Chapter 7
Web Page Recommendations Based Web Navigation Prediction

K. R. Venugopal and Sejal Santosh Nimbhorkar

Abstract A huge amount of user request data is generated in Web log. Predicting users' future requests based on previously visited pages is important for Web page recommendation, reduction of latency and on-line advertising. These applications compromise with prediction accuracy and modelling complexity. In this chapter, a Web Navigation Prediction Framework for Web page Recommendation (WNPWR) which creates and generates a classifier based on sessions as training examples is proposed. As sessions are used as training examples, they are created by calculating the average time on visiting Web pages rather than the traditional method which uses 30 min as default time-out. The proposed method uses standard benchmark datasets to analyse and compare our framework with two-tier prediction framework. Simulation results show that our generated classifier framework WNPWR outperforms two-tier prediction framework in prediction accuracy and time.

7.1 Introduction

A huge amount of data is generated when millions of users access Websites. One of the influential data source is log file of Web server, which traces the users' web-browsing actions. This data consists of repeatedly accessed Web pages by users within a period of time. Users' navigation history within a period of time is known as session. This session information is very important and helpful to find the user behaviour. Through this behaviour, user's next request can be predicted and recommendations can be made to reduce the browsing time of the Web pages. Recommending related pages to users reduces network traffic as it avoids visiting unnecessary pages.

The possibility of visiting a Web page by a user based on the history of earlier accessed Web pages is known as Web prediction. Prediction of Web user's behaviour

K. R. Venugopal (✉)
Bangalore University, Jnana Bharathi, Bengaluru 560056, India
e-mail: venugopalkr@gmail.com

S. Santosh Nimbhorkar
BNM Institute of Technology, Banashankari, Bengaluru 560070, India
e-mail: sej_nim@yahoo.co.in

K. R. Venugopal et al., *Web Recommendations Systems*,
https://doi.org/10.1007/978-981-15-2513-1_7

is critical in Web mining to enhance the performance of the search engine. The organization of the Web is modelled as a graph, where each node represents a Web portal, and the edge represents the user's navigation. Distribution of all Web pages visited can be calculated and utilized in re-weighting and re-ranking results. The information provided by the navigation path is of prime importance than the query given by the user. Web cache performance of search engine can be improved by storing predicted pages in the cache.

The Web users' usually spend more time on browsing and authoring than on search. Hence, the search engine cannot effectively predict users' search intention. Prediction is performed only after the users submit their queries to search engines, as the prediction is conducted in a passive manner and this navigation history is used for Web prediction. Web prediction can be used in recommendation system, in which the top *k* users are involved in similar activity.

Behavioural targeting is a key issue of predicting future behaviour of Web users. Behavioural targeting is a technique to improve efficiency of advertising by online website publishers and advertisers by extracting knowledge of web-browsing behaviour of users. Behaviour targeting selects advertisement to display with the help of web-browsing behaviour of users. The user analysis approach is the centre of interest in on-line advertising and properly targeted advertisements generate more consumer interest.

Predicting Web user's shopping behaviour has an important role in product recommendation. Product recommendations are the dynamic shopping recommendations across mobile, email and Web channels. It depends on each customer's past and current purchase behaviours. It also helps in website optimization, improves conversions and increases revenue by making related product recommendations to the customers.

The World Wide Web (WWW) has created tremendous opportunities to spread and accumulate huge online information. This motivates researchers to understand the navigation behaviour of Web site visitors from Web usage data to improve the quality of service of that site, to reduce access latency and Web page recommendations using efficient Web prediction technique. The Web log records the navigational behaviour of the user. Preprocessing of the raw data is required before giving the data as input to prediction model. Preprocessing challenges include session identification, handling huge amount of data and obtaining domain intelligence. Low accuracy and expensive training are fundamental issues in prediction.

Many researchers have developed several prediction models by fusing Support Vector Machine (SVM), Markov model, Association Rule Mining (ARM) and Artificial Neural Network (ANN). First, SVM and Markov improve prediction time, but this model uses the traditional method of session identification with 30 min time-out period for one session which decreases the prediction accuracy. Second, SVM and ARM are not scalable with large datasets. Third, ANN and SVM cannot handle the multi-class issue effectively on account of a large number of classes that are used in Web prediction.

Web navigation prediction framework for Web page recommendation is designed by using user request on the Web. This framework creates a classifier based on sessions as training data and classifiers generated by the *N*th-order Markov models.

Each session is mapped to the generated classifiers. If any session is mapped to more than one classifier, then each session is mapped to only one classifier according to PageRank algorithm during the filtering process. Filtered data is then used to train the SVM classifiers. Once SVM classifiers are trained, prediction accuracy and time are calculated on the test dataset. Finally, SVM classifiers can be used for page recommendation. In this work, Sessions are created by computing the average time on visiting Web pages rather than the traditional method which uses 30 min as default time-out. The PageRank algorithm is used in the filtering process, which results in an improvement in prediction accuracy and time.

7.2 Related Works

In this section, we have reviewed several papers related to Web page prediction and various prediction application.

7.2.1 Web Page Prediction

7.2.1.1 Markov Models for Web Page Prediction

Chimphlee et al. [1] developed a prediction method to access websites with First-order and Second-order Markov model by considering user Web access behaviour log file. This algorithm is used to cluster similar transition behaviours to further improve the efficiency of prediction. The First-order Markov model persistently gives the best performance to predict Web access behaviour. When the recall is less than 50%, the Second-order Markov model gives the worst performance. When precision is less than 50%, the association rule gives the worst performance. Next, Borges et al. [2] developed a method to measure variable-length Markov model's ability for summarizing Web sessions of users for the given length. Spearman footrule metric is used to determine the accuracy to characterize information content of the users' sessions. A prediction algorithm which eliminates few states of all kth Markov model selectively is developed to predict user's Web page access behaviour [3]. Support, Confidence and Error pruning technique are used to eliminate states. This technique has achieved better prediction accuracy than First, Second, Third and All kth Markov model.

7.2.1.2 Online Prediction

Guerbas et al. [4] proposed an approach for online navigational pattern prediction. Navigational patterns are discovered by density-based algorithm. A model is developed by modelling user's Web access information and by constructing weighted suffix tree from content of Web page [5]. This method requires less memory space and consistent computational training for user's activity. A two-step prediction model is developed that decreases the size of Web pages' candidate set and increases prediction accuracy's speed by using a hierarchical property of website [6].

7.2.1.3 Statistical Theory for Web Page Prediction

Many methods are developed to understand web-browsing behaviour using statistical theory. Dembczynski et al. [7] described the user-level models to predict Web users' behaviour by Statistical Decision Theory and Learning Theory. A model is proposed to understand web-browsing behaviour through Weibull distribution on *dwell time* [8]. *Dwell time* is the length of time a user spends on document. The log data is organized into sessions, each of which is determined as a sequence of Web pages browsed for 30 min or user closes browser before 30 min. *Dwell time* is computed for each page by leaving the last page in the session and Weibull distribution is applied to it. The prediction model is used to predict Web page *dwell time* distribution. Negative binomial distribution and inverse Gaussian models are used for qualitative comparison of session length to model the behaviour of visitors to an academic website [9].

White et al. [10] presented a log-based study to model user interests while interacting with the Web. The current page usage with other information like recent correspondence behaviour, hyperlinks, pages' relation to the present page which shares similar search engine queries, long-term interests of the present user with other users who also visit the present page are evaluated. Thwe [11] proposed a popularity- and similarity-based PageRank algorithm to predict Web page access behaviour. Different navigational attributes like size of the page, transition, frequency of page, similarity of the page, duration of the page and access time of the page are used. This model for next page prediction is a promising approach than Markov models.

7.2.1.4 Hybrid Techniques for Web Page Prediction

Different hybrid techniques are developed for Web page prediction. Khali et al. [12] presented a novel approach by incorporating association rules, Markov models and clustering for Web page prediction. The integration provides better prediction accuracy than using each technique individually. Awad et al. [13] studied composite models by combining various classification methods especially Artificial Neural Networks (ANN), All *k*th Markov Model by Dempster's rule and Markov Model to predict user's future request. Markov model lacks in predicting user's future request when training data is not available. In ANN, the prediction accuracy decreases when the number of classes increases. Markov model and ANNs combination handles above-mentioned drawbacks of individual model. This hybrid model is more effective in prediction accuracy than All *k*th Markov model, association rule mining and Markov model.

Dutta et al. [14] proposed a Web page prediction model by clustering user's interest and Markov model. Similar pages are aggregated with K-medoids clustering method and K is computed with the HITS algorithm. The predicted Web pages are saved with cellular automata scheme and are memory efficient. Awad et al. [15] developed a two-level prediction framework to identify user's web-browsing behaviour. Sessions are trained with nth-order Markov model and mapped to one or more orders of Markov models. Support Vector Machine (SVM) is used as a prediction technique to create Example Classifier (EC). A testing example is given as an input to EC to predict an appropriate classifier. This model predicts better web-browsing behaviour than Markov model and association rule mining.

7.2.2 Prediction Applications

Web Prediction is used in many applications like mobile users' movement, place prediction, service recommendations, online user behaviour, search prediction and image click prediction. Tseng et al. [16] designed a method to discover mobile users' sequential patterns to understand their movement in correlation with a desired service. Sequential Mobile Access Pattern (SMAP) tree is constructed to aggregate the access pattern. SMAP-Mine algorithm based on depth-first search is applied to find sequential patterns. A graph is created from session to understand behaviour strategy in mobile Internet, in which each Web page is vertex and edge indicates the number of transitions from one Web page from another Web page [17]. Random walk restart algorithm is applied on graph for prediction.

Semantic Place Prediction is a process to predict the semantic meaning of place. Huang et al. [18] have proposed a novel prediction framework which takes into account the spatial property, users' behaviour and environment for semantic place prediction. Several models like SVM, J48, etc., are used to build multilevel classification models. Decision Tree is used to discover the association between the results of different models and the real answer of place. Service recommendation system predicts the availability to atomic Web services by service Load, User Location, Service Class, Service Location Model [19]. This model predicts atomic services with collaborative filtering algorithm by using prior work availability. It predicts service availability by the geographic location of the service, the users' geographic location, the computational requirements of the service and the current load of the service provider.

Huang et al. [20] studied the usage of parallel browsing with the help of Web log. The degree of parallel browsing is identified by discovering browser pageview for outgoing clicks and tab switches. It is observed that 57.4% sessions with tab include parallel browsing and users are separating their browsing movement into various tabs rather than examining more pages. Goel et al. [21] developed a method to measure online behaviour of a Web user. This method demonstrates behavioural changes of the Web user with respect to time spent online. The heaviest users allot twice as much of their time to social media relative to a typical Web user. Linear Support Vector Machines(SVMs) is used to infer demographic attributes from the browsing history.

Cheng et al. [22] presented a technique to predict search intent from the user browsing behaviours. Queries are extracted from the Web pages that users read after issuing query from user browsing behaviour. Page-query bipartite graph is constructed and query visibility, query popularity and pattern frequency features are extracted from the graph to describe the users' interests. Query dissimilarity measure is also obtained from the bipartite graph to minimize the diversification of queries. Ranked list of the queries is obtained by Support Vector Machine (SVM) and suggest to users. Zhang et al. [23] designed task-centric click model to predict the user search behaviour. The sequence of queries and their clicks in search session is considered as a task. This model describes the user behaviour associated with a task as a collective

Table 7.1 A comparison of related works

Author	Concept	Advantages	Disadvantages
Awad et al. [13]	Predicting User's future request by combining Markov model and Artificial Neural Networks (ANNs)	This hybrid model is more effective in prediction accuracy than All kth Markov model, association rule mining Markov model	Requires more computation for prediction and training
Chimphlee et al. [1]	Predicting Web site access with user browsing history by hybrid Markov model	The 1st order Markov model constantly provides excellent prediction performance and model building is very easy	A particular order of Markov model is not able to predict for a session which was not examined during training because such session will have zero probability
Goel et al. [21]	Measures online browsing behaviour of a Web user by linear support vector machines (SVMs) from browsing history	Accurately measures individual activity with large scale data	SVM do not handle the multi-class problem efficiently
Awad et al. [15]	Identify user's web-browsing behaviour using nth order Markov model and support vector machine	Increases prediction accuracy and reduces prediction time compared to Markov and association rule mining models	Statistical features are not used to create sessions
WNPWR	Web navigation prediction framework for Web page recommendation	Enhances prediction accuracy and reduces prediction time compared to Awad et al. [15] method	

whole and shows improvement in prediction over the User Browsing Model and Dynamic Bayesian Networks.

Tian et al. [24] proposed a method which automatically predicts the feature of image search results for a query. First, features to measure image search quality is derived by examining the visual distribution attributes of bad and good search results of the training queries. The latent relationship between obtained features and the inherent query difficulty is mined during learning process to build query difficulty prediction model. This model is used to measure query difficulty for a new query. A method is designed to predict an image click based on hypergraph learning-based sparse coding method [25]. The acquired click data is used to re-rank images. Based on a group of the Web images with associated clicks and a new image without any clicks known as codebook, sparse coding is utilized to choose a few basic images as possible from the codebook in order to linearly reconstruct a new input image while minimizing reconstruction errors. A voting strategy is utilized to predict the click as a binary event from the sparse codes of the corresponding images.

Table 7.1 shows the comparison of closely related works with our proposed method.

7.3 Web Navigation Prediction Framework and WNPWR Algorithm

7.3.1 Problem Definition

Given a user navigation history, we convert user navigation history into sessions by calculating the average time of visiting Web pages. The objective is to generate Web navigational prediction framework with high prediction accuracy and reduced prediction time. The prediction accuracy and time is calculated only for the first four pages visited by the users. If the user visits more than four pages, then a sliding window of size four is applied.

7.3.2 Session Identification Method with Average Time of Visiting Web Pages

Session identification method with the average time of visiting Web pages that is used to create sessions to Web navigation prediction frameworks. Often the visitors access the same website frequently that is recorded in the log file. The process of segregating the page access of each user into a singular session is called session identification. It is presumed that the user starts a new session if the consecutive page request exceeds a certain time limit. Commercial websites usually have a default time-out of 30 min [26]. However, this time-out period may not be enough for some websites in which user reads articles and collects opinions about products. The time required to cover a certain amount of information is dependent on the profile of the users, for example, an elderly person follows information slowly. When a client wishes to purchase a product then he may spend more time to analyse the product and may exceed the 30 min time-out.

Dinuca et al. [27] proposed a new session identification method by computing the average time of visiting Web page. For each visited page, the duration of visit is computed as a difference between two successive timestamps for the same user and is identified either by username or IP. The highest timestamp among those visited pages by an user is assumed to be 20,000 s. A page average visit time is computed by calculating the mean of all the visit time spent on page. Time less than 2 s and larger than 20,000 s are not considered for computing the average visiting time. The method for session identification to compute the average time of visiting Web pages is given in Function 7.1.

Function 7.1: Session Identification

```
Function: session
Data: Consider the set of users by $U = U_1, U_2, ..., U_n$. The pages visited by the users $U_k$ is
      identified by $PU_k = PU_{k1}, PU_{k2}, ...$ and $TSPU_{ki}$ is the timestamp of $PU_{ki}$ page.
      $IDPU_{ki}$ is the session identification number allocate to pages $PU_{ki}$ with $ID$.
for each $U_k$ in U Repeat do
    $IDPU_{k1}$ = max($ID$)+1;
    I = 1;
    while ($I < |PU_k|$) do
        I=I+1;
        $IDPU_{ki} = IDPU_{k,i-1}$;
        $TMA_{ki}$ = max(2 * $TM_{ki}$, 300);
        if $TSPU_{ki} - TSPU_{k,i-1} > TMA_{ki}$ then
            $IDPU_{ki} = IDPU_{k,i-1} + 1$;
```

TM_{ki} is the average spent time on the page PU_{ki} by the users. TMA_{ki} is the time utilized to create sessions instead of 30 min time-out. The value of 300 is required to calculate TMA_{ki}, as the average time of some pages is very low that can negatively affect to identify the sessions.

For example, if the user X has visited different Web pages in a Web log as shown in Table 7.2. The visited Web pages are arranged in ascending order based on timestamp and duration of visited Web page is calculated as a difference between the timestamp of two successive Web pages as shown in Table 7.3. As discussed earlier, the timestamp difference of Web page with highest timestamp is assigned to 20,000. If the user Y has visited Web page B in his session and the visit duration is 40, then the average visit time of page B is $(53 + 40)/2 = 46.5$. Sessions are generated as described in Function 7.1.

Table 7.2 Web pages visited by a user

Web pages	Timestamp
A	[01/Jul/1995:01:40:52 -0400]
B	[01/Jul/1995:01:41:43 -0400]
C	[01/Jul/1995:01:42:36 -0400]
D	[01/Jul/1995:01:49:23 -0400]

Table 7.3 Timestamp difference between successive Web pages

Web pages	Timestamp difference
A	51
B	53
C	407
D	20000

7.3.3 Prediction Models

The Markov model, PageRank algorithm and Support Vector Machine (SVM) are used to generate classifiers in prediction framework.

Markov Model: Markov model is a stochastic process in which the next state relies on the former states. In Web prediction, the next state correlates to predicting the next visiting page and the former states correlate to the previously visited pages. Markov models are defined by three parameters, namely $< P, S, T >$ in Web prediction, where P is the previously visited Web pages by the users, S is the all possible states to build the Markov model; and T is a |S| × |P| Transition Probability Matrix (TPM) in which each entry t_{ij} represents the probability that a user visits page j when he has already visited i pages [3].

The simplest Markov model also known as first-order Markov model predicts the next page by only observing the previously visited page by the user. In the first-order model, the states represent a single page; in the second-order models, the states represent two successive pages and so on. In general, the Kth-order Markov model computes the probability of user visits kth page after he has visited $k - 1$ pages.

After determining the states of the Markov model, the TPM is estimated. The general approach is to use sessions as training set and measure each t_{ij} entry with the visited pages' frequency. For example, consider the session SE_6 (A, C, F, G, H) shown in Table 7.4. For the first-order Markov model, each state corresponds to a single page, so the first page A correlates to the state s_1 and second page C correlates to the state s_3. Since page A pursues the state s_3, the value of t_{13} in the TPM is amended. Equivalently, the next state is s_6 and the entry t_{36} is updated in TPM. Table 7.5 shows first-order Markov model TPM entries. In the higher order Markov model, each state is formed with more than one page. In the second-order Markov model, for session SE_6 the first state is (A, C) and the page F pursues the

Table 7.4 Session data

Session	Visited pages
SE_1	(A, B, C, D, E)
SE_2	(A, B, C, D, F)
SE_3	(A, B, C, D, E)
SE_4	(A, B, C, D, F)
SE_5	(A, B, F, G, H)
SE_6	(A, C, F, G, H)
SE_7	(A, B, C, D, E)

Table 7.5 First-order Markov model

1st order	A	B	C	D	E	F	G	H
$S_1 = \text{A}$	–	6	1	–	–	–	–	–
$S_2 = \text{B}$	–	–	5	–	–	1	–	–
$S_3 = \text{C}$	–	–	–	5	–	1	–	–
$S_4 = \text{D}$	–	–	–	–	3	2	–	–
$S_5 = \text{E}$	–	–	–	–	–	–	–	–
$S_6 = \text{F}$	–	–	–	–	–	–	2	–
$S_7 = \text{G}$	–	–	–	–	–	–	–	2
$S_8 = \text{H}$	–	–	–	–	–	–	–	–

state (A, C), the TPM value equivalent to the state (A, C) and page F is amended. Markov model has two advantages: (i) model construction efficiency and (ii) better prediction time performance. In our framework, we have used first-, second-, third- and fourth-order of Markov models.

PageRank Algorithm: Brin [28], designed a PageRank algorithm that ranks pages returned by a search engine. It allocates a numerical value to Web pages to compute their corresponding position in the Web pages' set. The significance of a page is equivalent to the total significant scores of Web pages linked to it. *PageRank* of a given page is the number of times the user has accessed the given page divided by the total number of pages the user has visited. The method for *Page Rank* is given in Function 7.2.

For example, if user X has visited different Web pages in a session as shown in Table 7.4, then pagerank of the page A is $7/8$, page B is $6/8$, page E is $3/8$ and page F is $5/8$.

Support Vector Machine (SVM): SVMs are used for classification which identifies patterns based on statistical learning theory. A classification method commonly separates data in testing and training sets. Each example in the training set includes one *target value* (i.e. the class labels) and various *attributes* (i.e. the observed or features variables). Given a set of training instances, each is assigned to one of the two categories. An SVM training algorithm generates non-probabilistic binary linear

Function 7.2: Page Rank

```
Function: PageRank
Data: Consider the set of users $U = U_1, U_2,...,U_n$. The pages visited by the user is identified
      by $PU_k = PU_{k1}, PU_{k2},...,PU_{km}$. Page rank of page $PU_{ki}$ is $PRPU_{ki}$.
for each $U_k$ in $U$ repeat do
    for each page in $PU_k$ repeat do
        Let $visit_{pages}$ = number of times the $U$ has visited given page ($PU_{ki}$)
            $$PRPU_{ki} = \frac{visit_{pages}}{\mid PU_k \mid}$$
```

classifier which designates new instance to one of the categories. An SVM model represents instances as points in space and mapped in such a way that the instances with different categories are divided by a clear wide gap. New instances are then outlined into that same space and used for prediction. It avoids the curse of dimensionality problem and works well with high-dimensional data. The method to find SVM model is as shown in Function 7.3.

Function 7.3: Support Vector Machine

```
Function: SVM
Data: Consider Data instances with Class Label $C_{label}$ and Attributes $C_{attributes}$
begin
    Segregate data instances into Training and Testing Dataset
    Convert each data instance as a vector of real numbers
    Apply scaling on Training and Testing Datasets
    Select SVM Kernel
    SVM produces a model $SVM_{model}$ which predicts the target values
    Give test data attribute as input to $SVM_{model}$ to predict the target value
    Use target value for recommendation
```

In our framework, the first-, second-, third- and fourth-order of Markov prediction models are used as data instances with predicted page as C_{label} and sessions as $C_{attributes}$. Here, LIBSVM package is used for SVM implementation and is discussed in experiments.

7.3.4 Two-Tier Prediction Framework

A two-tier prediction framework is discussed that is compared with our framework, as this framework has also used Markov model and SVM. Awad et al. [15] developed two-tier prediction framework to identify user's web-browsing behaviour. Sessions are trained with *n*th-order Markov model and mapped to one or more orders of Markov models and later used in prediction. Pruning is applied to those examples that are mapped to more than one classifier and by choosing the classifier that predicts accurately with maximum probability. Support Vector Machine (SVM) is used as a prediction technique to create Example Classifier (EC). A testing example is given to EC as an input for appropriate classifier prediction.

Web navigation prediction framework for Web page recommendation is explained in this section. Figure 7.1 represents different stages of this framework. All classifiers are trained on the sessions as training set S and N trained classifiers are derived. In mapping phase, each training session s in S is mapped to one or more classifiers which can be used to predict its target. If any training session s is mapped to more than one classifier, then each trained session is mapped to only one classifier according to the page rank algorithm during filtering process. This PageRank algorithm selects the classifier that predicts correctly with higher incoming page request. If the classifiers have equivalent incoming page request, one classifier is randomly selected. Next, SVM classifier is trained with filtered data. Once SVM classifiers are trained, prediction accuracy and time is calculated on the test dataset. Finally, SVM classifiers are used for page recommendation.

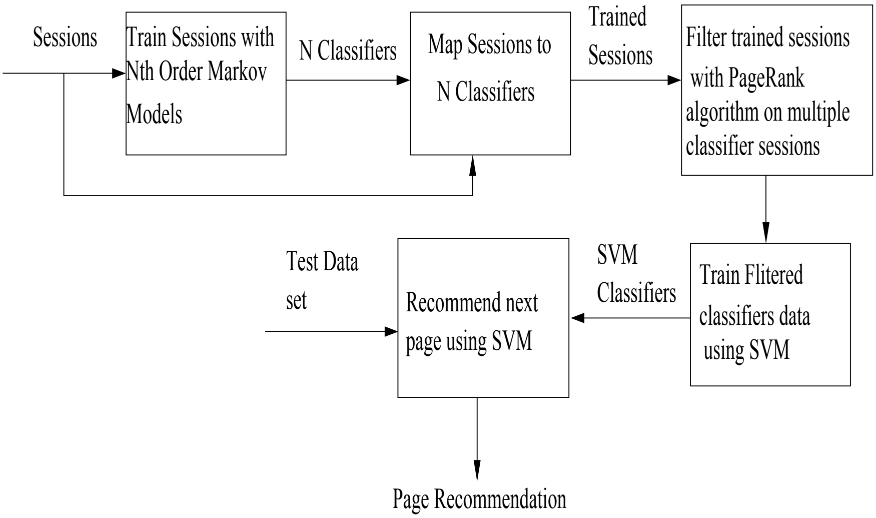
Sessions
Train Sessions with Nth Order Markov Models
N Classifiers
Map Sessions to N Classifiers
Trained Sessions
Filter trained sessions with PageRank algorithm on multiple classifier sessions
Train Flitered classifiers data using SVM
SVM Classifiers
Test Data set
Recommend next page using SVM
Page Recommendation

Fig. 7.1 Web navigation prediction framework

7.3.5 WNPWR Algorithm

The algorithm of Web Navigation Prediction for Web page Recommendation (WNPWR) is given in Algorithm 7.1.

Algorithm 7.1: WNPWR: Web Navigation Prediction Framework for Web Page Recommendation

```
Input : $M$ is the set of classifier models of size $N$. $S$ is the set of sessions generated from
        $session$. $T$ is a test set data.
Output: Recommended Pages
begin
    For each classifier model $m$ in $M$, train $m$ on $S$ with $N$-order Markov model.
    For each session $s$ in $S$ and a classifier model $m$ in $M$
    If $m$ predicts the target $s$ correctly then map $s$ to $m$.
    For each $s$ in $S$, if $s$ mapped to more than one classifier then filter it with PageRank().
    Train filtered classifier using SVM.
    For each $t$ in $T$ using SVM classifier
    If classifier predicts $t$ correctly then recommend next page and calculate prediction
    accuracy.
```

7.4 Experiments

7.4.1 Data Collection

The University of Saskatchewan's (UOFS) and the NASA datasets [29] are used to construct Web navigation prediction framework. The NASA dataset is divided into two groups in order to study the effect of size of the dataset in prediction accuracy and time: $NASA_{Low}$ and $NASA_{Medium}$. In the present work, UOFS data is called $UOFS_{High}$. Log files contain information of the user requests on a particular Web site. The main idea is to analyse HTTP user request and predict Web page access user behaviour. So, all entries with the extension type .gif, .GIF, .jpeg, .JPEG, .jpg are removed. Even status code other than 200, i.e. redirect (300 series), failure (400 series) and state error (500 series) are removed. The statistics of both the dataset is given in Table 7.6.

Each line of HTTP user request contains information about host making the request, timestamp, user request, HTTP reply code and bytes in the reply. If the hostname is not available, the Internet address is considered as host. The timestamp is in the form of *DAY/MON/YYYY: HH:MM:SS -0600*, where *DAY* is the day of the

Table 7.6 Summary of the datasets

	$NASA_{Low}$	$NASA_{Medium}$	$UOFS_{High}$
Total log records	132838	343234	1021891
Number of pages	608	713	3595
Data reduction in % after cleaning	79.09	78.79	58.65

Table 7.7 Sample of HTTP user request

130.54.25.198	01/Jun/1995:00:28:36 -0600	GET/macphed/finite/feresources/node1.html	200	9651
128.171.197.73	01/Jun/1995:00:34:50 -0600	GET/scottp/hawaii	200	29106
130.54.25.198	01/Jun/1995:00:35:01 -0600	GET/macphed/finite/feresources/node59.html	200	2042
sabre45.sasknet.sk.ca	26/Sep/1995:23:48:55 -0600	GET/davs/scn/next.gif HTTP/1.0	200	3003
sabre45.sasknet.sk.ca	26/Sep/1995:23:49:04 -0600	GET/davs/scn/scn3.htm HTTP/1.0	200	1136

month, *MON* is the name of the month, *YYYY* is the year, *HH:MM:SS* is the time of day using a 24-h clock, the timezone is −0600. Table 7.7 shows the sample of the HTTP user request.

7.4.2 User and Session Identification

In HTTP user request, the user is identified by either direct hostname or IP address. As discussed in background, pages accessed by users are divided into distinct session through a time-out in session identification. Sessions are identified by two methods described in background: Method 1, where the time between pages requests exceeds 30 min [26] and Method 2, where the average time of visiting Web pages [27] is computed. The statistics of session identification with Method 1 is given in Table 7.8 and with Method 2 is given in Table 7.9. Sessions are identified by calculating the average time of visiting Web pages for our Web navigation framework as Method 1 has a few disadvantages as discussed earlier.

Table 7.8 Statistics of session identification with traditional method (30 min time-out)

	$NASA_{Low}$	$NASA_{Medium}$	$UOFS_{High}$
Total sessions	5117	13298	66886
Average session length	4.55	4.60	5.30

Table 7.9 Statistics of session identification with average time of visiting Web pages

	$NASA_{Low}$	$NASA_{Medium}$	$UOFS_{High}$
Total sessions	5392	13890	70016
Average session length	3.97	4.02	4.37

7.4.3 Experimental Setup

The setup of Web Navigation Prediction Framework (WNPFramework) is as follows. Sessions are created by calculating the average time on visiting Web pages. Fourth-order Markov models are generated by employing sliding windows on the session set. Four classifiers are represented with these prediction models, namely first-, second-, third- and fourth-order Markov models. Each session is then mapped to one or more orders of Markov models. If any session is mapped to more than one Markov model, then pruning/filtering process is initiated to map to only one classifier. Filtering process is conducted based on the PageRank algorithm, i.e. by selecting the classifier which accurately predicts with the maximum incoming page request. If classifiers have equivalent incoming page request, one classifier is selected randomly. SVM Classifiers are generated by applying training on the filtered data. LIBSVM package [30] is used for SVM implementation. C-SVC and RBF kernel are used as values of the parameters svm_type and $kernel\_type$, respectively. For other parameters, default values are used.

Prediction accuracy and time is computed to suggest page recommendation. For prediction, given a session s of length L, prediction is conducted using *(L-1)* gram Markov model. The last page of s is concluded to measure the accuracy of the model. If session s is longer than fourth-order Markov model (as fourth-order Markov model is used in our experiments) then sliding window of size four is applied. For example, suppose $s = A, B, C, D, E, F$ then break s into A, B, C, D, E and B, C, D, E, F. Once SVM classifiers are trained and prediction setup is set, prediction accuracy and time are calculated on test dataset. Finally, SVM classifiers can be used for page recommendation. From this instance, setup is called as *WNPFramework + AvgtimeSession*.

The setup of Two-tier Prediction Framework (*TPFramework*) is as follows. Sessions are created by setting 30 min as a default time-out and by calculating the average time on visiting Web pages. Fourth-order Markov models are generated on both session data. Each session is mapped to one or more orders of Markov models and pruning process is applied by selecting the classifier which accurately predicts with the maximum probability. SVM classifiers are generated on the pruned dataset. Once SVM classifiers are trained, prediction time and accuracy is measured (*TPFramework + TraditionalSession*).

In order to study the effect of sessions created by computing the average time on visiting Web pages, prediction accuracy and time is calculated by two-tier prediction framework again but sessions are created by calculating average time on visiting Web pages and are used as training dataset (*TPFramework* + *AvgtimeSession*).

7.4.4 Results Comparison

In this section, the results of our experiments are presented and discussed. Web Navigation Prediction Framework (*WNPFramework*) and Two-tier Prediction Framework (*TPFramework*) are studied and compared. Experiments have been conducted on 4GB memory and Intel(R) Core(TM)2 Duo P7450@2.13 GHz processor. Dataset used for both *WNPFramework* and *TPFramework* are same as discussed in data collection.

Figures 7.2, 7.3 and 7.4 present a comparison between *TPFramework* and *WNPFramework* in terms of prediction accuracy by using different training dataset percentages on all the three datasets. The average accuracy for *WNPFramework* is increased to 44.07, 11.10 and 4.51% for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets, respectively, in comparison to *TPFramework*. It is observed from Fig. 7.3 that accuracy increases for both the frameworks with the increase in the size of the training dataset. But for $NASA_{Medium}$ dataset, at 50 and 60% training dataset, accuracy decreases for *TPFramework*. In *TPFramework*, sessions are created with the traditional method with 30 min time-out. It is observed in $NASA_{Medium}$ dataset that the actual user sessions are more than 30 min, hence these sessions have misclassified classifiers and accuracy is decreased. In *WNPFramework*, sessions are created

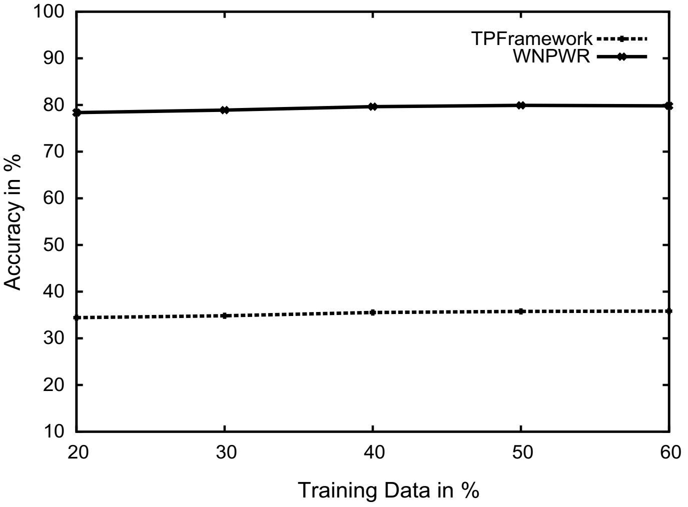

Fig. 7.2 Prediction accuracy comparison between TPFramework and WNPFramework for $NASA_{Low}$

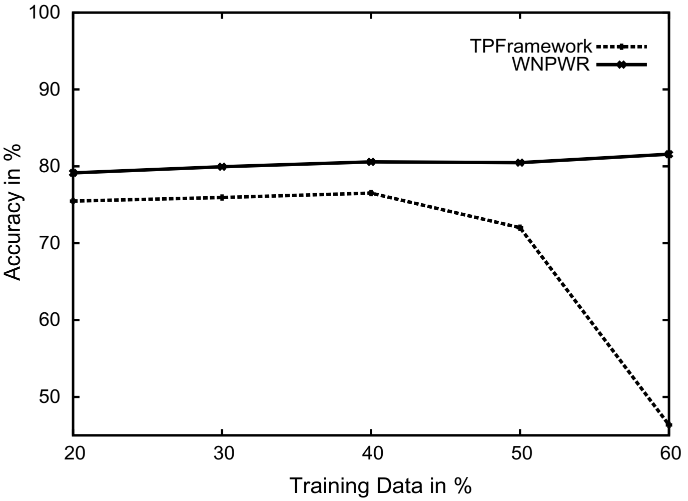

Fig. 7.3 Prediction accuracy comparison between TPFramework and WNPFramework for $NASA_{Medium}$

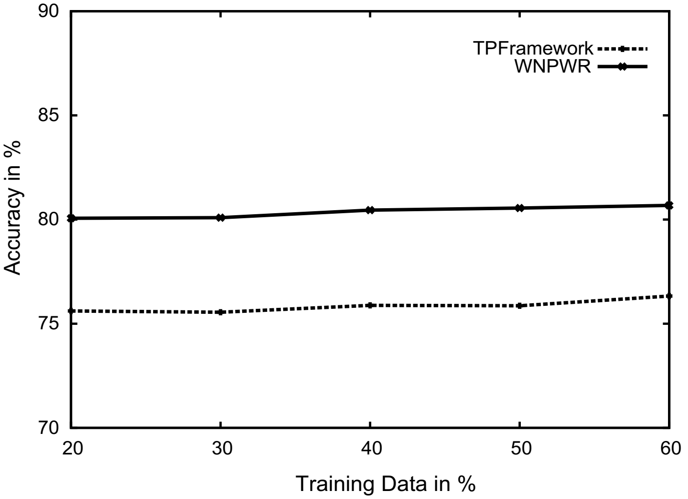

Fig. 7.4 Prediction accuracy comparison between TPFramework and WNPFramework for $UOFS_{High}$

by calculating the average time of visiting Web pages, hence accuracy increases with an increase in training datasets.

Figures 7.5, 7.6 and 7.7 present a comparison between *TPFramework* and *WNPFramework* in terms of prediction time by using different training dataset percentages on all the three datasets. It is observed from Figs. 7.5, 7.6 and 7.7 that

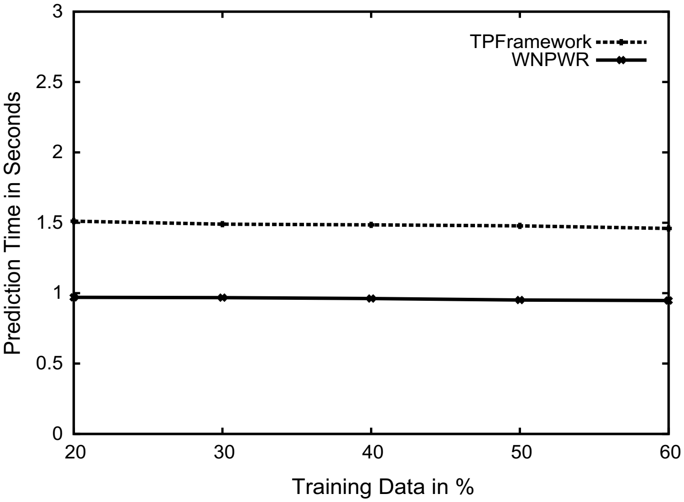

Fig. 7.5 Prediction time comparison between TPFramework and WNPFramework for $NASA_{Low}$

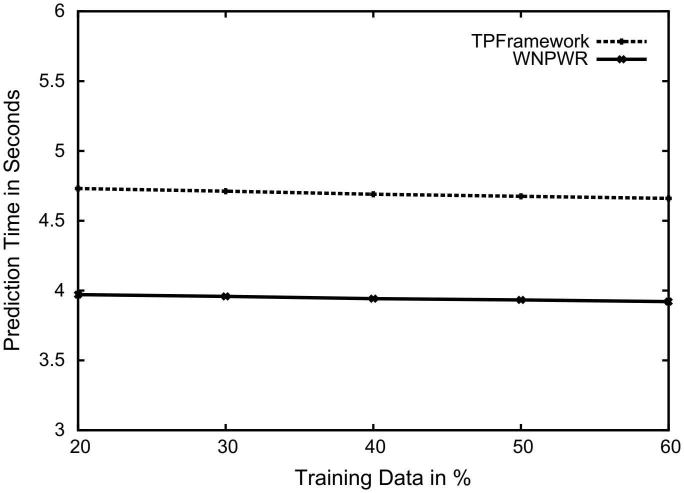

Fig. 7.6 Prediction time comparison between TPFramework and WNPFramework for $NASA_{Medium}$

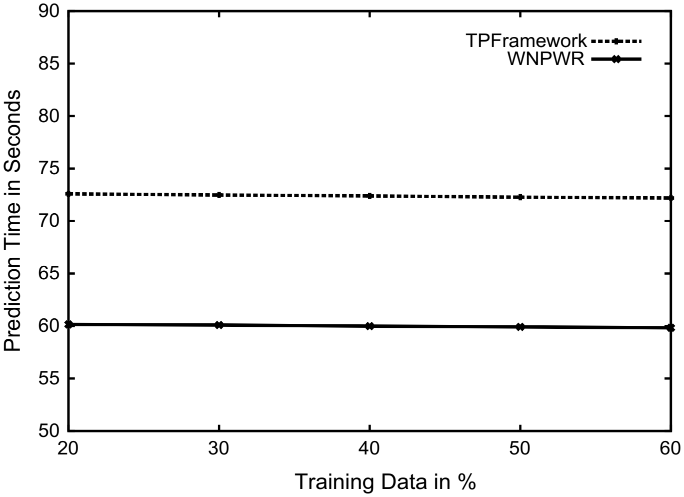

Fig. 7.7 Prediction time comparison between TPFramework and WNPFramework for $UOFS_{High}$

Table 7.10 Prediction accuracy comparison of TPFramework for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets with traditional and average time sessions

		Training data in %				
		20	30	40	50	60
$NASA_{Low}$	TPFramework [15] + Traditional session	1.511	1.490	1.485	1.478	1.460
	TPFramework [15] + Average time session	0.982	0.971	0.965	0.955	0.950
$NASA_{Medium}$	TPFramework [15] + Traditional session	4.731	4.712	4.690	4.675	4.660
	TPFramework [15] + Average time session	3.982	3.970	3.962	3.951	3.941
$UOFS_{High}$	TPFramework [15] + Traditional session	72.591	72.474	72.391	72.265	72.198
	TPFramework [15] + Average time session	60.888	60.781	60.677	60.591	60.481

prediction time decreases with increase in training datasets and decrease in testing datasets. The average prediction time for *WNPFramework* is decreased to 35.35, 15.94 and 17.11% for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets, respectively, in comparison to *TPFramework*.

Table 7.10 presents prediction accuracy comparison between *TPFramework* and *WNPFramework* for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets with traditional and average time sessions. It is observed from Table 7.10 that the average

Table 7.11 Prediction time in seconds comparison of TPFramework for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets with traditional and average time sessions

		Training data in %				
		20	30	40	50	60
$NASA_{Low}$	TPFramework [15] + Traditional session	34.431	34.838	35.520	35.774	35.838
	TPFramework [15] + Average time session	78.366	78.899	79.658	79.919	80.217
$NASA_{Medium}$	TPFramework [15] + Traditional session	75.483	75.943	76.520	72.025	46.367
	TPFramework [15] + Average time session	79.145	79.950	80.587	80.478	81.578
$UOFS_{High}$	TPFramework [15] + Traditional session	75.613	75.557	75.883	75.863	76.338
	TPFramework [15] + Average time session	80.061	80.094	80.457	80.552	80.681

accuracy is increased to 44.12, 11.08 and 4.51% for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets, respectively, for *TPFramework + AvgtimeSession* in comparison to *TPFramework + TraditionalSession*.

Table 7.11 presents prediction time (in seconds) comparison between *TPFramework* and *WNPFramework* for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets with traditional and average time sessions. It is observed from Table 7.11 that the average prediction time is decreased to 35.03, 15.60 and 16.16% for $NASA_{Low}$, $NASA_{Medium}$ and $UOFS_{High}$ datasets, respectively, for *TPFramework + AvgtimeSession* in comparison to *TPFramework + TraditionalSession*.

The major difference between our proposed framework and two-tier framework is the methodology of session identification (in our framework Method 2 which is discussed in user and session identification section). From Table 7.9, it can be observed that Method 2 has generated more number of sessions compared to Method 1 which is used as training dataset. In Method 2, the average visiting time depends on the visiting Web pages, so sessions are mapped to a realistic value than when a single constant value is used. *TPFramework* and *WNPFramework* with AvgtimeSession results in high prediction accuracy on account of these reasons. Even the average session length is small in Method 2 compared to Method 1, that results is lower prediction time. It is also observed that session generation time with Method 2 is more compared to Method 1.

In *WNPFramework*, if any session is mapped to more than one Markov models, then filtering is used by selecting the classifier which accurately predicts with the maximum incoming page request. In the two-tier framework, filtering is used by selecting the classifier which accurately predicts with the maximum probability. For example, user *X* has visited different Web pages in different sessions as shown in Table 7.4. For the test session *(A, B, C, D)*, two classifiers *E* and *F* are suggested.

During filtering E is selected according to the highest probability of classifier, while F is selected as there is more traffic towards F according to the highest incoming page request. *TPFramework* and *WNPFramework* have been compared with AvgtimeSession and it is observed that there is not much difference in prediction accuracy, but the overall prediction time is lower in *WNPFramework* in comparison to *TPFramework*.

7.5 Summary

In this chapter, we have proposed Web Navigation Prediction Framework for Web page Recommendation which creates and generates a classifier based on sessions as training examples. Sessions are created by calculating the average time on visiting Web pages, which maps to the realistic better value than when a single constant value is used. Each session is mapped to one or more generated classifiers and filtering process is applied by the PageRank algorithm. Simulations are performed on UOFS and NASA dataset and are compared with two-tier prediction framework. The WNPWR algorithm outperforms two-tier prediction framework by providing high prediction accuracy with reduced prediction time. Further, it is planned to extend this chapter for online Web page recommendations for mobile applications.

References

1. S. Chimphlee, W. Chimphlee, N. Salim, M.S.B. Ngadiman, Using hybrid markov model for web access prediction. J. Inf. Technol. **3**(3), 86–91 (2012)
2. J. Borges, M. Levene, Evaluating variable-length markov chain models for analysis of user web navigation sessions. IEEE Trans. Knowl. Data Eng. **19**(4), 441–452 (2007)
3. M. Deshpande, G. Karypis, Selective markov models for predicting web-page accesses. ACM Trans. Internet Technol. **4**(2), 163–184 (2004)
4. A. Guerbas, O. Addam, O. Zaarour, M. Nagi, A. Elhajj, M. Ridley, R. Alhajj, Effective web log mining and online navigational pattern prediction. J. Knowl. Based Syst. **49**(2), 50–62 (2013)
5. C. Dimopoulos, C. Makris, Y. Panagis, E. Theodoridis, A. Tsakalidis, A web page usage prediction scheme using sequence indexing and clustering techniques. J. Data Knowl. Eng. **69**(4), 371–382 April (2010)
6. C.-H. Lee, Y.-L. Lo, Y.-H. Fu, A novel prediction model based on hierarchical characteristics of web site. Int. J. Expert. Syst. Appl. **38**(4), 3422–3430 (2011)
7. K. Dembczyński, W. Kotłowski, M. Sydow, Effective prediction of web user behavior with user-level models. J. Fundam. Inform. **89**(3), 189–206 (2008)
8. C. Liu, R.W. White, S. Dumais, Understanding web browsing behaviors through Weibull analysis of dwell time, in *The Proceedings of 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10* (2010), pp. 379–386
9. F.K.H. Phoa, J. Sanchez, Modelling the browsing behavior of world wide web users. Open J. Stat. **3**(2), 145–154 (2013)
10. R.W. White, P. Bailey, L. Chen, Predicting user interests from contextual information, in *The Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval* (2009), pp. 363–370

11. P. Thwe, Proposed approach for web page access prediction using popularity and similarity based pagerank algorithm. Int. J. Sci. Technol. Res. **2**(3), 240–246 (2013)
12. F. Khali, J. Li, H. Wang, Integrating recommendation models for improved web page pediction accuracy, in *The Proceedings of the 31st Australasian Conference on Computer Science, ACSC '08* (2008), pp. 91–100
13. M.A. Awad, L.R. Khan, Web navigation prediction using multiple evidence combination and domain knowledge. IEEE Trans. Syst., Man Cybern.-Part A: Syst. Hum. **37**(6), 1054–1062 (2007)
14. R. Dutta, A. Kundu, D. Mukhopadhyay, Clustering-based web page prediction. Int. J. Knowl. Web Intell. **2**(4), 257–271 (2011)
15. M.A. Awad, I. Khalil, Prediction of user's web-browsing behavior: application of markov model. IEEE Trans. Syst., Man Cybern.-Part B: Cybern. 42(4), 1131–1142 (2012)
16. V.S. Tseng, K.W. Lin, Efficient mining and prediction of user behavior patterns in mobile web systems. J. Inf. Softw. Technol. **48**(6), 357–369 (2006)
17. G. Zhao, W. Lai, Predicting user behavior in mobile internet based on random walk. J. Comput. Inf. Syst. **9**(22), 9157–9164 (2013)
18. C.-M. Huang, J.J.-C. Ying, V.S. Tseng, Mining users' behaviors and environments for semantic place prediction, in *Mobile Data Challenge Workshop* (2012)
19. M. Silic, G. Delac, I. Krka, S. Srbljic, Scalable and accurate prediction of availability of atomic web services. IEEE Trans. Serv. Comput. **7**(2), 252–264 (2014)
20. J. Huang, R.W. White, Parallel browsing behavior on the web, in *The Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, HT'10* (2010), pp. 13–18
21. S. Goel, J.M. Hofman, M.I. Sirer, Who does what on the web: a large-scale study of browsing behavior, in *The Proceedings of 6th AAAI International Conference on Weblogs and Social Media, AAAI' 12* (2012), pp. 130–137
22. Z. Cheng, B. Gao, T.-Y. Liu, Actively predicting diverse search intent from user browsing behaviors, in *The Proceedings of 19th International Conference on World Wide Web, WWW '10* (2010), pp. 221–230
23. Y. Zhang, W. Chen, D. Wang, Q. Yang, User-click modeling for understanding and predicting search-behavior, in *The Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11* (2011), pp. 1388–1396
24. X. Tian, Y. Lu, L. Yang, Query difficulty prediction for web image search. IEEE Trans. Multimed. **14**(4), 951–962 (2012)
25. J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding. IEEE Trans. Image Process. **23**(5), 2019–2032 (2014)
26. R. Cooley, B. Mobasher, J. Srivastava, Data preparation for mining world wide web browsing patterns. J. Knowl. Inf. Syst. **1**(1), 5–32 (1999)
27. C.E. Dinuca, D. Ciobanu, Improving the session identification using the mean time. Int. J. Math. Model. Methods Appl. Sci. **6**(2), 265–272 (2012)
28. S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine. J. Comput. Netw. **56**(18), 3825–3833 (2012)
29. Internet Traffic Archive, http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html
30. C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–39 (2011)