}$, where each $s_i \in S$ is a subset of P. To facilitate the clustering operation, we represent each session s as an m-dimensional vector over the space of web pages, $s = \{w(p_1, s), w(p_2, s), \cdots, w(p_m, s)\}$, where $w(p_i, s)$ is a weight assigned to the i^{th} web page $(1 \leq i \leq m)$ visited in a session s. Note that it is allowed that a web page $p_i \in P$ repeats in each user access session $s_i \in S$. Regardless of the navigation sequence, we concentrate on the specific web pages visited in a session.

The weight $w(p_i, s)$ needs to be appropriately determined to capture a user's interest in a web page. In general, all the accessed page can be considered interesting to various degrees because users visited them. We propose a weight measure for approximating the interest degree of a web page to a user. First, let us introduce two concepts related to this measure, "*Frequency*" and "*Duration*".

"*Frequency*" is the number of visits of a web page. It seems natural to assume that web pages with a higher frequency are of stronger interest to users. The formula of "*Frequency*" is given in the Eq.(3.1), which is normalized by the total number of

visits of web pages in the session.

$$Frequency(Page) = \frac{Number\ of\ Visits\,(Page)}{\sum_{Page \in VisitedPages}(Number\ of\ Visits\,(Page))} \tag{3.1}$$

"*Duration*" is defined as the time spent on a page, i.e., the difference between the requested time of two adjacent entries in a session. We conjecture that the longer time a user spends on a page, the likelier the user is interested in the page. If a web page is not interesting, a user usually jumps to another page quickly [38]. However, a quick jump might also occur due to the short length of a web page. Hence, it is more appropriate to accordingly normalize "*Duration*" by the length of the web page. We use Eq.(3.2) to measure the "*Duration*" of a web page,

$$Duration(Page) = \frac{TotalDuration\,(Page)\,/Size\,(Page)}{\max_{Page \in VisitedPages}(TotalDuration\,(Page)\,/Size\,(Page))}, \tag{3.2}$$

where "*Duration*" of a web page is further normalized by the max "*Duration*" of pages in the session. For the last access web page in each user access session, it is not possible for us to estimate its duration by calculating the difference of requested time. We have used the average duration of the relevant session as the estimated duration for the last access event.

In this work, "*Frequency*" and "*Duration*" are considered two strong indicators of users' interest. Therefore, in the weight measure we devised, "*Frequency*" and "*Duration*" are valued equally. We decide to use the harmonic mean of "*Frequency*" and "*Duration*" to represent the interest degree of a web page to a user in the session, shown as below.

$$Interest(Page) = \frac{2 \times Frequency\,(Page) \times Duration\,(Page)}{Frequency\,(Page) + Duration\,(Page)} \tag{3.3}$$

Eq.(3.3) guarantees that "*Interest*" of a page is high only when "*Frequency*" and "*Duration*" are both high. Meanwhile, the value of "*Interest*" is normalized to be between 0 and 1, which is not only convenient for understanding but also suitable for session clustering.

In the end, every user access session is successfully transformed into an m-dimensional vector of weights of web pages, i.e. $s = \{w_1, w_2, \cdots, w_m\}$, where m is the number of web pages visited in all user access sessions. However, if the number of dimensions m

exceeds a reasonable size, it not only consumes large amounts of processing time when clustering sessions, but also limits the applicability of the system in the real world. For reducing dimensions, we have used a frequency threshold f_{min} as a constraint to filter out web pages that are accessed less than f_{min} times in all access sessions. For our system, we found that 80% of web pages appearing in the training access sessions were visited at least 10 times. We consider that these web pages are representative pages which drew intensive attention of users. Therefore, we finally set $f_{min} = 10$.

3.3.2 Session clustering

Given the transformation of user access sessions into a multi-dimensional space as vectors of web pages, standard clustering algorithms can partition this space into groups of sessions that are close to each other based on a distance measure. We choose K-means as the base method to cluster vectors because K-means is a commonly used, relatively efficient clustering method. WEKA 3.4 machine learning toolkit [39] is used to perform the K-means algorithm. In addition, the most popular *Euclidean distance* is adopted as the distance measure, which is defined in the Eq.(3.4) as

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2}, \tag{3.4}$$

where $i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \cdots, x_{jp})$ are two p-dimensional data objects. Algorithm 1 shows the details of the K-means algorithm.

Algorithm 1 K-means

Input: $D = \{t_1, t_2, \ldots, t_n\}$ //Set of elements
k //Number of desired clusters

Output: K //Set of clusters

Assign initial values for means $m_1, m_2, \ldots, m_k$;
repeat
Assign each item t_i to the cluster which has the closest mean;
Calculate new mean for each cluster;
until Some termination criteria is met.

As a rule, the repeating steps in the K-means algorithm terminates when some termination criterion is met. These criteria includes that:

(1) No element is moved from one cluster to another.

(2) The algorithm reaches the maximum number of iterations.

(3) Some evaluation shows the algorithm performance has reached maximum.

In our system, the K-means algorithm terminates when either the criteria (1) or (2) is satisfied.

The clustering result is a set of clusters, $C = \{c_1, c_2, \cdots, c_k\}$, in which each $c_i\,(1 \leq i \leq k)$ is a subset of the set of user access sessions S, and k is the number of clusters. We compute a mean vector m_c for each session cluster $c \in C$ as its representation. Each mean vector represents the representative user navigation pattern of a cluster in which a particular set of web pages are accessed. The mean value for each web page in the mean vector is computed as the average weight of the web pages across total access sessions in the cluster. Therefore, the mean value is also between 0 and 1. Also, a weight threshold for the mean vector of each session cluster, w_{min}, is set as a constraint to filter out web pages with mean value below the threshold in the cluster. Web pages remained in each cluster are considered of more interest to users, and then become the representative navigation pattern of the cluster.

In our system, *user navigation patterns* are described as the common browsing characteristics among a group of users. Since many users may have common interests up to a point during their navigation, navigation patterns should capture the overlapping interests or the information needs of these users. In addition, navigation patterns should also be capable to distinguish among web pages based on their different significance to each pattern. *user navigation pattern* is defined as follows:

Definition 3.3: A user navigation pattern np captures an aggregate view of the behaviour of a group of site users based on their common interests or information needs. As the results of session clustering, $NP = \{np_1, np_2, \cdots, np_k\}$ is used to represent the set of user navigation patterns, in which each np_i is a subset of P, the set of web pages.

3.3.3 Identification of the optimal number of clusters

Like some other partitioning clustering algorithms, for instance, PAM, Genetic and CURE, the number of clusters k is necessary to be specified in advance as the input

of the K-means algorithm. For our system, we need to identify the optimal number of clusters on the access sessions from the clustering results in an unsupervised way because this number determines how many representative navigation patterns will be extracted from user access sessions, and how many user profiles are supposed to be further constructed in the next system module. The *optimal number* means that the partition of user access sessions can best reflect the distribution of sessions, and can also be validated by user's inspection [32].

Our approach is to apply the K-means method to the access sessions using k values ranging from 2 to n (n is a number above 2). For each k value, we evaluate the quality of the clustering result using the measures Cmp and Sep, which are defined in the Section 2.3. In this way, we are able to observe the varying pattern of the score values according to the change of k. Naturally, the most satisfactory quality score indicates the best partition of access sessions, while the corresponding k value suggests the optimal number of clusters on the access sessions. In addition, we will give the inspection and interpretation using our domain knowledge to assist in identifying the optimal number of clusters.

3.4 Building Navigation Pattern Profiles

As the key module of the whole system, user profiling attempts to integrate the representative user navigation patterns obtained from the clustering operation with contents of the corresponding web pages to construct the user navigation profiles. In this section, we first introduce the preprocessing operation on web contents. Then, we propose our method on how to combine user navigation patterns with web contents based on N-gram representations. Finally, we give the process of constructing N-gram-based user navigation profiles.

3.4.1 Web content cleaning

As mentioned in the Section 3.3.1, $P = \{p_1, p_2, \cdots, p_m\}$ is a set of web pages accessed by users in all web log entries. Each user navigation pattern obtained from the clustering operation, a small set of web pages, is a subset of P. In consideration of the peculiarities of web pages differing from plain text documents, web pages of P

need to be cleansed before including page contents into the corresponding navigation patterns.

It can be observed that web pages tend to follow some fixed layouts or presentation styles in a standard website [11]. However, what we are interested are the actual textual contents of pages. In order to extract page contents efficiently, we perform several cleaning procedures on the web pages of P before taking further operations. These cleaning steps include:

- Removing HTML, XML or SGML tags;
- Filtering out all punctuations in contents like comma, full stop, quotation mark, etc., only except the underscore in-between words;
- Eliminating all digital numbers;
- Transferring all characters to upper case;
- Deleting all blank lines.

We use $PC = \{pc_1, pc_2, \cdots, pc_m\}$ to represent the set of web pages after cleaning.

3.4.2 Using N-grams to combine web usage mining with content mining

In order to analyze the influence on the site user profiling when combining web usage with content mining, a character N-gram-based approach is proposed to combine user navigation patterns with web contents. In our approach, we attempt to use N-grams to represent the contents of every web page of user navigation patterns. Thus, each navigation pattern is composed of a collection of N-grams, which appear in the web pages of the pattern. To understand the distribution of N-grams in each navigation pattern, two kinds of frequencies, *term frequency* and *document frequency* [9], are computed to be associated with the N-grams.

Defined in the Eq.(3.5), term frequency $tf\,(x_i, j)$ is the normalized frequency of N-gram x_i in the pattern $j \in NP$.

$$tf_{x_i,j} = \frac{freq_{x_i,j}}{\sum_{x_l \in j} freq_{x_l,j}} \tag{3.5}$$

$freq_{x_i,j}$ is the raw frequency of N-gram x_i in the pattern j (i.e., the number of times the N-gram x_i is mentioned in the web pages of the pattern j), and the sum of the raw frequencies of all N-grams mentioned in the pattern j is computed for normalization. As such, Eq.(3.6) defines document frequency $df_{x_i,j}$, which is the number of web pages that N-gram x_i occurs in the pattern $j \in NP$.

$$df_{x_i,j} = \frac{n_{x_i}}{N_j} \tag{3.6}$$

The total number of web pages in the pattern j, N_j, is used for normalization. It makes $df_{x_i,j}$ between 0 and 1.

Therefore, each navigation pattern $j \in NP$ can be represented by a collection of N-gram triples $\{(x_1, tf_{x_1}, df_{x_1}), (x_2, tf_{x_2}, df_{x_2}), \cdots, (x_n, tf_{x_n}, df_{x_n})\}$. The algorithm for transforming each user navigation pattern to its collection of N-gram triples is given in the Algorithm 2.

According to the definition given in the Section 2.1.4, our approach of combining web usage mining with content mining can be categorized into the *pre-mining integration*.

3.4.3 User navigation pattern profiling

In order to understand the representative user navigation patterns, recognize users' particular visiting style and thus cater to the need of upcoming users, we need to build a profile for each representative navigation pattern. If we build the pattern profile based on the whole collection of N-gram triples of each pattern, the profile could be too large and too general. As a result, it might not accurately capture users' interest. We conjecture that the reason why users tend to follow a similar navigation pattern is some contents are intrinsically correlated or in common among the web pages being visited. Therefore, we try to base the pattern profiles on the N-grams which are qualified to be the representatives of each pattern. We attempt to use document frequency $df_{x_i,j}$ to filter out the N-grams from each profile, which are less important to the corresponding pattern, and maintain the size of each profile at the same time.

Given in the Eq.(3.6), document frequency $df_{x_i,j}$ is always between 0 and 1. If an N-gram's $df_{x_i,j}$ is high, it means this N-gram occurs in most of web pages of the

Algorithm 2 N-gram Triples

Input: $j \in NP$ //Set of web pages of the navigation pattern j

Output: N-gram triples for j

V=Φ //Vocabulary of N-grams

for all page $p \in j$ **do**

Extract(p, PC)

//Extracting the corresponding clean page p from PC, the clean page set defined

//in the Section 3.4.1

V$\leftarrow$ V $\bigcup$ N-grams(p)

//N-grams() is the function to produce N-gram tables of each web page [40]

end for

for all N-grams $x_i \in V$ **do**

Build N-gram pair as (x_i, tf_{x_i})

for all page $p \in j$ **do**

if x_i appears in p **then**

n_{x_i}++ //Initial value of n_{x_i} is 1

end if

end for

Build N-gram triple as $(x_i, tf_{x_i}, df_{x_i})$

end for

pattern, and may carry more representative information of the pattern. Otherwise, it might not be appropriate to be included into the profile as the representative N-gram. In our system, we try to build different pattern profiles by varying the threshold value of $df_{x_i,j}$. Then, we use these profiles to perform the experiments of classification and prediction on the testing data. Based on the performance comparison of profiles with different document frequency values, we will find out which $df_{x_i,j}$ will generate pattern profiles that achieve the best experimental results.

3.5 Classification and Prediction

The objectives of the experimental system are to classify user navigation patterns and predict users' future requests. Once we achieve the profiles of user navigation patterns, we perform the experiments of classification and prediction on the testing set of sessions.

For the task of classifying user navigation patterns, we aim to classify user access sessions into the categories to which they respectively belong. Each user session will be assigned a class label of patterns, so users' navigation activities can be clearly identified. For the task of predicting users' future requests, we attempt to predict future requests of an active user session. According to the prediction results, reasonable recommendations can be provided to the active session to better meet the user' need. In our work, these two tasks are all performed on the testing set of sessions. Testing sessions can be directly used for the classification experiment. However, for the prediction experiment, we have to use testing sessions to simulate active user sessions in the real world. Our approach is to divide a testing session into two parts. The first part of the session is simulated as an active session of the current user. So, it is natural that the second part becomes the contents that the user will request. That is, we use the first part of the session to predict its second part.

Inspired by the success of the system for the task of Authorship attribution [26], we try to apply the similar techniques to the experiments of classification and prediction in our system. Namely, given profiles of user navigation patterns and a session profile, we need to determine the pattern profile to which the session profile most likely belongs. The basic idea is simple: for the obtained set of N-gram-based profiles of user navigation patterns $P_i, i = 1, 2, \ldots, k$, we build another N-gram-based profile p for a

user access session and calculate the dissimilarity measures $D(p, P_i), i = 1, 2, \ldots, k$. If the value $D(p, P_s)$ is the smallest one, then the conjecture is that the session with profile p belongs to the navigation pattern with profile P_s. Essentially, this is the well-known k Nearest neighbours (kNN) classification method, with $k = 1$. We believe that profiles with a similar navigation pattern share a similar distribution of character N-grams.

The procedure of constructing the session profile is somewhat similar to the way of user navigation pattern profiling in the last section. Instead of N-gram triples, the session profile is only composed of N-grams pairs $\{(x_1, tf_{x_1}), \cdots, (x_n, tf_{x_n})\}$, where x_i is the character N-grams extracted from accessed web pages of the session while $tf(x_i)$ is the normalized term frequency of x_i. When computing the $tf(x_i)$, we attempt two different methods: *equal weight* and *linear weight*. Same as the way of building user navigation pattern profiles, the equal weight method assumes that all web pages in a testing session are equally important to the profile construction of the session. Thus, N-grams extracted from all the web pages of the session are given equal weights when calculating the $tf(x_i)$. Contrarily, the linear weight method considers that the web pages later accessed in a session capture the user' intention of the session better than the pages accessed earlier. Hence, N-grams are given a linear incremental weight in computing the $tf(x_i)$ according to the access sequence of the web pages, from which N-grams are extracted. We apply these two methods to the calculation of $tf(x_i)$ when constructing session profiles. We want to know which method will lead to better results in the experiments of classification and prediction.

There is also a difference in the procedures of building session profiles between the classification and the prediction. We build the profile for the classification based on the total web pages accessed in each testing session. For the prediction, the web pages accessed in each testing session are divided into two parts. Web pages in the first part are simulated as the total web pages accessed by a current active session, while web pages in the second part are simulated as the next requested pages in the active session that we try to predict in the real world. That is, we construct the session profile only based on the web pages in the first part of each testing session.

To be more specific, we respectively define the two experiments: *classification* and *prediction* as follows:

Definition 3.4: let s be a testing session containing n accessed web pages. For the classification experiment, we build an N-gram-based profile p for the session s based on total n web pages in it, and will determine the navigation pattern to which s belongs by comparing the dissimilarity $D(p, P_i)$ between the session profile p and pattern profiles $P_i, i = 1, 2, \ldots, k$. If the value $D(p, P_s)$ is the smallest one, the session s belongs to the navigation pattern with profile P_s.

Definition 3.5: let s be a testing session containing n accessed web pages. For the prediction experiment, the web pages of the session s are divided into two parts. Web pages in the first part are simulated as the total accessed web pages of an active session a. We build an N-gram-based profile p for the session a based on the first $n-1$ web pages of the session s. Then, we will determine the navigation pattern to which a belongs by comparing the dissimilarity $D(p, P_i)$ between the session profile p and pattern profiles $P_i, i = 1, 2, \ldots, k$. If the value $D(p, P_s)$ is the smallest one, the simulated active session a belongs to the navigation pattern with profile P_s.

Algorithm 3 is applied to calculating the dissimilarity $D(p, P_i)$ between a session profile and navigation pattern profiles. Given two profiles, the algorithm returns a positive number, which is a measure of dissimilarity.

Algorithm 3 Profile Dissimilarity $D(p, P_i)$

Input: Session profile p and pattern profile P_i

Output: Dissimilarity score between two profiles

$sum \leftarrow 0$

for all N-grams x_i contained in profile p or profile P_i **do**

Let tf_p and tf_{P_i} be term frequencies of x_i in profile p and profile P_i (zero if they are not included)

$d(tf_p, tf_{P_i})$ //Dissimilarity measure

$sum \leftarrow sum + d(tf_p, tf_{P_i})$

end for

Return $-$ sum

It is observed that the quality of the algorithm completely relies on the appropriateness of the dissimilarity measure we choose. In our system, we respectively apply the three dissimilarity measures below, d_1, d_2, and d_3, to the calculation of profile

dissimilarity. These dissimilarity measures are the Eq.(2.5), Eq.(2.7) and Eq.(2.8), which have been introduced in the Section 2.6.

$$d_1\left(tf_p, tf_{P_i}\right) = \sum_{x_i \in profile} \left(\frac{2 \times \left(tf_p\left(x_i\right) - tf_{P_i}\left(x_i\right)\right)}{tf_p\left(x_i\right) + tf_{P_i}\left(x_i\right)} \right)^2$$

$$d_2\left(tf_p, tf_{P_i}\right) = \sum_{x_i \in profile} \frac{2 \times \left(tf_p\left(x_i\right) - tf_{P_i}\left(x_i\right)\right)^2}{\left(tf_p\left(x_i\right) + tf_{P_i}\left(x_i\right)\right)}$$

$$d_3\left(tf_p, tf_{P_i}\right) = \sum_{x_i \in profile} \frac{\left|tf_p\left(x_i\right) - tf_{P_i}\left(x_i\right)\right|}{\sqrt{tf_p\left(x_i\right) \times tf_{P_i}\left(x_i\right)} + 1}$$

We will evaluate these measures in the Chapter 4 according to their performance on the experiments of classification and prediction.

3.6 System Performance Evaluation

The last module of the system is the evaluation module, which aims to evaluate the experimental results of classification and prediction. The evaluation of the classification is about whether the navigation pattern that the system assigned to a testing session is the pattern to which the testing session is supposed to belong. For the prediction, the evaluation measures if the navigation pattern assigned to a simulated active session, namely the first part of a testing session, is accordant with the pattern to which the whole testing session should belong.

We base the system evaluation on two measures: classification accuracy $A\left(C\right)$ and prediction accuracy $A\left(P\right)$. The classification accuracy measures the proportion of the number of correctly classified testing sessions to the total number of testing sessions. Once a testing session is correctly labeled with a pattern, we can further understand the users' navigation characteristics by studying the pattern profile. The prediction accuracy describes the ratio of the number of simulated active sessions that share the same navigation patterns with their original testing sessions to the total number of testing sessions. If a simulated active session, i.e. the first part of a testing session, shares the same navigation pattern with the whole testing session, it can be concluded that the contents of web pages in the second part of the testing session also fall into the category of that navigation pattern. Therefore, we can rely on the

profile of a real active user session to predict the user's future requests. Specifically, we can recommend web pages in the navigation pattern that have not been accessed to the real active user session as his most wanted pages.

For the system, the larger the accuracies, the better the results. However, it is not possible for us to calculate the accuracies without correctly pre-labeled testing sessions. Hence, we decide to manually classify the testing set of sessions in advance by assigning each testing session an appropriate label of navigation patterns based on the patterns achieved from web usage mining. The principle of the manual classification work is that the navigation pattern assigned to each testing session must be in accordance with the whole intention of the session.

Chapter 4

Experimental Results

4.1 Dataset and Environment

For our experiments, it is necessary to use such a dataset that allows us to analyze both web log data and web pages. Our experiments have been conducted on an Apache server log access file from the graduate Web server of the Faculty of Computer Science at Dalhousie University. Although there are some widely used, public datasets containing only web pages, we are not able to get both log data and web pages from them. In addition, it is also often seen that experiments in other published work are based on their departmental Web server.

We extracted access entries of two-month period, September and October 2004, from the server log file as our experimental dataset. In this period, there are 1,248,675 access entries in September producing a 226MB log file, and 1,370,373 access entries in October producing a 245MB log file. Access entries of September are used as the training dataset, while access entries of October are prepared as the testing dataset.

All the experiments were executed on a Sun Solaris server at the CS Faculty of Dalhousie University. The server type is SunOS sparc SUNW, Sun-Fire-880. The experimental system was mainly implemented using Perl except that Java was used for implementing the K-means clustering algorithm.

4.2 Web-log Preprocessing Results

In the preprocessing procedures, the original Web server log data are cleansed, formatted, and finally grouped into meaningful user sessions. Table 4.1 presents some statistics of the experimental dataset, including both training and testing sets after the preprocessing operations.

We can see that for the training dataset, 116,166 clean entries are extracted. Meanwhile, there are 12,931 different users who accessed the Web server in September

Attributes	Training set	Testing set
Total access entries	1,248,675	1,370,373
Clean access entries	116,166	111,477
Different access users	12,931	13,062
Accessed web pages (total)	792	804
Accessed web pages ($\geq$ 10 times)	616	623
Identified sessions (total, session duration)	23,791	23,242
Identified sessions ($\geq$ 2 requests, session duration)	12,402	10,204
Identified sessions (total, page-stay time)	24,756	24,303
Identified sessions ($\geq$ 2 requests, page-stay time)	12,675	10,546

Table 4.1: Statistics of Experimental Dataset

2004 based on the techniques of user differentiation. In this period, 792 web pages were visited and 616 of them were accessed at least 10 times. As introduced in the Section 2.2, we applied two time-oriented heuristic methods to the procedure of identifying access sessions. One is based on a 30-minute threshold of the session duration, and the other method relies on a 10-minute threshold of the page-stay time. Although totally 23,791 sessions were identified from the training set by the session-duration-based method, only 12,402 of them contain more than 2 requests. Furthermore, it is observed that the number of sessions identified by the page-stay-time-based method is generally more than the number of sessions identified by the session-duration-based method for both the training and testing sets. We will determine one session identification method which leads to better experimental results.

We assume that identified sessions containing more than 2 requests are more suitable for our experiments since it might carry more information about users' intention. Therefore, only these sessions in the training set are chosen as the experimental training set for web usage mining, while only these sessions in the testing set are prepared as the experimental testing set for the classification and the prediction.

4.3 Web Usage Mining Results

As mentioned in the Section 3.3.1, we used a frequency threshold $f_{min} = 10$ in our system as a constraint to filter out web pages that were accessed less than 10 times in the training dataset. Therefore, the dimension size of the vector representing each

training session is reduced to an appropriate range. As shown in the Table 4.1, only 616 web pages were accessed more than 10 times in the training dataset. Hence, each training session is represented as a 616-dimensional vector after the session vectorization.

WEKA 3.4 machine learning toolkit [39] was used to perform the K-means clustering algorithm. To facilitate the operation of WEKA, we transformed all the training session vectors into an $ARFF$ file, which is a standard input file format of WEKA. Since we tried two methods of identifying sessions, two $ARFF$ files were produced respectively. Namely, one is for sessions identified by the session-duration-based method, and the other is for sessions identified by the page-stay-time-based method.

For each of the $ARFF$ files, we applied the K-means algorithm using k values ranging from 2 to 25 as the input number of desired clusters. For each k value, we computed the *cluster compactness* (Cmp), the *cluster separation* (Sep) and the combination measure *overall cluster quality* (Ocq) to evaluate the quality of the corresponding clustering result. These are the evaluation measures of clustering quality, Eq.(2.1), Eq.(2.2), Eq.(2.3) and Eq.(2.4), which have been introduced in the Section 2.3, and they are also given as follows:

$$Cmp = \frac{1}{C}\sum_{i}^{C}\frac{v\left(c_i\right)}{v(X)}$$

where C is the number of clusters generated on the data set X, $v\left(c_i\right)$ is the variance of the cluster c_i, and $v\left(X\right)$ is the variance of the data set X.

$$v\left(X\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d^2\left(x_i, \overline{x}\right)}$$

where $d\left(x_i, x_j\right)$, for example the *Euclidean distance*, is a distance measure between two vectors x_i and x_j, N is the number of members in X, and $\overline{x}$ is the mean of X.

$$Sep = \frac{1}{C\left(C-1\right)}\sum_{i=1}^{C}\sum_{j=1,j\neq i}^{C}\exp(-\frac{d^2(x_{c_i}, x_{c_j})}{2\sigma^2})$$

where σ is the standard deviation of the data set X, C is the number of clusters, x_{c_i} is the centroid of the cluster c_i, $d\left(\right)$ is the distance measure used by the clustering system, and $d(x_{c_i}, x_{c_j})$ is the distance between the centroid of c_i and the centroid of c_j.

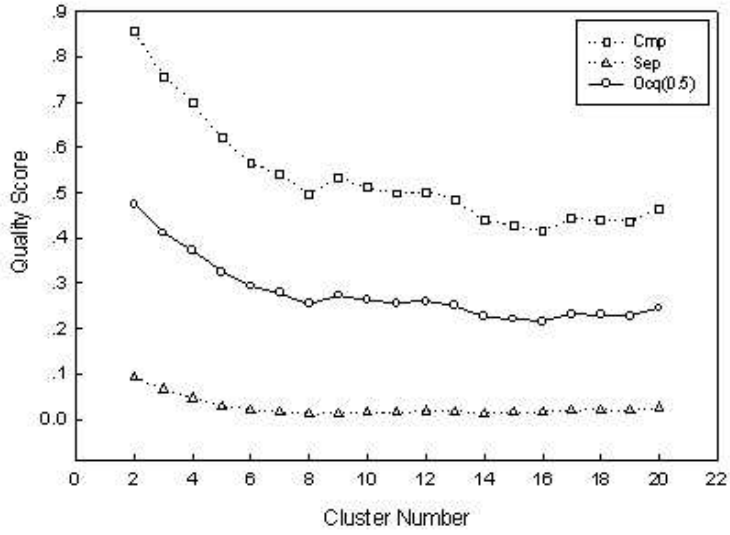


Figure 4.1: *Cmp*, *Sep*, and *Ocq* (0.5) of K-means on the Training Sessions Identified by the Session-duration-based Method

$$Ocq\,(\beta) = \beta \cdot Cmp + (1 - \beta) \cdot Sep$$

where $\beta \in [0, 1]$ is the weight that balances the measures *Cmp* and *Sep*.

We found that when the k exceeds 20, some partitions of clustering results contain access sessions less than 1% of the total training sessions. We consider that the navigation patterns that these partitions present are not representative patterns of the total training sessions. Thus, we focused on the clustering results with k values not exceeding 20. Figure 4.1 and Figure 4.2 respectively depict the change of *cluster compactness*, *cluster separation*, as well as *overall cluster quality* on the clustering results obtained by two session identification methods with varying k from 2 to 20. $2\sigma^2 = 0.25$ was used for the ease of evaluation in the *cluster separation*, and $\beta = 0.5$ was adopted for computing *overall cluster quality* to give equal weights to *cluster compactness* and *cluster separation*.

Shown in the Figure 4.1, it is noted when k increases, the *Cmp* gradiently decreases and tends to go up for the last several k values, while the *Sep* slightly increases after the initial decrease (not obvious due to the scale). It is natural that the increase of partitions on the dataset generally leads to the decrease of the size of each partition, which results in higher compactness in each partition. Also, the decrease of the distances among the partition centroids, which causes lower separation of partitions.

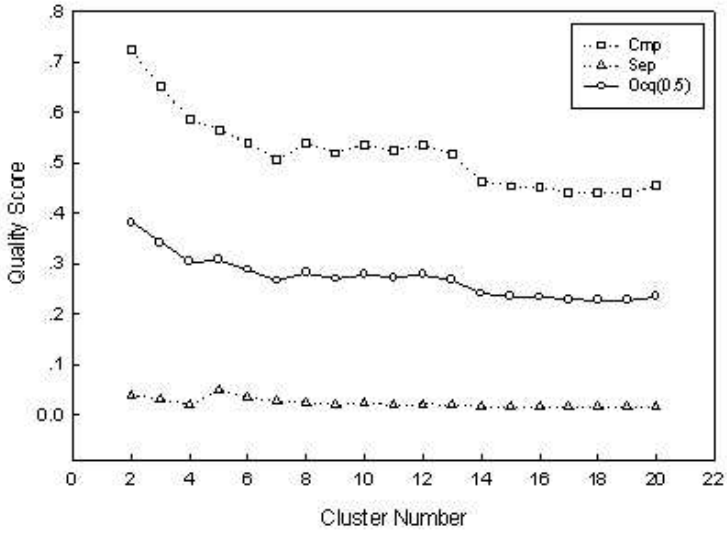


Figure 4.2: *Cmp*, *Sep*, and *Ocq* (0.5) of K-means on the Training Sessions Identified by the Page-stay-time-based Method

However, as the exceptions, the *Cmp* presents two locally minimal values at $k = 8$ and $k = 16$, and the decreasing trends of *Cmp* at $k = 8$ and $k = 16$ are evidently different from those at different k values. Furthermore, the *Ocq* also shows the locally minimal values at $k = 8$ and $k = 16$. This indicates that for the training sessions identified by the session-duration-based method, there are two optimal numbers of clusters, 8 and 16, based on the *Euclidean distance*. As introduced in the Section 2.3, a smaller value of the combination measure *Ocq* indicates a higher quality of the overall output clusters. Since the *Ocq* at $k = 16$ is even lower than the *Ocq* at $k = 8$, we chose 16 as the final number of clusters on the training session set. This number is also supported by the inspection in terms of our domain knowledge. We noticed that 16 clusters avoided the situation that happened among 8 clusters, in which some of important clusters were not partitioned from the others.

Similarly, in the Figure 4.2, both the *Cmp* and the *Ocq* present the locally minimal values at $k = 7$ and $k = 18$. Hence, two optimal numbers of clusters can be recognized on the training sessions identified by the page-stay-time-based method, 7 and 18, in terms of the *Euclidean distance*. After comparing the *Ocq* values, 18 was finally chosen as the number of clusters on the training session set.

We also gave a comparison between the clustering results of two session identification methods by comparing the combination measure *Ocq* of two methods with

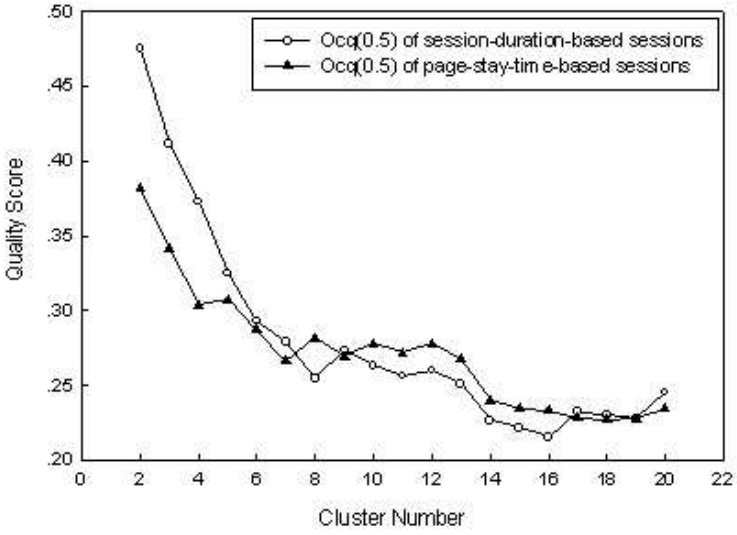


Figure 4.3: $Ocq(0.5)$ Comparison between Clustering Results of Two Methods

varying k from 2 to 20. Shown in the Figure 4.3, it is obvious that when k ranges between 8 and 19, the session-duration-based method generally achieve a better clustering quality than the page-stay-time-based method. It is also noted that the Ocq at k=8 and $k = 16$, two optimal numbers of clusters in terms of the session-duration-based method, are all higher than the Ocq at $k = 7$ and $k = 18$, two optimal numbers of clusters based on the page-stay-time-based method. In addition, when we took a close look at the clustering results, we noticed that at $k = 18$, for the clustering result of the page-stay-time-based method, some different partitions shared same or similar topics. This means some clusters were over-partitioned. Therefore, we drew the conclusion that the training sessions identified by the session-duration-based method produced the better clustering results than the training sessions identified by the page-stay-time-based method. As a result, 16 clusters became the final clustering result of the web usage mining on the training session set. Meanwhile, as our final choice for session identification, we used the testing sessions identified by the session-duration-based method as the experimental testing session set for the rest of our experiments.

For the 16 clusters, each cluster is a subset of the training session set. As introduced in the Section 3.3.2, we computed a mean vector for each cluster. Then, we extracted the corresponding web pages accessed in each cluster based on the weight values of its mean vector. A weight threshold for the mean vector of each cluster was

set to limit the number of web pages in the cluster. Finally, the number of web pages in every cluster was all within 150. Meanwhile, we extracted a brief topic summary for each cluster in terms of our understanding of the contents of the web pages remained in each cluster. Table 4.2 gives a specific description on each of the obtained 16 clusters, including the proportion of training sessions, the number of web pages and the topic summary.

Cluster label	Proportion of training session	Number of web pages	Topic summary
1	3%	43	CS scholarships related
2	7%	75	Graduate programs related
3	2%	34	Student services
4	5%	84	Java programming related
5	3%	41	CS news
6	4%	47	Interviews of CS professors
7	5%	75	CS professors' personal sites
8	10%	13	Webcams related
9	28%	147	Miscellanies
10	3%	31	Administration related
11	3%	46	Prospective students related
12	6%	56	Descriptions and slides of courses
13	7%	48	Technical reports
14	5%	24	Undergraduate program descriptions
15	4%	53	Reseach projects related
16	5%	23	CS pictures

Table 4.2: Description on Each Cluster

From the table, we can see that the 9^{th} cluster accounts for the largest proportion of training sessions, 28%, and contains the most web pages, 147, among all 16 clusters. These indicate that this cluster stands for the most frequent navigation pattern of the training dataset. The topic of the 9^{th} cluster is "miscellanies", which proves that lots of users tended to browse various kinds of information on the Dalhousie CS website, and did not show their interest on any particular topics. It is also worth noticing that although the 8^{th} cluster contains only 13 web pages, it possesses of the second largest proportion of training sessions. This suggests that web pages of the 8^{th} cluster did draw a lot of users' attention. The topic of this cluster is "Webcams related", which means that a good amount of users accessed the webcam related web

pages, including webcam of university constructions, webcam of streets, webcam of computer servers and administration of webcams. In fact, the new webcams of our CS faculty started to take effect in September 2004. Each webcam shows different scenes around or inside the Dalhousie Computer Science building to provide convenience. Our experimental dataset recorded the browsing activities of users who were interested in the webcams at that time. In the meantime, it is also seen that other clusters are evenly proportioned with different specific topics, and all include a certain number of web pages.

4.4 User Profiling Results

Following the methodology in the Chapter 3, we extracted character N-grams from contents of web pages of each cluster, and computed the *term frequency* and the *document frequency* of N-grams to construct the collections of N-gram triples for each cluster. The Perl package Text::Ngrams [40] was used to produce N-gram tables of each web page. Experiments [26] demonstrated that processing N-grams of sizes larger than 10 was slow and only got comparable experimental results with N-grams of smaller sizes. Therefore, we built the N-gram triple collections on N-grams sizes from 1 to 10. We want to find out which N-gram size will produce user profiles that lead to the better results in the experiments of classification and prediction.

We used the *document frequency* to filter out the less important N-grams, and maintain the number of N-grams in the user profiles. We attempted seven different *document frequency* values to build user profiles for each of 16 clusters. These seven *df* values are: 5%, 10%, 25%, 33%, 50%, 66% and 75%. For instance, 5% denotes that the N-grams remained in the user profile of the cluster at least appeared in 5% of the web pages of the cluster. Therefore, the smaller the *df* value, the more the N-grams in the profiles. We attempt to figure out which *df* value will produce user profiles that achieve the best results in the experiments of classification and prediction. For each *df* value, we need to build user profiles of N-grams sizes from 1 to 10 for each of 16 clusters, that is, totally 160 profiles. However, when we used the *df* value 75% to perform the experiment, we found that some built cluster profiles were totally empty. Namely, no N-grams showed up at least in 75% of the web pages in these clusters. As a result, we only used the other six *df* values to construct user profiles. For each

df value, we calculated the number of N-grams contained in each of 160 profiles, and the detailed results are listed in the Appendix A.

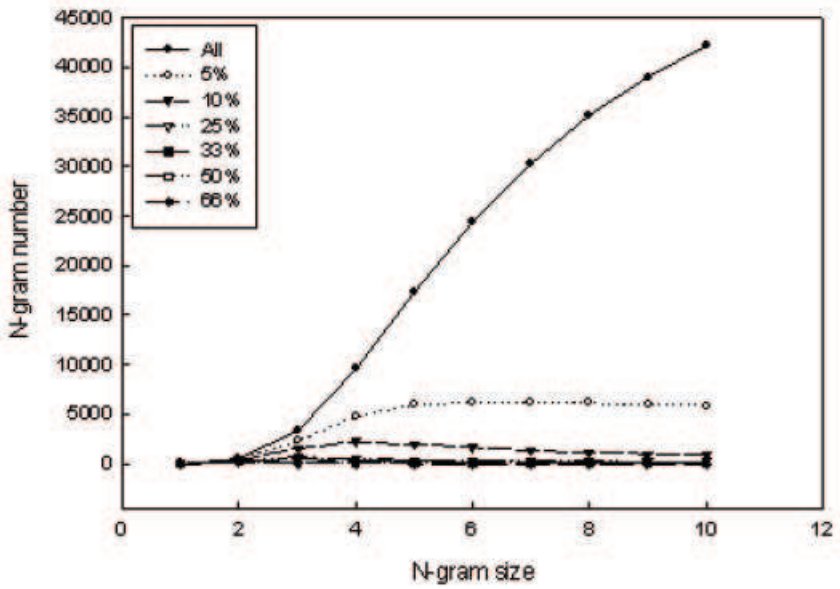
In order to have an overall understanding of the distribution of N-gram numbers in user profiles, for each *df* value, we computed the mean values of N-gram numbers of each of 10 N-gram sizes by finding the average of the N-gram numbers of each N-gram size across 16 clusters. In addition, for comparison, we also computed the mean values of the numbers of all N-grams in each size before using *df* to filter out any N-grams. Here, we used "All" to denote them. Table 4.3 lists the obtained mean values.

Profile size	N-gram size									
	1	2	3	4	5	6	7	8	9	10
All	27	495	3291	9606	17334	24296	30218	35144	38985	42128
5%	27	428	2275	4795	5934	6243	6278	6119	5990	5867
10%	27	373	1561	2230	1959	1645	1374	1150	1007	899
25%	27	285	688	540	401	325	269	241	223	212
33%	27	252	463	308	229	180	148	131	119	111
50%	25	192	223	132	96	80	68	61	54	50
66%	24	142	110	55	39	31	26	23	20	18

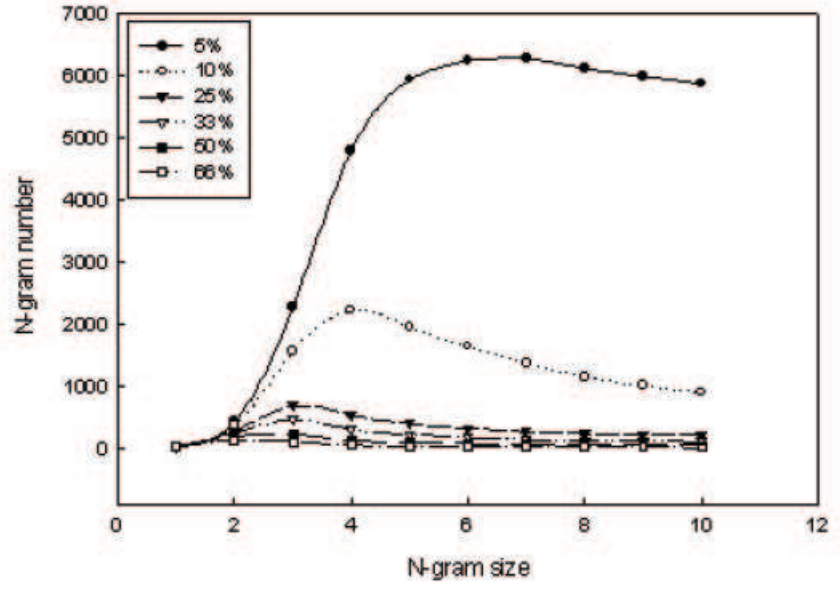
Table 4.3: Mean Values of N-gram Numbers of Profiles

According to the data in the Table 4.3, Figure 4.4 was drawn to illustrate the distribution of N-gram numbers in user profiles of each *df* value. Figure 4.4(a) describes the changing patterns of average numbers of N-grams in the "All" and in the profiles of each *df* value, with the increase of the N-gram size. Since the numbers of N-grams of "All" are generally much more than the numbers of N-grams of all the *df* values, it is hard for us to clearly see all the changing curves under the same scaling. Hence, Figure 4.4(b) is also provided to show the curves of only six *df* values.

It is obvious in the Figure 4.4 that if *df* is not adopted in the construction of user profiles, the number of N-grams increases linearly with the N-gram size. However, when *df* is applied to the user profiling, at the beginning the number of N-grams in the profiles increases sharply to a peak value, and then tends to slowly decrease to a certain level with the increase of the N-gram size. We consider if a profile contains a large number of N-grams, the profile could be too general. Thus, it might lead to more computational cost but might not accurately capture users' interest. Contrarily,



(a) "All" and 6 *df* values



(b) Only 6 *df* values

Figure 4.4: Distribution of N-gram Numbers in User Profiles of (a) "All" and six *df* values (b) only six *df* values

if a profile contains few N-grams, the information that the profile carries might not adequate enough to reflect users' desires. Therefore, it is concluded that the size of user profiles can be efficiently controlled by the *document* $frequency$.

4.5 Results of Classification and Prediction

By constructing the N-gram-based user profiles, we successfully associated the web page contents with the obtained user navigation patterns. Then, we performed the experiments of classification and prediction based on the achieved user profiles. Our testing dataset includes 10,204 sessions identified by the session-duration-based method. According to the methods introduced in the Section 3.5, we built the session profiles by two methods: *equal weight* and *linear weight*. For each method, we respectively performed three dissimilarity measures (Section 2.6) on the testing sessions for both classification and prediction experiments.

Finally, the 10,204 testing sessions were successfully labeled with 16 obtained clusters based on the different experimental requirements of classification and prediction. We named all the labeled testing sessions "Session 2" because all these sessions contain at least 2 access requests. In order to study the influence of session length on the experimental results of classification and prediction, we further extracted two kinds of labeled sessions from all the labeled testing sessions. One contains at least 3 access requests, and the other contains at least 4 access requests. These two kinds of labeled sessions were respectively named "Session 3" and "Session 4". "Session 4" is a subset of "Session 3", while both "Session 3" and "Session 4" are subsets of "Session 2".

4.6 System Evaluation and Result Analysis

To measure the classification accuracy $A(C)$ and the prediction accuracy $A(P)$, the correctly pre-labeled testing sessions are required. Since the total testing session set is very big, we decided to extract 1,500 sessions from it as the sample set for our system evaluation. We manually pre-labeled the 1,500 sessions with 16 resulting clusters. The distribution of cluster labels of the sample sessions is shown in the Table 4.4.

These 1,500 sample sessions stand for the session set "Session 2". We further extracted the sets "Session 3" and "Session 4" from "Session 2", respectively including

Cluster label	Number of sessions	Proportion of sessions
1	27	2%
2	64	4%
3	150	10%
4	42	3%
5	61	4%
6	32	2%
7	128	9%
8	413	28%
9	109	7%
10	137	9%
11	45	3%
12	153	10%
13	36	2%
14	31	2%
15	44	3%
16	28	2%

Table 4.4: Label Distribution of Sample Sessions

759 and 469 testing sessions. It is seen in the Table 4.4 that the 8^{th} cluster accounts for the largest proportion, 28%, of the pre-labeled sample sessions. We defined this largest proportion as the baseline of the classification accuracy and prediction accuracy on the set "Session 2". The baselines of the $A(C)$ and $A(P)$ on "Session 3" and "Session 4" are also given in the Table 4.5. It can be further calculated out that there are 741(1500 minus 759) sample sessions containing exactly 2 access requests, and 290(759 minus 469) sample sessions containing exactly 3 requests out of the total 1,500 sample sessions.

Set label	Number of sessions	Baseline	Cluster label of baseline
Session 2	1500	0.28	8
Session 3	759	0.30	8
Session 4	469	0.29	8

Table 4.5: Baselines of "Session 2", "Session 3" and "Session 4"

4.6.1 Evaluation on classification results

For the classification, we computed $A(C)$ for the results obtained by both *equal weight* and *linear weight* methods. For these two methods, we observed that the $A(C)$ of the results based on the dissimilarity measure Eq.(2.5) are all generally lower than the $A(C)$ of the results based on the Eq.(2.7) and the Eq.(2.8). In addition, the classification results based on the Eq.(2.7) achieve the comparable $A(C)$ with the results based on the Eq.(2.8). However, the highest $A(C)$ is only reached by the classification results based on the Eq.(2.8). Table 4.6 and Table 4.7 respectively list the classification accuracies for both *equal weight* and *linear weight* methods based on the Eq.(2.8). For comparison, the classification accuracies for both *equal weight* and *linear weight* methods based on the other two dissimilarity measures are also given in the Appendix B. Each table respectively shows the $A(C)$ on the sets "Session 2", "Session 3" and "Session 4'. The highest classification accuracies have been accentuated in the bold font style.

For the *equal weight* method, it is observed that the classification accuracies in all three session sets are much higher than the corresponding baseline given in the Table 4.5. This indicates that the sample sessions classified by the system were not only assigned to the most frequent cluster. The best $A(C)$ of 71.1% appears in the set "Session 3", in which the $A(C)$ are generally higher than the $A(C)$ of "Session 2" and "Session 4". According to the Table 4.5, there are 741 sessions containing only 2 access requests, nearly the half of the sample sessions in "Session 2", while all the 759 sessions in "Session 3" contain at least 3 requests. Naturally, it is more difficult to conclude a user' navigation pattern based on his only two access requests than three or more requests. We believe that this is the reason why the $A(C)$ of "Session 3" are generally higher than the $A(C)$ of "Session 2". For the set "Session 4", all the sessions include at least 4 access requests. We conjecture that users might have multiple intentions during navigation when they make more access requests. Therefore, it is hard to conclude a user's activities of multiple intentions into one specific navigation pattern. We think that this explains why the $A(C)$ of "Session 4" are lower than the $A(C)$ of "Session 3". Since "Session 2" stands for the 1,500 sample sessions, the classification accuracies in "Session 2" reflect the overall classification accuracies of the 10,204 testing sessions. It is seen in the Table 4.6 that the classification accuracies

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.391	0.492	0.512	0.569	0.490	0.474	0.457	0.452	0.447	0.443
50%	0.402	0.495	0.548	0.637	0.641	0.624	0.580	0.577	0.566	0.549
33%	0.404	0.504	0.567	0.643	0.675	0.648	0.636	0.597	0.541	0.497
25%	0.419	0.543	0.628	0.655	0.693	**0.699**	0.675	0.648	0.594	0.580
10%	0.418	0.550	0.629	0.653	0.694	**0.699**	0.679	0.653	0.593	0.577
5%	0.408	0.533	0.574	0.642	0.671	0.652	0.638	0.599	0.543	0.495

Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.457	0.536	0.561	0.560	0.497	0.471	0.448	0.444	0.442	0.437
50%	0.479	0.604	0.635	0.654	0.624	0.615	0.567	0.545	0.527	0.525
33%	0.473	0.614	0.651	0.673	0.682	0.691	0.680	0.620	0.588	0.566
25%	0.475	0.619	0.675	0.688	0.698	0.709	0.703	0.689	0.673	0.660
10%	0.477	0.618	0.677	0.689	0.699	**0.711**	0.707	0.692	0.674	0.658
5%	0.469	0.600	0.644	0.663	0.679	0.689	0.673	0.631	0.611	0.579

Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.441	0.463	0.500	0.496	0.474	0.466	0.453	0.441	0.434	0.421
50%	0.472	0.564	0.571	0.615	0.608	0.610	0.573	0.545	0.538	0.516
33%	0.474	0.610	0.602	0.626	0.641	0.654	0.634	0.620	0.593	0.586
25%	0.462	0.603	0.635	0.648	0.656	0.663	0.660	0.655	0.630	0.621
10%	0.470	0.605	0.641	0.648	0.659	**0.664**	0.658	0.652	0.627	0.623
5%	0.464	0.557	0.589	0.617	0.638	0.653	0.639	0.617	0.588	0.579

Table 4.6: Classification Accuracy of Equal Weight Classification Results based on Dissimilarity Measure Eq.(2.8)

of "Session 2" are generally higher than 55%, and the highest accuracy is nearly 70%.

For the *equal weight* method, it is also noticed that the classification accuracies vary with the increase of both N-gram and profile sizes. Figure 4.5 illustrates the distribution curves of classification accuracies according to the change of N-gram and profile sizes for "Session 2", "Session 3" and "Session 4". It is clear that for every curve in the figure, the accuracies increase until reaching a peak value, and then decrease to a certain level with the increase of the N-gram size. For all three sets, profiles with $df = 10\%$ always achieve better accuracies than profiles with other df values, while the highest accuracies are all reached for the N-gram size 6. This

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.386	0.539	0.544	0.565	0.529	0.501	0.466	0.453	0.441	0.440
50%	0.399	0.523	0.587	0.623	0.627	0.622	0.593	0.562	0.556	0.534
33%	0.407	0.577	0.634	0.601	0.659	0.634	0.581	0.551	0.529	0.522
25%	0.397	0.598	0.649	0.620	0.655	0.661	0.628	0.612	0.600	0.587
10%	0.403	0.595	0.648	0.631	0.657	**0.663**	0.631	0.615	0.603	0.592
5%	0.405	0.588	0.641	0.614	0.653	0.642	0.607	0.585	0.574	0.556

Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.449	0.500	0.508	0.527	0.502	0.478	0.450	0.451	0.448	0.441
50%	0.471	0.580	0.586	0.636	0.600	0.592	0.589	0.545	0.552	0.530
33%	0.457	0.615	0.620	0.652	0.683	0.670	0.663	0.610	0.581	0.553
25%	0.458	0.613	0.662	0.681	0.687	0.683	0.691	0.664	0.661	0.645
10%	0.456	0.616	0.664	0.680	0.689	0.684	**0.692**	0.667	0.662	0.648
5%	0.462	0.618	0.627	0.671	0.679	0.674	0.665	0.623	0.601	0.576

Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.408	0.448	0.483	0.493	0.453	0.454	0.438	0.436	0.420	0.417
50%	0.450	0.545	0.546	0.614	0.600	0.589	0.561	0.535	0.522	0.518
33%	0.451	0.595	0.599	0.620	0.647	0.641	0.643	0.619	0.585	0.559
25%	0.446	0.591	0.613	0.624	0.665	0.660	**0.669**	0.651	0.647	0.621
10%	0.453	0.593	0.609	0.627	0.663	0.662	**0.669**	0.653	0.649	0.622
5%	0.448	0.589	0.597	0.619	0.650	0.644	0.647	0.628	0.617	0.592

Table 4.7: Classification Accuracy of Linear Weight Classification Results based on Dissimilarity Measure Eq.(2.8)

indicates that profiles with specific N-gram and profile sizes could lead to the best classification accuracies by the *equal weight* method.

Compared to the classification accuracies obtained by the *equal weight* method, the classification accuracies achieved by the *linear weight* method are generally a little lower except that some accuracies of "Session 4" are even higher. It suggests that only when a user makes more than 4 requests in a session, the web pages later accessed in the session might better capture the user's intention of the session than the pages accessed earlier. In this situation, we assume, the user has to locate his wanted pages through some other pages. Same as the accuracies of the *equal weight* method, it is

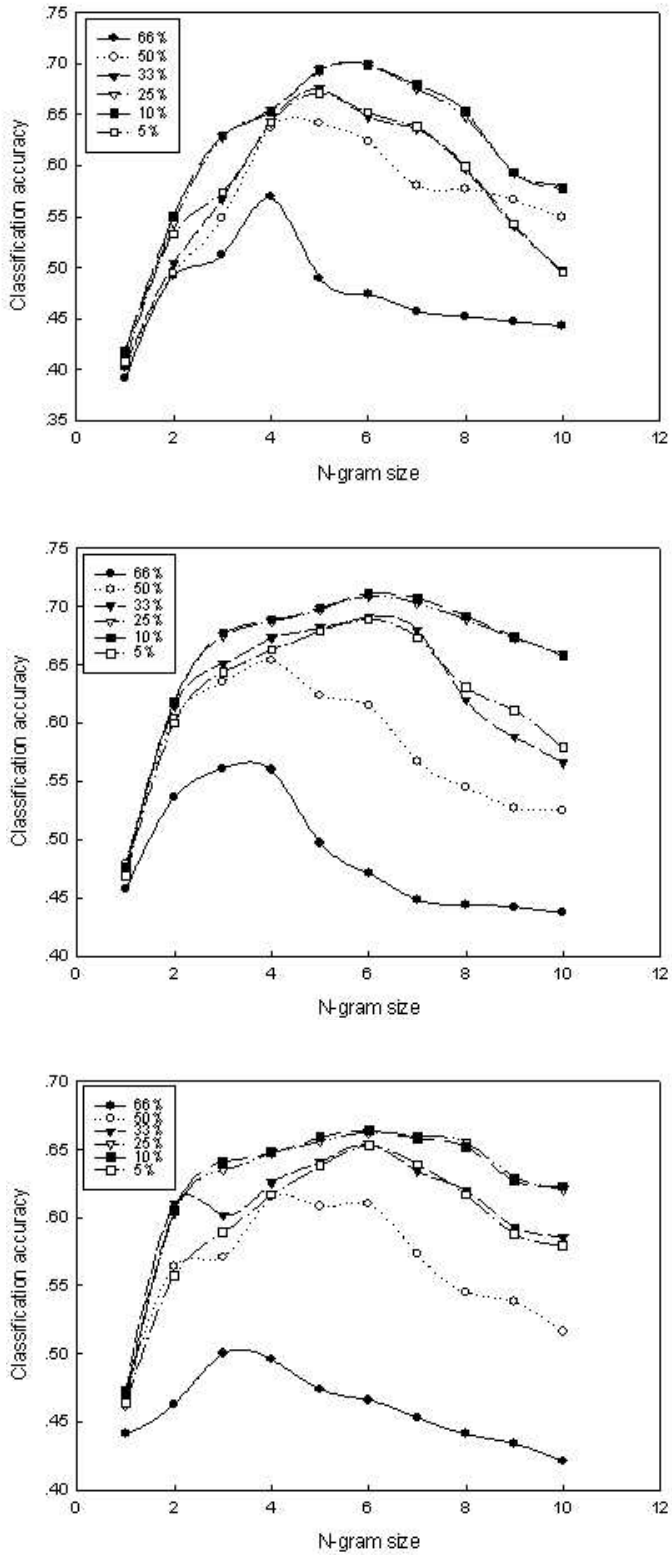


Figure 4.5: Distribution of Classification Accuracy $A(C)$ of "Session 2", "Session 3" and "Session 4" from the top down

seen in the Table 4.7 that accuracies of "Session 3" are also generally higher than the accuracies of "Session 2" and "Session 4". Moreover, the *linear weight* method share the same distribution pattern of classification accuracies with the *equal weight* method in terms of the change of N-gram size and profile size on each session set. For all three sets, profiles with $df = 10\%$ widely achieve better classification accuracies than profiles with other df values, and the best accuracies are reached for the N-gram sizes 6 & 7.

By analyzing the results of classification, we can draw the conclusions:

- The geometric-mean-based dissimilarity measure, Eq.(2.8), achieves the best classification results among three dissimilarity measures.
- The *equal weight* method generally outperforms the *linear weight* method in classification.
- The set "Session 3" always obtains the better classification results than the sets "Session 2" and "Session 4".
- Profiles with specific N-gram and profile sizes can reach the best classification accuracies: In the classification experiments, the N-gram size is 6 or 7, and the profile size is $df = 10\%$.

4.6.2 Evaluation on prediction results

For prediction, we computed $A(P)$ for the results obtained by both *equal weight* and *linear weight* methods. Similar to the classification accuracies, for the two methods, the $A(P)$ of the results based on the dissimilarity measure Eq.(2.5) are all generally lower than the $A(P)$ of the results based on the Eq.(2.7) and the Eq.(2.8). Furthermore, the prediction results based on the Eq.(2.7) achieve the comparable $A(P)$ with the prediction results based on the Eq.(2.8). However, the highest $A(P)$ is only achieved by the prediction results based on the Eq.(2.8). Table 4.8 and Table 4.9 respectively list the prediction accuracies for both *equal weight* and *linear weight* methods based on the Eq.(2.8). For comparison, the prediction accuracies for both *equal weight* and *linear weight* methods based on the other two dissimilarity measures are also given in the Appendix C. Each table respectively shows the $A(P)$ on the sets

"Session 2", "Session 3" and "Session 4'. The highest prediction accuracies have been accentuated in the bold font style.

Profile size	Session 2 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.383	0.418	0.424	0.434	0.406	0.396	0.334	0.331	0.320	0.303
50%	0.411	0.459	0.467	0.479	0.483	0.487	0.451	0.438	0.410	0.388
33%	0.408	0.476	0.499	**0.553**	0.533	0.477	0.452	0.423	0.397	0.356
25%	0.404	0.474	0.496	0.503	0.538	0.488	0.487	0.479	0.464	0.461
10%	0.406	0.476	0.498	0.523	0.539	0.491	0.486	0.477	0.470	0.463
5%	0.404	0.453	0.493	0.517	0.528	0.483	0.471	0.434	0.431	0.406

Profile size	Session 3 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.466	0.506	0.523	0.515	0.477	0.464	0.453	0.449	0.441	0.438
50%	0.485	0.553	0.577	0.601	0.585	0.569	0.551	0.528	0.512	0.503
33%	0.486	0.590	0.620	0.639	0.633	0.607	0.585	0.543	0.500	0.464
25%	0.479	0.573	0.623	0.641	0.638	0.633	0.631	0.621	0.614	0.606
10%	0.481	0.575	0.621	**0.643**	0.637	0.634	0.630	0.626	0.621	0.614
5%	0.483	0.565	0.617	0.638	0.629	0.613	0.592	0.545	0.507	0.460

Profile size	Session 4 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.430	0.469	0.503	0.493	0.464	0.448	0.447	0.433	0.430	0.401
50%	0.449	0.546	0.556	0.609	0.589	0.580	0.556	0.521	0.526	0.489
33%	0.477	0.575	0.599	0.626	0.601	0.581	0.588	0.574	0.550	0.527
25%	0.448	0.569	0.633	0.640	0.642	0.631	0.633	0.620	0.611	0.608
10%	0.453	0.571	0.636	0.641	**0.644**	0.633	0.634	0.622	0.615	0.606
5%	0.452	0.565	0.596	0.627	0.600	0.591	0.585	0.576	0.563	0.542

Table 4.8: Prediction Accuracy of Equal Weight Prediction Results based on Dissimilarity Measure Eq.(2.8)

For the *equal weight* method, it is seen that the $A(P)$ in all session sets are much higher than the corresponding baseline given in the Table 4.5. This indicates that the simulated active sessions of the sample sessions were not only predicted into the most frequent cluster. Compared to the classification accuracies, the prediction accuracies are generally a little lower. We chose the first $n-1$ accessed web pages of a testing session containing n accessed web pages as the simulated active session to build the session profile for predicting the navigation pattern of the whole session.

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.383	0.424	0.423	0.430	0.401	0.377	0.331	0.320	0.314	0.303
50%	0.410	0.469	0.461	0.489	0.474	0.469	0.449	0.439	0.424	0.379
33%	0.399	0.463	0.500	**0.557**	0.524	0.465	0.463	0.417	0.363	0.331
25%	0.407	0.466	0.498	0.492	0.533	0.483	0.492	0.480	0.462	0.458
10%	0.406	0.467	0.501	0.508	0.538	0.492	0.498	0.483	0.470	0.461
5%	0.408	0.466	0.484	0.515	0.519	0.502	0.493	0.464	0.421	0.404

Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.443	0.501	0.506	0.505	0.484	0.453	0.444	0.441	0.439	0.427
50%	0.477	0.571	0.572	0.616	0.579	0.565	0.545	0.530	0.527	0.516
33%	0.464	0.572	0.623	**0.647**	0.632	0.603	0.581	0.535	0.502	0.492
25%	0.471	0.566	0.621	0.615	0.630	0.620	0.627	0.614	0.608	0.603
10%	0.469	0.568	0.619	0.617	0.632	0.623	0.626	0.617	0.613	0.606
5%	0.473	0.570	0.606	0.623	0.629	0.625	0.615	0.603	0.585	0.563

Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.423	0.460	0.481	0.487	0.462	0.445	0.433	0.420	0.417	0.407
50%	0.454	0.565	0.557	0.608	0.577	0.573	0.550	0.535	0.521	0.509
33%	0.455	0.578	0.594	0.611	0.613	0.607	0.587	0.581	0.563	0.545
25%	0.448	0.550	0.603	0.607	0.617	0.610	0.619	0.605	0.600	0.593
10%	0.450	0.557	0.599	0.608	0.617	0.613	**0.620**	0.606	0.597	0.591
5%	0.449	0.563	0.601	0.609	0.612	0.608	0.601	0.587	0.565	0.557

Table 4.9: Prediction Accuracy of Linear Weight Prediction Results based on Dissimilarity Measure Eq.(2.8)

Since prediction results are not based on the total number of web pages of a session, the simulated active sessions carry less user navigation information than the whole testing sessions. We consider, therefore, that this explains why the prediction accuracies are generally lower than the classification accuracies. However, for the overall prediction accuracies of the sample testing set, shown in the Table 4.8, the $A(P)$ of "Session 2" are widely higher than 45%, and the highest $A(P)$ is nearly 54%. In fact, the prediction accuracies of "Session 2" are generally lower than the prediction accuracies of "Session 3" and "Session 4", while "Session 3" and "Session 4" have the comparable prediction accuracies. Since the best $A(C)$ of 64.4% appears in the set

"Session 4", we conjecture that the more requests a user makes in a session, the more navigation information could be utilized for a more accurate prediction of the user's future requests.

For the *equal weight* method, we also noticed that the prediction accuracies also vary with the increase of both N-gram and profile sizes. Figure 4.6 describes the distribution curves of $A(P)$ according to the change of N-gram and profile sizes for "Session 2", "Session 3" and "Session 4". It is obvious that for all three sets, profiles with $df = 10\%$ widely achieve better accuracies than profiles with other df values, and the highest accuracies are all reached for the N-gram sizes 4 & 5. This indicates that profiles with specific N-gram and profile sizes could also lead to the best prediction accuracies by the *equal weight* method.

Compared to the prediction accuracies obtained by the *equal weight* method, the accuracies achieved by the *linear weight* method are generally a little lower in all three session sets. It suggests that the *linear weight* method cannot achieve better prediction accuracies than the *equal weight* method in our experiments. However, the prediction accuracies obtained the *linear weight* method share the same distribution pattern with the prediction accuracies obtained by the *equal weight* method in terms of the change of both N-gram and profile sizes for each session set. For all three sets, profiles with $df = 10\%$ still generally achieve better accuracies than profiles with other df values, and the best accuracies are reached for the N-gram sizes 4 & 7.

By analyzing the results of prediction, we can draw the conclusions:

- The geometric-mean-based dissimilarity measure, Eq.(2.8), achieves the best prediction results among three dissimilarity measures.
- The *equal weight* method outperforms the *linear weight* method in prediction.
- The set "Session 4" obtains the better prediction results than the sets "Session 2" and "Session 3".
- Profiles with specific N-gram and profile sizes can reach the best prediction accuracies: In the prediction experiments, the N-gram size is 4, 5 or 7, and the profile size is $df = 10\%$.

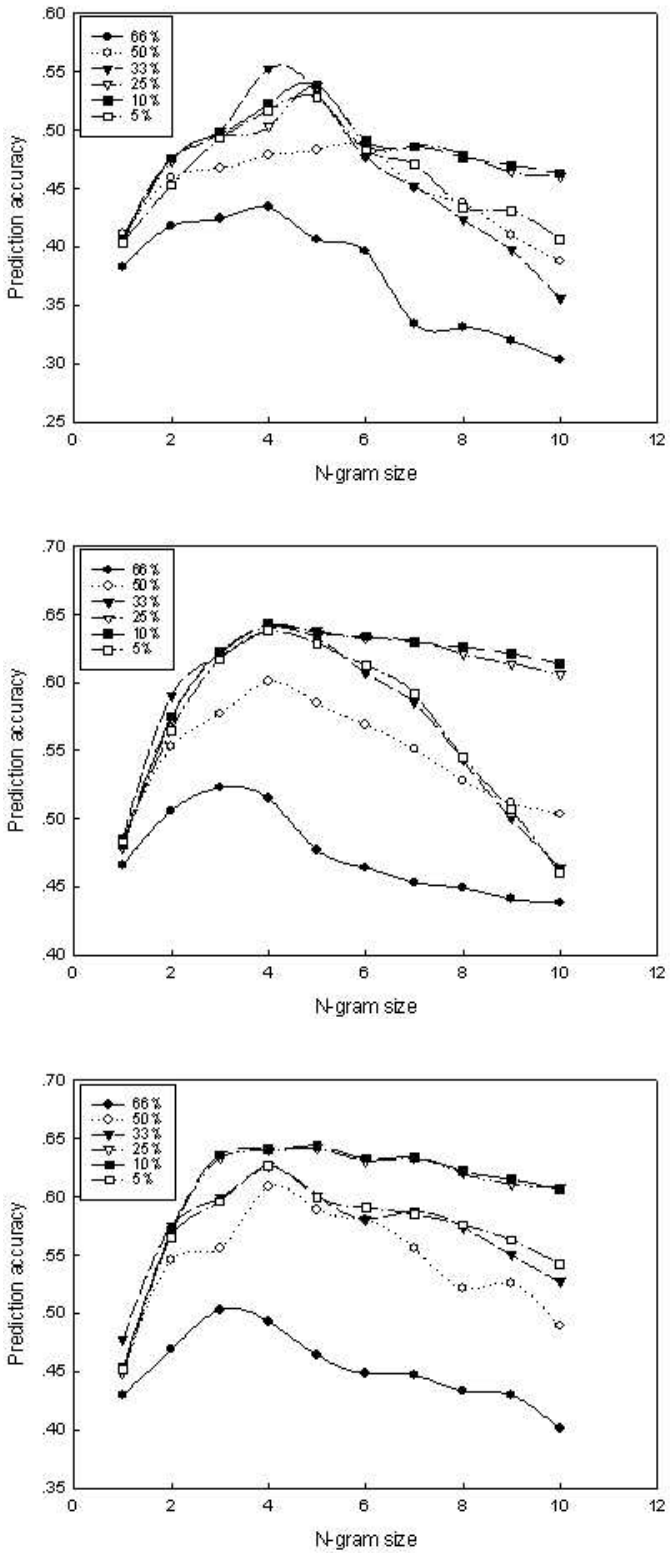


Figure 4.6: Distribution of Prediction Accuracy $A(P)$ of "Session 2", "Session 3" and "Session 4" from the top down

Chapter 5

Conclusion and Future work

5.1 Conclusion

In this thesis, a novel approach is presented to classifying user navigation patterns and predicting users' future requests by combined mining of Web server logs and web contents. We have conducted the experiments on our designed experimental system, which consists of five major modules: web-log preprocessing, web usage mining, navigation pattern profiling, classification and prediction, and system performance evaluation. The dataset used in this thesis is the log access file of our departmental Web server for a two-month period.

First, we attempted two methods to identify user sessions in the web-log preprocessing. To facilitate the web usage mining, we then vectorized the identified sessions based on a proposed weight measure which captures the interest degree of a web page to a user. We extracted navigation patterns of site users by clustering the vectorized sessions with the K-means clustering algorithm. We evaluated the quality of clustering results and identified an optimal number of clusters on the training dataset. This determined the number of the representative user navigation patterns. Next, we associated the mined navigation patterns with the character N-gram based representation of corresponding web pages to build profiles of user navigation patterns. Meanwhile, we used document frequency of N-grams to ensure the quality of pattern profiles. We also tried two means to build both testing session and simulated active session profiles. Based on the profiles of navigation patterns, we then applied the kNN classification method to classify testing and simulated active sessions into the navigation patterns to which they belong. Prediction of users' future requests were further experimented on the classification results of simulated active sessions. Three different dissimilarity measures were tried to implement the kNN classification method. Finally, we evaluated the overall performance of the experimental system in terms of two accuracy measures we defined.

Compared to the page-stay-time-based method, the session-duration-based method is more suitable for identifying sessions during preprocessing Web server logs. This is seen from the experimental results of web usage mining, from which it is also concluded that 16 is identified as the optimal number of clusters on the training dataset. Based on the accuracies obtained by evaluating the 1,500 sample testing sessions, we conjecture that our system improves the classification accuracy to 70% and achieves the prediction accuracy of about 65%. It is much higher than the accuracy reported in the experiments of other studies [7]. Noticeably, three findings have been also achieved in the thesis. Firstly, we have found that the *equal weight* method to build profiles of the testing and simulated active sessions achieves both better classification and prediction accuracy than the *linear weight* method. Secondly, the kNN classification method implemented with the geometric-mean-based dissimilarity measure uniformly obtains the best accuracies in both classification and prediction for the system. Thirdly, the highest classification or prediction accuracy is reached by the profiles with a specific N-gram size and a profile size, which is controlled by the document frequency of N-grams.

5.2 Future work

The following topics are of considerable research interest to us for improving the experimental system in future work:

- During the session vectorization, "*Frequency*" and "*Duration*" are the only two factors in the weight measure due to they are considered two strong indicators for capturing the interest degree of a web page to a user. Some other factors, for example, the sequence of accessed web pages may also have influence on users' interest. We are interested in incorporating more influencing factors into the weight measure of the session vectorization.

- In this thesis, one of the partitioning clustering algorithm, K-means, is used as the base clustering method for web usage mining. Therefore, the number of clusters is necessary to be specified in advance as the input of the algorithm. This leads to more efforts on identifying the optimal number of clusters. We

would like to perform our experiments with some more sophisticate hierarchical clustering methods in which the number of clusters can be automatically determined, and see if better clustering results can be obtained.

- Currently, we build the profiles of navigation patterns based on the total web pages accessed in the training sessions of each resulting cluster. Then, we base both the experiments of classification and prediction on the same pattern profiles. We intend to specifically build pattern profiles for predicting users' future requests. We will build the pattern profiles only based on the first $n-1$ web pages accessed in each session of each cluster and use these profiles to classify the simulated active sessions. We expect the prediction accuracy could be improved to some extent in this way.

- We have attempted two methods to build profiles of both testing and simulated active sessions: *equal weight* and *linear weight*. Although the *equal weight* method outperforms the *linear weight* method in the experiments, we insist that the sequence of accessed web pages reflects the importance of web pages to a user. We want to try some more sophisticate weighting schema for building profiles of both testing and simulated active sessions in the future.

- Borrowing the idea of Guo *et al*, we also attempt to transform user access sessions into "content-enhanced" sessions by clustering web page contents first, and then apply usage mining techniques to build integrated user navigation profiles. We expect that better classification and prediction results could be achieved in this way.

- We consider that the associations among web pages of each obtained user navigation pattern would be useful for capturing togetherness of accessed web pages, and could be used to further discover suitable web page visiting sequences within each pattern, which will assist in the recommendation sequence of web pages to users. In the future, we will apply the association rule mining to the resulting navigation patterns of our system to see if some interesting page visiting rules will be discovered.

- At present, we perform the prediction of users' future requests on the simulated active sessions extracted from testing sessions, and have obtained a quite good prediction accuracy. We would like to incorporate our current off-line mining system into an on-line web recommendation system to observe the degree of real users' satisfaction on the generated recommendations which will be derived from the predicted requests by our system.

Bibliography

[1] Baoyao Zhou, Siu Cheung Hui, and Kuiyu Chang. An intelligent recommender system using sequential web access patterns. In *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–3, 2004.

[2] Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz. Collaborative recommendation via adaptive association rule mining. In *WEBKDD 2000 - Web Mining for E-Commerce – Challenges and Opportunities, Second International Workshop*, August 2000.

[3] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Effective personalization based on association rule discovery from web usage data. In *WIDM '01: Proceedings of the 3rd international workshop on Web information and data management*, pages 9–15. ACM Press, 2001.

[4] Bamshad Mobasher. A web personalization engine based on user transaction clustering. In *Proceedings of the 9th Workshop on Information Technologies and Systems*, 1999.

[5] Dhananjay S. Phatak and Rory Mulvaney. Clustering for personalized mobile web usage. In *Proceedings of the IEEE FUZZ'02*, pages 705–710, 2002.

[6] Dou Shen, Yan Cong, Jian-Tao Sun, and Yu-Chang Lu. Studies on chinese web page classification. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, volume 1, pages 23–27, 2003.

[7] Miriam Baglioni, U. Ferrara, A. Romei, Salvatore Ruggieri, and Franco Turini. Preprocessing and mining web log data for web personalization. In *AI*IA*, pages 237–249, 2003.

[8] Margaret H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.

[9] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[10] Bamshad Mobasher. Web content mining. Accessed in Jun.2005. `http://maya.cs.depaul.edu/~mobasher/webminer/survey/node3.html`.

[11] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 296–305. ACM Press, 2003.

[12] Ron Kohavi and Rajesh Parekh. Ten supplementary analyses to improve e-commerce web sites. In *Proceedings of the Fifth WEBKDD workshop*, 2003.

[13] Brian D. Davison. Predicting web actions from html content. In *Proceedings of the The Thirteenth ACM Conference on Hypertext and Hypermedia (HT'02)*, pages 159–168, College Park, MD, June 2002.

[14] Oh-Woog Kwon and Jong-Hyeok Lee. Web page classification based on k-nearest neighbor approach. In *IRAL '00: Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, pages 9–15. ACM Press, 2000.

[15] "AWStats" official web site. Accessed in Jun.2005. `http://www.awstats.org/`.

[16] "Webalizer" official web site. Accessed in Jun.2005. `http://www.mrunix.net/webalizer/`.

[17] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, and Hua Zhu. Mining access patterns efficiently from web logs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 396–407, 2000.

[18] Alexandros Nanopoulos, Yannis Manolopoulos, Maciej Zakrzewicz, and Tadeusz Morzy. Indexing web access-logs for pattern queries. In *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*, pages 63–68. ACM Press, 2002.

[19] Paulo Batista and Mfirio J. Silva. Mining on-line newspaper web access logs. In *Proceedings of the AH 2002 Workshop on Recommendation and Personalization in eCommerce*, pages 100–108, 2002.

[20] Cyrus Shahabi, Farnoush Banaei Kashani, Yi-Shin Chen, and Dennis McLeod. Yoda: An accurate and scalable web-based recommendation system. In *CooplS '01: Proceedings of the 9th International Conference on Cooperative Information Systems*, pages 418–432. Springer-Verlag, 2001.

[21] Hiroshi Ishikawa, Toshiyuki Nakajima, Tokuyo Mizuhara, Shohei Yokoyama, Junya Nakayama, Manabu Ohta, and Kaoru Katayama. An intelligent web recommendation system: A web usage mining approach. In *ISMIS*, pages 342–350, 2002.

[22] Honghua Dai and Bamshad Mobasher. A road map to more effective web personalization: Integrating domain knowledge with web usage mining. In *International Conference on Internet Computing*, pages 58–64, 2003.

[23] Bamshad Mobasher, Honghua Dai, Tao Luo, Yuqing Sun, and Jiang Zhu. Integrating web usage and content mining for more effective personalization. In *EC-WEB '00: Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, pages 165–176. Springer-Verlag, 2000.

[24] Jiayun Guo, Vlado Kešelj, and Qigang Gao. Integrating web content clustering into web log association rule mining. In *Proceedings of Canadian AI'2005*, Victoria, BC, Canada, May 2005.

[25] William B. Cavnar. Using an n-gram-based document representation with a vector processing retrieval model. In *TREC*, 1994.

[26] Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics*, Nova Scotia, Canada, 2003.

[27] Andrija Tomovic, Predrag Janicic, and Vlado Kešelj. N-gram-based classification and hierarchical clustering of genome sequences. Submitted.

[28] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[29] Bettina Berendt, Bamshad Mobasher, Myra Spiliopoulou, and Jim Wiltshire. Measuring the accuracy of sessionizers for web usage analysis. In *Proceedings of the Workshop on Web Mining at the First SIAM International Conference on Data Mining*, 2001.

[30] Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS J. on Computing*, 15(2):171–190, 2003.

[31] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Technique*. Morgan Kaufmann, 2000.

[32] Ji He, Ah-Hwee Tan, Chew-Lim Tan, and Sam-Yuan Sung. On quantitative evaluation of clustering systems. In Weili Wu and Hui Xiong, editors, *Information Retrieval and Clustering*. Kluwer Academic Publishers, 2002. In press.

[33] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *KDD workshop on Text Mining*, 2000.

[34] William B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA, 1994.

[35] Yingbo Miao, Vlado Kešelj, and Evangelos E. Milios. Comparing document clustering using n-grams, terms and words. Master's thesis, Dalhousie University, 2004.

[36] Robot List. Accessed in Jun.2005. `http://www.robotstxt.org/`.

[37] The Personal Information Protection and Electronic Documents Act (PIPEDA). Accessed in Jun.2005.
http://www.privcom.gc.ca/legislation/02_06_01_e.asp.

[38] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–281. Springer-Verlag New York, Inc., 1994.

[39] Weka: Machine learning software in Java. Accessed in Jun.2005.
http://www.cs.waikato.ac.nz/~ml/weka/.

[40] Vlado Kešelj. Perl package Text::Ngrams, 2003. Accessed in Jun.2005.
http://www.cs.dal.ca/~vlado/srcperl/Ngrams or
http://search.cpan.org/author/VLADO/Text-Ngrams-1.1/.

Appendix A

Number of N-grams

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	27	494	3305	9624	17133	23462	28729	33037	36300	38966
2	27	555	3815	11567	21183	29831	37174	43235	47956	51803
3	27	504	3168	8722	15280	21137	26081	30203	33417	36104
4	27	538	3940	12442	23986	35162	45000	53450	60102	65535
5	27	495	3296	9541	16826	23153	28453	32750	36053	38785
6	27	520	3555	10587	19624	28077	35414	41657	46473	50387
7	27	533	3831	11805	21531	29726	36421	41786	45914	49281
8	26	292	962	1512	1857	2095	2273	2414	2537	2636
9	27	599	5055	17909	38091	59800	79661	97149	111256	122734
10	27	456	2776	7151	11196	14261	16795	18798	20393	21720
11	27	505	3393	9850	17017	22855	27533	31262	34123	36491
12	27	511	3584	10981	20565	29162	36371	42309	46775	50362
13	27	529	3771	11434	20746	28588	34936	39929	43701	46763
14	27	471	2850	6959	10506	13002	14902	16321	17425	18342
15	27	488	3145	8664	14622	19528	23545	26813	29396	31569
16	27	424	2203	4939	7182	8889	10200	11189	11944	12566

Table A.1: Number of N-grams Contained in Each of 160 Profiles for "All"

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	27	444	2466	5459	6714	6801	6471	5845	5339	4898
2	27	415	2043	3555	3429	2954	2448	1995	1704	1450
3	27	429	2267	4814	6060	6257	6010	5479	4932	4387
4	27	422	2183	4197	4317	3759	3103	2414	1954	1585
5	27	438	2432	5327	6560	6639	6314	5715	5180	4679
6	27	464	2568	5842	7684	7953	7596	6732	5812	4999
7	27	415	2053	3549	3268	2729	2253	1803	1524	1298
8	26	291	962	1512	1857	2095	2273	2414	2537	2636
9	27	434	2245	4209	4013	3245	2520	1856	1445	1124
10	27	456	2776	7151	11196	14262	16795	18798	20393	21720
11	27	448	2502	5352	6467	6531	6296	5866	5527	5210
12	27	429	2288	4621	5107	4773	4248	3604	3159	2817
13	27	460	2626	5949	7342	7141	6575	5809	5162	4606
14	27	471	2850	6959	10506	13002	14902	16321	17425	18342
15	27	405	1931	3291	3234	2857	2443	2064	1796	1555
16	27	424	2203	4939	7182	8889	10200	11189	11944	12566

Table A.2: Number of N-grams Contained in Each of 160 Profiles for *df* 5%

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	27	387	1759	2707	2369	1912	1497	1158	953	797
2	27	365	1444	1798	1405	1086	857	670	557	473
3	27	395	1790	3094	3166	2774	2361	1911	1556	1259
4	27	368	1483	1886	1445	1071	776	550	437	360
5	27	391	1754	2685	2351	1862	1410	1064	847	685
6	27	370	1525	2225	1854	1404	1027	717	539	417
7	27	371	1487	1716	1183	842	618	488	409	353
8	26	292	962	1512	1857	2095	2273	2414	2537	2636
9	27	370	1474	1733	1241	864	597	424	335	282
10	27	365	1506	2100	1794	1542	1282	1064	894	766
11	27	393	1742	2547	2143	1729	1389	1122	926	782
12	27	385	1749	2661	2347	1938	1578	1264	1083	947
13	27	383	1617	2156	1697	1353	1072	881	766	683
14	27	403	1847	2971	2962	2729	2492	2274	2107	1962
15	27	373	1444	1794	1382	1085	849	662	536	440
16	27	352	1389	2093	2153	2035	1899	1739	1625	1535

Table A.3: Number of N-grams Contained in Each of 160 Profiles for *df* 10%

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	27	304	781	586	397	296	223	193	174	163
2	27	279	584	365	231	152	103	81	65	54
3	27	302	873	824	547	380	259	179	131	100
4	27	278	612	383	253	177	122	93	76	66
5	27	305	883	746	488	346	245	190	155	132
6	27	297	824	685	473	348	247	198	170	154
7	27	277	543	315	254	218	195	186	178	172
8	25	238	568	681	735	776	797	814	827	840
9	27	284	642	419	289	218	174	153	140	131
10	27	270	615	430	346	293	260	233	215	201
11	27	289	657	443	316	251	204	187	175	165
12	27	299	776	603	362	238	147	106	80	65
13	27	301	741	550	342	225	143	106	80	65
14	27	302	787	699	585	543	505	482	470	459
15	27	264	516	338	276	233	207	193	183	175
16	27	269	600	573	529	498	475	458	452	448

Table A.4: Number of N-grams Contained in Each of 160 Profiles for *df* 25%

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	27	271	542	327	213	139	103	82	67	58
2	26	239	377	194	117	80	59	50	44	39
3	27	279	681	524	353	237	152	109	84	70
4	26	243	379	196	125	88	60	49	39	32
5	27	274	542	311	206	141	101	81	68	59
6	26	270	546	325	196	122	77	57	48	43
7	26	235	342	234	200	184	177	170	166	163
8	25	211	407	470	491	497	496	490	487	483
9	27	254	394	210	133	95	67	53	43	38
10	27	238	424	291	250	216	197	184	177	171
11	27	264	462	263	190	138	111	95	83	76
12	27	257	497	279	169	111	75	59	49	42
13	27	251	463	250	143	92	66	53	46	41
14	27	265	515	380	300	242	189	158	133	117
15	27	230	347	240	209	187	177	171	167	164
16	26	246	489	429	368	310	263	227	200	179

Table A.5: Number of N-grams Contained in Each of 160 Profiles for *df* 33%

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	26	209	241	118	72	59	52	48	44	40
2	25	176	156	69	44	33	28	26	24	22
3	26	229	320	165	96	65	45	36	31	28
4	25	178	155	60	36	29	25	22	19	16
5	26	216	273	127	78	64	52	45	38	32
6	26	218	273	135	78	57	43	38	32	27
7	24	168	175	130	104	84	71	62	54	48
8	25	181	280	261	235	207	179	158	139	126
9	25	181	171	74	44	37	33	30	26	23
10	24	182	220	192	184	178	173	169	166	163
11	26	190	187	92	56	45	40	38	35	33
12	26	204	235	103	65	50	38	30	24	20
13	25	181	189	82	52	43	36	32	28	25
14	25	196	259	150	96	68	44	29	21	16
15	24	172	191	168	150	141	134	129	125	121
16	25	183	246	185	143	112	90	75	62	55

Table A.6: Number of N-grams Contained in Each of 160 Profiles for *df* 50%

Cluster label	N-gram size									
	1	2	3	4	5	6	7	8	9	10
1	24	153	108	48	35	30	27	25	23	21
2	24	118	64	30	23	17	15	13	11	9
3	24	171	164	60	28	21	15	13	11	9
4	24	122	57	23	17	13	12	11	10	9
5	24	158	115	46	29	23	19	16	14	12
6	24	151	123	45	29	22	19	16	14	12
7	24	131	102	56	44	37	34	31	28	25
8	25	132	150	115	97	84	76	69	64	60
9	24	129	70	32	24	17	15	13	11	9
10	24	149	156	129	98	82	70	62	55	49
11	24	141	90	51	40	31	27	22	18	14
12	24	144	100	38	27	19	15	13	11	9
13	24	129	78	35	26	18	15	12	11	9
14	25	151	139	43	22	12	8	6	4	3
15	24	141	118	73	46	39	34	31	28	25
16	25	143	129	57	35	23	16	11	7	5

Table A.7: Number of N-grams Contained in Each of 160 Profiles for *df* 66%

Appendix B

Classification Accuracy

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.306	0.281	0.528	0.526	0.459	0.455	0.452	0.438	0.439	0.434
50%	0.372	0.374	0.520	0.558	0.550	0.538	0.502	0.484	0.482	0.478
33%	0.382	0.452	0.456	0.588	0.528	0.545	0.545	0.537	0.513	0.470
25%	0.380	0.466	0.503	0.500	0.538	0.513	0.512	0.516	0.516	0.519
10%	0.380	0.472	0.507	0.509	0.550	0.527	0.546	0.520	0.516	0.494
5%	0.378	0.442	0.448	0.473	0.529	0.514	0.534	0.508	0.501	0.473
Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.356	0.220	0.497	0.490	0.438	0.438	0.435	0.428	0.431	0.427
50%	0.427	0.419	0.489	0.526	0.511	0.518	0.494	0.472	0.467	0.461
33%	0.452	0.490	0.501	0.575	0.592	0.601	0.600	0.604	0.585	0.543
25%	0.444	0.523	0.593	0.598	0.652	0.655	0.647	0.633	0.629	0.637
10%	0.442	0.525	0.596	0.600	0.657	0.654	0.648	0.630	0.630	0.629
5%	0.442	0.481	0.529	0.574	0.590	0.601	0.609	0.595	0.580	0.542
Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.320	0.196	0.471	0.461	0.410	0.412	0.412	0.403	0.405	0.399
50%	0.395	0.371	0.461	0.503	0.478	0.488	0.465	0.450	0.444	0.437
33%	0.439	0.450	0.459	0.522	0.557	0.569	0.580	0.582	0.569	0.527
25%	0.427	0.491	0.544	0.533	0.616	0.614	0.625	0.633	0.627	0.633
10%	0.430	0.493	0.545	0.542	0.620	0.617	0.629	0.632	0.629	0.628
5%	0.422	0.457	0.468	0.491	0.553	0.572	0.584	0.573	0.566	0.519

Table B.1: Classification Accuracy of Equal Weight Classification Results based on Dissimilarity Measure Eq.(2.5)

Profile size	Session 2 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.384	0.532	0.540	0.549	0.487	0.466	0.456	0.454	0.450	0.449
50%	0.418	0.494	0.580	0.614	0.636	0.624	0.592	0.569	0.560	0.555
33%	0.408	0.545	0.536	0.654	0.666	0.653	0.626	0.584	0.518	0.494
25%	0.408	0.549	0.596	0.625	0.687	0.690	0.670	0.637	0.589	0.586
10%	0.408	0.550	0.589	0.632	0.690	0.691	0.673	0.641	0.593	0.589
5%	0.408	0.531	0.554	0.622	0.654	0.657	0.632	0.600	0.524	0.484

Profile size	Session 3 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.452	0.502	0.519	0.514	0.485	0.473	0.453	0.452	0.445	0.442
50%	0.485	0.575	0.585	0.631	0.622	0.607	0.569	0.550	0.528	0.519
33%	0.473	0.619	0.631	0.664	0.683	0.683	0.663	0.626	0.584	0.560
25%	0.472	0.626	0.679	0.685	0.692	0.701	0.698	0.692	0.665	0.656
10%	0.473	0.624	0.680	0.687	0.693	0.703	0.700	0.691	0.660	0.651
5%	0.473	0.596	0.599	0.639	0.663	0.677	0.673	0.618	0.578	0.570

Profile size	Session 4 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.429	0.469	0.495	0.495	0.463	0.448	0.435	0.431	0.420	0.416
50%	0.478	0.548	0.552	0.608	0.597	0.593	0.561	0.533	0.518	0.501
33%	0.471	0.608	0.599	0.614	0.633	0.644	0.627	0.616	0.589	0.576
25%	0.459	0.601	0.638	0.646	0.655	0.659	0.655	0.659	0.638	0.629
10%	0.463	0.600	0.637	0.648	0.657	0.661	0.659	0.654	0.644	0.627
5%	0.461	0.576	0.586	0.596	0.615	0.643	0.632	0.591	0.589	0.575

Table B.2: Classification Accuracy of Equal Weight Classification Results based on Dissimilarity Measure Eq.(2.7)

Profile size	Session 2 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.262	0.184	0.529	0.526	0.456	0.454	0.452	0.439	0.439	0.434
50%	0.315	0.370	0.521	0.556	0.550	0.538	0.502	0.483	0.484	0.483
33%	0.381	0.459	0.472	0.580	0.514	0.539	0.520	0.539	0.529	0.494
25%	0.370	0.464	0.526	0.494	0.522	0.494	0.469	0.476	0.478	0.478
10%	0.366	0.463	0.531	0.510	0.519	0.497	0.473	0.480	0.479	0.477
5%	0.373	0.461	0.521	0.511	0.518	0.488	0.477	0.481	0.468	0.463

Profile size	Session 3 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.286	0.127	0.496	0.489	0.438	0.439	0.435	0.431	0.431	0.427
50%	0.341	0.407	0.489	0.522	0.511	0.519	0.494	0.471	0.468	0.469
33%	0.449	0.503	0.510	0.584	0.588	0.597	0.602	0.597	0.579	0.542
25%	0.428	0.523	0.601	0.598	0.637	0.643	0.630	0.623	0.621	0.623
10%	0.433	0.527	0.603	0.601	0.639	0.644	0.634	0.628	0.622	0.622
5%	0.430	0.513	0.557	0.579	0.593	0.607	0.600	0.592	0.581	0.557

Profile size	Session 4 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.277	0.100	0.471	0.461	0.412	0.414	0.414	0.407	0.405	0.399
50%	0.339	0.360	0.459	0.497	0.482	0.491	0.467	0.446	0.444	0.444
33%	0.442	0.469	0.469	0.535	0.561	0.565	0.584	0.582	0.565	0.529
25%	0.410	0.480	0.552	0.537	0.610	0.616	0.621	0.629	0.623	0.618
10%	0.420	0.483	0.561	0.544	0.613	0.616	0.624	0.631	0.625	0.620
5%	0.418	0.473	0.523	0.545	0.588	0.592	0.597	0.604	0.589	0.561

Table B.3: Classification Accuracy of Linear Weight Classification Results based on Dissimilarity Measure Eq.(2.5)

Profile size	Session 2 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.374	0.522	0.537	0.545	0.506	0.480	0.455	0.450	0.446	0.444
50%	0.394	0.501	0.580	0.619	0.617	0.616	0.588	0.564	0.562	0.554
33%	0.395	0.566	0.632	0.592	0.652	0.622	0.570	0.533	0.509	0.516
25%	0.400	0.608	0.658	0.610	0.639	0.642	0.626	0.606	0.598	0.590
10%	0.400	0.607	0.659	0.615	0.640	0.644	0.628	0.610	0.600	0.591
5%	0.395	0.588	0.620	0.600	0.631	0.628	0.583	0.554	0.531	0.519

Profile size	Session 3 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.436	0.490	0.510	0.507	0.478	0.469	0.444	0.440	0.435	0.435
50%	0.469	0.576	0.588	0.626	0.589	0.588	0.563	0.536	0.532	0.523
33%	0.455	0.618	0.621	0.648	0.681	0.666	0.652	0.611	0.567	0.542
25%	0.456	0.610	0.658	0.679	0.685	0.680	0.689	0.665	0.659	0.650
10%	0.460	0.613	0.656	0.680	0.684	0.682	0.691	0.667	0.657	0.651
5%	0.454	0.603	0.631	0.649	0.680	0.671	0.653	0.634	0.621	0.602

Profile size	Session 4 N-gram size									
	1	2	3	4	5	6	7	8	9	10
66%	0.407	0.452	0.482	0.484	0.452	0.450	0.429	0.427	0.418	0.416
50%	0.450	0.535	0.542	0.610	0.591	0.580	0.557	0.529	0.527	0.516
33%	0.446	0.593	0.595	0.618	0.642	0.636	0.633	0.608	0.589	0.563
25%	0.442	0.587	0.614	0.625	0.663	0.657	0.668	0.644	0.640	0.627
10%	0.447	0.590	0.619	0.633	0.665	0.656	0.671	0.644	0.641	0.629
5%	0.450	0.582	0.607	0.617	0.644	0.639	0.631	0.623	0.613	0.601

Table B.4: Classification Accuracy of Linear Weight Classification Results based on Dissimilarity Measure Eq.(2.7)

Appendix C

Prediction Accuracy

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.294	0.198	0.416	0.412	0.308	0.301	0.298	0.292	0.292	0.290
50%	0.380	0.388	0.422	0.416	0.425	0.414	0.351	0.342	0.337	0.334
33%	0.373	0.415	0.434	0.359	0.332	0.362	0.374	0.364	0.346	0.324
25%	0.402	0.396	0.439	0.339	0.338	0.311	0.292	0.297	0.326	0.338
10%	0.403	0.400	0.441	0.342	0.340	0.332	0.298	0.330	0.331	0.336
5%	0.378	0.395	0.431	0.338	0.333	0.341	0.339	0.337	0.336	0.329
Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.374	0.191	0.496	0.486	0.434	0.434	0.432	0.426	0.427	0.423
50%	0.444	0.430	0.509	0.511	0.522	0.527	0.501	0.472	0.468	0.463
33%	0.446	0.503	0.521	0.550	0.497	0.506	0.528	0.506	0.490	0.448
25%	0.440	0.490	0.552	0.551	0.542	0.519	0.480	0.492	0.502	0.507
10%	0.442	0.491	0.557	0.550	0.544	0.523	0.494	0.498	0.501	0.503
5%	0.441	0.484	0.508	0.521	0.502	0.501	0.511	0.505	0.494	0.463
Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.350	0.122	0.478	0.465	0.420	0.414	0.416	0.405	0.407	0.401
50%	0.414	0.388	0.480	0.495	0.497	0.501	0.491	0.461	0.456	0.448
33%	0.433	0.478	0.478	0.516	0.520	0.523	0.557	0.546	0.537	0.488
25%	0.418	0.465	0.535	0.527	0.559	0.572	0.569	0.563	0.567	0.576
10%	0.420	0.467	0.537	0.530	0.561	0.571	0.570	0.565	0.566	0.569
5%	0.417	0.475	0.482	0.506	0.523	0.531	0.552	0.548	0.540	0.493

Table C.1: Prediction Accuracy of Equal Weight Prediction Results based on Dissimilarity Measure Eq.(2.5)

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.370	0.420	0.422	0.426	0.374	0.356	0.299	0.300	0.295	0.293
50%	0.416	0.455	0.458	0.474	0.476	0.474	0.454	0.441	0.428	0.373
33%	0.411	0.471	0.497	0.542	0.518	0.458	0.456	0.417	0.353	0.320
25%	0.408	0.466	0.497	0.499	0.531	0.476	0.486	0.478	0.466	0.463
10%	0.410	0.470	0.495	0.503	0.537	0.481	0.488	0.480	0.463	0.462
5%	0.411	0.466	0.493	0.512	0.519	0.464	0.462	0.434	0.421	0.397
Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.449	0.507	0.517	0.505	0.467	0.448	0.439	0.440	0.434	0.431
50%	0.484	0.547	0.573	0.594	0.579	0.576	0.548	0.527	0.522	0.509
33%	0.485	0.585	0.614	0.633	0.622	0.604	0.588	0.548	0.482	0.461
25%	0.475	0.576	0.625	0.630	0.633	0.623	0.630	0.623	0.602	0.604
10%	0.473	0.577	0.628	0.629	0.635	0.627	0.633	0.621	0.601	0.602
5%	0.475	0.580	0.611	0.628	0.624	0.606	0.600	0.565	0.510	0.482
Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.410	0.478	0.497	0.488	0.456	0.437	0.424	0.424	0.414	0.412
50%	0.465	0.533	0.548	0.591	0.572	0.578	0.542	0.529	0.516	0.499
33%	0.476	0.576	0.587	0.606	0.589	0.584	0.593	0.578	0.546	0.523
25%	0.452	0.565	0.621	0.631	0.627	0.627	0.636	0.627	0.606	0.604
10%	0.466	0.570	0.618	0.633	0.630	0.629	0.635	0.630	0.608	0.604
5%	0.463	0.567	0.588	0.608	0.584	0.592	0.589	0.580	0.552	0.527

Table C.2: Prediction Accuracy of Equal Weight Prediction Results based on Dissimilarity Measure Eq.(2.7)

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.265	0.200	0.419	0.410	0.307	0.299	0.297	0.292	0.290	0.289
50%	0.351	0.386	0.422	0.416	0.423	0.416	0.354	0.344	0.337	0.336
33%	0.368	0.418	0.439	0.366	0.336	0.369	0.370	0.372	0.358	0.332
25%	0.398	0.397	0.438	0.338	0.342	0.315	0.294	0.298	0.327	0.340
10%	0.398	0.403	0.441	0.343	0.339	0.327	0.306	0.310	0.332	0.340
5%	0.379	0.399	0.436	0.368	0.341	0.347	0.362	0.372	0.361	0.341

Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.318	0.194	0.501	0.482	0.432	0.430	0.430	0.426	0.424	0.422
50%	0.386	0.424	0.510	0.511	0.518	0.532	0.506	0.474	0.468	0.468
33%	0.435	0.509	0.531	0.564	0.505	0.521	0.522	0.522	0.513	0.464
25%	0.431	0.492	0.551	0.548	0.550	0.527	0.482	0.493	0.505	0.513
10%	0.432	0.501	0.549	0.549	0.549	0.532	0.501	0.497	0.508	0.510
5%	0.433	0.503	0.529	0.558	0.512	0.528	0.529	0.517	0.508	0.483

Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.284	0.124	0.482	0.456	0.418	0.416	0.412	0.405	0.403	0.399
50%	0.343	0.386	0.478	0.495	0.495	0.510	0.493	0.463	0.454	0.456
33%	0.420	0.482	0.495	0.533	0.520	0.540	0.546	0.557	0.550	0.495
25%	0.407	0.463	0.529	0.529	0.569	0.572	0.563	0.559	0.565	0.576
10%	0.409	0.467	0.531	0.530	0.572	0.574	0.564	0.560	0.563	0.562
5%	0.413	0.471	0.508	0.529	0.524	0.553	0.542	0.556	0.547	0.521

Table C.3: Prediction Accuracy of Linear Weight Prediction Results based on Dissimilarity Measure Eq.(2.5)

Profile	Session 2 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.370	0.414	0.416	0.426	0.374	0.355	0.301	0.300	0.296	0.290
50%	0.413	0.468	0.458	0.483	0.475	0.468	0.452	0.441	0.430	0.377
33%	0.402	0.466	0.501	0.546	0.520	0.455	0.450	0.408	0.358	0.330
25%	0.405	0.460	0.494	0.489	0.528	0.472	0.485	0.472	0.468	0.463
10%	0.408	0.469	0.497	0.521	0.527	0.476	0.488	0.471	0.469	0.464
5%	0.409	0.470	0.488	0.516	0.523	0.477	0.473	0.452	0.423	0.419

Profile	Session 3 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.449	0.493	0.505	0.505	0.468	0.447	0.443	0.439	0.435	0.424
50%	0.478	0.572	0.573	0.611	0.577	0.564	0.543	0.527	0.526	0.517
33%	0.468	0.575	0.622	0.642	0.626	0.597	0.577	0.530	0.493	0.481
25%	0.469	0.564	0.618	0.610	0.626	0.614	0.627	0.610	0.607	0.604
10%	0.472	0.570	0.617	0.622	0.629	0.608	0.628	0.613	0.608	0.604
5%	0.464	0.573	0.621	0.638	0.628	0.604	0.598	0.570	0.541	0.523

Profile	Session 4 N-gram size									
size	1	2	3	4	5	6	7	8	9	10
66%	0.410	0.459	0.482	0.488	0.456	0.435	0.431	0.422	0.418	0.410
50%	0.456	0.561	0.557	0.606	0.580	0.572	0.542	0.525	0.522	0.512
33%	0.450	0.574	0.589	0.610	0.610	0.599	0.586	0.574	0.561	0.544
25%	0.446	0.548	0.599	0.604	0.614	0.604	0.618	0.601	0.599	0.599
10%	0.453	0.543	0.602	0.607	0.618	0.608	0.622	0.606	0.603	0.600
5%	0.448	0.565	0.593	0.609	0.613	0.598	0.593	0.587	0.577	0.556

Table C.4: Prediction Accuracy of Linear Weight Prediction Results based on Dissimilarity Measure Eq.(2.7)