Modeling user hidden navigational behavior for Web recommendation

Guandong Xu a,*, Lin Li b, Yanchun Zhang a, Xun Yi a and Masaru Kitsuregawa c

a School of Engineering & Science, Victoria University, PO Box 14428, VIC 8001, Australia
E-mail: {Guandong.Xu, Yanchun.Zhang, Xun.Yi}@vu.edu.au

b School of Computer Science & Technology, Wuhan University of Technology, China
E-mail: Cathylilin@whut.edu.cn

c Institute of Industrial Science, University of Tokyo, Japan
E-mail: Kitsure@tkl.iis.u-tokyo.ac.jp

Abstract. Web users exhibit a variety of navigational interests through clicking a sequence of Web pages. Analyses of Web usage data will lead to discovering Web user access patterns, and in turn, facilitating users to locate more preferable Web contents via collaborative recommendation techniques. In the context of Web usage mining, Latent Semantic Analysis (LSA) based on probability inference provides a promising approach to capture not only user hidden navigational patterns, but also the associations between users, pages and hidden navigational patterns. The discovered user access patterns could be used as a usage reference base for identifying the new user’s access preferences and making usage-based collaborative Web recommendations. In this paper, we propose a novel usage-based Web recommendation framework, in which Latent Dirichlet Allocation (LDA) is employed to learn the underlying task space from the training Web log data and infer the task distribution for a target user via task inference. The main advantages of the adapted LDA model are its capabilities of efficiently learning the semantic usage information from the Web log data and effectively inferring the access preference of the target user even with a few Web clicks that might be unseen in the training data. In this paper, we also investigate the determination of an optimizing task number, which is another important problem commonly encountered in latent semantic analysis. Experiments conducted on a real Web log dataset show that this approach can achieve better recommendation performance in comparison to other existing techniques. And the discovered task-simplex expression can also provide a better interpretation for Web designers or users to better understand the user navigational preference.

Keywords: User modeling, Web recommendation, hidden navigational task mining, latent Dirichlet allocation

1. Introduction

With the massive influx of information onto the World Wide Web, the Internet has become a huge platform to facilitate information dissemination, information retrieval as well as business execution, e.g., e-commerce [30,35]. By efficiently utilizing the operational options provided by Web sites, Web users are able to achieve their interest-oriented tasks. For example, imagine a Web site devoted to online service regarding sports goods. Generally, there exist various types of user groups associated with different interests while they were visiting such an e-commerce website. One type of customers intends to browse specific category products, for example footwears, by browsing various content pages representing different brands, while another just exhibits more interest in purchasing products belonging to one particular famous brand, such as “Nike”, rather than one specific sports product category. In this manner, different types of clicks-stream data will be recorded in Web access log files, which may convey the usage pattern information. As a result, delivering the customized Web contents to various kinds of Web users will significantly improve the functionality and operatability of an e-commerce site.
due to its attractiveness, in turn, leading to the increase in user click-throughs.

On the other hand, because of the explosive growth of information over the Internet in last decades, information overload is becoming a big challenge suffering Web users in finding the needed information from the massive data available on the Web. For example, in the context of Web search, how to locate the desirable Web information that Web users are really interested in or need, is emerging as a very popular topic in Web search, which attracts much attention not only from the researcher communities, but also the industrial partners [35]. A variety of research efforts have been contributed to tackle the above problems. Amongst these developed techniques, Web recommendation is probably a promising and effective way that is capable of presenting the preferable Web contents to various type of users based on predicting the user navigational interest and referring to the access preferences of other like-minded users, which are derived from the Web log files.

Web recommendation is proposed and developed with the benefits of research progresses on recommender systems [12, 24]. The traditional recommender systems have been intensively studied in artificial intelligence and user interface research communities in the last decade. To-date, there are two main kinds of approaches and techniques commonly used in recommender systems, namely Content-Based (CB) filtering [21] and Collaborative Filtering (CF) systems [15]. Content-based filtering systems generate the recommendations on a basis of the pre-constructed user profiles by measuring the similarity of Web contents to these profiles. In contrast, collaborative filtering systems make the recommendations by referring to the access preferences exhibited by other users who are like-minded or share similar access interests. For example, MovieLens developed by the University of Minnesota, is a typical recommender system based on collaborative filtering. This system is able to recommend the preferable movies to visitors upon the initial input of preference by visitors. It starts at creating the item rating or user correlation matrices from the expressed opinions about movies by visitors. When a new target user comes in, the system then determines a like-minded ‘neighborhood’ of the target user from other users by measuring the similarity between the rating vectors of the target user and other users. Ratings from these neighbors are used to create personalized recommendations for the target user (the typical recommendation scheme based on collaborative filtering is shown in Fig. 1). Since the collaborative filtering techniques refer to the common interests of the like-minded users rather than the only content relevance to make recommendations via a statistical learning procedure, they are able to present more preferable Web contents in a broader extent to the user. The main difficulty of the collaborative filtering based recommender systems is the similarity computation for finding the nearest neighbors, that usually needs to be completed in real time. This difficulty will largely affect the practical performance of recommendation, especially in the settings of a large volume of rating or visiting databases.

To address the online recommendation difficulties incurred in real recommender systems, especially in Web recommender systems, researchers in Web search areas proposed to combine Web Usage Mining (WUM), one kind of Web mining approaches, with collaborative filtering approaches for usage-based Web recommendation. For example, a two-phase recommendation framework was proposed by Mobasher et al. [25], in which an offline user profiling procedure is executed first to discover user session aggregates via clustering user sessions; and then an online recommendation calculation is performed to select the closest user profile to the target user session and determine the recommendation score for each Web page. Because the most time-consuming part of the approach is happened in the offline stage, the proposed clustering based recommender system makes it possible to generate the needed Web pages promptly and accurately in real time. The generic description of a Web recommendation system is summarized as follows: it first discovers various usage patterns from the collected Web usage data using the WUM approaches, and builds up various types of user profiles as a user navigational

![Fig. 1. The typical scheme of recommender systems based on collaborative filtering.](image-url)
knowledge base. Then when a new target user is coming, the system predicts the possible access interest of the target user, and refers to the common visiting preferences of the like-minded users collaboratively, and finally makes the recommendations. The whole process of WUM and Web recommendation consisting of three stages is depicted in Fig. 2.

However, the current Web recommender systems suffer from the following major concerns. The first common difficulty is how to measure the similarity of navigation interest between various Web users and how to partition the like-minded Web users into various user groups. Since the measuring and partitioning operations are performed directly on the data of ratings or clicks of Web items by Web users, this kind of computations is usually very time-consuming in computation as a result of the high dimensional nature of the input space. For example, because the analyzed data may contain millions of user ratings or user clicks on various Web items, it is difficult to calculate the user-based item similarity explicitly by using the users as the columns in user clustering [25]. As a consequence, this obstacle greatly affects the efficiency of the online recommender systems. The second challenge is resulting from the sparse and short input (for example, 2 or 3 keywords input and clicks) from the user. With the only sparse and short query input, it sometime is unlikely to accurately determine which user group is the most like-minded one matching the target user’s needs based on simply calculating the similarity between them, and to make the recommendations accordingly. The third one is the lack of the capability of the semantic analysis on Web usage data. Even we can partition various users into the corresponding groups based on their access behaviors, we still can not understand the proper latent semantic property of each user group, i.e. why such users are grouped together in this cluster.

To address the above problems, Latent Semantic Analysis (LSA) based on statistical learning was proposed for conducting semantic analysis, for example, the traditional Latent Semantic Indexing (LSI) approaches are first proposed for dealing with the topic identification in Digital Library. Based on various mathematical foundation and formulation, many variants of LSA are proposed and developed, such as Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) in Data Mining and Machine Learning. The LSA algorithms could well tackle the concerns discussed above. First, the LSA algorithms are often utilized as an efficient dimensionality reduction approach, such as Principal Component Analysis (PCA), which is able to capture the principle and inherent correlation amongst the co-occurrence observations in a reduced co-occurrence space. Furthermore, the LSA algorithms intend to introduce the concept of navigational task or factor into usage mining, which makes it possible for LSA to have the capability of latent semantic analysis. With the transformation of the original usage data into a new data expression over the more implicit task space, the visiting preferences of users could be accurately captured and predicted via referring to the choices by other like-minded users, even in the case of the user clicking only a few items. LSA has achieved great successes in information retrieval [9], topic estimation [16] and text classification [27]. pLSA [16] is another kind of latent semantic analysis based on probability uncertainty theory. The main characteristics of pLSA model is an aspect model, which assumes the Web user behavior is governed by a set of navigational tasks. The aim of pLSA model is to estimate the probability distribution of co-occurrence observations over the tasks, in turn, to generate user profiles. In our previous work [31], we investigate the study of employing pLSA model for user profiling via Web usage mining. The Experimental study validated that the usage-based user profiles could be efficiently utilized for Web recommendation. But the pLSA model suffers from some difficulties, for example, the overfitting problem due to its intrinsic weakness in statistical foundation [4] and inability for inferring the navigational task for a new user [4]. In this paper, we aim to devise a novel framework of Web recommendation based on modeling and inferring user hidden navigational tasks via the LDA model, and...
making the usage-based Web recommendations. The main contributions of this paper are highlighted as follows:

- We devise a versatile Web recommendation framework, in which the Latent Dirichlet Allocation (LDA) model is introduced to model user hidden navigational tasks, in turn, for Web recommendation. To our best knowledge, LDA is initially proposed as a latent analytical approach for text analysis; it is probably the first to be employed in modeling user access behaviors.

- To determine the user navigational preference, we aim to perform a task inference process rather than simply calculating the similarity between the sparse and short input of the target user and the learned user access patterns. Due to the strength of the complete generative process of the LDA model, it can largely improve the performance of predicting the user visit preference.

- Investigation on choosing an optimizing number of user groups. In the research of unsupervised clustering or latent semantic analysis, selecting an appropriate and optimal number of clusters is often crucial and difficult. In this paper, we aim to employ a measure of modularity $Q$ function to evaluate the quality of user clusters.

- Experimental evaluations are manipulated to validate the efficiency of the employed approach in terms of recommendation precision of the proposed framework. The comparative study with other existing approaches of Web recommendation has shown the advantages of the employed approach we claim here.

The remainder of the paper is structured as follows: we introduce the backgrounds and mathematical expressions of the study in terms of Web usage mining and LDA in Section 2. Section 3 intensively discusses the methodology of using the LDA model to capture user hidden navigational tasks and make collaborative Web recommendations. The involved algorithms are presented in this section as well. We address choosing an optimizing number of user clusters via calculating $Q$ function in Section 4. Experiments and results on a real weblog dataset are presented in Section 5. Related work is reviewed in Section 6. And finally we conclude the paper and outline the future work in Section 7.

Fig. 3. The framework of usage-based Web recommendation with hidden navigational tasks.

2. Background

2.1. The general usage-based collaborative Web recommendation framework

In this section, we first give a general description of the proposed framework of the usage-based collaborative Web recommendation based on the hidden user navigational task model (shown in Fig. 3). The framework consists of four major components: (a) Web log data collection, and training and test dataset preparation, (b) user hidden navigational task estimation using training data and building up user profiles, (c) navigational task inference for test usage data, (d) recommending Web contents based on the identified hidden navigational tasks.

Among the four steps, collecting a large Web log data is probably the most important step for successful hidden navigational task estimation and Web recommendation. The processed usage data should cover as many and generic trained tasks as possible in the study domain for further task inference, and it also needs to be consistent with the test data. The Web log data is usually obtained from Web log servers, in the expressions of many data forms, such as adjacency matrices, usage matrices, click sequences and neighboring graphs and so on. Among these expression forms, usage matrix is a mostly used one in Web usage mining, which reflects the observations of user clicks on various Web items. More details regarding the usage matrix (or called the characteristic matrix) will be discussed in the following Section 2.3. To evaluate our proposed framework and algorithms, we need to carry out experiments with test data. We randomly select a portion of the collected Web log data as a test dataset, and use the rest of the dataset for the purpose of hidden navigational task estimation training.

The second step, doing the hidden navigational task estimation from the training Web log data, is imple-
mented by using one of the well-known hidden topic learning models, i.e. LDA model. Essentially LDA is proposed to learn the topic distribution of contents in text mining, but in the context of Web usage mining we adapt the concept of “topic” as the navigational task. Another reason why we choose the LDA model is its advantages of the low overfitting learning and the complete topic generation [4]. The detailed description of the LDA model will be discussed in the following Section 2.2. After task estimation by LDA model, we eventually obtain the underlying associations, which are characterized by two estimated matrices between users and hidden tasks, and hidden tasks and Web pages. The former is used to construct the usage-based user profiles, while the latter is employed for task inference in the next step.

Task inference for the test usage data of the target users (c) is another core component of the whole process. This step is to calculate the navigational task distributions of the target user. By selecting the significant tasks that have the assigned probabilities exceeding a predefined threshold, we can determine those tasks as the dominant visiting “themes” of the target users.

Finally by incorporating the user profile knowledge derived from step (b) with the identified visiting themes, we sort all Web items in a descending order, in which the ranking is dependent on the corresponding contribution weights calculated by a top-N scoring approach. We then select the top-N Web items in the list as the Web recommendations to the target users (d). In Section 5, we demonstrate the evaluation of the proposed framework on a real Web log file.

2.2. Hidden navigational task analysis model

Latent Dirichlet Allocation (LDA), firstly proposed by Blei et al. [4], is a probabilistic generative model that can be used to estimate topic assignments from co-occurrence observations by using an iterative learning process. In a generalized expression, LDA is a variant of so-called Latent Semantic Analysis (LSA). The term LSA has been coined by Deewester [9] for text mining, in which the co-occurrence of terms in text documents was used to derive a latent topic space hidden in the document collection via matrix decomposition operations. More theoretically, LSA is on a basis of an assumption that there exists an unseen structure of “topics” or “themes” in the text corpus, which governs the co-occurrence observations. As such, the intuition behind LSA is to discover this latent topic structure via extracting the “abstracts” of the original co-occurrence activities. The extracted forms of the latent topics can help to better handle the linguistic phenomenon like synonymy and polysemy as they represent the textual information on a semantic level rather than by lexical occurrence. Probabilistic Latent Semantic Analysis (pLSA) is also a sibling variant of LSA, which shares the similar principle of hidden topic discovery. pLSA was firstly formulated and proposed by Hoffman, and was successfully employed in applications such as document topic modeling [16], text categorization [27] and collaborative filtering [17]. However, due to the completeness of the generative process, LDA has demonstrated to be superior to pLSA in terms of avoiding overfitting problems and topic inference by some studies [4]. In this study, we aim to employ the LDA model to conduct the hidden task analysis.

2.2.1. Latent Dirichlet allocation (LDA)

LDA is a generative probabilistic model of co-occurrence observations. As for a typical application in text analysis, here we utilize the document corpus as the co-occurrence observations to conduct the following formulation. As discussed above, the basic idea of the LDA model is that the documents within the corpus are represented as a random mixture over the latent topics and each topic is characterized by a distribution over the words in the documents. The graphical illustration of the generative procedure of LDA is shown in Fig. 4. LDA is performed via a sequence of the document generation processes (shown in Fig. 4 and Algorithm 1). The notations used and the generative procedure of LDA model are outlined as follows.

Notations:
- $M$: the number of documents
- $K$: the number of topics
- $V$: the size of vocabulary
- $\alpha, \beta$: Dirichlet parameters
- $\theta_m$: the topic assignment of the document $m$
Algorithm 1 (Generation Process of LDA).

for each of topics

sample the mixture of words \( \phi_k \sim \text{Dir}(\beta) \)
end for each of documents \( m = 1 : M \)

sample the mixture of topics \( \theta_m \sim \text{Dir}(\alpha) \)
sample the lengths of documents \( N_m \sim \text{Pois}(\xi) \) for each word \( n = 1 : N_m \) in the document \( m \)
sample the topic index of \( z_{m,n} \sim \text{Mult}(\theta_m) \)
sample the weight of word \( w_{m,n} \sim \text{Mult}(\phi_{z_{m,n}}) \)
end

In LDA, a document \( d_m = \{w_{m,n}, n = 1, \ldots, N_m\} \) is generated by picking a distribution over the topics from a Dirichlet distribution (\( \text{Dir}(\alpha) \)). And given the topic distribution, we pick the topic assignment of each specific word. Then the topic assignment for each word placeholder \( [m, n] \) is calculated by sampling a particular topic from the multinomial distribution of \( z_{m,n} \). And finally, a particular word of \( w_{m,n} \) is generated for the placeholder \( [m, n] \) by sampling its weight from the multinomial distribution of \( \text{Mult}(\phi_{z_{m,n}}) \).

Known from the above description, given Dirichlet parameters \( \alpha \) and \( \beta \), we can formulate a joint distribution of a document \( d_m \), a topic mixture of \( d_m \), i.e. \( \theta_m \), and a set of \( N_m \) topics, i.e. \( z_m \), as follows.

\[
P_r(\theta_m, z_m, d_m | \alpha, \beta) = P_r(\theta_m | \alpha) \prod_{n=1}^{N_m} P_r(w_{m,n} | \phi_{z_{m,n}}) P_r(z_{m,n} | \theta_m).
\]

And integrating over \( \theta_m, \phi_{z_{m,n}} \) and summing over \( z_m \), we obtain the likelihood of the document \( d_m \):

\[
P_r(d_m | \alpha, \beta) = \int P_r(\theta_m | \alpha) \times \prod_{n=1}^{N_m} P_r(w_{m,n} | \phi_{z_{m,n}}) P_r(z_{m,n} | \theta_m) \, d\theta_m.
\]

Finally the likelihood of the document corpus \( D = \{d_m, m = 1, \ldots, M\} \) is a product of the likelihood of all documents in the corpus.

\[
P_r(D | \alpha, \beta) = \prod_{m=1}^{M} P_r(d_m | \alpha, \beta).
\]

2.2.2. Dirichlet parameter estimation and topic inference

In general, estimating the parameters of LDA is performed by maximizing the likelihood of the whole documents. In particular, given a corpus of documents \( D = \{d_m, m = 1, \ldots, M\} \), we aim to estimate the parameters of \( \alpha \) and \( \beta \) that maximize the log likelihood of the data:

\[
(\alpha_{\text{est}}, \beta_{\text{est}}) = \max (\alpha, \beta) = \max \sum_{m=1}^{M} \log P_r(d_m | \alpha, \beta).
\]

However the direct computing for the parameters \( \alpha \) and \( \beta \) is intractable due to the nature of the computation. The solution to this is to use various alternative approximate estimation methods. Here we employ the variational EM algorithm [4] to estimate the variational parameters that maximize the total likelihood of the corpus with respect to the model parameters of \( \alpha \) and \( \beta \). The variational EM algorithm is briefly described as follows:

1. (E-step) For each document, find the optimizing values of variational parameters \( \theta_m^* \) and \( \phi_{z_{m,n}}^* \).

2. (M-step) Maximize the resulting low bound on the likelihood with respect to model parameters \( \alpha \) and \( \beta \). This corresponds to finding the maximum likelihood estimate with the approximate posterior which is computed in the E-step.

The E-step and M-step are executed iteratively until a maximum likelihood value reaches. Meanwhile, the calculated estimation parameters can be used to infer topic distribution of a new document by performing the variational inference. More details with respect to the variational EM algorithm are referred to [4].

2.3. Web usage data model

Since we aim to apply the technique used in the context of text mining to Web usage mining, the most important point that we have to start is to construct a usage data model and then map it into an analogous document-word space, where a generative model will be employed to derive the hidden user access patterns.
First of all, we thus need to define what words and documents are actually equivalent to in the context of Web usage processing.

In general, the exhibited user access interests may be reflected by the varying degrees of visits on different Web pages during one user session. Thus, we can represent a user session as a weighted vector of pages visited by the user during a particular period. In this manner, the analogues of a word and a document addressed in text mining domain could be viewed as a Web pageview and a Web user session respectively, in turn, resulting in a Web pageview collection and a user session collection corresponding to word vocabulary and document corpus concepts in text mining context. With the introduction of Web session-pageview expression, we further generalize a Web usage data model for Web usage mining and Web recommendation. In this paper, we use the following notations to model the co-occurrence activities of Web users and pages:

- \( S = \{ s_i, i = 1, \ldots, m \} \): a set of \( m \) user sessions.
- \( P = \{ p_j, j = 1, \ldots, n \} \): a set of \( n \) Web pageviews.

For each user, a user session is represented as a vector of visited pageviews with corresponding weights: \( s_i = \{ a_{ij}, j = 1, \ldots, n \} \), where \( a_{ij} \) denotes the weight for page \( p_j \) visited in \( s_i \) user session. The corresponding weight is usually determined by the number of hits or the amount of time spent on the specific page. Here, we use the latter to construct the usage data for the dataset used in experimental study.

- \( SP_{m \times n} = \{ a_{ij}, i = 1, \ldots, m, j = 1, \ldots, n \} \): the ultimate usage data in the form of a weight matrix with a dimensionality of \( m \times n \).

Generally, the element in the session-pageview matrix, \( a_{ij} \), is the normalized weight associated with the page \( p_j \) in the user session \( s_i \). The session normalization is able to capture the relative significance of a page within one user session with respect to others pages accessed by same user. Figure 5 depicts a usage data snapshot from a weblog file along with its corresponding usage data in the form of the normalized page weight vector [19]. Particularly, the session begins with a line of the form:

SESSION \#n (USER_ID = k)

where \( n \) is the session number, and \( k \) is the user id. Within a given session, each line corresponds to a list of specific page accesses. Each line in a session is a tab d elimited sequence of three fields: time stamp, page accessed, and the referer. The time stamp represents the number of seconds relative to January 1, 2002. For each session, the pageview duration of the last pageview in that session, was estimated to be the average duration of that pageview across all sessions. To eliminate the influence of the difference in the visit duration each element in the usage matrix is normalized by calculating the ratio of the visiting time on the corresponding page to the total visiting time, e.g. \( \frac{w_{ect2002.pdf}}{14 + 68 + 326 + 6 + 55 + 94} \times 100 = 9.77 \ldots \) and so on.

3. Modeling user navigational task via LDA

Similar to capturing the underlying topics over the word vocabulary and each document’s probability distribution over the mixing topic space, LDA is also able to be used to discover the hidden access tasks and the user preference mixtures over the uncovered task space from user navigational observations. That is, from the usage data, LDA identifies the hidden tasks and represents each task as a simplex of Web pages. Furthermore, LDA characterizes each Web user session as a simplex of these discovered tasks. In other words, LDA reveals two aspects of underlying usage information to us: the hidden task space and the task mixture distribution of each Web user session, which reflect the underlying correlations between Web pages as well as Web user sessions. With the discovered task-simplex expressions, it is viable to model the user access patterns in the forms of weighted page vectors, in turn, to predict the target user’s potentially interested pages by employing a collaborative recommendation algorithm. In the following part, we first discuss how to discover the user access patterns in terms of task-simplex expression based on LDA model and how to infer a target user’s navigational preference by using the estimated task-simplex
space, and then present a collaborative recommendation algorithm by incorporating the inferred navigational task distribution of the target user with the discovered user access patterns to make Web recommendations.

3.1. Discovering user navigational tasks based on LDA

Similar to the implementation of document-topic expression in text mining discussed above, viewing Web user sessions as mixtures of tasks makes it possible to formulate the problem of identifying the set of underlying tasks hidden in a collection of user clicks. Given $m$ Web user sessions containing $t$ hidden tasks expressed over $n$ distinctive pageviews, we can represent $P_t(\{p|z\})$ with a set of $t$ multinomial distributions $\phi$ over the $n$ Web pages, such that for a hidden task $z_k$, the associative likelihood on the page $p_j$ is $P_t(p_j|z_k) = \phi_{k,j}$, $j = 1, \ldots, n$, $k = 1, \ldots, t$, and $P_t(z|s)$ with a set of $m$ multinomial distributions $\theta$ over the $t$ tasks, such that for a Web session $s_i$, the associative likelihood on the task $z_k$ is $P_t(z_k|s_i) = \theta_{i,k}$.

Here we use the LDA model described above to generate $\phi$ and $\theta$ resulting in the maximum likelihood estimates of LDA model in the Web usage data. The complete probability model is as follows:

$$
\begin{align*}
\theta_i & \sim \text{Dirichlet}(\alpha), \\
\phi_k & \sim \text{Discrete}(\beta), \\
p_j | \phi_k & \sim \text{Discrete}(\phi_k).
\end{align*}
$$

(3)

Here, $z_k$ stands for a specific task, $\theta_i$ denotes the Web session $s_i$’s navigational preference distribution over the tasks and $\phi_k$ represents the specific task $z_k$’s association distribution over the pages. Parameters $\alpha$ and $\beta$ are the hyper-parameters of the prior variables of $\theta$ and $\phi$.

We use the variational inference algorithm described above to estimate each Web session’s correlation with the multiple tasks ($\theta$), and the associations between the tasks and Web pages ($\phi$), with which we can capture the user visit preference distribution exhibited by each user session and identify the semantics of the task space. Interpreting the contents of the prominent pages related to each task based on $\phi$ will eventually result in defining the nature of each task. Meanwhile, the task-based user access patterns are constructed by examining the calculated user session’s associations with the multiple tasks and aggregating all sessions whose associative degrees with a specific task are greater than a threshold. We describe our approach to discovering the task-based user access patterns below.

Given this representation, for each latent task, (1) we can consider user sessions with $\theta_{i,k}$ exceeding a threshold as the “prototypical” user sessions associated with that task. In other words, these top user sessions contribute significantly to the navigational pattern of this task, in turn, are used to construct this task-specific user access pattern; (2) we select all Web pages as contributive pages of each task dependent on the values of $\phi_k$, and capture the semantics of the task by interpreting the contents of those pages.

Thus, for each latent task corresponding to one access pattern, we choose all user sessions with $\theta_{i,k}$ exceeding a certain threshold as the candidates of this specific access pattern. As each user session is represented by a weighted page vector in the original usage data space, we can create an aggregate of user sessions to represent this task-specific access pattern in the form of weighted page vector. The algorithm of generating the task-specific access pattern is described as follows:

Algorithm 2 (Building Task-Specific User Access Patterns).

Input: the discovered session-task preference distribution matrix $\theta$, $m$ user sessions $S = \{s_i, i = 1, \ldots, m\}$, and the predefined threshold $\mu$.

1. For each latent task $z_k$, choose all user sessions with $\theta_{i,k} > \mu$ to construct a user session aggregation $R_k$ corresponding to $z_k$:

$$R_k = \{s_i|\theta_{i,k} > \mu, k = 1, \ldots, t\}.$$ 

(4)

2. Within the $R_k$, compute the aggregated task-specific user access pattern in terms of a weighted page vector by taking the sessions’ associations with $z_k$, i.e. $\theta_{i,k}$, into account:

$$ap_k = \frac{\sum_{s_i \in R_k} \theta_{i,k} \cdot s_i}{|R_k|}.$$ 

(5)

where $|R_k|$ is the number of the chosen user sessions in $R_k$.

3. Output a set of task-specific user access patterns TAP corresponding to $t$ tasks, $TAP = \{ap_k, k = 1, \ldots, t\}$. In this expression, each user access pattern is represented by a weighted page vector, where the weights indicate the relative visit preferences of pages exhibited by all associated user sessions for this task-specific access pattern.
3.2. Collaborative Web recommendation using hidden task inference

After we have built up a set of task-based user access patterns in terms of weighted page vectors, we can utilize these patterns to make collaborative Web recommendations by identifying the target user’s visit task preference and integrating it with the learned task-based user access patterns to make Web recommendations.

The recommendation process consists of two sub-tasks. The first stage is to identify the target user’s visit preference distribution over the hidden tasks by using the task inference of LDA model. With a few inputs or clicks on pages, we can predict the main access themes of the target user based on the learned associations between tasks and Web objects (i.e. the estimated values of \( \theta \) and \( \phi \)). The inferred task assignment of the target user session indicates how preferable the tasks are intended by the target user. Secondly, we choose the most dominant tasks by examining whether the probabilities are exceeding a certain value, then refer to the corresponding user access patterns that discovered in Section 3.1, and finally sum up the page weights in the identified user access patterns. As the summed page weights reflect the overall likelihoods of the pages visited by the target user, we then sort the weights in a descending order to rank the predicted recommendation scores of the pages. Here we will use a top-N ranking approach to predict the target user’s potentially interested pages, that is, the top N pages with the N highest ranking scores are the recommended Web pages. Given a set of user access patterns and the current target user session, the algorithm of generating the top-N Web recommendation is described as follows:

**Algorithm 3** (Top-N Collaborative Web Recommendation using Hidden Task Inference).

Input: A target user session, a set of the learned user access patterns and a threshold \( \delta \).

Output: The top-N recommended Web pages

1. Given a target user session \( s_t = \{ p^t_j, j = 1, \ldots, j_t \} \), we determine the access task distribution of the target user session by performing the task inference using the associations between the hidden tasks and the Web objects learned from the above task estimation process, and choose the dominant access tasks by selecting the corresponding tasks with probability values exceeding the threshold value.

\[
Z^t = \{ z^t_k | Pr ( z_k | s_t ) > \delta, k = 1, \ldots, k_t \}.
\]

2. Refer to the page weights in the selected dominant task-oriented access patterns, and sum up each page weight multiplied by the probability value of the corresponding task, then utilize the weights as the recommendation scores \( RS(p_j) \):

\[
RS(p_j) = \frac{1}{k_t} \sum_{k=1}^{k_t} Pr ( z_k | s_t ) \cdot ap_{k,j}
\]  \( (6) \)

where \( z_k \) denotes the dominant access task and \( ap_{k,j} \) is the weight of \( p_j \) in the access pattern corresponding to the task \( z_k \).

3. Sort the calculated recommendation scores in step 2 in a descending order, i.e. \( RS^t = ( p^t_{j_1}, p^t_{j_2}, \ldots, p^t_{j_{N_t}} ) \), and select the N pages with the top-N highest recommendation scores to construct the top-N recommendation set:

\[
WR(s_t) = \{ p^t_j | RS ( p^t_j ) > RS ( p^t_{j+1} ), j = 1, \ldots, N - 1 \}.
\]  \( (7) \)

4. Determining an optimizing number of hidden tasks

4.1. Modular \( Q \) function

The task estimation process of LDA model results in finding the task-specific user access patterns (or user profiles) by partitioning various user sessions into the corresponding user groups. Then a crucial question is arising from the partition of the user sessions that how to determine the number of the hidden task as the varying in this number will lead to various partitions that considerably affect the quality of the user profiling, in turn, having positive or negative impacts on the recommendation performance. Similar difficulties are commonly encountered in other latent semantic analysis approaches like pLSA, in which the number of the latent factors also needs to be preset before the parameter estimation. This problem could also be viewed as an equivalent challenge of finding an appropriate number of clusters in clustering. One possible solution to this is to introduce the idea of evaluating the strength of community structure and selecting an optimizing value, which is utilized in community analysis [26]. Since each user profile consisting of a number of user sessions could be considered a user session community [13], the mutual similarities among the members
within the community reflect the strength of the community. Thus a better partition of user sessions means a closer internal association within the community, but a stronger disjoining from the members of this community to those in the other communities. There are many metrics that can evaluate the strength of community structure, such as min-cut used in [11] and edge ratio defined in [3]. Recently the modularity $Q$ function proposed by Newman and Girvan in [26], was frequently used and recognized as a standard metric of evaluating the quality of a community structure. This metric measures the difference between a community and its randomly constructed community that has the same structure but random connections between vertices. Here we choose it as a measure to validate the inherent structure, such as min-cut used in [11] and edge ratio proposed by [34] to update the formulation of modularity $Q$ function. The modified modularity $Q$ function is defined as follow.

**Definition 1.** If a networking $G = (V, E)$, where $V$ and $E$ are vertexes and edges of $G$ respectively, and the size of $V$ is $n$, has a partition of $P_k$ ($k$ is the number of communities), its modularity $Q$ function is

$$Q(P_k) = \sum_{c=1}^{k} \left[ \frac{A(V_c, V_c)}{A(V, V)} - \left( \frac{A(V_c, V)}{A(V, V)} \right)^2 \right]$$

where $A(V', V'') = \sum_{u \in V', v \in V''} w(u, v)$.

The $w(u, v)$ is the weight of the edge between vertexes $u$ and $v$. In the case of Web usage mining, we can use a cosine function of two user sessions $u$ and $v$ as their mutual weight. Known from the task estimation of LDA model, one user session could be viewed as a multi-peak distribution over the task space. That is, one user session indeed has a task assignment to various task-based user profiles with varying probabilities. In other words, such kind of partitions on the user sessions is actually a soft assignment rather than a crisp or hard partition. Thus it is necessary to take into consideration of the individual weights contributed by all members within the cluster when we evaluate the quality of the community structure. Here we employ a measuring method proposed by [34] to update the formulation of modularity $Q$ function. The modified modularity $Q$ function is described as follows.

**Definition 2.** Given the networking having $k$ communities and each vertex has a soft-assignment to various communities, we define a corresponding $n \times k$ soft assignment matrix $U_k = [u_1, \ldots, u_n]^T$, $u_i = [u_{i,1}, \ldots, u_{i,k}]$ with $0 \leq u_{i,c} \leq 1$ for each $c = 1, \ldots, k$, and $\sum_{c=1}^{k} u_{i,c} = 1$ for each $i = 1, \ldots, n$. With this expression we can further define a membership of each community as $V_c = \{ i | u_{i,c} > \lambda, i = 1, \ldots, n \}$, where $\lambda$ is a threshold to select the dominant candidates. As a result, we can define a modified modularity $Q$ function as

$$\tilde{Q}(U_k) = \sum_{c=1}^{k} \left[ \frac{A(\hat{V}_c, \hat{V}_c)}{A(V, V)} - \left( \frac{A(\hat{V}_c, \hat{V})}{A(V, V)} \right)^2 \right]$$

where $U_k$ is a multi-task partition of vertices into $k$ communities, $A(\hat{V}_c, \hat{V}_c) = \sum_{i \in \hat{V}_c, j \in \hat{V}_c} ((u_{i,c} + u_{j,c})/2)w(i, j)$, $A(\hat{V}_c, \hat{V}) = A(\hat{V}_c, \hat{V}) + \sum_{i \in \hat{V}_c, j \in \hat{V} \setminus \hat{V}_c} ((u_{i,c} + u_{j,c})/2)w(i, j)$ and $A(V, V) = \sum_{i,j \in \hat{V}} w(i, j)$.

In this case, our ultimate aim of determining an optimal number of tasks is to find a multi-task soft partition of user sessions which maximizes the value of the modified $Q$ function with an appropriate $k$.

**4.2. Optimizing the number of hidden tasks in LDA model**

In the case of LDA model, we need to specify the exact variables used in $Q$ function, here we formulate the following quantities to replace the corresponding variables in Eq. (8):

**Definition 3.** Given two user session $s_i$ and $s_j$, the weight between them is defined as the cosine function of their page vectors:

$$w(i, j) = \cos(s_i, s_j) = (s_i \cdot s_j)/\|s_i\|_2 \|s_j\|_2$$

where $(s_i \cdot s_j) = \sum_{k=1}^{m} a_{i,k} \times a_{j,k}$, $\|s_i\|_2 = \sqrt{\sum_{k=1}^{m} (s_{i,k})^2}$ and $\|s_j\|_2 = \sqrt{\sum_{k=1}^{m} (s_{j,k})^2}$.

**Definition 4.** Given the task estimation in LDA model, the soft partition of user sessions is defined as $\hat{U}_k = [u_1, \ldots, u_m]^T$, $u_i = [\theta_{i,1}, \ldots, \theta_{i,k}]$ where $m$ is the total number of the user sessions and $k$ is the number of the tasks, respectively. $\sum_{c=1}^{k} \theta_{i,c} = 1$ for $c = 1, \ldots, k$.

We give the calculation results on $Q$ values in Section 5.3.

**5. Experiments and discussions**

In order to evaluate the effectiveness of the employed Latent Dirichlet Allocation model for task-
Based on user access pattern mining and usage-based collaborative Web recommendation, we conducted experimental evaluations on a real Web log dataset. Firstly, we demonstrate the determination of an optimizing number of the user hidden navigational tasks. Based on the optimized hidden task number, we employ the LDA model to extract the semantics of the tasks via interpreting the contents of the dominant pages with the significant probabilities, and then, to learn the task-based user access patterns using Algorithm 2 discussed above. Furthermore, we aim to make the usage-based collaborative Web recommendations by inferring the dominant task assignment of the target user, and referring to the learned user access models corresponding to the significant hidden tasks of the target user. In addition, evaluation metrics are introduced to compare the effectiveness of the proposed framework and algorithms with other existing conventional Web recommendation methods.

5.1. Dataset

The Web log dataset is from an academic website log files [19]. The data is based on a 2-week Web log file during April of 2002. After data preprocessing, the filtered data contains 13745 sessions and 683 pages. This data file is expressed as a session-page matrix where each column is a page and each row is a session represented as a page vector. The entries in the table correspond to the amount of time (in seconds) spent on pages during a given session. For the purpose of convenience, we refer this data as “CTI data”. We randomly choose 1000 user sessions from the whole dataset as the test dataset, whereas the remainder part is selected as the training set for finding the user hidden navigational tasks and constructing the user profiles.

5.2. Evaluation metric

In the context of Web recommendation, the effectiveness of recommendation scheme is normally evaluated by the precision of recommendation. Here, we exploit a metric called hit ratio [19] to measure the top-N recommendation performance: Given a user session in the test set, we extract the first \( j \) pages of the testing session as an active target user session, and generate a top-N recommendation set via the procedure described in Section 3.2. We then compare the \((j+1)th\) page of the testing session with the recommendation list of N pages. If the \((j+1)th\) page is appeared in the recommended set, it is considered a hit. Here we choose the test session window size being \( j = 3 \). We count the total number of the hits, and calculate the hit ratio by averaging it by the total number of testing session in the test set, i.e. \( \text{hit ratio} = |\text{hit}|/|T| \), where \(|\text{hit}|\) and \(|T|\) represent the number of hits and testing data in the whole test set, respectively. Thus, \( \text{hit ratio} \) indicates the performance of the Web recommendation process. Obviously, a bigger value of \( N \) (number of recommendations) results in a higher hit precision.

5.3. Determination of the optimizing number of hidden tasks

Here we utilize the modularity \( Q \) function to determine the optimizing number of hidden tasks by using Eq. (8). Considering the coverage and scope of the user navigational activities, we choose \( k = 2, 3, \ldots, 20, 25 \) and calculate out the \( Q \) values (the calculated results are illustrated in Fig. 6). From the figure, we can see that the curve of \( Q \) climbs dramatically at the beginning increase of the task number, and then inclines to a flat line smoothly. The top value of \( Q \) is appearing at the position of \( k = 5 \), thus we choose the number of five as the optimizing number of the hidden tasks, to conduct the further Web usage mining and Web recommendation.

5.4. Hidden task space and user navigational task distribution

As stated in Section 3, we can use LDA to identify the semantics of the hidden tasks from the contents of prominent pages that contributing significantly to various tasks. We first present the identified five hidden task space in Tables 1 and 2. To better illustrate these tasks, we also list the URLs of the prominent pages as well as their corresponding probabilities (based on \( \phi \)) in this table respectively. We now derive the semantics of each task space. From the ta-
Table 1
The five hidden task space discovered from CTI dataset (1)

<table>
<thead>
<tr>
<th>Task # 1</th>
<th>URL</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>359</td>
<td>/cti/studentprofile/studentprofile.asp?section=mycti</td>
<td>0.135</td>
</tr>
<tr>
<td>55</td>
<td>/authenticate/login.asp?section=mycti&amp;title=mycti&amp;urlahead=studentprofile/studentprofile</td>
<td>0.070</td>
</tr>
<tr>
<td>187</td>
<td>/courses/syllabus.asp?course=447-21-301&amp;q=3&amp;y=2002&amp;id=109</td>
<td>0.056</td>
</tr>
<tr>
<td>195</td>
<td>/courses/syllabus.asp?course=450-96-303&amp;q=3&amp;y=2002&amp;id=290</td>
<td>0.035</td>
</tr>
<tr>
<td>388</td>
<td>/news/default.asp</td>
<td>0.031</td>
</tr>
<tr>
<td>363</td>
<td>/cti/ug_scholarship/nsfform.asp?section=advising</td>
<td>0.027</td>
</tr>
<tr>
<td>158</td>
<td>/courses/syllabus.asp?course=361-98-601&amp;q=3&amp;y=2002&amp;id=326</td>
<td>0.027</td>
</tr>
<tr>
<td>92</td>
<td>/courses/syllabus.asp?course=212-21-605&amp;q=3&amp;y=2002&amp;id=211</td>
<td>0.022</td>
</tr>
<tr>
<td>326</td>
<td>/cti/core/core.asp?section=news</td>
<td>0.021</td>
</tr>
<tr>
<td>523</td>
<td>/people/facultyinfo.asp?id=365</td>
<td>0.020</td>
</tr>
<tr>
<td>200</td>
<td>/courses/syllabus.asp?course=452-21-301&amp;q=3&amp;y=2002&amp;id=144</td>
<td>0.019</td>
</tr>
<tr>
<td>536</td>
<td>/people/facultyinfo.asp?id=618</td>
<td>0.019</td>
</tr>
<tr>
<td>171</td>
<td>/courses/syllabus.asp?course=416-21-306&amp;q=3&amp;y=2002&amp;id=776</td>
<td>0.016</td>
</tr>
<tr>
<td>397</td>
<td>/news/news.asp?theid=583</td>
<td>0.016</td>
</tr>
<tr>
<td>452</td>
<td>/people/facultyinfo.asp?id=191</td>
<td>0.015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task # 2</th>
<th>URL</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>67</td>
<td>/courses/schedule.asp</td>
<td>0.129</td>
</tr>
<tr>
<td>20</td>
<td>/admissions/requirements.asp</td>
<td>0.040</td>
</tr>
<tr>
<td>586</td>
<td>/programs/2002/gradcs2002.asp</td>
<td>0.040</td>
</tr>
<tr>
<td>389</td>
<td>/news/jobs.asp</td>
<td>0.040</td>
</tr>
<tr>
<td>500</td>
<td>/programs/2002/gradis2002.asp</td>
<td>0.027</td>
</tr>
<tr>
<td>388</td>
<td>/news/default.asp</td>
<td>0.025</td>
</tr>
<tr>
<td>22</td>
<td>/advising/</td>
<td>0.023</td>
</tr>
<tr>
<td>150</td>
<td>/courses/syllabus.asp?course=347-21-901&amp;q=3&amp;y=2002&amp;id=561</td>
<td>0.021</td>
</tr>
<tr>
<td>586</td>
<td>/programs/2002/gradect2002.asp</td>
<td>0.020</td>
</tr>
<tr>
<td>6</td>
<td>/admissions/general.asp</td>
<td>0.019</td>
</tr>
<tr>
<td>214</td>
<td>/courses/syllabus.asp?course=483-94-301&amp;q=3&amp;y=2002&amp;id=333</td>
<td>0.019</td>
</tr>
<tr>
<td>416</td>
<td>/people/</td>
<td>0.019</td>
</tr>
<tr>
<td>346</td>
<td>/cti/gradapp/step2.asp</td>
<td>0.018</td>
</tr>
<tr>
<td>16</td>
<td>/admissions/international.asp</td>
<td>0.018</td>
</tr>
<tr>
<td>106</td>
<td>/courses/syllabus.asp?course=224-21-907&amp;q=3&amp;y=2002&amp;id=305</td>
<td>0.018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task # 3</th>
<th>URL</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>/courses/</td>
<td>0.251</td>
</tr>
<tr>
<td>72</td>
<td>/courses/syllabilist.asp</td>
<td>0.080</td>
</tr>
<tr>
<td>388</td>
<td>/news/default.asp</td>
<td>0.050</td>
</tr>
<tr>
<td>267</td>
<td>/cti/advising/display.asp</td>
<td>0.034</td>
</tr>
<tr>
<td>287</td>
<td>/cti/advising/display.asp?page=intranetnews</td>
<td>0.031</td>
</tr>
<tr>
<td>354</td>
<td>/cti/gradassist/assistantship_form.asp?section=news</td>
<td>0.030</td>
</tr>
<tr>
<td>128</td>
<td>/courses/syllabus.asp?course=313-94-601&amp;q=3&amp;y=2002&amp;id=618</td>
<td>0.023</td>
</tr>
<tr>
<td>317</td>
<td>/cti/advising/login.asp</td>
<td>0.018</td>
</tr>
<tr>
<td>32</td>
<td>/advising/graduation.asp?section=advising</td>
<td>0.017</td>
</tr>
<tr>
<td>29</td>
<td>/advising/grad_scholarships.asp</td>
<td>0.016</td>
</tr>
<tr>
<td>355</td>
<td>/cti/gradassist/assistsubmit.asp</td>
<td>0.015</td>
</tr>
<tr>
<td>292</td>
<td>/cti/advising/display.asp?page=recommendations</td>
<td>0.014</td>
</tr>
<tr>
<td>30</td>
<td>/advising/grad_scholarships.asp?section=advising</td>
<td>0.013</td>
</tr>
<tr>
<td>156</td>
<td>/courses/syllabus.asp?course=353-97-905&amp;q=3&amp;y=2002&amp;id=307</td>
<td>0.012</td>
</tr>
<tr>
<td>134</td>
<td>/courses/syllabus.asp?course=321-21-602&amp;q=3&amp;y=2002&amp;id=800</td>
<td>0.012</td>
</tr>
</tbody>
</table>
Table 2
The five hidden task space discovered from CTI dataset (2)

<table>
<thead>
<tr>
<th>Task #4</th>
<th>URL</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>388</td>
<td>/news/default.asp</td>
<td>0.374</td>
</tr>
<tr>
<td>59</td>
<td>/calendar/calendar.asp</td>
<td>0.043</td>
</tr>
<tr>
<td>1</td>
<td>/admissions/</td>
<td>0.030</td>
</tr>
<tr>
<td>242</td>
<td>/courses/syllabus.asp?course=560-96-301&amp;q=3&amp;y=2002&amp;id=104</td>
<td>0.026</td>
</tr>
<tr>
<td>325</td>
<td>/cti/core/core.asp</td>
<td>0.025</td>
</tr>
<tr>
<td>386</td>
<td>/news/</td>
<td>0.024</td>
</tr>
<tr>
<td>668</td>
<td>/research/</td>
<td>0.018</td>
</tr>
<tr>
<td>579</td>
<td>/programs/2002/bachelors2002.asp</td>
<td>0.017</td>
</tr>
<tr>
<td>350</td>
<td>/cti/gradapp/logon.asp?opt=3</td>
<td>0.016</td>
</tr>
<tr>
<td>237</td>
<td>/courses/syllabus.asp?course=554-96-301&amp;q=3&amp;y=2002&amp;id=179</td>
<td>0.016</td>
</tr>
<tr>
<td>387</td>
<td>/news/campusdirections.asp</td>
<td>0.015</td>
</tr>
<tr>
<td>541</td>
<td>/people/facultyinfo.asp?id=653</td>
<td>0.015</td>
</tr>
<tr>
<td>256</td>
<td>/courses/syllabus.asp?course=582-97-302&amp;q=3&amp;y=2002&amp;id=190</td>
<td>0.014</td>
</tr>
<tr>
<td>448</td>
<td>/people/facultyinfo.asp?id=186</td>
<td>0.013</td>
</tr>
<tr>
<td>509</td>
<td>/people/facultyinfo.asp?id=306</td>
<td>0.012</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task #5</th>
<th>URL</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>394</td>
<td>/news/news.asp?theid=580</td>
<td>0.091</td>
</tr>
<tr>
<td>575</td>
<td>/programs/</td>
<td>0.055</td>
</tr>
<tr>
<td>4</td>
<td>/admissions/costs.asp</td>
<td>0.051</td>
</tr>
<tr>
<td>277</td>
<td>/cti/advising/display.asp?page=coursehistory</td>
<td>0.050</td>
</tr>
<tr>
<td>679</td>
<td>/resources/tutoring.asp</td>
<td>0.039</td>
</tr>
<tr>
<td>63</td>
<td>/courses/distancelearning.asp</td>
<td>0.035</td>
</tr>
<tr>
<td>388</td>
<td>/news/default.asp</td>
<td>0.029</td>
</tr>
<tr>
<td>401</td>
<td>/news/news.asp?theid=587</td>
<td>0.025</td>
</tr>
<tr>
<td>593</td>
<td>/programs/2002/gradtc2002.asp</td>
<td>0.024</td>
</tr>
<tr>
<td>308</td>
<td>/cti/advising/display.asp?tab=aptsfaculty&amp;page=browseapts</td>
<td>0.024</td>
</tr>
<tr>
<td>190</td>
<td>/courses/syllabus.asp?course=449-21-302&amp;q=3&amp;y=2002&amp;id=101</td>
<td>0.020</td>
</tr>
<tr>
<td>402</td>
<td>/news/news.asp?theid=588</td>
<td>0.018</td>
</tr>
<tr>
<td>437</td>
<td>/people/facultyinfo.asp?id=148</td>
<td>0.018</td>
</tr>
<tr>
<td>311</td>
<td>/cti/advising/display.asp?tab=home</td>
<td>0.018</td>
</tr>
<tr>
<td>220</td>
<td>/courses/syllabus.asp?course=511-98-303&amp;q=3&amp;y=2002&amp;id=131</td>
<td>0.017</td>
</tr>
</tbody>
</table>

ble, we find these co-occurrence observations are almost with respect to the educational and academic activities by students or academic staff, such as enrolling in course subjects, searching for faculty information, applying for scholarships and so on, due to the nature of the website functionality and coverage, from which the Web log data is collected. In particular, we can perceive that task #1 is about the activities that students use “mycti” (probably a portal entry of student service) to search for courses, faculties as well as scholarship information, which may represent a typical kind of navigational behaviors in student daily browsing. Task #2 reflects the specific interests of students in searching for course subject information, enrollment requirement and admission information in postgraduate programs in computing and telecommunication related disciplines. For example, international students are particularly interested in such kinds of programs. Compared to the above two, task #3 is more related to one specific category of the student navigational activities happened frequently, that is, browsing the information on assistantship or scholarship opportunities, application requirements and carrying out application procedures. These navigational trends were usually demonstrated along with browsing postgraduate programs and courses. Likewise task #5 is more concentrated on the various concerns about the course learning itself by students, such as course costs, tutor-
Fig. 7. Five specific user session-task preference distributions over the discovered tasks.

ing and course syllabus information. It is very common for students to know as much information as possible when they planned to start their studies. Different from the previously described tasks, it seems that task #4 has a broad and diverse spectrum of browsing interests, which ranges from course, program faculty information to admission, academic calendar, and campus directions. This may be caused by the fact of the existence of multiple subtasks in this category, which indicates that it is likely to separate this task into two or more sub-categories. This finding is reasonable and acceptable as we only choose the number of hidden tasks as five rather than a big number. The phenomenon is also often encountered in clustering.

Meanwhile, the inference of each user session’s association with multiple tasks \( \theta \) could be used to model a new target user’s navigational preference over the task space. Figure 7 depicts the navigational preference distributions of five user sessions over the five hidden tasks. The five user sessions are listed as follows:

\( S_1: \)
# 574- /programs/:2
# 604- /programs/courses.asp?deptcode=21 & deptmme=csc& courseid=211 :5
# 679- /search/newsearch.asp :4

\( S_2: \)
# 0- /admissions/:17
# 385- /news/:166
# 389- /news/news.asp:43
# 675- /resources/:749

\( S_3: \)
# 387- /news/default.asp:37
# 557- /people/search.asp?sort=ft:14
# 415- /people/:29
# 484- /people/facultyinfo.asp?id=269:10
# 577- /people/search.asp?sort=ft:14

\( S_5: \)
# 387- /news/default.asp:14
# 0- /admissions/:68
# 19- /admissions/requirements.asp:326
# 574- /programs/:6

For example, the fifth user session is corresponding to that user session illustrated in Fig. 5. From the column chart, we can conclude this user session is merely in a highly close relation to task #2, which is of interest in being enrollment in the program of telecommunication disciplines. This conclusion is also consistent and evidenced by the nature of the user’s navigational contents. More interestingly, in opposite to user #5, user #4 has exhibited a wide range of browsing preferences, exhibiting navigational interests over 4 tasks. On the other hand, with the calculated task distribution, the original sparse usage vector expressed by only three clicked pages is, therefore, expanded to the inclusion of distribution of five hidden tasks resulting in the improvement of user profiling.

5.5. Quantitative analysis

To conduct the quantitative analysis with the employed LDA-based approach, we employ the evaluation metric aforementioned to compute the recommendation performance. The results are shown in Fig. 8. In order to compare our approach with other existing methods, we also carry out experiments on the CTI dataset with other conventional clustering-based and pLSA-based approaches. In a similar manner, the usage-based session clusters by performing k-means clustering and probability inference with pLSA model [25,31] are constructed to aggregate user sessions with similar access preferences, and the centroids of clusters are derived as the aggregated users access patterns. Using the described user profiling approach, we then, could determine the most matched user profile, and generate the top \( N \) recommendations via the top-N ranking algorithm.

The results demonstrate that the employed LDA-based technique consistently outperforms the standard clustering-based and pLSA-based algorithms in terms of hit ratio parameter. From this comparison, it can be concluded that the employed approach is capable
of making Web recommendations more accurately and effectively against the conventional methods based on aggregating like-minded user sessions (i.e. user profiling). In addition to high recommendation performance, this approach is able to identify the hidden navigational tasks behind the usage data, which indicate why such user sessions or Web pages are grouped together in various categories.

6. Related work

Web recommendation research has become a hot research topic in the context of Web data management and information service in the last decade despite of the fact that recommender systems have been well studied in user interface and artificial intelligence areas. Recommender system is a specific type of information filtering techniques that aims to present the tailored information items that are likely of interest to users. To-date, there are two main kinds of Web recommender systems, namely content-based filtering and collaborative filtering systems. Content-based filtering systems such as WebWatcher [21] generally generate recommendations based on the pre-constructed user profiles by measuring the similarity of the Web contents to these profiles, while collaborative filtering systems make recommendations on information items they had not yet considered by referring to other users’ preferences that are closely similar to the current one [15]. The core component of recommender systems is user profiling, which is to build up different kinds of user profiles (or sometimes called user interest) for implementing information filtering [8,18,29,33].

In the context of Web applications, user profiling is often conducted by modeling user navigational behavior. As a result, Web usage mining has been proposed and employed as an efficient and practical means to fulfill such demands [25]. Particularly, with the great progresses in data mining and machine learning domains, many computational intelligence and data mining techniques, such as Neural Network [7], k-Nearest Neighbour (kNN) [14], Web user or page clustering [25], Markov Models [6] association rule mining [1] and sequential pattern mining techniques [2] have been extensively employed in Web recommender systems and substantially broaden and deepen the application background with satisfactory performance. For example, Liu et al. [23] proposed a framework for forming communities in a peer-to-peer communication environment by analyzing the client-side Web browsing history. This framework is based on an order statistics-based approach. [10] employed relevance feedback from users to deal with user modeling enhancing personalized news information delivery. In [5], Bose et al. incorporated the ontology in the form of concept hierarchy into usage based recommendation systems to reinforce the recommendation by taking semantics into recommendation, but obviously, the developed recommender system is heavily relying on the application domain. In [28] the authors proposed to combine the model-based and memory-based CF algorithms into a hybrid system to improve the recommendation performance without incurring high computational costs. And Mobasher et al. [25] introduced an aggregate user profiling approach for Web usage mining, which was used to make recommendations via collaborative filtering. [20] devised a Maximum Entropy algorithm into the recommendation scoring algorithm to achieve better recommendation performance. These techniques are both based on Web clustering on Web users or pages, however suffering from the problems of Clustering. In particular, in the event of Web usage mining, the usage data has the nature of extremely high dimensionality, thereby resulting in the computational difficulty in clustering and the lack of prompt response in real online applications. On the other hand, capturing the hidden navigational task space that governs user navigational activities is another interesting concern in the context of Web usage mining. To deal with this, using latent semantic analysis via statistical learning is emerging as an effective way recently. For example, [32] utilized semantic relations of terms to retrieve more relevant documents via a thesaurus. In [31] Xu et al. introduced the pLSA approach for Web us-
age mining, and proposed an algorithm of Web recommendation based on the user profiles derived from pLSA model. [22] presented an approach to mine the unseen navigational factors from Web logs for personalized Web search based on pLSA model. The major strengths of the family of latent semantic analysis is its capability of not only representing the high dimensional co-occurrence space in a low compact projective space with the maximum approximation of original space, but also interpreting the latent semantic task space [17].

7. Conclusion and future work

In this paper, we proposed a usage-based collaborative Web recommendation framework by incorporating Web user access patterns into collaborative Web recommendation based on LDA model. With the LDA model, the associations between user sessions and multiple tasks and the associations between tasks and Web pages hidden in the user click history are estimated via variational inference techniques. Interpreting the predominant Web pages with the significant contributive probabilities results in revealing the semantics of the underlying navigational task space, and examining the associations between user sessions and multiple tasks leads to discovering user navigational preference distributions over the task space, in turn, provides a better way for identifying the common user access patterns by aggregating user sessions with similar access preferences. Furthermore, inferring the task distributions upon the sparse and short inputs from target users makes it possible to accurately predict the navigational preferences of the new target users, and recommend the more preferable Web contents to the users. The main advantages of the employed LDA-based Web usage mining and Web recommendation over other existing approaches are (1) it can better handle the similarity calculation on sparse inputs by using task inference rather than the traditional similarity measures; (2) it can identify the navigational task distributions for the unseen usage data due to completeness of the generative process; (3) it is able to capture the usage-based access patterns and make collaborative recommendations. In this paper, we have also investigated the determination of the optimizing number of hidden tasks via an evaluation metric (i.e. modularity $Q$ function), which is to measure the strength of community structure. By selecting an optimal value of $Q$, we can quantitatively determine the hidden task number. The experiments on a real weblog dataset have shown that this approach can achieve a better recommendation performance in comparison to the existing techniques.

In future, we aim to combine the content information of Web pages into pattern learning procedure to reinforce semantic analysis. Moreover, we are also interested in exploring a new recommendation scoring schema in Web recommendation ranking.

References


