Expert Systems with Applications 37 (2010) 2911–2922

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Combination of Web page recommender systems

Murat Göksedef [1], Şule Gündüz-Öğüdücü *,[1]

Department of Computer Engineering, Istanbul Technical University, 34469 Istanbul, Turkey

ARTICLE INFO

Keywords:
Web usage mining
Web page recommendation
Hybrid recommender systems

ABSTRACT

With the rapid growth of the World Wide Web (www), finding useful information from the Internet has become a critical issue. Web recommender systems help users make decisions in this complex information space where the volume of information available to them is huge. Recently, a number of Web page recommender systems have been developed to extract the user behavior from the user's navigational path and predict the next request as s/he visits Web pages. However, each of these systems has its own merits and limitations. In this paper, we investigate a hybrid recommender system, which combines the results of several recommender techniques based on Web usage mining. We conduct a detailed comparative evaluation of how different combined methods and different recommendation techniques affect the prediction accuracy of the hybrid recommender. We then discuss the results in terms of using a hybrid recommender system instead of a single recommender model. Our results suggest that the hybrid recommender system is better in predicting the next request of a Web user.

1. Introduction

Most Web users complain about finding useful information on Web sites. Web page recommender systems predict the information needs of users and provide them with recommendations to facilitate their navigation. Given a user's current actions, the goal is to determine which Web pages will be accessed next. Many Web sites on Internet use Web page recommender systems to increase their usability and user satisfaction. Traditional methods for recommendation are based on Web usage and Web content mining techniques (Agrawal & Srikant, 1995; Cadez, Heckerman, Meek, Smyth, & White, 2003; Deshpande & Karypis, 2004; Nanopoulos, Katsaros, & Manolopoulos, 2001; Sarukkai, 2000).

In recent years, there has been an increasing interest in applying Web content mining techniques to build Web recommender systems. However, the Web content mining techniques are unable to handle constantly changing Web sites, such as news sites, and dynamically created Web pages. Thus, using Web content mining techniques in a recommender model leads to update the model frequently. For this reason, in this work our aim is to derive a predictive model for Web pages which is based on Web usage mining techniques.

The performance of a recommender model depends on the structure of the Web site besides the specific technique that it uses. Furthermore, different users may have different navigation strategies. Thus, it could be difficult to estimate a single best model for recommendation. Each of the single methods has its advantages, but also limitations and disadvantages. Therefore, combining different methods to overcome disadvantages and limitations of a single method may improve the performance of recommenders. Hybrid recommender systems combine two or more techniques to improve recommender performance. Burke (2002) proposed hybrid recommenders for collaborative filtering that combine information across different sources. To date, most of the research on hybrid recommenders is on collaborative filtering approaches such as combining these approaches with content based approaches rather than combining multiple recommender techniques. For this reason, we especially focused in this paper on how the recommendation accuracy may be improved by combining different recommendation models based on the Web usage mining techniques.

Web page recommendation systems have been extensively explored in Web usage mining. However, the quality of recommendations and the user satisfaction with such systems are still not optimal. Recommender systems based on Web usage mining techniques have also strengths and weaknesses (Kazienko & Kolodziejski, 2006). Since different methods focus on different characteristics of Web users, they produce different prediction accuracies on the same data set. Combination of different methods may result in better accuracy (Burke, 2002). For example, consider two recommender models, model *A* and model *B*, applied to a Web site to predict three pages one of which could be the next request of a Web user surfing on this site. Both models use different methods to predict the next request of a user. Let two users, user 1 and user 2, are requesting p_1 and p_4, respectively, where p_i is a Web page on the Web site. Model *A* generates the same

* Corresponding author. Tel./fax: +90 212 285 3597.
E-mail addresses: goksedef@itu.edu.tr (M. Göksedef), sgunduz@itu.edu.tr (Ş. Gündüz-Öğüdücü).

[1] The authors are given in alphabetical order.

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.09.046

recommendation set, $RS_A = \{p_1, p_2, p_3\}$, for both users, where model *B* generates $RS_B = \{p_2, p_4, p_5\}$ for the same users. In this case, model *A* is successful for user 1 and model *B* is successful for user 2. However, if it is possible to combine these two models into one model to generate a recommendation set, for example $RS = \{p_1, p_4, p_2\}$, this combined model can generate correct recommendations for both users.

In this study, we analyze the performance of hybrid recommender models by combining the results of different recommender techniques using four different hybridization (combination) methods. We especially focus on answering the questions of whether using hybrid approaches increases the recommendation accuracy and what types of combination methods are likely to be successful in predicting the next request of users. We use the structure information of the Web site for ranking the pages in the recommendation set. For this purpose, we implement and compare different hybrid recommenders including some novel combinations. The different recommender techniques implemented in this work are called the modules of the hybrid recommender model. We conduct a detailed comparative evaluation of different recommender techniques which can be used as the modules of the hybrid recommender system. For this purpose, four recommender techniques which use different data mining approaches based on different characteristics of user sessions are implemented (Cadez et al., 2003; Gündüz & Özsu, 2003; Mobasher, Dai, Luo, & Nakagawa, 2001, 2002a). We combine the modules by using four different combination methods. We propose a new combination method using Google's PageRank (Brin & Page, 1998) and HITS algorithms (Kleinberg, 1999) that make use of the topological structure of the Web sites. We also examine whether it is worthwhile to use a hybrid recommender instead of a single recommender model.

The hybrid recommender is implemented as an experimental system and its performance is evaluated based on the correct prediction of the next request of a user, namely Hit-Ratio. Our detailed experimental results show that when choosing appropriate combination methods and modules, hybrid approaches achieve a better prediction accuracy.

The rest of the paper is organized as follows: In Section 2, we introduce the related work. Section 3 describes the architecture of the experimental system designed for predicting the next request of users by a hybrid recommender. In Section 4, we introduce the distinct techniques which are integrated to build a hybrid recommendation model. The methods for combining different recommender techniques are explicated in Section 5. Section 6 provides detailed experimental results. Finally, in Section 7 we conclude and discuss future work.

2. Related work

Various Web usage mining techniques have been used to develop efficient and effective recommendation systems. One of the most successful and widely used technologies for building recommendation systems is Collaborative Filtering (CF) (Resnick & Varian, 1997). CF techniques predict the utility of items of an active user by matching, in real-time, the active user's preferences against similar records (nearest neighbors) obtained by the system over time from other users (Breese, Heckerman, & Kadie, 1998). A shortcoming of these approaches is that it becomes hard to maintain the prediction accuracy in a reasonable range while handling the large number of items in order to decrease the on-line prediction cost. Some hybrid approaches are proposed to handle these problems, which combine aspects of both pattern discovery methods and CF. O'Conner and Herlocker (1999) use existing data partitioning and clustering algorithms to partition the set of items based on user rating data. Predictions of items are then computed independently within each partition.

Another example to hybrid approaches is content-based systems, which work by comparing text descriptions or other representations associated with an item. Balabonović and Shoham (1995) describe a system that helps users to discover new and interesting sites that are of interest to them. As an alternative to the methods discussed above, the system proposed by Pazzani (1999) is designed to create a framework. In this work three major recommendation approaches including content based (using the product attributes), demographic filtering (using the customer attributes) and CF techniques are combined. Furthermore, hybrid approaches, that utilize content information, are proposed by Burke (2002) to overcome some of the shortcomings of CF such as the cold start problems for new users and new products. In this work six different hybridization methods are surveyed and implemented.

There have been attempts to combine content, structure and usage information to generate hybrid systems. One recent work in this area is Li and Zaïane's work (Li & Zaïane, 2004). They propose an algorithm that combines and utilizes of usage, content and structure data. In Nakagawa and Mobasher (2003), a recommender system is proposed that adopts a clustering technique to obtain both the site usage and site content profiles. In this work, the authors use association rule mining and sequential pattern mining to generate navigational patterns of Web users. A switching (Burke, 2002) hybridization method is used to integrate the navigational patterns of Web users in order to generate a recommendation set. In Göksedef and Gündüz-Öğüdücü (2007) a recommendation model, which is called consensus recommender, is developed. In the study, several recommendation models based on Web usage mining techniques are integrated. Results of the study show that consensus model achieves a better prediction accuracy compared to it's individual components.

3. System design

In this work, we have designed an experimental system to assist in our examination of whether combining multiple recommender techniques increases the prediction accuracy of a recommendation system. Fig. 1 depicts the overall process that we consider in our system. As with most recommender systems, ours is composed of two parts: an off-line part and an on-line part. The off-line part has two components, namely data preparation and pattern extraction. The on-line part also consists of two components: recommendation engine and hybridization.

3.1. Off-line part

3.1.1. Data preparation

The task of this component is to prepare the raw Web access log data for mining the usage patterns. Generally, several preparation steps need to be performed. For our work, these include data cleaning, user identification and session identification. These preprocessing steps are the same for any Web usage mining problem and fundamental methods of them have been well studied in Cooley, Mobasher, and Srivastava (1999) Srivastava, Cooley, Deshpande, and Tan (2000). In the data cleaning step, first the irrelevant log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG are eliminated. Next, the URLs in the log file are normalized in order to determine same Web pages which are represented by syntactically different URLs. A common form for each page is chosen using a Web crawler (Mohr, Kimpton, Stack, & Ranitovic, 2004). Only links that point to the Web pages within the site are added to the list of pages to explore. Comparing the content of pages pro-

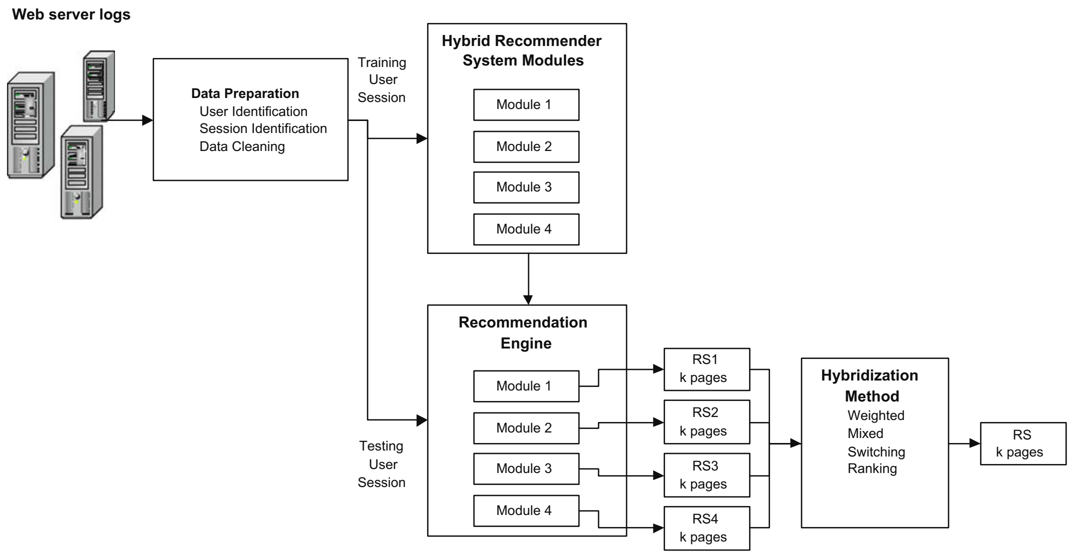

Fig. 1. Hybrid recommender system.

vides a way to determine different URLs belonging to the same Web page. The *visiting page time*, which is defined as the time difference between consecutive page requests, is calculated for each page.

In our study, we perform the user and session identification steps in parallel. Since cookies are not available in our Web logs, we use a heuristic method which identifies an unique IP as an user, bearing in mind that a single IP can be used by a group of users. A new session is created when a new IP-address is encountered or if the visiting page time exceeds 30 min for the same IP-address. For the last page of the user session, we set the visiting page time to be the mean of the times for that page taken across all sessions in which it is not the last page request. Thus, a session consists of ordered sequence of page visits. As in other studies (Mobasher et al., 2001, 2002a) short user sessions are removed. For this study, we eliminate sessions whose *session length*[2] is less than or equal to 2 in order to eliminate the effect of random accesses to the Web site. It is important to note that these are only heuristics to identify users and user sessions, and other heuristics may be employed in future studies.

The visiting times are normalized across the visiting times of the pages in the same session as proposed in Mobasher et al. (2001). Eventually, this component produces a set of URL's $P = \{p_1, \ldots, p_n\}$ and a set of user sessions $S = \{s_1, s_2, \ldots, s_{|S|}\}$ where $|S|$ is the number of sessions in S. Each module of the hybrid recommender may represent the user session in a different form. In general a user session s_i of a length m is a sequence:

$$s_i = \left\langle \left(p_1^i, t_{p_1^i}\right), \left(p_2^i, t_{p_2^i}\right), \ldots, \left(p_m^i, t_{p_m^i}\right) \right\rangle \quad (1)$$

where $p_j^i = p_l$ for some $l \in \{1, \ldots, n\}$ and $t_{p_j^i}$ is its corresponding normalized visiting time in session s_i.

3.1.2. Pattern extraction

The pattern extraction component consists of modules each of which is a recommender using a different technique. These modules are discussed further in the next Section. The recommenders are first trained using the extracted session information. During regular recommendation runs, each module, ($module_i$, $i = 1, \ldots, 4$), extracts usage patterns from the training user sessions. These patterns will be further used in the recommendation engine.

3.2. On-line part

3.2.1. Recommendation engine and hybridization

Given a user's current request, each module generates, in parallel, a recommendation set (RS_i) consisting of k pages, that it thinks the user will visit as the next page in his or her session. These individual recommendation sets are aggregated using the same hybridization method to generate a final recommendation set (RS) with k pages as well:

$$RS = RS_1 \circ RS_2 \circ RS_3 \circ RS_4 \quad (2)$$

where "$\circ$" denotes the combination method. The different methods are discussed in Section 5 in detail.

4. Hybrid recommender system modules

In this section, we briefly present the distinct recommender techniques that are included as modules in our hybrid recommender system. Four recommender techniques that use different data mining approaches based on different characteristics of user sessions are implemented. The first technique (Mobasher et al., 2002a) is based on clusters of pages found from the server logs for a site. This model uses the time information without considering the visiting order of the pages. The second technique (Gündüz & Özsu, 2003) uses the order information of pages in a session. A new similarity measure is proposed in (Gündüz & Özsu, 2003) to calculate pairwise similarities between user sessions. The user sessions are then clustered using a graph-based clustering algorithm. The third technique (Cadez et al., 2003) clusters user sessions by learning a mixture of first order Markov models using the Expectation-Maximization algorithm. The last recommender system is based on association rule discovery among user sessions (Mobasher et al., 2001). The reason of choosing these recommender techniques as the modules of the hybrid is that their high recommendation accuracy have been reported in many papers and that many other recommender systems are based on these models (Bose, Beemanapalli, Srivastava, & Sahar, 2006; Eirinaki, Vazirgiannis, & Kapogiannis, 2005; Mobasher, Dai, Luo, & Nakagawa, 2002b). Furthermore, they can be classified into three groups

[2] The length of a session is determined by the number of pages requested by one user within a server session.

according to the data structure they use for representing user sessions:

1. Those that represent user sessions using only the time that a user spends on each page during his or her visit (Mobasher et al., 2002a);
2. Those that represent user sessions by using only the visiting order of the Web pages (Cadez et al., 2003; Gündüz & Özsu, 2003);
3. Those that represent user sessions by association rules which capture the relationships among pages based on their patterns of co-occurrence across user sessions (Mobasher et al., 2001).

All of those recommender models are based on Web usage mining techniques. Since they use different data mining approaches for modeling user behavior they may complement each other's shortcomings. Each of these techniques is implemented as a module of the hybrid recommender model and they are described in the following subsections.

4.1. Recommender model based on clustering user sessions

The model proposed in (Mobasher et al., 2002a) (k-means-Model) clusters user sessions according to the visiting time of pages. After the data preparation, a user session s_i is represented by an n-dimensional vector of visited pages over the space of page references:

$$\overrightarrow{s_i} = \langle w(p_1, s_i), \ldots, w(p_k, s_i), \ldots, w(p_n, s_i) \rangle \tag{3}$$

where n is the total number of unique pages in the Web site and $w(p_k, s_i) = t_{p_j^i}$ for some $j \in \{1, \ldots, n\}$ if p_k appears in session s_i and 0 otherwise.

The session data is clustered by a simple k-means algorithm (MacQueen, 1967) based on vector distances. The usage pattern for each cluster is represented by the center of that cluster. The center of a cluster c_t can be computed easily by calculating the mean vectors of the sessions assigned to the cluster:

$$\overrightarrow{\mu_t} = \langle w(p_1), w(p_2), \ldots, w(p_n) \rangle$$

where $w(p_j)$ is given by

$$w(p_j) = \frac{1}{|c_t|} \cdot \sum_{s_i \in c_t} w(p_j, s_i)$$

In the recommendation step, a similarity value is calculated between each cluster center $\overrightarrow{\mu_t}$ and the active user session $\overrightarrow{s_a}$ using the cosine similarity metric. The cluster with the highest similarity value, $sim(\overrightarrow{s_a}, \overrightarrow{\mu_t})$, is selected as the best matching cluster. To recommend pages, the recommendation algorithm uses the center vector of the best matching cluster. A recommendation score is calculated by multiplying each weight in the cluster center vector by the similarity value of that cluster. The recommendation score of a page $p_i \in P$ is calculated as follows:

$$rec(\overrightarrow{s_a}, p_i) = \sqrt{w(p_i) \times sim(\overrightarrow{s_a}, \overrightarrow{\mu_t})}$$

The first k pages with the highest recommendation score are added to the individual recommendation set of this model, (RS_i).

4.2. Recommender model based on click-stream tree

In the click-stream tree model (CST-Model), the recommendations are generated using the recommender technique proposed in (Gündüz & Özsu, 2003). In this technique a user session in the data set is represented as a sequence of pages visited in that session:

$$s_i = p_1^i p_2^i \cdots p_m^i \tag{4}$$

The pairwise similarities between user sessions are calculated using a sequence alignment method (Charter, Schaeffer, & Szafron, 2000). Using pairwise similarity values, a graph is constructed whose vertices are user sessions. An edge connecting two vertices in the graph has a weight equal to the similarity between these two user sessions. Using an efficient graph-based clustering algorithm (Demir, Uyar, & Gündüz-Ögüdücü, 2007) the user sessions are clustered, and each cluster is then represented by a click-stream tree (CST) whose nodes are pages of user sessions of that cluster. Each user session in a cluster is a branch of the corresponding CST. Each CST has a *root* node, which is labeled as "null". Each node except the *root* node consists of three fields: *data*, *count* and *next_node*. *Data* field consists of page information. *Count* field registers the number of sessions represented by the portion of the path arriving at that node. *Next_node* links to the next node in the CST that has the same *data* field or null if there is any node with the same *data* field. Each CST has a *data_table*, which consists of two fields: *data* field and *first_node* that links to the first node in the CST with the same *data* field. The children of each node in the click-stream tree is ordered in the count-descending order such that a child node with bigger count is closer to its parent node. The resulting click-stream trees are then used for recommendation.

When a request is received from an active user, a recommendation set is produced consisting of k different pages that the user has not yet visited using the best matching user session.[3] For the first two requests of an active user session, all clusters are explored to find the one that best matches the active user session. After the second request of the active user, top-N clusters that have higher recommendation scores among other clusters are selected for producing further recommendation sets. For the remaining requests, the best matching user session is found by exploring the top-N clusters that have the highest N similarity values computed using the first two requests of the active user session. The rest of the recommendations for the same active user session are made by using the top-N clusters.

4.3. Recommender model based on Markov model

In this model (Markov-Model) (Cadez et al., 2003), the user sessions are represented as sequences of Web pages as in Eq. (4) and are partitioned into clusters according to the order of Web pages in each session. A model based clustering approach is employed to cluster user sessions. In particular, the user sessions are clustered by learning a mixture of first order Markov models using a standard learning technique, the Expectation-Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). Each cluster has a different Markov model which consists of a (sparse) matrix of state transition probabilities, and the initial state probability vector. The proportion of user sessions assigned to each cluster as well as the parameters of each Markov model is learned using EM algorithm. The user sessions are modeled as follows: (1) a user session is assigned to a particular cluster with some probability, (2) the order of Web pages requested in that session is generated from a Markov model with parameters specific to that cluster.

The parameters of the Markov-Model consists of: (1) the probabilities of assigning user sessions to various clusters ($p(c_g)$ where c_g is the gth cluster), (2) the parameters of each cluster. The parameters of each cluster are composed of a set of states called state space, initial state probabilities, and transition probabilities t_{ij} between two adjacent states x_i and x_j (probability of moving from x_i to x_j). In our case, the state space of the Markov model is the

[3] The user session that has the highest similarity to the active user session is defined as the best matching session.

set of pages that make up the Web site. A transition probability t_{ij} between state x_i and state x_j corresponds to the probability of visiting page p_j after visiting page $p_i(p(p_j|p_i))$. The Markov-Model of our recommender system uses a first order Markov model and predicts the probability of the next action of a user by only considering the user's previous action. Let $s_i = (p^i_1, p^i_2, \dots, p^i_m)$ be user session of length m. The Markov-Model assumes that the user session s_i is being generated by a mixture of Markov models as follows:

$$
\begin{aligned}
p(s_i) &= \sum_{g=1}^{G} p(s_i|c_g)p(c_g) \\
p(s_i|c_g) &= p(p^i_1|c_g)\prod_{j=2}^{m} p(p^i_j|p^i_{j-1}, c_g) \qquad (5) \\
p(c_g|s_i) &= \frac{p(s_i|c_g)p(c_g)}{\sum_j p(s_i|c_j)p(c_j)}
\end{aligned}
$$

where G is the number of clusters, and $p(p^i_1|c_g)$ is the initial state probability of the gth cluster for page p^i_1.

Since the study in Cadez et al. (2003) focuses on visualization of navigational patterns rather than on predicting the next request of Web users, the proposed model does not have a recommendation part. For this reason, we developed in Göksedef and Gündüz-Ögüdücü (2007) a recommendation engine for the Markov-Model. A recommendation set consisting of k pages is generated from Markov-Model as follows: the active user session is assigned to one of the clusters (c_a) that has the highest probability calculated using Eq. (5). The recommendation set for the current user is generated using the transition matrix of c_a. All the transition entries t_{ij} of c_a are sorted in descending order, where state x_i is equal to the last visited page in the active user session. The top k pages are selected to get a recommendation set from the Markov-Model.

4.4. Recommender model based on association rule discovery

The work in Mobasher et al. (2001) presents a recommender technique based on the association rule mining method (AR-Model). In the work, the user sessions are represented as follows:

$$s_i = \langle w(p_1, s_i), \dots, w(p_k, s_i), \dots, w(p_n, s_i)\rangle \qquad (6)$$

where the feature weights $w(p_j, s_i)$ $j = 1, 2, 3, \dots, n$ of the session vector s_i will be binary values: 0 if page p_j is not visited during the session, 1 otherwise.

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions. In the case of Web sessions, association rules capture relationships among visited pages. In this study, we use the Apriori algorithm (Agrawal & Srikant, 1994) to find groups of pages occurring frequently together in many user sessions.

The recommendation engine uses the resulting frequent pages to make a recommendation according to the user's actions. A fixed-size sliding window over the current active session is used to capture the current user's behavior. For example, if the current session (with a window size of 3) is $\langle p_1, p_2, p_3\rangle$, and the user references the pageview p_4, then the new active session becomes $\langle p_2, p_3, p_4\rangle$. The recommendation engine matches the current user session window with frequent pages to find candidate pageviews for giving recommendations. Given an active session window w, all frequent itemsets of size $|w| + 1$ which contain the active user session window are considered. The recommendation score of each candidate pageview is calculated using the confidence value of the corresponding association rule whose consequent is the singleton containing the pageview to be recommended. The k pages with the highest recommendation scores are added to the recommendation set of this module.

5. Methods for combining recommendations

The aim of a hybrid recommender system is to combine multiple recommender techniques or to combine multiple results of different recommender models together to produce a single output. In Burke (2002) six different types of hybridization methods are introduced. However, some of the combination methods described in that paper are appropriate only for combining information from different domains such as content and usage data. Besides, some methods are designed for use in collaborative filtering techniques where ratings for items are available. For this reason, we implemented only three of them with novel features, namely *weighted*, *mixed*, *switching*, and a new method that we propose: *ranking*. This section describes the combination methods used in this paper.

5.1. Weighted

As stated in Burke (2002) a weighted Web recommender is the simplest design of hybrid recommenders in which each component of the hybrid scores a given item based on the item ratings or weights. After each component generates a candidate set, the union or the intersect of the candidate sets is used as the final candidate set. Each component scores the items in the final candidate set and the linear combination of two scores computed becomes the item's prediction rate.

In our case the items correspond to the Web pages on a Web site $(P = \{p_1, p_2, \dots, p_n\})$. Among the modules of the hybrid, only the recommender model based on clustering user sessions (Mobasher et al., 2002a) and the recommender model based on association rule discovery (Mobasher et al., 2001) are able to score a given Web page as a candidate. Therefore it is not possible to use the weighted hybrid as proposed. For this reason, we modify the weighted hybrid as follows. Each module of the hybrid generates a recommendation set consisting of k different pages. The pages in each recommendation set is ordered such that an individual recommender thinks that the first page in the recommendation set is most likely accessed next. Thus each page in each recommendation set is weighted by a score based on its rank in the recommendation set. Each individual recommendation set is defined by a tuple:

$$RS_i = (PAGES_i, WEIGHTS_i)$$

where $PAGES_i$ is a subset of P that the $module_i$ generates as a recommendation set and $WEIGHTS_i$ is the associated weights of pages in $PAGES_i$:

$$
\begin{aligned}
PAGES_i &= \{p^i_1, \dots, p^i_k\} \\
WEIGHTS_i &= \{weight_{p^i_1}, \dots, weight_{p^i_k}\}
\end{aligned}
$$

where $p^i_j = p_l$ for some $l \in \{1, \dots, n\}$ and $weight_{p_j}$ is the weight of page p_j if $p_j \in PAGES_i$. The hybrid recommender generates then the combined recommendation set $(CRS = (PAGES, WEIGHTS))$ where $PAGES$ is the union of $PAGES_i$ for $i = \{1, \dots, 4\}$ and $WEIGHTS$ is the combined recommendation scores. The combined recommendation score of a page p_i is calculated by summing up the weights of this page in each recommendation set. The pages in CRS are then sorted by the combined recommendation score and the top k pages shown to the user as the final recommendation set RS.

5.2. Mixed

A mixed hybrid presents recommendations of its different modules side-by-side in a combined list (Burke, 2002). However, the challenge of these types of hybrids is the integration of ranked pages in each recommendation set into a final recommendation set. This can be accomplished as follows (Göksedef & Gündüz-Ögüdücü, 2007). Initially, equal weights (m_i) are assigned to each module of the hybrid assuming that each module generates equally accurate recommendations at the beginning. Each module of the hybrid generates a recommendation set consisting of k pages

$$RS_i = \{p_1^i, p_2^i, \ldots, p_k^i\} \tag{7}$$

where $p_j^i = p_l$ for some $l \in \{1, \ldots, n\}$. Then the individual recommendation sets are combined to get a final recommendation set, which also consists of k pages, as follows:

$$RS = \{p_i \mid p_i \in RS_1 \text{ and } i = 0, \ldots, m_1\} \cup \cdots \cup \{p_j \mid p_j \in RS_4 \text{ and } j = 0, \ldots, m_4\}$$

where $m_1, \ldots, m_4 \in \{0, 1, 2, 3\} \wedge \sum_i m_i = k$ and $p_0 = \emptyset$. Note that each module has m_i pages in the final recommendation set and at least two of the modules contribute to the final recommendation set. After each recommendation, the weights of the modules are updated by evaluating their performance. Three different methods are proposed to update these weights (Göksedef & Gündüz-Öğüdücü, 2007):

Method 1. Find best and worst modules according to their HitRatio for the last user session.
Method 2. Find best and worst modules according to their HitRatio for the first two pages in the active user session.
Method 3. Find best and worst modules according to their HitRatio until the current user session.

In all of the three methods the weight of the best module is increased whereas the weight of the worst module is decreased, considering minimum and maximum weight values of the modules.

5.3. Switching

The switching hybrid selects one of its modules as appropriate in the current situation based on a switching criterion. The idea is that the modules may have not consistent performance for all types of users. These hybrid requires a reliable switching criterion based on either the performance of the individual recommenders or some alternative measure. For the switching hybrid, each of the modules generates its individual recommendation set as in Eq. (7). Then, the switching hybrid selects one of the individual recommendation sets as the final recommendation set, namely $RS = RS_i$, based on its switching criterion.

In this paper, we use tree switching criteria based on the performance of the modules:

Criterion 1. Based on the length of user sessions.
Criterion 2. Based on the average confidence value which is calculated by dividing the confidence of all association rules having the same set of pages at the left-hand-side over the number of these rules extracted from a predefined portion of training user sessions.
Criterion 3. Based on the confidence values of association rules extracted from a predefined portion of training user sessions.

The details of these criteria are given in Section 6.4. The individual recommendation set of one of the modules that performs better according the selected criterion is set as the final recommendation set.

5.4. Ranking

The ranking hybrid first combines the individual recommendation sets of its modules into one recommendation set and then applies a ranking method to sort the pages in this set. We propose in this paper three ranking methods using two different techniques that are inspired by two algorithms based on the connectivity of Web pages, namely PageRank (Brin & Page, 1998) and HITS (Hyperlink Induced Topic Search) (Kleinberg, 1999). First, each of the modules of the hybrid generates a recommendation set as in Eq. (7). The combined recommendation set is obtained by the union of the individual recommendation sets:

$$CRS = \bigcup_{i=1}^{4} RS_i \tag{8}$$

Then, the pages are ranked according to the scores computed using the three methods described in this subsection. The first k pages are recommended to the active user as the final recommendation set RS. For all of the three methods, the scores of the pages are calculated completely off-line, thus the methods for ranking do not cause any runtime overhead.

Method 1. The first method for ranking pages is based on the original PageRank algorithm (Brin & Page, 1998). This is a link analysis algorithm that assigns a numerical weight to each element of a hyperlinked set of documents. In order to rank the pages in the combined recommendation set CRS, we first calculate the PageRank scores of the pages in this set. The first k pages with highest PageRank scores are selected as the final recommendation set.

Method 2. The second ranking method is based on the users' navigational behavior instead of the structure of the Web site. In this method, a modified PageRank score is assigned to each page on the Web site that reflects its *popularity*. The score of each page is influenced by the total number of visits of that page. We modified the original PageRank algorithm as follows. The Web logs induce a directed graph G with nodes representing Web pages in the Web site and arcs between Web pages p_i to p_j whenever there is a visit in the log from page p_i to p_j. The weights of the arcs are set to 1. Then the modified PageRank score of each page is computed on this graph using the PageRank algorithm. These scores are ranked in descending order and top k pages with the highest scores are selected as the final recommendation set. The details of this method is given in Göksedef and Gündüz-Öğüdücü (2008).

Method 3. The last combination method for ranking pages is based on the HITS algorithm (Kleinberg, 1999) developed by Jon Kleinberg. In HITS, a query is used to select a subgraph from the Web. In our method the subgraph is constructed only from the Web site for which we want to generate recommendations. From this graph, two kinds of nodes are identified: authoritative pages to which many pages link, and hub pages that consist of links to valuable pages on the subject. Thus every page p_i has two distinct measures of merit, its hub score $h[p_i]$ and its authority score $a[p_i]$. In this work, authority scores and hub scores of each Web page on the Web site are calculated using the structure of the Web site. The pages that have high authority scores are the prominent sources of the primary content of the Web site. Since the main intent of a recommender system is to guide users to the pages that contain a portion of the information content that is related to the interest of the user at that time, only the authority scores are used to rank the pages in the recommendation set. The final recommendation set consists of k pages that have higher authority scores among the other candidates in CRS.

6. Performance evaluation

6.1. Data sets

We report our experiments conducted on four different data sets. The first data set is from the NASA Kennedy Space Center (NASA) server[4] (with 92 pages and 15,359 user sessions). This data

[4] http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html.

set is used, because it is a well-known public data set which has been used in other studies as well. The second server log (BIDB) is from the Web site of Computer Center of the Istanbul Technical University (ITU).[5] This Web site contains information about the Computer Center of ITU and technical information about computer systems. This server log is collected over the months of October 2006–April 2007. After the preprocessing steps the BIDB data set consists of 1238 user sessions with 323 different pages. The third server log (CE) is from the Department of Computer Engineering of the Istanbul Technical University.[6] This server log is obtained from the access entries of 2-month period between February and April, 2007. It consists of 3484 user sessions with 500 different pages. The last server log (TISC) is collected in March 2007 by a Internet Service Company. This data set contains 9833 user sessions with 1908 different pages. Approximately 30% of the cleaned sessions extracted from each data set are randomly selected as the test set, and the remaining part as the training set. In order to conduct comparative empirical studies of the relative performance of different modules of the hybrid recommender we perform the following analysis of the data sets used to train and test: we calculate the pairwise similarities $sim(s_i, s_j)$ as proposed in Gündüz and Özsu (2003), where s_i and s_j are user sessions in the test and training sets, respectively. We then take the maximum similarity value for every user session in the test set. The frequencies of every different similarity values are given in Fig. 2.

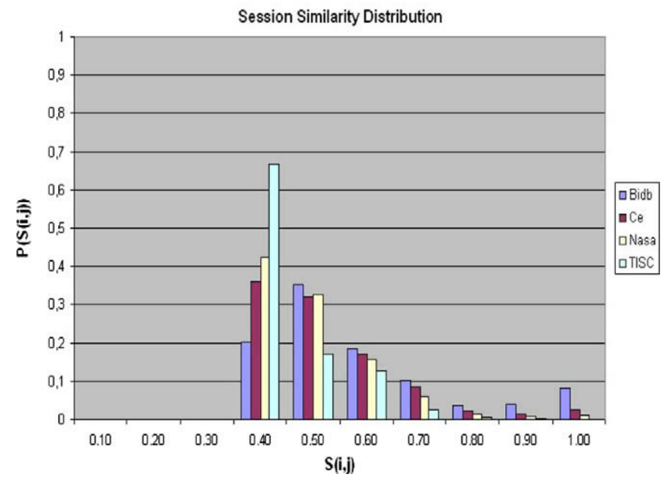

Fig. 2. Pairwise session similarity distribution.

6.2. Experimental setup

Given the data partitioned as described above, we carried out the following empirical evaluation; the training set is used for building our model. The number of pages in the final recommendation set, k, is set to four. The reason for this setting is that our detailed experiments show that the probability of visiting the fourth page of the recommendation set is less than 3% for the hybrids. For each session $s_i = \left(p_1^i, \ldots, p_j^i, p_{j+1}^i, \ldots, p_m^i\right)$ in the test set, we take each page request p_j^i and our model generates a recommendation list ($RS(s_{ij})$) consisting of four pages using the prefix of s_i that is defined as $H(s_{ij}) = \left(p_1^i, \ldots, p_j^i\right)$. To measure the recommendation accuracy we use Hit-Ratio (HR) (Jin, Mobasher, & Zhou, 2005) as evaluation metric. Hit-Ratio is defined as follows: A hit is declared if any one of the four recommended pages is the next request (p_{j+1}^i) of the user. The Hit-Ratio is the number of hits divided by the total number of recommendations made by the system.

We conducted the experiments with a single CST for BIDB and CE data sets without clustering user sessions. Although using a single CST extends the time period required to generate the recommendation set, it is acceptable for those data sets that have a small number of sessions. The results obtained by using a single CST gives us the upper bound of the prediction accuracy of the CST-Model (Gündüz & Özsu, 2003). In that case we do not have any side effects of the clustering algorithm or of the assumptions we made for assigning the active user session to a cluster, since the entire tree is searched. For the NASA data set, the number of clusters is chosen to be five, as proposed in Gündüz and Özsu (2003).

To determine the parameters of the Markov-Model and k-means-Model, we conducted the experiments on the training sets with different number of clusters and choose the one with highest Hit-Ratio. We perform ten runs (each with different initial parameter settings for the EM algorithm and each with different initial cluster centers for k-means algorithm) for each different number of clusters and report the results with the highest Hit-Ratio.

6.3. Effects of using different recommendation techniques

In this subsection, we investigate the effects of using different recommender techniques as the modules of the hybrid recommender. For this purpose, four different recommender techniques are implemented: k-means-Model (KM in tables) (Mobasher et al., 2002a), CST-Model (CST in tables) (Gündüz & Özsu, 2003), Markov-Model (MM in tables) (Cadez et al., 2003), AR-Model (AM in tables) (Mobasher et al., 2001) as explained in Section 4. A set of experiments are conducted with all the combinations of these techniques. We only present the results of the experiments in which the Hit-Ratio of the hybrid recommender is not 30% smaller than one of its individual modules.

Table 1 shows the Hit-Ratio results of the NASA, BIDB, CE and TISC data sets. The results for the hybrid recommender are given for the weighted combination method since it outperforms the other types of hybridization methods when using k-means-Model and AR-Model as one of the modules. Although k-means-Model and AR-Model generate recommendations based on different aspects, they could not increase the recommendation accuracy of the hybrid recommender. The reason for that could be that recommender techniques that consider the order of visiting pages have a better performance compared with the other models that represent user sessions in a different way (e.g., time spent on page or co-occurred pages) (Demir, Goksedef, & Uyar, 2007). Our further experiments confirm that, as expected, the recommendation performance of the modules have direct influence on the performance of the hybrid system – a low performance technique that is used as a module reduces the total performance. Therefore, we report only the results obtained by the combination of the better performing two modules, namely CST-Model and Markov-Model.

6.4. Determination of switching criterion

We conducted further experiments to investigate the performance of various modules over the different data sets whose results in turn would be helpful in deciding the switching criterion (Section 5.3). A validation set is extracted from the training data to find a suitable criterion for switching. We experiment by varying the size of the validation set. The recommendation models are trained on the remaining training data and evaluated on the validation set for determining a switching criterion. Once this is done, the recommender models are trained again using the entire training data.

[5] http://www.bidb.itu.edu.tr/.
[6] http://www.ce.itu.edu.tr/.

Table 1
Hit-Ratio in % of the data sets.

Data set	AM	KM	MM	CST	KM + MM	KM + CST	KM + AM	AM + MM	AM + CST	KM + AM + CST	KM + AM + MM	CST + MM
NASA	45	2	65	65	54	53	33	55	56	58	55	**69**
BIDB	42	11	78	69	75	67	38	69	62	58	64	**79**
CE	33	1	43	48	41	42	29	43	45	42	41	**51**
TISC	8	2	20	17	19	17	07	19	13	19	16	**21**

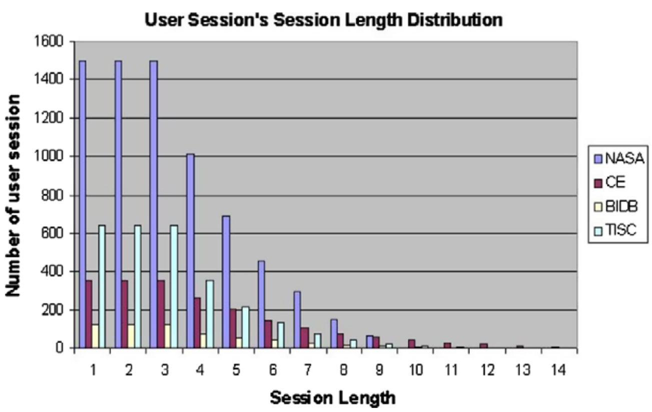

Fig. 3. Session length histogram for each dataset.

We first investigate the impact of the user session characteristics on the modules of the hybrid recommender. Fig. 3 depicts the histogram of the occurrence of each of the possible session lengths for each test data set used in our experiments. To generate the histogram, for a user session with 4 pages, we increase the number of the user sessions which have session lengths 1, 2, 3 and 4 by 1. As can be seen from the figure, the user characteristics of all of the Web sites reflect relatively short sessions. Fig. 4 shows the change of the Hit-Ratio of the CST-Model and Markov-Model with the session length. These results show that CST-Model has a slightly better Hit-Ratio for shorter sessions for the NASA data set. However, Markov-Model outperforms CST-Model for BIDB and TISC data sets, whereas in the CE data set CST-Model has a better Hit-Ratio almost for every session length. Thus, we decided to use the session length as a switching criterion for the NASA data set. We denoted this switching criterion as "Switch S" in the experimental tables. But, we can not use the same switching criterion for the other data sets.

We further analyze the recommendation efficiency of individual modules according to the confidence levels of association rules extracted from the user sessions as proposed for association rule mining (Agrawal & Srikant, 1994). In the context of Web usage mining, association rules refer to sets of pages that are accessed together with a support value exceeding some specified threshold. We have demonstrated our results about the confidence levels of association rules extracted from the user sessions on two sets of experiments. For both sets of experiments, we extract from the training set all of the association rules, $X \Rightarrow Y$, with a minimum

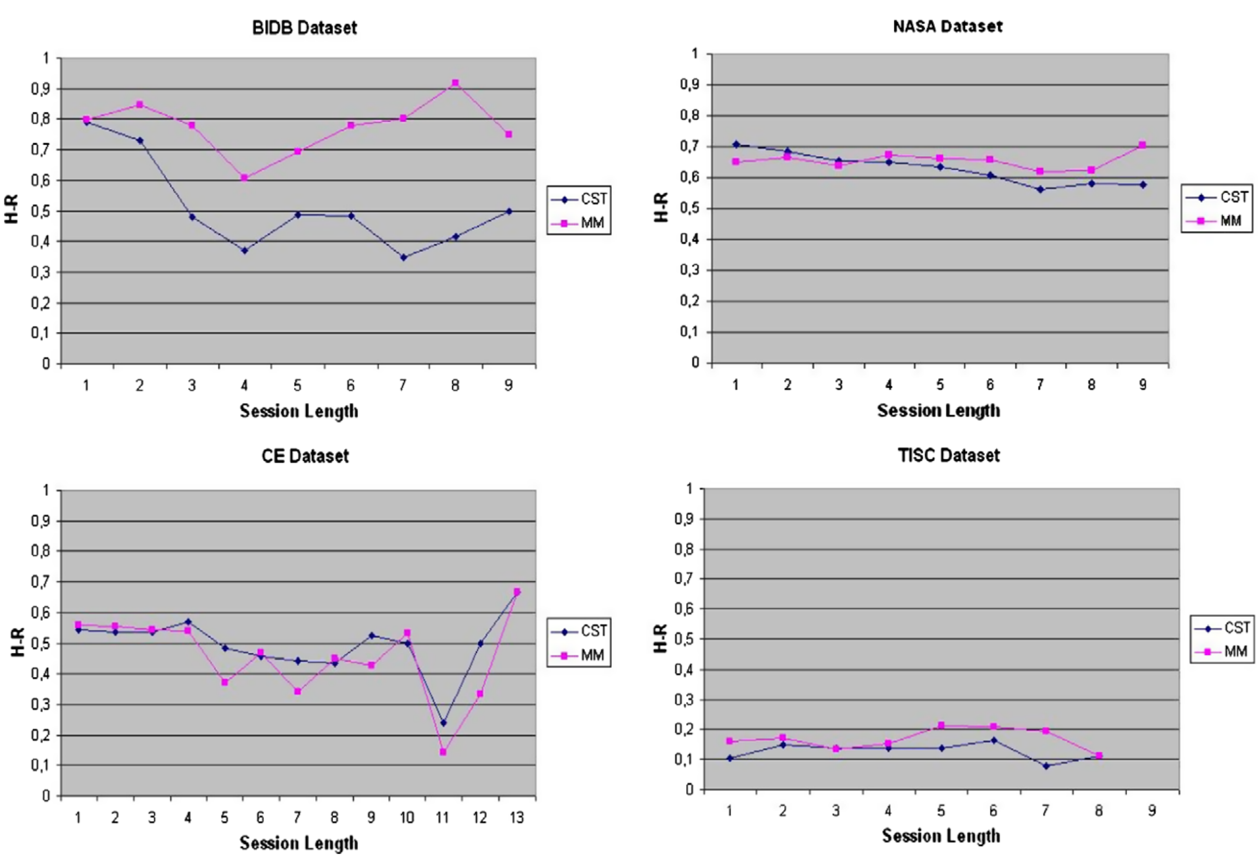

Fig. 4. Recommender performance according to session length for each dataset.

support 0.001. The reason of choosing a low support value is that very few rules are found for higher support values in these data sets. This is likely due to the fact that Web logs are sparse data sets (Yang, Wang, & Kitsuregawa, 2006) with many unique pages with only few repetitions of them in the sessions. The left-hand-side of the rule, namely X, contains two pages and the right-hand-side, Y, contains only one Web page. We then extract from the validation set all the association rules with the same left-hand-sides as the association rules extracted from the training set. These rules could have different right-hand-sides. For example, suppose we extracted an association rule $X \Rightarrow Y$ from the training set. Then, we search for all rules in the validation set that have the same left-hand-side, namely X. But the right-hand-side of these rules could be different than Y. We then calculate the confidence values of each of the association rules extracted from the validation set.

In the first set of experiments, we calculate the average confidence by dividing the sum of the confidences of all association rules having the same set of pages at the left-hand-side (X) over the number of these rules. For every possible average confidence value we calculate the Hit-Ratio of the modules over the validation set when the user session, to which the module generates recommendations, contains the pages that are in the set of X. Fig. 5 illustrates the Hit-Ratio of the modules according to the changes of the average confidence values of the association rules for the NASA, BIDB,CE and TISC data sets. As can be seen from the figure, we can not determine a switching criterion that uses average confidence for the NASA, CE and TISC data sets. However, in the BIDB data set, the Markov-Model outperforms the CST-Model for the average confidence values greater than 0.4. Thus, we decided to use this average confidence level as a switching criterion for the BIDB data set. In the experimental tables ‘‘Switch CFA(FI)’’ denotes this criterion.

In the second set of experiments for confidence value, we followed the same methodology as the last experiment set, but using confidence values of the rules instead of average confidence values. We have obtained a switching criterion for the BIDB data set where the Markov-Model outperforms for the confidence values greater than 0.8. This method for switching is represented in the experimental tables as ‘‘Switch CF(FI)’’. As a result of two sets of experiments about confidence values, we come to the conclusion that the confidence values of association rules can not be used as a switching criterion for the CE, NASA and TISC data sets. For this reason, we repeated the same experiments for the (average) confidence values of the frequent sequences. Sequential pattern mining is based on association rule mining concepts of support and confidence. In this case, we look for a switching criterion that is based on the (average) confidence of the frequent sequences (Fig. 6). We can not determine a switching criterion for BIDB and CE data sets since Markov-Model outperforms the CST-Model for every possible (average) confidence values in the data sets. In the NASA data set, the switching criterion for the confidence of frequent sequences is found as 0.2 and 0.5 since CST-Model has a better performance for the confidence values between these values. The CST-Model generates recommendations between the confidence values 0.2 and 0.5, otherwise Markov-Model generates recommendations. In the TISC data set, Markov-Model outperforms the CST-Model for the average confidence values greater than 0.4. In the tables, the criteria that use the average confidence and confidence values for frequent sequences are termed ‘‘Switch CFA(FS)’’ and ‘‘Switch CF(FS)’’, respectively. As a result of these experiments, we can only determine switching criteria for BIDB, NASA and TISC data sets, but we cannot determine any criterion for CE data set.

There are two conclusions to be drawn from the performance of the modules. One is that the modules do not have consistent performance for all types of user behavior. The performance of the modules vary between the data sets for the same types of user behavior. For example, we can not come up with one model that performs always better for short sessions (or long sessions). The

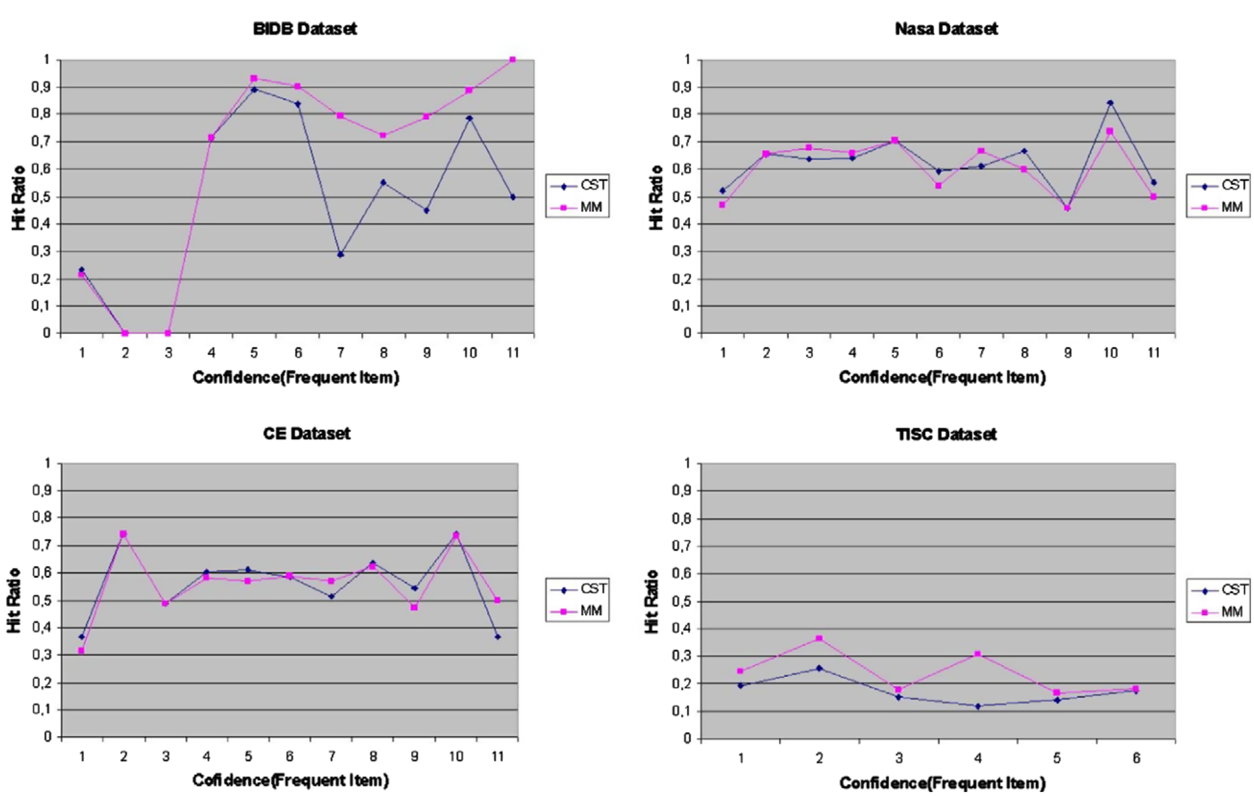

Fig. 5. Recommender performance according to the average confidence value (frequent item).

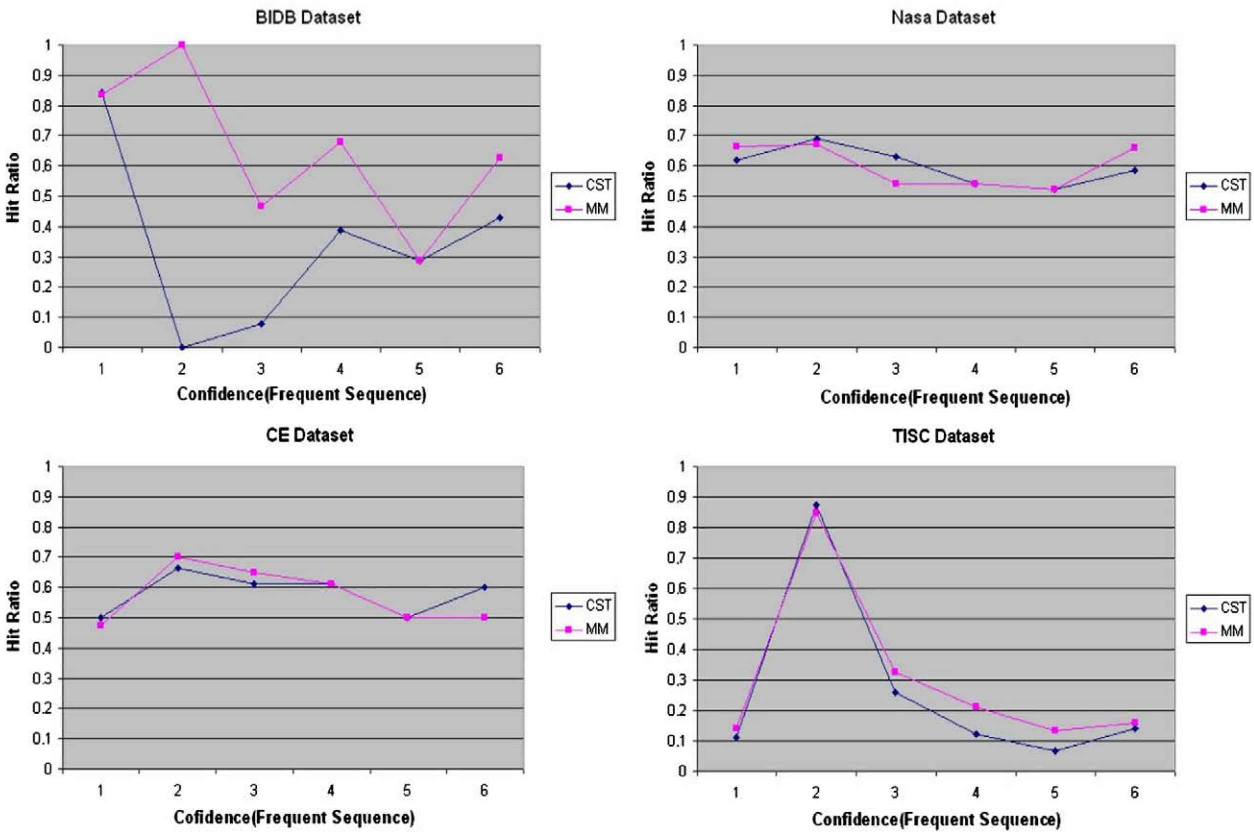

Fig. 6. Recommender performance according to the average confidence value (frequent sequence).

second point is that the performance of each module should be evaluated over each data set separately in order to decide a switching criterion when used switching as a hybridization method.

6.5. Effects of the hybridization methods

This subsection briefly describes the general results of the hybridization methods. There are no results for "switch" methods according to session length criterion for the CE, BIDB and TISC data sets and according to confidence value criterion for the CE data set. The reason is that we found out in the previous subsection that the performance of both of the modules do not depend on these criteria in these data sets. The logs of the NASA data sets were collected in 1995. As with other publicly available data sets for Web usage mining, there is no information for this Web site structure in 1995. Thus, "ranking" methods which use original PageRank and HITS algorithm are not examined for this data set.

Table 2 shows the Hit-Ratio for the hybrid recommenders and for the individual modules. The Hit-Ratio of the TISC data set is low. This may be due to the low similarity values between the user sessions in the test set and the user sessions in the training set (Fig. 2). Our aim in this paper is to examine the effects of the hybrid recommenders. For this reason, we do not especially concentrate on increasing the Hit-Ratio of the individual modules of the hybrid on a data set; rather we are interested in the results of the hybrid recommender against the results of its modules. Any improvement of the Hit-Ratio of the modules will also have a positive impact on the performance of the hybrid recommender that uses these modules since, as noted earlier, the performance of the modules are positively correlated with the results of the hybrid recommender. According to the table, the mixed hybrids (Method 1 and Method 3) have a lower Hit-Ratio than the individual modules for the BIDB data set. Mixed hybrid when using Method 2 increases the Hit-Ratio for this data set, however this increase is not as significant as in the other data sets. There is a bigger performance difference between the modules of the hybrid for short sessions in this data set than the other data sets (Fig. 4). As stated before, the user's activities tend towards short session lengths for all of the data sets. For this reason, a big performance difference for the short sessions have a major effect over the performance of the hybrid recommender.

In general, the weighted and switching hybridization methods increase the Hit-Ratio. Especially, for the NASA and CE data sets the weighted hybrid performs better since the individual modules of the hybrid show the same performance behavior (see Figs. 4 and 5). The weighted hybrids work under the assumption that the

Table 2
Hit-Ratio in % of the hybrid methods.

Recommender	Method	BIDB	CE	NASA	TISC
CST		69	48	65	17
MM		78	43	65	20
Weighted		79	51	69	21
Mixed	Method 1	74	50	68	21
	Method 2	79	50	68	21
	Method 3	77	50	69	20
Switching	Switch S			67	
	Switch CFA(FI)	80			
	Switch CFA(FS)				20
	Switch CF(FI)	79			
	Switch CF(FS)	79		67	
Ranking	Method 1	78	48		18
	Method 2	77	41	56	17
	Method 3	79	50		18

individual modules have uniform performance across the recommendation space. For this reason, this type of combination method results in a higher prediction accuracy for these data sets. However, in this case it is impossible to determine a switching criterion for this behavior. When determining a correct switching criterion, the switching method increases the recommendation accuracy. For TISC data set, the switching method does not increase the recommendation accuracy, because there are few cases that satisfy the switching criterion. In most cases, the recommendations are generated by the Markov-Model for this data set.

These results also show that using ranking methods which utilize HITS and PageRank algorithms do not increase the Hit-Ratio. This may suggest that the structure information incorporated by using HITS or PageRank algorithms are not suited for recommendation models. Furthermore, modified PageRank algorithm do not increase the prediction accuracy when using as a ranking method.

We further evaluate the hybrid recommender in terms of precision and coverage metrics defined for recommender models (Li & Zaïane, 2004). The precision is defined as follows:

$$PR = \frac{\sum_{s_i} \frac{|\cup_{p_j^i} T(s_{ij}) \cap RS(s_{ij})|}{|\cup_{p_j^i} RS(s_{ij})|}}{|S|}$$

where $T(s_{ij}) = (p_{j+1}^i, \dots, p_m^i)$ is denoted as the suffix of s_i and $|S|$ is the number of user sessions in the test set.

The coverage is defined as follows (Li & Zaïane, 2004):

$$C = \frac{\sum_{s_i} \frac{|\cup_{p_j^i} (T(s_{ij}) \cap RS(s_{ij}))|}{|\cup_{p_j^i} T(s_{ij})|}}{|S|}$$

Tables 3 and 4 illustrate, respectively, the precision and coverage results of the hybrids. The coverage[7] of a recommender system measures the ability of a system to produce all pages that are likely to be visited by the users. For this reason, the coverage increases when the Hit-Ratio increases. As expected, the precision decreases for the hybrids when the coverage increases. As observed in the tables, in general, weighted and switching hybrids tend to produce more accurate recommendations while decreasing precision within a reasonable range. Ranking hybrids lead to lower precision levels while not providing any advantage in terms of higher coverage.

6.6. Summary of the results

As a results of our experiments we can summarize that using a recommender model as a module of hybrid recommender system, which has a lower performance relative to the other modules of the hybrid recommender, decreases the final recommendation performance. The hybrid recommender provides successful recommendation when the recommended page is generated by all the modules of the hybrid. This is not surprising since a page generated by a large group of modules may be more reliable. However, it may be that using recommender techniques as the modules of the hybrid system that have high prediction accuracy, but have less pages in common in their recommendation sets could increase the final Hit-Ratio of the hybrid recommender. By analyzing the results of the hybrid recommenders we can draw the following conclusions:

- The weighted and the switching hybrids show good performance.
- The performance of each module should be evaluated over each data set separately in order to decide a switching criterion.

Table 3
Precision in % of the hybrid recommenders.

Recommender	Method	BIDB	CE	NASA	TISC
CST		48	25	28	9
MM		40	15	29	10
Weighted		39	18	29	11
Mixed	Method 1	48	24	29	10
	Method 2	47	23	28	11
	Method 3	47	23	27	11
Switching	Switch S			28	
	Switch CFA(FI)	40			
	Switch CFA(FS)				9
	Switch CF(FI)	39			
	Switch CF(FS)	40		29	
Ranking	Method 1	39	17		9
	Method 2	38	15	24	9
	Method 3	39	18		8

Table 4
Coverage in % of the hybrid recommenders.

Recommender	Method	BIDB	CE	NASA	TISC
CST		30	15	27	8
MM		38	15	28	10
Weighted		38	15	29	11
Mixed	Method 1	33	16	29	10
	Method 2	34	17	30	10
	Method 3	34	16	29	10
Switching	Switch S			29	
	Switch CFA(FI)	39			
	Switch CFA(FS)				9
	Switch CF(FI)	39			
	Switch CF(FS)	39		29	
Ranking	Method 1	38	15		8
	Method 2	38	15	24	8
	Method 3	38	16		9

- The weighted and switching hybrids tend to produce more accurate recommendations while decreasing precision within a reasonable range.
- Ranking methods that utilize HITS and PageRank algorithms in order to combine Web site structure do not provide any advantage for recommendation.
- Ranking hybrids lead to lower precision levels while not providing any advantage in terms of higher coverage.

7. Conclusion and future work

In this paper, we examine the performance of hybrid recommender models including some novel combinations in order to combine several recommender techniques. We investigate four different combination methods and propose several modifications to these methods. Our hybrid recommender combines the results from multiple recommender techniques, which we call the modules of our recommender system, and generates a single recommendation set for a new user. We see significant differences between the combination methods. For example, if the hybridization method is wrongly chosen, then the performance of the hybrid can be lower than its components. For this reason, it is particularly important to examine the performance of each module of the hybrid on the data set. Among the different combination methods, the weighted and switching methods perform better in terms of Hit-Ratio. However, for the switching methods, a detailed examination of the performance of the modules on the data sets is needed in order to determine a switching criterion. In choosing a recommender model as a module of the hybrid, the individual performance of the modules should be taken into consideration. Our

[7] It is similar to the concept *recall* in information retrieval.

detailed experiments show that using a recommender technique as a module of hybrid recommender system, which has a lower performance relative to the other modules of the hybrid recommender, decreases the final recommendation performance.

At present, we perform the prediction of the next request of the users on the test user sessions simulating active user sessions. Since our results are promising, we are now implementing the weighted hybrid recommender model to serve real-time recommendations to the Web site of the Department of Computer Engineering of the Istanbul Technical University. We are planning to use the results of this system for improvement of the weighting strategy.

Acknowledgments

The authors were supported by the Scientific and Technological Research Council of Turkey (TUBITAK) EEEAG project 105E162.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), *Proceedings of the 20th international conference on very large data bases, VLDB* (pp. 487–499). Morgan Kaufmann.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the international conference on data engineering ICDE* (pp. 3–14).

Balabonović, M., & Shoham, Y. (1995). Learning information retrieval agents: Experiments with automated web browsing. In *Proceedings of the AAAI spring symposium on information gathering from heterogenous, distributed resources* (pp. 13–18).

Bose, A., Beemanapalli, K., Srivastava, J., & Sahar, S. (2006). Incorporating concept hierarchies into usage mining based recommendations. In *WebKDD 2006: KDD workshop on web mining and web usage analysis, in conjunction with the 12th ACM SIGKDD international conference on knowledge discovery and data mining KDD 2006* (pp. 110–126).

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th annual conference on uncertainty in artificial intelligence* (pp. 43–52).

Brin, S., & Page, L. (1998). The anatomy of large-scale hypertextual web search engine. In *Proceedings of international world wide web conference (WWW'98)* (pp. 107–117).

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331–370.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2003). Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery, 7*(4), 399–424.

Charter, K., Schaeffer, J., & Szafron, D. (2000). Sequence alignment using FastLSA. In *Proceedings of the International conference on mathematics and engineering techniques in medicine and biological sciences* (pp. 239–245).

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

Demir, G. N., Goksedef, M., & Uyar, A. S. (2007). Effects of session representation models on the performance of web recommender systems. In *Proceedings of the workshop on data mining and business intelligence* (pp. 931–936).

Demir, G.N., Uyar, A.S., & Gündüz-Ögüdücü, S. (2007). Multiobjective evolutionary clustering of Web user sessions: a case study in Web page recommendation. *Soft Computing Journal*. doi:10.1007/s00500-009-0428-y.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, 39*(1), 1–38.

Deshpande, M., & Karypis, G. (2004). Selective markov models for predicting web page accesses. *ACM Transactions on Internet Technology (TOIT), 4*(2), 163–184.

Eirinaki, M., Vazirgiannis, M., & Kapogiannis, D. (2005). Web path recommendations based on page ranking and markov models. In *WIDM '05: Proceedings of the 7th annual ACM international workshop on web information and data management, New York, NY, USA* (pp. 2–9). ACM.

Göksedef, M., & Gündüz-Ögüdücü, S. (2007). A consensus recommender for web users. In *ADMA 2007: The 3rd international conference on advance data mining and applications* (pp. 287–299).

Göksedef, M., & Gündüz-Ögüdücü, S. (2008). Integration of the pagerank algorithm into web recommendation system. In *IADIS European conference on data mining*.

Gündüz, S., & Özsu, M. T. (2003). A web page prediction model based on click-stream tree representation of user behavior. In *Proceedings of 9th ACM international conference on knowledge discovery and data mining (KDD), Washington, DC, USA, August* (pp. 535–540).

Jin, X., Mobasher, B., & Zhou, Y. (2005). A web recommendation system based on maximum entropy. *ITCC '05: Proceedings of the international conference on information technology: Coding and computing (ITCC'05), Washington, DC, USA* (Vol. I, pp. 213–218). IEEE Computer Society.

Kazienko, P., & Kolodziejski, P. (2006). Personalized integration of recommendation methods for e-commerce. *International Journal of Computer Science and Applications, 3*(3), 12–26.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46*(5), 604–632.

Li, J., & Zaïane, O. R. (2004). Combining usage, content, and structure data to improve web site recommendation. In *Proceedings of 5th international conference on electronic commerce and web* (pp. 305–315).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings 5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

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

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002b). Using sequential and non-sequential patterns in predictive web usage mining tasks. In *ICDM '02: Proceedings of the 2002 IEEE international conference on data mining (ICDM'02), Washington, DC, USA* (pp. 669–672). IEEE Computer Society.

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2001). Effective personalization based on association rule discovery from web usage data. In *Web information and data management* (pp. 9–15).

Mohr, G., Kimpton, M., Stack, M., & Ranitovic, I. (2004). Introduction to heritrix, an archival quality web crawler. In *Proceedings of the 4th international web archiving workshop*.

Nakagawa, M., & Mobasher, B. (2003). A hybrid web personalization model based on site connectivity. In *Proceedings of WebKDD* (pp. 59–70).

Nanopoulos, A., Katsaros, D., & Manolopoulos, Y. (2001). Effective prediction of web-user accesses: A data mining approach. In *Proceedings of WEBKDD workshop, San Francisco, CA, USA*.

O'Conner, M., & Herlocker, J. (1999). Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR workshop on recommender systems, Berkeley, CA*.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review, 13*(5–6), 393–408.

Resnick, P., & Varian, R. H. (1997). Recommender systems. *Communications of the ACM, 40*(3), 56–58.

Sarukkai, R. R. (2000). Link prediction and path analysis using markov chains. In *Proceedings of the 9th international world wide web conference* (pp. 377–386) *Amsterdam*.

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

Yang, Z., Wang, Y., & Kitsuregawa, M. (2006). An effective system for mining web log. In *Proceedings of the 8th Asia-Pacific web conference APWeb* (pp. 40–52).