Available online at www.sciencedirect.com
ScienceDirect
Data & Knowledge Engineering 61 (2007) 304–330
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests

Haibin Liu, Vlado Kešelj *

Faculty of Computer Science, Dalhousie University, 6050 University Ave, Halifax, NS, Canada B3H 1W5

Received 19 August 2005; received in revised form 3 May 2006; accepted 10 June 2006
Available online 7 July 2006

Abstract

We present a study of the automatic classification of web user navigation patterns and propose a novel approach to classifying user navigation patterns and predicting users' future requests. The approach is based on the combined mining of Web server logs and the contents of the retrieved web pages. The textual content of web pages is captured through extraction of character *N*-grams, which are combined with Web server log files to derive user navigation profiles. The approach is implemented as an experimental system, and its performance is evaluated based on two tasks: classification and prediction. The system achieves the classification accuracy of nearly 70% and the prediction accuracy of about 65%, which is about 20% higher than the classification accuracy by mining Web server logs alone. This approach may be used to facilitate better web personalization and website organization.

Keywords: Web usage mining; Web content mining; User navigation profiles; Classification; Prediction

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 useful tool to solve these problems. The basis of our approach is extraction of navigation profiles that capture similar behaviour of site users. 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.

* Corresponding author. Tel.: +1 902 494 2893; fax: +1 902 492 1517.
E-mail address: vlado@cs.dal.ca (V. Kešelj).

0169-023X/$ - see front matter
doi:10.1016/j.datak.2006.06.001

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. We estimate the best profile describing navigation behaviour of the current user, and find related unrequested web pages of 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 one view of the web site content, a site 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.

Recently, 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].

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. 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, but the results were not very good with lower classification accuracy. One reason for such results discussed by the authors was they did not exploit content mining techniques. However, the contents of accessed pages may reveal topics related to visitors' profiles, which can improve the classification accuracy. In fact, it is often observable that web pages a visitor requested during his navigation are correlated and can be bound to a specific content. Thus, it is more likely that this type of content could be accessed in the further extension of this navigation. We therefore hypothesize that looking into web page contents will better capture site users' interests and increase the classification accuracy 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 paper, we used character N-grams to represent the contents of web pages, and combined them with user navigation patterns by building user navigation profiles composed of a collection of N-grams. This character N-gram-based approach is a new way to integrate web usage mining and web content mining. Meanwhile, we tried different ways when building user profiles and attempted different parameters when experimenting on classification and prediction. Furthermore, we found that some parameters could influence the classification and prediction results. The existence of the optimal parameters reveals a clue on how to build desired user profiles and also becomes a guide for further experiments. We evaluated the performance of the experimental system using two defined measures: *classification accuracy* and *prediction accuracy*. We concluded that our system can achieve good experimental results although it is only a proof-of-concept prototype of the idea of combining both web usage and content mining.

The rest of this paper is organized as follows: In Section 2, we review recent research advances in both web usage and content mining. Section 3 describes the architecture of the experimental system proposed for classifying user navigation patterns and predicting users' future requests. The approaches and algorithms applied in the system are also explicated. The overall performance of the system is evaluated in Section 4. Finally, Section 5 summarizes the paper and introduces future work.

2. Related work

Web mining is the mining of data related to the World Wide Web. It is categorized into three active research areas according to what part of web data is mined [8]: Content mining, which is the process of extracting knowledge from the content of websites, for example, contents of documents or their descriptions; Structure mining, which 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, which studies user access information from logged server data in order to extract interesting usage patterns.

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

Moreover, based on textual contents of recently requested web pages, Davison [9] proposed an approach to 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, intentions of visitors might change during their browsing, so 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, and also limits contents of the predicted pages to recently requested pages.

In order to overcome the inherent limitations of both content and usage mining, combining these two areas is a natural way of solving 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 [1,10–12]. They generally consist of two major processes: off-line mining and on-line recommendation. 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 the on-line recommendation process, user's requests from his current active session are recorded. By comparing these requests with the models obtained from the off-line mining, appropriate personalized recommendations are generated. Our experiments in this paper mainly center around the off-line mining process. We first combine usage and content mining to process web logs for building user navigation profiles, and then use these profiles to classify the site users. Next, we simulate the requests from active sessions to construct current navigation profiles of users. By matching the current profile with the navigation profiles built in the off-line process, we are able to predict users' future requests. Corresponding recommendations can be generated by web recommendation systems during the on-line recommendation process.

According to the definitions given in [13], there are at least two ways to integrate content features of web pages into usage mining results: pre-mining integration, which involves the transformation of normal user access sessions into "content-enhanced" sessions containing the semantic features of the web page contents, and post-mining integration, which denotes performing usage mining and content mining independently and then combining their mining results. Mobasher et al. [14] made an attempt to integrate both usage and content attributes of a site into a web mining framework for web personalization. A "post-mining" type approach was implemented to obtain the uniform representation for both site usage and site content profiles to facilitate the real-time personalization. However, the techniques proposed in [14] were limited to the use of clustering to separately build site usage and content profiles.

An experimental system was designed by Guo et al. [15] to investigate combined mining of both web logs and web contents. The methods of document clustering are first applied to the website contents for grouping

web pages into a certain number of clusters. Then, the representative information of page clusters is integrated into original web log entries as the content indicator. Finally, the association rule mining algorithm is applied to the "content-enhanced" data source. The system demonstrates that novel association rules can be discovered from the integrated log data, while they could not be mined from original log entries. Furthermore, Li et al. [16] proposed a web recommendation system which combines not only usage data, content data, but structure data in a web site to generate user navigational models. These models are then fed back into the system to recommend users shortcuts or page resources. Similar to the method in Guo et al. [15], normal user access sessions are first transferred into so-called "missions", namely "content-enhanced" sessions. The structure data is later used to provide connectivity information among web pages to improve the navigational patterns obtained by performing web usage mining on derived "missions". Preliminary experiments in the paper prove that the system can provide recommendations of good quality.

In this paper, we focus on exploring the application of some of the newer successful techniques in text clustering and classification [17–19] to web usage and content mining to build integrated *N*-gram-based profiles for both more effective classification of user navigation patterns and prediction of users' future requests. The approach that we propose in the paper to integrate usage and content attributes belongs to *post-mining integration*.

3. System design

In this work, 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. Fig. 1 illustrates the overall data flow of the system, which consists of five major modules: web-log preprocessing, web usage mining, navigation pattern profiling, classification and prediction, and system performance evaluation.

In this scheme, we start with the primary web-log preprocessing to extract user navigation sessions from dataset. From these, we 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 *N*-gram-based user 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 subsections, we will describe the algorithms and implementations for each component of the system in detail.

3.1. Web-log preprocessing

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, such as: client IP address, request time, requested URL, HTTP status code, referrer, etc.

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. [20]. The original server logs are cleansed, formatted, and then grouped into meaningful sessions before being utilized by web usage mining.

3.1.1. 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, which includes:

- Requests executed by automated programs [21], such as web robots, spiders and crawlers; The traffic to websites these programs generate can dramatically bias site statistics [22] and are also not the desired category which web usage mining investigates.

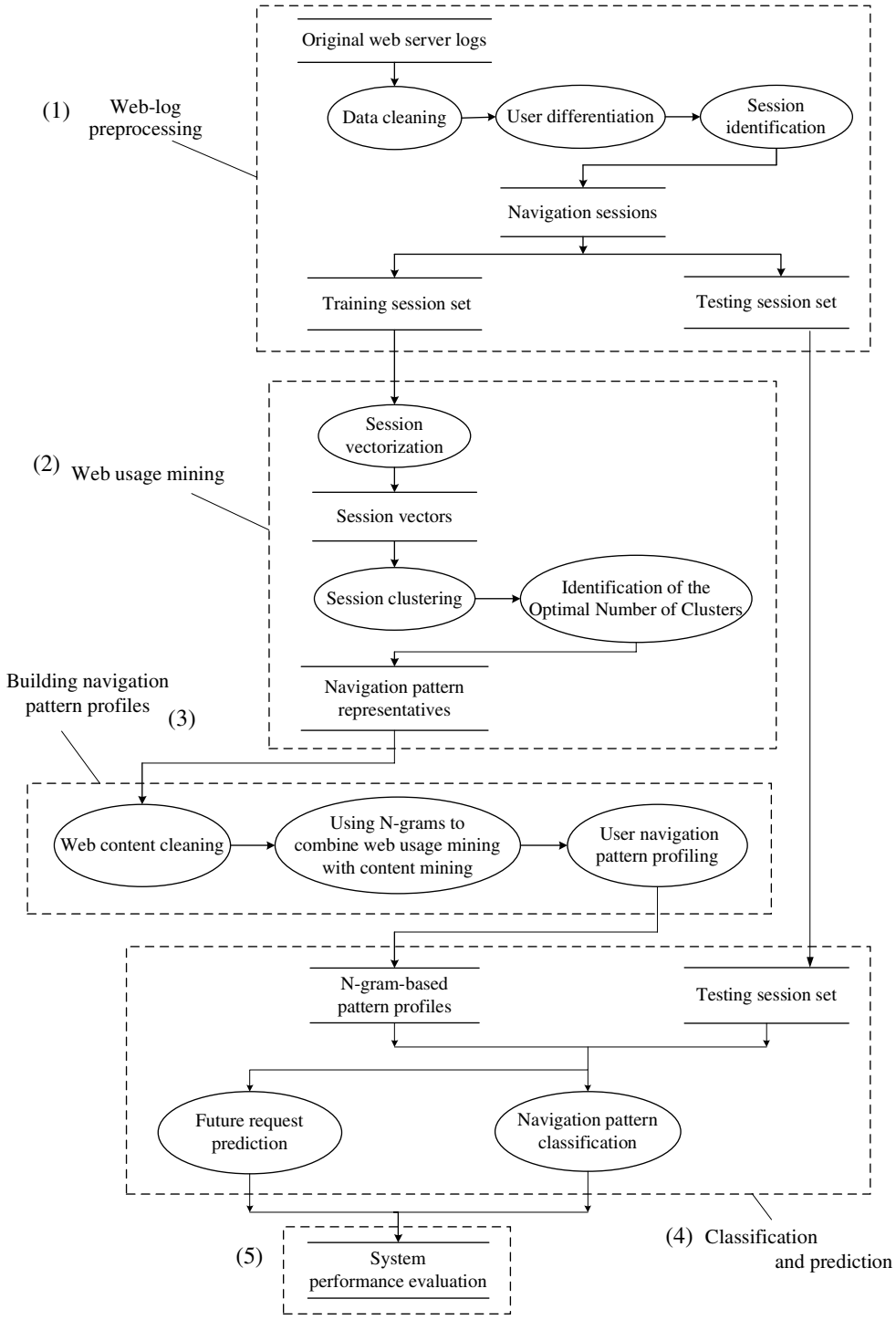


Fig. 1. System dataflow diagram.

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

- Entries with unsuccessful HTTP status codes; HTTP status codes are used to indicate the success or failure of a requested event, and we only consider successful entries with codes between 200 and 299.
- Log entries with request methods except "Get" and "Post".

3.1.2. 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 [23]. 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 [23]. 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 [23]. However, the use of cookie needs to comply with existing laws [24], at least that users must be clearly made aware of its presence. Therefore, the proactive strategy is not always a feasible option.

In our system, 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.

3.1.3. Session identification

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 individual access sessions [20]. Two time-oriented heuristic methods, *session-duration-based* method and *page-stay-time-based* method, have been specifically proposed by Mobasher et al. [20,23,25] for session identification.

The *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-min threshold for total session duration has been recommended [25,23]. For 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 min, has been proposed to capture the time for loading and studying the contents of a page [23,25]. In our system, we apply these two heuristic methods to the task of identifying sessions, and will explore which method leads to better experimental results in the later system modules.

The identified sessions are further split up into two sets based on the date of log entries: training set and testing set. The training set is used to build user profiles in the web usage mining, while the testing set is prepared for the experiments of classification and prediction, which are objectives of our system.

3.2. Web usage mining

After the preprocessing step, we then perform web usage mining on the derived user access sessions. As an important operation of web usage mining, clustering 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. These patterns will be further used to facilitate the user profiling process of our system. In this system module, we will introduce how we perform the session clustering and how we identify the optimal number of clusters from clustering results.

3.2.1. Session vectorization

Let P be a set of web pages accessed by users in Web server logs, $P = \{p_1, p_2, \ldots, p_m\}$, each of which is uniquely represented by its associated URL. Let S be a set of user access sessions. Hence, $S = \{s_1, s_2, \ldots, s_n\}$, 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, $s = \{w(p_1,s), w(p_2,s), \ldots, w(p_m,s)\}$, where $w(p_i,s)$ is a weight assigned to the ith web page ($1 \leqslant i \leqslant m$) visited in a session s. Note that it is allowed that a web page $p_i \in P$ repeats in each user access session $s_i \in S$. We assume that all the accessed web pages are equally important to user navigation pattern profiles. Therefore, 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. Inspired by Chan and coworkers [26,27], 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 Eq. (1), which is normalized by the total number of visits of web pages in the session:

$$Frequency(Page) = \frac{NumberOfVisits(Page)}{\sum_{Page \in VisitedPages}(NumberOfVisits(Page))}. \tag{1}$$

"*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 [28]. 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, that is, the total bytes of the page. We use Eq. (2) to measure the "*Duration*" of a web page,

$$Duration(Page) = \frac{TotalDuration(Page)/Length(Page)}{\max_{Page \in VisitedPages}(TotalDuration(Page)/Length(Page))}, \tag{2}$$

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 our system, "*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 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:

$$Interest(Page) = \frac{2 \times Frequency(Page) \times Duration(Page)}{Frequency(Page) + Duration(Page)}. \tag{3}$$

Eq. (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., $s = \{w_1, w_2, \ldots, w_m\}$, 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 here use 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 $f_{\min} = 10$.

3.2.2. 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. The well-known K-means algorithm is applied as the base method to cluster

vectorized sessions. WEKA 3.4 machine learning toolkit [29] is used to perform the *K*-means algorithm. In addition, the most popular *Euclidean distance* is adopted as the distance measure.

Session clustering results in a set of clusters, $C = \{c_1, c_2, \ldots, c_k\}$, in which each c_i $(1 \leqslant i \leqslant k)$ 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. Meanwhile, 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. For our system, since the least mean value is always far smaller than the second least and the third least mean values, we choose the second least mean value of each mean vector as the $w_{\min}$ of each session 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* in this work is defined as follows:

Definition 3.1. 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, $NP = \{np_1, np_2, \ldots, np_k\}$ 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.

3.2.3. 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. In 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 constructed next. 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 [30].

Our approach is to apply the *K*-means method to the access sessions using k values ranging from 2 to n $(n > 2)$. For each k value, we first evaluate the quality of the clustering result using *internal* quality measures which are always the quantitative evaluation and do not rely on any external knowledge. Here, we use the internal evaluation functions proposed in [30], *cluster compactness* (*Cmp*), *cluster separation* (*Sep*) and combined measure *overall cluster quality* (*Ocq*), to evaluate both the intra-cluster homogeneity and the inter-cluster separation of the clustering result. The definitions of these functions are given below:

$$Cmp = \frac{1}{C}\sum_{i}^{C}\frac{v(c_i)}{v(X)}, \tag{4}$$

where C is the number of clusters generated on the data set X, $v(c_i)$ is the deviation of the cluster c_i, and $v(X)$ is the deviation of the data set X:

$$v(X) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d^2(x_i, \bar{x})}, \tag{5}$$

where $d(x_i, x_j)$, 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 $\bar{x}$ is the mean of X. The smaller the *Cmp* value, the higher the average compactness in the output clusters:

$$Sep = \frac{1}{C(C-1)} \sum_{i=1}^{C} \sum_{j=1, j\neq i}^{C} \exp\left(-\frac{d^2(x_{c_i}, x_{c_j})}{2\sigma^2}\right) \tag{6}$$

where σ is the standard deviation of the data set X, C is the number of clusters, exp() is the function that returns an exponential value, i.e., $\exp(x) = e^x$, x_{c_i} is the centroid of the cluster c_i, $d()$ 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:

$$Ocq(\beta) = \beta \cdot Cmp + (1-\beta) \cdot Sep, \tag{7}$$

where $\beta \in [0,1]$ is the weight that balances the measures *Cmp* and *Sep*. $Ocq(0.5)$ is often used to give equal weights to the two measures for overcoming the deficiency of each measure and assess the overall performance of a clustering system. Therefore, the lower the *Ocq* value, the better quality of the overall output clusters.

In this way, we are able to observe the varying pattern of the score values according to the change of input number k. Naturally, the most satisfactory quality score indicates the best partition of access sessions, while the corresponding k suggests the optimal number of clusters on the access sessions. As a supplement, the quality of the clustering results are also evaluated using external quality measures, which rely on some external knowledge such as the inspection and interpretation in terms of our domain knowledge, to assist in identifying the optimal number of clusters.

3.3. Building navigation pattern profiles

This system module attempts to integrate the representative user navigation patterns obtained from the clustering operation with contents of the corresponding web pages to construct user navigation profiles. We first introduce the preprocessing operation on web contents. Then, we propose a new 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.

3.3.1. Web content cleaning

As mentioned earlier, $P = \{p_1, p_2, \ldots, p_m\}$ 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 [31]. 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 $PC = \{pc_1, pc_2, \ldots, pc_m\}$ to represent the set of web pages after cleaning.

3.3.2. Using N-grams to combine web usage mining with content mining

A character N-gram is an N-character substring of a longer string [17]. 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 Table 1.

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 [17]. 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 character N-gram representation was also used in the Authorship attribution tasks [18]. An optimal

Table 1
Example of character *N*-grams

	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_

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. In addition, the classification and hierarchical clustering of biological genome sequences were also performed based on the *N*-gram representation of genome sequences [19]. The character *N*-gram representation has three distinct advantages: it provides a robust representation and easy to control numerosity; it is language and topic independent and requires no special preparations; it can be applied to different file formats.

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* [32], are computed to be associated with the *N*-grams.

Defined in Eq. (8), term frequency $tf(x_i,j)$ is the normalized frequency of *N*-gram x_i in the pattern $j \in NP$:

$$tf_{x_i,j} = \frac{freq_{x_i,j}}{\sum_{x_l \in j} freq_{x_l,j}}, \tag{8}$$

$freq_{x_i,j}$ 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. (9) defines document frequency $df_{x_i,j}$, which is the number of web pages that *N*-gram x_i occurs in the pattern $j \in NP$:

$$df_{x_i,j} = \frac{n_{x_i}}{N_j}. \tag{9}$$

The total number of web pages in the pattern j, N_j, is used for normalization which 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 $\{(x_1, tf_{x_1}, df_{x_1}), (x_2, tf_{x_2}, df_{x_2}), \ldots, (x_n, tf_{x_n}, df_{x_n})\}$. The algorithm for transforming each user navigation pattern to its collection of *N*-gram triples is given in Algorithm 1.

According to the definition given in [13], our approach of combining web usage mining with content mining can be categorized into *post-mining integration*.

Algorithm 1. *N*-gram Triples

Input: $j \in NP$ // Set of web pages of the navigation pattern j
Output: *N*-gram triples for j
  $V = \Phi$ // 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
    $V \leftarrow V \cup N - \text{grams}(p)$
    // *N*-grams() is the function to produce *N*-gram tables of each web page [33]
  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 ∈ 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
```

3.3.3. 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 Eq. (9), 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 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.

3.4. 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's 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 [18], 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 $\{(x_1, tf_{x_1}), \ldots, (x_n, tf_{x_n})\}$, 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*. As 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's 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 3.2. 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 3.3. 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 $n - j$ ($j \geqslant 1$) 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 2 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 2. 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. 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.

In our system, we respectively apply the three dissimilarity measures below, d_1, d_2, and d_3, to the calculation of profile dissimilarity:

$$d_1(tf_p, tf_{P_i}) = \sum_{x_i \in profile} \left(\frac{2 \times (tf_p(x_i) - tf_{P_i}(x_i))}{tf_p(x_i) + tf_{P_i}(x_i)} \right)^2, \tag{10}$$

$$d_2(tf_p, tf_{P_i}) = \sum_{x_i \in profile} \frac{2 \times (tf_p(x_i) - tf_{P_i}(x_i))^2}{(tf_p(x_i) + tf_{P_i}(x_i))}, \tag{11}$$

$$d_3(tf_p, tf_{P_i}) = \sum_{x_i \in profile} \frac{|tf_p(x_i) - tf_{P_i}(x_i)|}{\sqrt{tf_p(x_i) \times tf_{P_i}(x_i)} + 1}. \tag{12}$$

These dissimilarity measures have been applied to the experiments in literature and achieved success in solving different problems [18,34,19]. In fact, these measures are only different in the normalization schema. d_1 and d_2 use average (arithmetic mean value $= (tf_p(x_i) + tf_{P_i}(x_i))/2$) frequency for a given N-gram as the normalization scheme, while d_3 is normalized by geometric mean value. We will evaluate these measures in our system according to their performance on the experiments of classification and prediction.

3.5. 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 $A(C)$ and prediction accuracy $A(P)$. 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, web pages in the navigation pattern that have not been accessed have great potential to be the next pages that the user wants to see, and we can rely on the profile of a real active user session to predict the user's future requests. Specifically, we can recommend unrequested web pages in the navigation pattern 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. This work was done by three independent people, and an agreement was reached before pre-labeling the sessions, that is, the navigation pattern assigned to each testing session must be in accordance with the whole intention of the session. We studied the contents of the web pages visited in each session, judged the principal idea of the session, and then concluded the navigation pattern assigned to the session.

4. Results and evaluation

In this section, we start with the description of the experimental dataset in Section 4.1. Web-log preprocessing results are then presented followed by usage mining results in which the optimal number of clusters, 16, is identified in Section 4.3. Next, we report how *document frequency* contributes to the construction of

N-gram-based user profiles by exploring the number of *N*-grams in user profiles in function of the *document frequency* threshold. After performing a thorough evaluation in Section 4.6 on classification and prediction tasks, it is found that the geometric-mean-based dissimilarity measure achieves the best performance while the *equal weight* method of building profiles outperforms the *linear weight* method in both classification and prediction. At the end of Section 4.6, in order to verify the proposed hypothesis of the paper, we conducted another experiment to classify user navigation patterns without considering web contents. The comparison between results indicates that the inclusion of a content mining approach improves the classification performance.

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. The typical users of the graduate Web server are professors and graduate students, who mostly have their own computers and do not share IP addresses. 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 226 MB log file, and 1,370,373 access entries in October producing a 245 MB 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 and Java.

4.2. Web-log preprocessing results

Table 2 presents some statistics of the preprocessed experimental dataset, including both training and testing sets.

For the training dataset, 116,166 clean entries are extracted, and there are approximate 12,931 different users who accessed the Web server in September 2004. In this period, 792 web pages were visited and 616 of them were accessed at least 10 times. Although totally 23,791 sessions were identified by the session-duration-based method from the training set, 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 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

Table 2
Statistics of experimental dataset

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 ($\geqslant$10 times)	616	623
Identified sessions (total, session duration)	23,791	23,242
Identified sessions ($\geqslant$2 requests, session duration)	12,402	10,204
Identified sessions (total, page-stay time)	24,756	24,303
Identified sessions ($\geqslant$2 requests, page-stay time)	12,675	10,546

are chosen as the training set for web usage mining, while only these sessions in the testing set are prepared as the testing set for the experiments of classification and prediction.

4.3. Web usage mining results

We used a frequency threshold $f_{\min} = 10$ 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 Table 2, 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.

Since we tried two methods of identifying sessions, the *K*-means clustering algorithm was respectively performed on two kinds of sessions obtained by the session-duration-based method and the page-stay-time-based method. For each kind of sessions, 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.

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. We gave a comparison between the clustering results of two session identification methods by comparing the combination measure *Ocq* of two methods with varying k from 2 to 20. Shown in Fig. 2, it is obvious that when k ranges between 8 and 19, the session-duration-based method uniformly achieves better clustering quality than the page-stay-time-based method. Since the *Ocq* obtained the lowest value at $k = 16$ in terms of the session-duration-based method, 16, therefore, became the optimal number of clusters for 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. We computed a mean vector for each cluster and extracted the corresponding web pages accessed in each cluster based on the weight values of its mean vector. We also 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 3 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.

It is seen in the table that the ninth 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 ninth cluster is ''miscellanies'', which

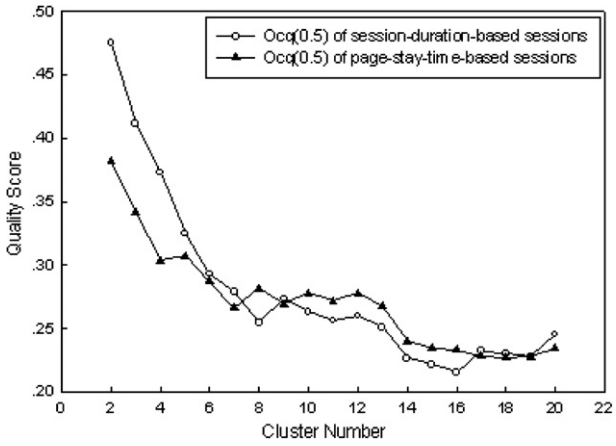


Fig. 2. *Ocq*(0.5) comparison between clustering results of two methods.

Table 3
Description on each cluster

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	Research projects related
16	5	23	CS pictures

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 eighth cluster contains only 13 web pages, it possesses of the second largest proportion of training sessions. The topic of this cluster is "Webcams related". 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, 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 [33] was used to produce N-gram tables of each web page. Experiments [18] 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.

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 lists the obtained mean values.

Table 4
Mean values of *N*-gram numbers of profiles

Profile size (%)	*N*-gram size									
	1	2	3	4	5	6	7	8	9	10
All	27	495	3291	9606	17,334	24,296	30,218	35,144	38,985	42,128
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

According to the data in Table 4, Fig. 3 was drawn to illustrate the distribution of *N*-gram numbers in user profiles of each *df* value. Fig. 3(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, Fig. 3(b) is also provided to show the curves of only six *df* values.

It is obvious in Fig. 3 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

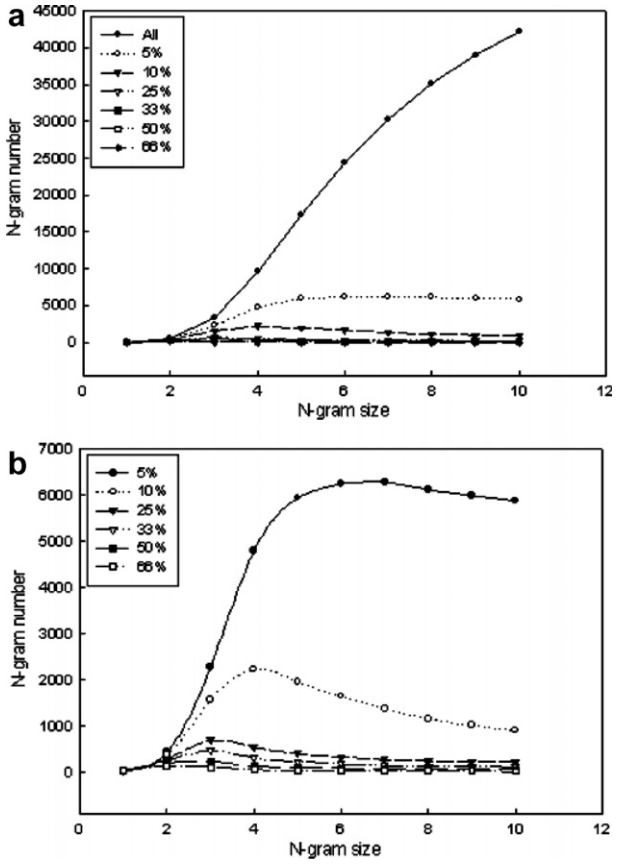


Fig. 3. Distribution of *N*-gram numbers in user profiles of (a) "All" and six *df* values and (b) only six *df* values.

capture users' interest. Contrarily, 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 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. By the observation on the sessions, we notice that a large proportion of them contains exactly 2 access requests. In our prediction experiments, therefore, we set $j = 1$, i.e., the profiles are built based on the first $n - 1$ web pages of sessions. We built session profiles by two methods: *equal weight* and *linear weight*. For each method, we respectively performed three dissimilarity measures 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 1500 sessions from it as the sample set for our system evaluation. We manually pre-labeled the 1500 sessions with 16 resulting clusters. The distribution of cluster labels of the sample sessions is shown in Table 5.

These 1500 sample sessions stand for the session set "Session 2". We further extracted the sets "Session 3" and "Session 4" from "Session 2", respectively including 759 and 469 testing sessions. It is seen in Table 5 that the eighth 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 Table 6. 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 1500 sample sessions.

Table 5
Label distribution of sample sessions

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 6
Baselines of "Session 2", "Session 3" and "Session 4"

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

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 d_1 are all generally lower than the $A(C)$ of the results based on the d_2 and the d_3. In addition, the classification results based on the d_2 achieve the comparable $A(C)$ with the results based on the d_3. However, the highest $A(C)$ is only reached by the classification results based on the d_3. Table 7 lists the classification accuracies for *equal weight* method based on the d_3. It 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 Table 6. 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 Table 6, 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's 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 more and more access requests are made. 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" and "Session 2", and the $A(C)$ of "Session 2" are only little lower than the $A(C)$ of "Session 3". Since "Session 2" stands for the

Table 7
Classification accuracy of equal weight classification results based on dissimilarity measure d_3

Profile size (%)	Session 2 *N*-gram 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
	Session 3 *N*-gram size									
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
	Session 4 *N*-gram size									
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

1500 sample sessions, the classification accuracies in "Session 2" reflect the overall classification accuracies of the 10,204 testing sessions. It is seen in Table 7 that the classification accuracies 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. Fig. 4 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

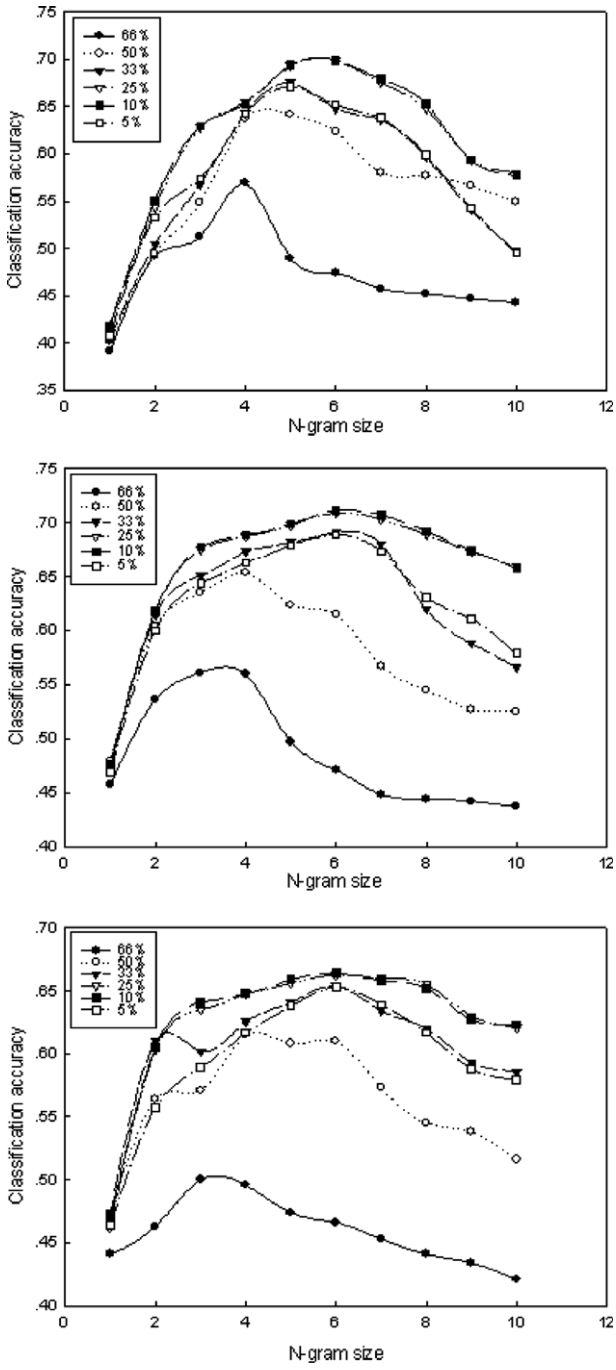


Fig. 4. Distribution of classification accuracy $A(C)$ of "Session 2", "Session 3" and "Session 4" from the top down.

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 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. As same as the accuracies of the *equal weight* method, the 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 and 7.

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

- The geometric-mean-based dissimilarity measure, d_3, 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 d_1 are all generally lower than the $A(P)$ of the results based on the d_2 and the d_3. Furthermore, the prediction results based on the d_2 achieve the comparable $A(P)$ with the prediction results based on the d_3. However, the highest $A(P)$ is only achieved by the prediction results based on the d_3. Table 8 lists the prediction accuracies for *equal weight* method based on the d_3. It 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.

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 Table 6. 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. 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 Table 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 much 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. Fig. 5 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

Table 8
Prediction accuracy of equal weight prediction results based on dissimilarity measure d_3

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
	Session 3 *N*-gram size									
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
	Session 4 *N*-gram size									
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

are all reached for the *N*-gram sizes 4 and 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 size 4.

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

- The geometric-mean-based dissimilarity measure, d_3, 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 or 5, and the profile size is $df = 10\%$.

4.6.3. Performance comparison experiment

In order to verify the proposed hypothesis that looking into web page contents will better capture users' interests and improve the classification accuracy of user navigation patterns, we further conducted an experiment on the same dataset to classify user navigation patterns based only on regular web usage mining for comparison, i.e., without the inclusion of content features of web pages. We focused on exploring the application of different supervised machine learning algorithms to effectively classify the 1500 pre-labeled sample testing sessions, "Session 2", into the 16 obtained representative navigation patterns without taking web page contents into account. For this classification experiment, the sample sessions were first vectorized following

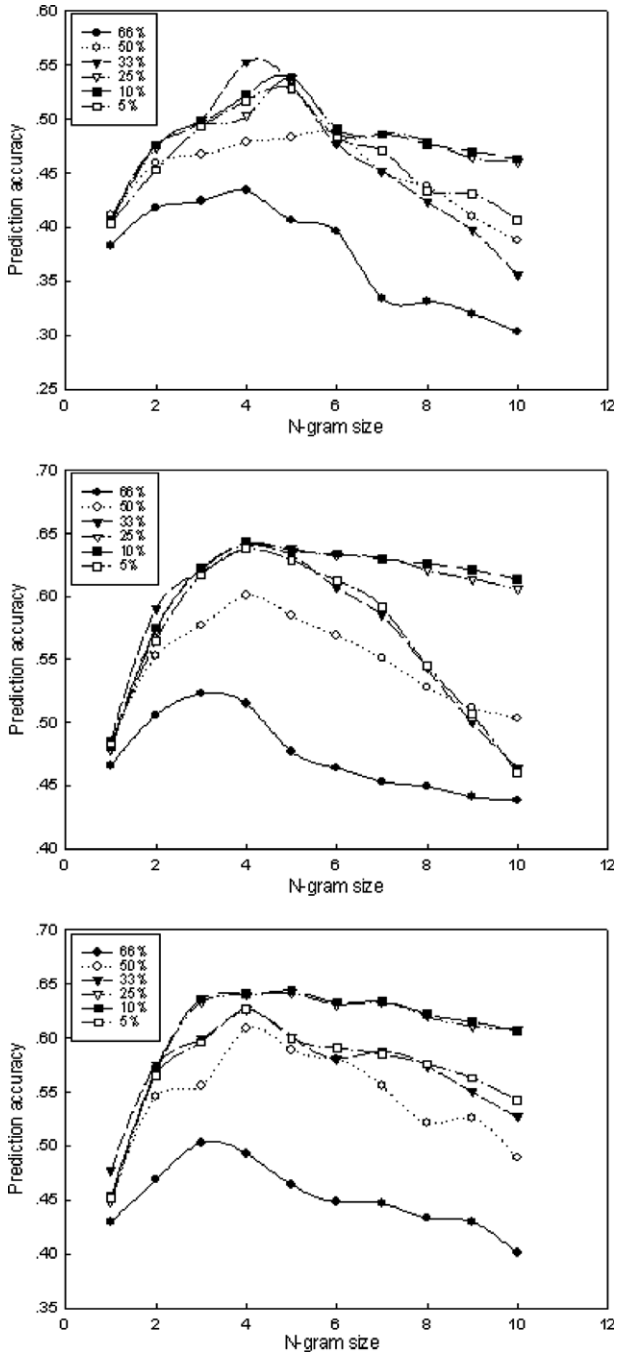


Fig. 5. Distribution of prediction accuracy $A(P)$ of "Session 2", "Session 3" and "Session 4" from the top down.

the procedure introduced in Section 3.2.1, then further divided into training and testing sets in the proportion of 2:1. We experimented on the sample sessions with six commonly used classification algorithms, which are Naive Bayes classifier, C4.5 decision tree classifier, k Nearest neighbours (kNN) classification method, AdaBoost M1 classifier, Support vector machines and Rule Part classifier. We conducted a large number of experiments by tuning the parameters of each algorithm in order to achieve the optimal experimental results. Three widely used standards, Precision, Recall and F-score [32] are adopted to evaluate the performance of

algorithms while the classification accuracies $A(C)$ are also reported. The classification results on training and testing sets are presented in Table 9.

It is shown that C4.5, kNN and SVM uniformly outperform the other three algorithms with classification accuracies over 50%. It is also observable that kNN achieves the highest F-score and $A(C)$ on the training set among six algorithms. However, the performance of kNN on the testing set suggests higher overfitting than the other algorithms. Therefore, the classification performance of C4.5 and SVM classifiers are considered more reliable, and C4.5 even outperforms SVM on both training and testing session sets.

In general, $A(C)$ on the training set are about 18% lower than the accuracies of "Session 2" reported in Table 7, for which user navigation patterns were classified based on the combined mining of Web server logs and web page contents. The classification model extracted from the training set can be evaluated on the testing set for which classification accuracies are treated as prediction accuracies $A(P)$ of the model. Although the experiment was performed based on the total n accessed web pages of each session, the $A(P)$ are generally lower than the $A(P)$ reported in Table 8, for which only the contents of first $n-1$ accessed web pages in each session are considered in the prediction experiment. Fig. 6 describes the performance comparison between the approaches of combined mining and the usuage mining alone in terms of the classification accuracy on the "Session 2" set. In the figure, CA represents the highest classification accuracy by the combined mining

Table 9
Classification performance of different algorithms

Classification algorithms	Training set			
	Precision	Recall	F-score	$A(C)$
Naive Bayes	0.658	0.395	0.494	0.428
C4.5	0.729	0.527	0.612	0.546
AdaBoost M1	0.583	0.479	0.526	0.441
SVM	0.672	0.494	0.570	0.527
KNN	0.753	0.524	0.618	0.566
Rule Part	0.571	0.442	0.498	0.414
	Testing set			
Naive Bayes	0.565	0.378	0.453	0.397
C4.5	0.674	0.531	0.594	0.529
AdaBoost M1	0.556	0.437	0.489	0.413
SVM	0.669	0.471	0.553	0.521
KNN	0.617	0.546	0.579	0.516
Rule Part	0.549	0.462	0.502	0.396

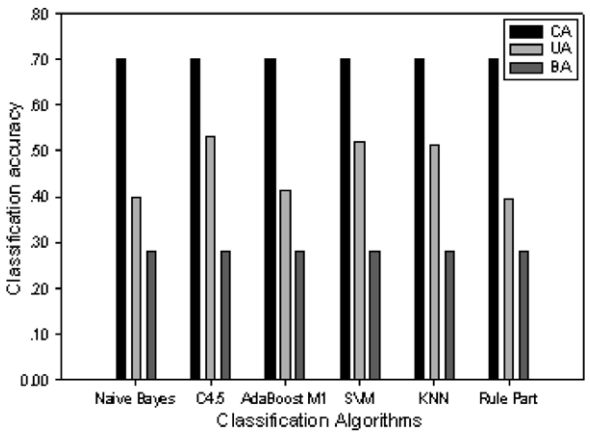


Fig. 6. Performance comparison of two mining approaches.

approach; *UA* stands for the classification accuracy of six algorithms on the testing set based on the usage mining only; *BA* denotes the baseline of the classification accuracy for "Session 2" defined in Table 6. Furthermore, we also conducted the same experiment on both "Session 3" and "Session 4" sets respectively. The classification accuracies of two sets are all nearly 20% lower than the corresponding results presented in Table 7.

The procedure of this experiment is equivalent to the sex prediction experiment of Baglioni et al. [7], and the results of both experiments are also comparable. However, for the experiment of predicting web site users' sex in [7], the registration information of users was required to assist in building classification models, while in this experiment all the information for building models to classify user navigation patterns was inferred from the Web server log itself. Meanwhile, although the classification attributes are all defined as the web pages visited by a session for both experiments, the value of attributes is defined differently. The attribute value was defined as the visit existence or the visit frequency of a web page in [7], while we assigned a weight to each web page to approximate its interest degree to a user, which considers more influencing factors. Furthermore, compared to the two-class type classification of the sex prediction experiment in [7], there are 16 classes in our experiment, which make the classification model more sophisticated. Since the results of both experiments are not very good with classification accuracies all under 57%, we conjecture that accurately capturing users' navigation behaviour is difficult by anaylzing Web server logs alone. As the approach of looking into web page contents evidently outperforms the regular web usage mining on user navigation pattern classification in our experiments, although Baglioni et al. and we experimented on different types of Web server logs, we infer that the inclusion of a content mining approach will improve the classification performance of their experiments as well.

5. Conclusion and future work

In this paper, a novel approach is presented to classify user navigation patterns and predict users' future requests by combined mining of Web server logs and web contents. We have conducted the experiments on our designed experimental system. The dataset used in the system is the log access file of our departmental Web server for a two-month period.

By evaluating 1500 sample testing sessions, we conjecture that our system achieves the classification accuracy of nearly 70% and the prediction accuracy of about 65%. It improves the classification accuracy of performing regular usage mining alone by 20%. Noticeably, three important findings have been also achieved. 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. The existence of the optimal parameters reveals a clue on how to build desired user navigation profiles, and also becomes a guide for the further experiments.

The system we designed is a proof-of-concept prototype of the idea of combining both web usage and content mining, and there are some aspects in which it can be improved in our future work:

- In 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 implicit factors [27], for example, the sequence of accessed web pages may also indicate users' interests and preferences. We are interested in incorporating more influencing factors into the weight measure of the session vectorization.
- 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 and calculate the degree of real users' satisfaction on the generated recommendations, which are derived from the predicted requests, by our system.

References

[1] B. Zhou, S.C. Hui, K. Chang, An intelligent recommender system using sequential web access patterns, in: Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 2004, pp. 1–3.

[2] W. Lin, S.A. Alvarez, C. Ruiz, Collaborative recommendation via adaptive association rule mining, in: WEBKDD2000 – Web Mining for E-Commerce – Challenges and Opportunities, Second International Workshop, Boston, MA, USA, 2000.

[3] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Effective personalization based on association rule discovery from web usage data, in: Proceedings of the 3rd International Workshop on Web Information and Data Management, ACM Press, Atlanta, GA, USA, 2001, pp. 9–15.

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

[5] D.S. Phatak, R. Mulvaney, Clustering for personalized mobile web usage, in: Proceedings of the IEEE FUZZ'02, Hawaii, USA, 2002, pp. 705–710.

[6] D. Shen, Y. Cong, J.-T. Sun, Y.-C. Lu, Studies on Chinese web page classification, in: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, vol. 1, 2003, pp. 23–27.

[7] M. Baglioni, U. Ferrara, A. Romei, S. Ruggieri, F. Turini, Preprocessing and mining web log data for web personalization, in: AI*IA, 2003, pp. 237–249.

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

[9] B.D. Davison, Predicting web actions from html content, in: Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (HT'02), College Park, MD, 2002, pp. 159–168.

[10] R. Burke, Hybrid recommender systems: survey and experiments, User Modeling and User-Adapted Interaction 12 (4) (2002) 331–370.

[11] C. Shahabi, F.B. Kashani, Y.-S. Chen, D. McLeod, Yoda: An accurate and scalable web-based recommendation system, in: Proceedings of the 9th International Conference on Cooperative Information Systems, Springer-Verlag, 2001, pp. 418–432.

[12] H. Ishikawa, T. Nakajima, T. Mizuhara, S. Yokoyama, J. Nakayama, M. Ohta, K. Katayama, An intelligent web recommendation system: A web usage mining approach, in: ISMIS, 2002, pp. 342–350.

[13] H. Dai, B. Mobasher, A road map to more effective web personalization: Integrating domain knowledge with web usage mining, in: International Conference on Internet Computing, 2003, pp. 58–64.

[14] B. Mobasher, H. Dai, T. Luo, Y. Sun, J. Zhu, Integrating web usage and content mining for more effective personalization, in: Proceedings of the First International Conference on Electronic Commerce and Web Technologies, Springer-Verlag, 2000, pp. 165–176.

[15] J. Guo, V. Kešelj, Q. Gao, Integrating web content clustering into web log association rule mining, in: Proceedings of Canadian AI'2005, Victoria, BC, Canada, May 2005.

[16] J. Li, O.R. Zaïane, Combining usage, content, and structure data to improve web site recommendation, in: EC-Web, 2004, pp. 305–315.

[17] W.B. Cavnar, Using an *n*-gram-based document representation with a vector processing retrieval model, in: TREC, 1994, pp. 269–278.

[18] V. Kešelj, F. Peng, N. Cercone, C. Thomas, *N*-gram-based author profiles for authorship attribution, in: Proceedings of the Conference Pacific Association for Computational Linguistics, Nova Scotia, Canada, 2003.

[19] A. Tomovic, P. Janicic, V. Kešelj, *N*-gram-based classification and hierarchical clustering of genome sequences, Computer Methods and Programs in Biomedicine (2005).

[20] R. Cooley, B. Mobasher, J. Srivastava, Data preparation for mining world wide web browsing patterns, Knowledge and Information Systems 1 (1) (1999) 5–32.

[21] Robot List, <http://www.robotstxt.org/> (accessed June 2005).

[22] R. Kohavi, R. Parekh, Ten supplementary analyses to improve e-commerce web sites, in: Proceedings of the Fifth WEBKDD workshop, 2003.

[23] M. Spiliopoulou, B. Mobasher, B. Berendt, M. Nakagawa, A framework for the evaluation of session reconstruction heuristics in web-usage analysis, INFORMS Journal on Computing 15 (2) (2003) 171–190.

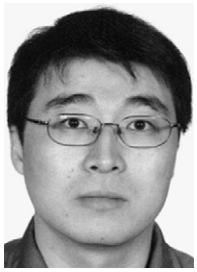
[24] The Personal Information Protection and Electronic Documents Act (PIPEDA), <http://www.privcom.gc.ca/legislation/02_06_01_e.asp> (accessed June 2005).

[25] B. Berendt, B. Mobasher, M. Spiliopoulou, J. 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, Chicago, IL, USA, 2001.

[26] P.K. Chan, A non-invasive learning approach to building web user profiles, in: Workshop on Web usage analysis and user profiling, Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, 1999.

[27] S. Dumais, T. Joachims, K. Bharat, A. Weigend, Sigir 2003 workshop report: implicit measures of user interests and preferences, ACM SIGIR Forum (Fall) (2003).

[28] M. Morita, Y. Shinoda, Information filtering based on user behavior analysis and best match text retrieval, in: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag New York, Inc., Dublin, Ireland, 1994, pp. 272–281.
[29] Weka: Machine learning software in Java, <http://www.cs.waikato.ac.nz/~ml/weka/> (accessed June 2005).
[30] J. He, A.-H. Tan, C.-L. Tan, S.-Y. Sung, On quantitative evaluation of clustering systems, in: W. Wu, H. Xiong (Eds.), Information Retrieval and Clustering, Kluwer Academic Publishers, 2003, pp. 105–134.
[31] L. Yi, B. Liu, X. Li, Eliminating noisy information in web pages for data mining, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, Washington, DC, 2003, pp. 296–305.
[32] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999.
[33] V. Kešelj, Perl package Text::Ngrams, 2003, <http://www.cs.dal.ca/~vlado/srcperl/Ngrams> or <http://search.cpan.org/author/VLADO/Text-Ngrams-1.1/> (accessed June 2005).
[34] Y. Miao, V. Kešelj, E.E. Milios, Comparing document clustering using *n*-grams, terms and words, Master's thesis, Faculty of Computer Science, Dalhousie University, 2004.



Haibin Liu (BE, Chemical Engineering, Beijing University of Chemical Technology, China, 2000; BS, Computer Science and Technology, Tsinghua University, China, 2003; MEC, Electronic Commerce, Dalhousie University, Canada, 2005) is a PhD candidate at Faculty of Computer Science, Dalhousie University, Canada. His research interests concern web mining and text mining in Bioinformatics.

Dr. Vlado Keselj is Associate Professor in Computer Science at Dalhousie University, Canada. He obtained his B.Math (CS) at the University of Belgrade (Yugoslavia), MMath (CS) and PhD at the University of Waterloo (Canada). His research interests include Natural Language Processing and Text Mining. Web page: http://www.cs.dal.ca/~vlado