An Efficient Web-Page Recommender System using Frequent Pattern Discovery and Dynamic Markov Models

Tich Phuoc Tran[1], Thi Thanh Sang Nguyen[1], Kien Cuong Dang[2], Xiaoying Kong[1]
[1]Faculty of Engineering, Information Technology
University of Technology, Sydney
{Tich.Tran@uts.edu.au, tsang@it.uts.edu.au, Xiaoying.Kong@uts.edu.au}

[2]Faculty of Information Technology
Nong Lam University, HCMC
dkcuong@hcmuaf.edu.vn

ABSTRACT: *The Internet has recently become not only one of the most popular communication channels but also the most accessible and searchable information repository of different domains. Billions of users surf the Internet everyday to search for information or visit social network and e-commerce websites. The web usage behaviors of these users can be analyzed by Web Usage Mining (WUM) systems to discover useful knowledge that helps improve service performance. Despite recent successes, existing WUM systems still cannot cope with the growing dynamics and complexity of the Web, resulting in overwhelming overheads and low efficiency. In this paper, an innovative Web-Page Recommender System is proposed to model user web browsing behaviors, extract popular web paths and predict web navigation possibilities. Particularly, the main inference algorithm in this system integrates advanced Frequent Pattern Recognition methods and Stochastic Markov Models to achieve an optimal balance between superior predictive accuracy and excessively demanding computation in higher order models. Empirical analysis suggests that our system outperforms other conventional methods with respect to complexity reduction and accuracy improvement.*

Keywords: Web Usage Mining, Frequent Pattern Recognition, Markov models

Received: 5 September 2010, Revised 13 October 2010, Accepted 16 October 2010

1. Introduction

As many businesses are conducted online and more people use the Internet as a cheap and effective way to communicate and share information with each other, a tremendous volume of web usage data is collected by web servers and ready for further analysis. Such data encapsulated in log files contain not only simple user sessions, but also useful information which helps trace web usage patterns in relation to browsing behavior and recommend relevant web pages to users. To extract useful insights and actionable knowledge from this data, a Web Usage Mining (WUM) process can be used which implements advanced Knowledge Discovery techniques including clustering, classification, relationship mining and temporal sequence mining. Many WUM systems have been recently developed to better analyze web log data with highly advanced technologies. One of the most commonly used learning methods in WUM is Markov-based models [1-3]. However, there is a trade-off between predictive accuracy and model complexity which makes existing Markov-based models computationally extensive and in some cases, subject to overfitting problems [4]. In this paper, we propose an extension to conventional Markov models using advanced frequent pattern mining techniques. This algorithm is then implemented in a Web-Page Recommender

System to model web usage behaviors and predict next page to be visited by a user. This system is designed with a focus on scalability and processing efficiency. A number of benchmarking datasets are deployed to verify the effectiveness of our method and compare with other learning algorithms.

This paper is organized as follows. Section 2 introduces Web Mining and reviews some existing research related to Markov-Based learning models and sequence mining for Web applications. Section 3 explains the proposed WUM system including its motivations, relevant concepts and technologies. Section 4 presents experimental analysis conducted on benchmarking data to compare our methods and other algorithms. Section 5 concludes the paper and points out the future work.

2. Web Mining and Related Works

2.1 Web Mining

Web Mining is defined as a process of extracting potentially useful patterns and implicit knowledge from artifacts or activities related to the web structure, web content and web usage behaviors. This process applies advanced Data Mining, Machine Learning and high end data visualization technologies. There are three major branches from Web Mining research: (1) Web Content Mining processes published HTML (semi-structured), plaintext (unstructured) or XML documents (structured); (2) Web Structure Mining investigates hyperlink architecture and (3) Web Usage Mining analyzes user interactions with a web server, using web logs, click-stream and transactional data.

This paper focuses on recent advances in Web Usage Mining (WUM). Unlike other fields of Web Mining, WUM concerns with Web browsing patterns and correlations between the web pages that supports server performance improvement, hypertext structure optimization, web personalization, traffic analysis, and targeted advertising. Personalization is one of the most widely researched areas in WUM. This personalization characteristic can be achieved through the development of adaptive websites that automatically change their organization and navigational presentation according to user preferences [5]. Some typical applications of WUM include clustering web users [6], mining conceptual link hierarchies from web log files for adaptive website navigation [7], building frequent web access sequences [8] and predicting web navigations [9].

One of the early WUM systems is the Analog system [10] which comprises an offline component that clusters past user data and an online component which classifies active user sessions into identified clusters. Another typical WUM system is the WebWatcher system [11] which suggests hyperlinks to users based on their interests. This recommender system uses keyword-based method to identify user interest and information of selected hyperlinks and the site structure. Similarly, a WUM system in [12] also takes into account the site topology to cluster related hyperlinks. It proposes the Association Rule Hypergraph Partitioning technique with a fixed-width sliding window of current active sessions. Though these systems obtain a certain level of accuracy, they suffer from low scalability and efficiency in the case of large amount of data to be processed.

As a result of an increasing focus on human-centric web applications, many researches have been devoted to study web personalization techniques with new ideas recently emerged, such as flexibly combining data mining techniques, performance of mining process is significantly improved, and considering effective input data. These approaches are important but have not yet discovered much in previous works. A hybrid web personalization model was introduced in [13] that used the *localized connection measure* (LCM) to select amongst possible recommender models the most adaptive model to individual user's preferences. By flexibility of switching between recommender systems including such as Association Rules, Sequential Patterns, and Contiguous Sequential Pattern at each pageview in the active session, the system was found to obtain the highest precision and coverage.

2.2 Hybrid Markov-Based WUM Models

There has been a great deal of WUM research that uses Markov models to capture the sequential relationships hidden in web navigation histories and to estimate the probability of visiting web pages. Each web page is referred to as a state in the Markov model. The N-order Markov model predicts the next page to which a user most likely to navigate based on the previous N-1 visited pages. The higher the order of a Markov model is, the more accurate predictive capability it will get. However, its number of states also exponentially increases for higher order models which consequently lead to excessive model complexity, especially for a website with huge number of pages. Assume that the web structure is not at optimal configuration, there are usually a significant number of inactive, redundant or irrelevant pages included in the Markov model. Therefore, it is imperative to filter out such pages to remedy the complexity drawbacks of Markov model. Several works have

been proposed to construct hybrid Markov models with a hope to alleviate this problem. For example, a WUM system in [2] combines the Markov model with association rules and clustering techniques in order to enhance mining efficiency. This model is reported to predict accurately and have less state space complexity and fewer generated rules than the original Markov model. However, its performance depends on several uncertain factors such as the hyperlink structure and the support and confidence thresholds. A different approach is developed in [3] in which the first order Markov model is implemented using the Expectation-Maximization algorithm and the resultant model is then integrated with an adaptive model-based clustering method. Though the predictive accuracy is very comparative between these two models, they both share the scalability problem which is popular for Web Mining due to the vast amount of noisy and redundant data.

Recent novel Markov-based web mining systems often combine Markov models of different prediction levels which correspond to each state level, and then produce output symbols at each state transition. These output symbols are essential to discover user group interest path patterns. A hybrid Markov model was proposed in [14] for mining interest navigation patterns, which obtained high performance with respect to prediction overlay and correct rate in Markov model, and optimal navigation path and time when moving from one page to another. In this model, the interest path patterns were computed based on interest keywords determined by user access time length. Therefore, this approach achieves higher accuracy of predicting interest navigation patterns than traditional Markov models, and overcomes the state-space complexity of Markov models. However this hybrid Markov model has not yet completely resolved redundant data issues, making the computation of output symbols and interest patterns still complex.

2.3 Sequence Mining for Web Applications

In this Section, a family of sequence mining techniques will be discusses in details with special focus put on two typical algorithms, namely Generalized Sequential Pattern (GSP) [15] and Sequential pattern Discovery using Equivalence classes (SPADE) [16]. These techniques are commonly used in a number of web mining applications [17, 18].

Firstly, a sequence is simply an ordered list of item sets. The length of a sequence is the total number of item occurrences. Let S_A and S_B respectively denote two sequences $<A_1, \ldots, A_n>$ and $<B_1, \ldots, B_n>$ where A_i and A_j are item sets and $m \geq n$. If there exist integers $i_1<i_2<...<i_n$ such that $A_1 \subseteq B_{i_1}$,…, and $A_n \subseteq B_{n_1}$, it is said that S_A is subsequence of S_B or $S_A \prec S_B$.

2.3.1 Generalized Sequential Pattern (GSP)

GSP [15] discovers all frequent items (not item sets) by making multiple passes over the database. For each iteration, a two-phase process is executed.

- Candidate generation (generate-and-test)
  - Joining (k-1)-sequences with themselves to generate k-sequence. This is based on the anti-monotone property that all the subsequences of a frequent sequence must be frequent.
  - Pruning by removing candidate sequences whose subsequence is not frequent
- Support Counting (hash tree-based search)

This process is repeated until there is no frequent sequence left, or there is no candidate sequence generated.

$Freq\_seq\_List_1$:=Frequent atoms;
For $k=2$; $Freq\_seq\_List_k != empty$; $k = k + 1$;
C_k = set of candidates
For all records in database do
Increment count of $X \in C_k$ contained in S
End for
$Freq\_seq\_List_k = \{X \in C_k | support(X) > threshold\}$
End for
Set of all frequent sequences: $= \bigcup_k Freq\_seq\_List_k$

Table 1. GSP Algorithm

2.3.2. Sequential Pattern Discovery using Equivalence classes (SPADE)

Besides the high accuracy of GSP algorithm, it is rather computationally expensive to decompose sequences for counting their supports. Additionally, GSP normally results in a large number of candidate sequences. To overcome this problem, a novel sequence mining algorithm, namely Sequential pattern Discovery using Equivalence classes (SPADE) [16], is proposed. SPADE is faster and more efficient than GSP because it decomposes the problem (i.e. search space) into smaller sub-problems and then deploy parallel search in sub-search spaces. It first discovers frequent 1-sequences, and 2-sequences. A 2-phase *decomposition* and *enumeration* process is then executed:

- Decomposition of search space into equivalence classes and
- Enumeration of frequent sequences within each class

Another feature of SPADE that improves its performance is the introduction of the lattice concept to divide the candidate sequences into groups by items such that each group can be completely stored in the main memory. In addition, SPADE uses ID-list to reduce the costs for computing support counts. ID-list of a sequence keeps a list of pairs which indicate the positions that sequence appears in the database. In a pair, the first value identifies which *customer* has the sequence and the second refers to a *transaction* of that customer which contains the last item set of the sequence. A typical side effect of using ID-list is that it may be costly to repeatedly merge the ID-lists for a large number of frequent candidate sequences.

P_1 = all frequent atoms with their ID-list

P_2 = all frequent 2-sequences with their ID-list

E = All equivalence classes of atoms $\in P_1$

For all $[x] \in E$ do

  Construct pattern lattice of [X]

  Explore frequent sequences in [X] by

    Using either Depth first or Breath first search

End for

Table 2. SPADE Algorithm

3. The New WUM System

The first-order Markov models (Markov Chains) provide a simple way to model sequential relationships between states with only a single variable (current state) by computing the probability of each state and of transition between states. However, these models assume that the next state is determined by a function of the current states and therefore, neglect the "long-term memory" aspect of web usage behaviors [19]. To better capture dynamic and complex behaviors of web users, higher-order Markov models are developed with higher predictive accuracy for navigational paths, significantly increasing model coverage, however, at the price of exponential increase in state-space complexity. Moreover, such complex models may be susceptible to overfitting problem, affecting the overall system robustness and limiting their applicability to real-time and data-extensive (e.g. concerned with large state space) web mining tasks [19]. Several mixture approaches are also studied to combine Markov models of different orders. Unfortunately, such mixtures require excessive data and computation resources for processing and training [20]. This incompatibility of model coverage and model complexity motivates our research to seek for a compromised solution for this problem.

This system comprises four main components: (1) *the preprocessing module* that cleans the raw web log data, identifies user profiles and constructs usage sequences; (2) *the frequent pattern discovery module* that derives frequently-happened sequences; (3) *the predictive module* that models web usage behaviors and (4) *the recommendation module* that suggests possible next navigation steps based on user interests. The details of these modules are presented below:

[1] http://sol.cs.unwindsor.ca/~cezeife/webcleaner.tar.gz

3.1 Preprocessing Module

Web log data is a log file which records every request from a user's browser a website on a Web server. The format of the log file is the *common log format* (CLF or "clog") [21] includes the following fields: remote host field, date/time field, HTTP field,status code field, etc. Because web data is highly noisy due to wide range of data types and communication protocols allowed in the web environment, careful data preparation methods are imperative to facilitate the discovery of browsing patterns.

- *Data cleaning*: remove erroneous and invalid pages non-html files (multimedia and scripts) from the raw web log data. The WebCleaner[1] tool is used for this purpose.

- *User profile identification*: filter IP addresses, session ID based on a predefined timeout between two subsequent requests from the same user.

- *Web usage sequence construction*: for each session, a sequence of visited pages is constructed by encoding these pages with sequence item code. The resultant sequences are stored in the Web Access Sequence Database (WASD).

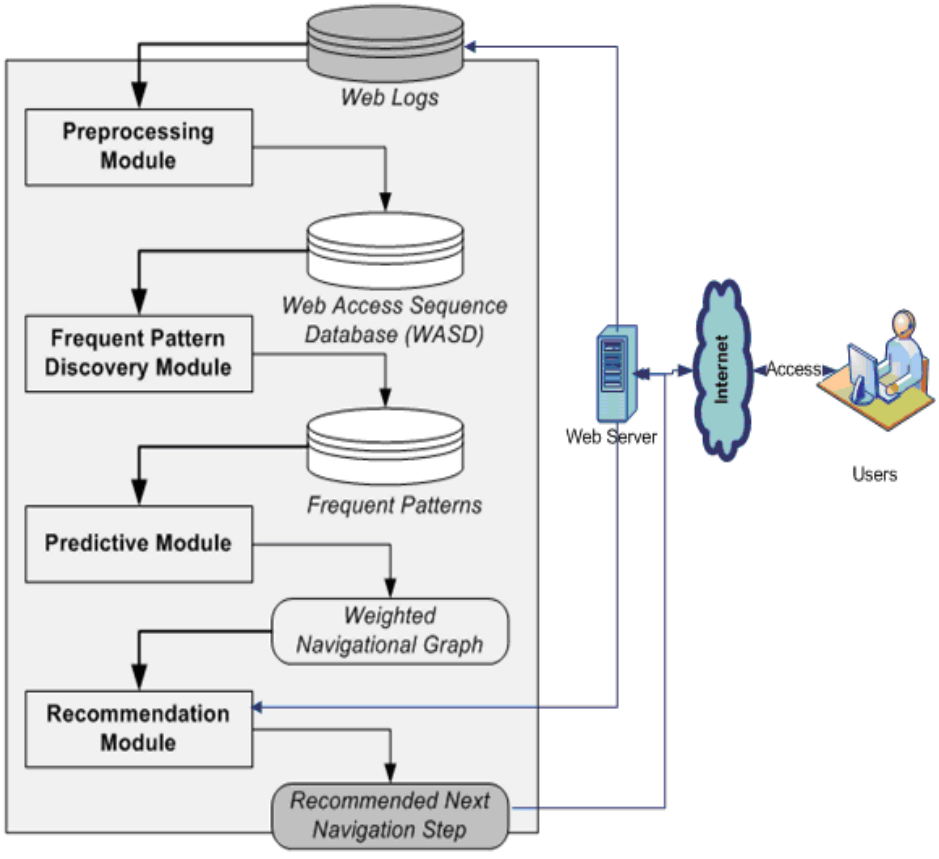

Figure 1. A novel WUM System

3.2 Frequent Pattern Discovery Module

This module utilizes a frequent pattern discovery algorithm to reduce the state-space for later Markov predictive module. In particular, we adapt a pre-order tree-based learning method [8] which is known to be robust to crawling attacks against navigation-based Web recommender systems [22]. Such attacks are generated by the crawling mechanisms in which fake user profiles are injected to clickstream navigation to manipulate the future behavior of the recommender system.

Let *I* be the set of all web pages and *S* be the set of all sequences. A *sequence* is an ordered list of subsets of *I*. A sequence

is denoted as $(\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_n)$ where each sequence element α_i is a subset of *I*. The length of sequence α is *n* and its *width* is the maximum size of any α_i for $1 \leq i \leq n$.

A sequence α is said to be *subsequent* of β, denoted as $\alpha \prec \beta$ if there exists integers $i_1 < i_2 < \ldots < i_n$ such that $\alpha_j \subseteq \beta_{i_j}$ for all α_j.

Let $FS = \{FS_1, FS_2, \ldots FS_m\}$ be the set of frequent sequences derived from *S*. A sequence FS_i is said to be frequent if $\sigma(FS_i) > \sigma_{min}$ where support is computed as follow:

$$\sigma(FS_i) = \frac{|\alpha \in S : FS_i \prec \alpha|}{|S|}$$

and $0 < \sigma_{min} < 1$.

We also consider other candidates for this module, including advanced frequent pattern recognition methods such as SPADE [16] and GSP [23]. Extensive experiments are presented in Section 4 to compare the effects of these algorithms on the overall system performance.

This module analyzes the frequent patterns generated by the previous module by implementing a dynamic clustering-based second-order Markov model [24]. Let $p_{i,kj}$ be the second-order probability of the transition (A_k, A_j) given that the previous transition was (A_i, A_k). The second-order probabilities are estimated as follows:

$$p_{i,kj} = \frac{w_{i,j,k}}{w_{i,k}}$$

where $w_{i,j,k}$ and $w_{i,k}$ correspond to weights of the transitions (A_i, A_j, A_k) and (A_i, A_k) respectively.

Output of this modeling process is the co-occurrence array *P* of all conditional probability (second-order) that $p_{i,kj}$ page *j* is visited given page *i* and page *k* are the previous and current pages respectively, in the same session. This array is then transformed into a directed navigational graph *G*, as depicted in Figure 2 which represents frequently visited pages as state nodes and the links between them are associated with a weight equal to the corresponding $p_{i,kj}$.

By introducing the concept of state cloning to duplicate states and using model-based clustering technique, this dynamic Markov model can achieve second-order accuracy with less additional states than a conventional second-order Markov model [24]. In particular, a state is cloned only when it is an inaccurate state, i.e. the difference between its first-order and second-order probabilities is greater than a certain threshold γ. This restriction is to leverage the performance of these states without overwhelming the total number of states. A k-mean clustering method is then deployed to assign in-links with similar second-order probabilities to the same clone. Essentially, at this point, users with similar navigation patterns are put into same clusters, each of which is represented by a Markov model.

3.4 Recommendation Module

This Recommendation Module suggests to web users the next navigation step which is potentially of their interests. Such recommendation is generated automatically by grouping active users based on their current states and their recorded previous states within the same session. The inference of the users' next possible state is conducted using the co-occurrence array *P* computed from the previous module.

In our model, at each step, a user may change his/her state from the current state to another state (or remain in the same state) according to a probability distribution. The changes of state are called transitions, and the probabilities associated with various state changes are called transition probabilities. The recommendation generated by this module is a stochastic

process, means that all state transitions are probabilistic rather than an absolute recommendation. This stochastic recommendation is visualized in a Recommended Next navigation Step diagram, as illustrated in Figure 3.

3.5. System Features

The proposed WUM system is developed based on a modular framework in which additional plug-in components can be used without the need of significant configuration changes. For instance, the pre-order tree-based algorithm in the Frequent Pattern Discovery Module can be replaced easily by other algorithms such as SPADE or GSP and the system will perform accordingly.

The key virtue of this system is that it attempts to further alleviate the lack of scalability of current Markov-based WUM systems, at the same time, enhance predictive accuracy. When the amount of data to be processed grows, the number of Markov states increases nonlinearly. This effect inevitably slows down the Markov-based WUM systems' performance and harmers their accuracy. An attempt to lift up the accuracy of Markov models by making them higher orders will also lead to the same problem of large number of additional states. Compared with a conventional second-order Markov model, our Predictive Module can achieve a comparative performance with less computation required due to the effect of conditional cloning and dynamic clustering component. Moreover, our system models web navigation patterns using only the frequently accessed web pages extracted from the web log rather than using all available pages as in [24] and other hybrid methods. As a result, we can significantly reduce the number of states and hence model complexity. This complexity reduction is contributed by the Frequent Pattern Discovery Module.

4. Experimental analysis

4.1 Design

The WUM system proposed in this paper uses a tree-based method for frequent pattern discovery and a dynamic clustering-based Markov model mining web navigation behaviors. We abbreviate this as a Frequent-Pattern Dynamic-Markov WUM system, or *FP-DM WUM system*.

To evaluate the effectiveness of our WUM system, two benchmarking datasets are tested, namely NASA and Kent. These publicly accessible data[1] are HTTP requests to the web servers at the Kennedy Space Center (USA) and Kent State University (USA) respectively. The experiments were run on a PC with Intel Core 2 Duo E8400 processor, 2.99 GHz and 3.25 GB of RAM.

Each dataset is input to the four modules in our proposed WUM system for preprocessing, frequent pattern discovery, web usage behavior modeling and recommendation respectively. From the conducted experiments, two performance metrics are used, including model complexity and model efficiency. In particular, processing time and memory requirement are used as model complexity indicators while the number of states resulted from our Predictive Module suggest the computational efficiency of the system.

In the experiments, from our proposed system using the tree-based frequent pattern discovery method (*FP-DM WUM tree-based*), we consider two variants of the proposed system in which SPADE (*FP-DM WUM SPADE*) and GSP (*FP-DM WUM GSP*) are used for frequent pattern discovery instead of the tree-based method. These variants are then compared against Borges' WUM system [24] which uses dynamic clustering-based Markov.

4.2 Results

a) Preprocessing

The datasets NASA and Kent are Web log files including the following fields: remote host field, date/time field, HTTP field, and status code field. We first apply preprocessing on these datasets as described in Section 3.1 to obtain the Web Access Sequence databases. The prepared data is detailed in the Table 3.

b) Model Efficiency

For the NASA dataset, amongst FP-DM WUM variants, the tree-based and GSP approaches are the best performing models

[2] http://www.web-caching.com/traces-logs.html

Data	# of users	# of sessions	# of Web pages	Period (# of days)
NASA	26,037	49,406	1,446	13
Kent	4,472	8,412	7,134	6

Table 3. Benchmarking Datasets

which extracted 50 frequent patterns out of 1,446 pages in total. The patterns extracted from the two approaches are same shows that these results are correct and reliable. These frequently accessed pages are then transferred to 67 states in the resultant Markov model. Compared with 2,352 states generated by the Borges' WUM system, our approach achieves much lower number of states, including all variants of FP-DM WUM system (e.g. variants using Tree-based, SPADE and GSP). A similar observation can be drawn from the Kent dataset with the tree-based method to achieve the lowest number states.

c) Model Complexity

Table 5 displays the processing time and memory required by the tested systems.

	Borges's WUM system	FP-DM WUM (Tree-Based)	FP-DM WUM (SPADE)	FP-DMWUM (GSP)
NASA data (1,446 pages) $\gamma = 0.1, \sigma_{min} = 0.01$				
# of Freq. Pages	NA	**50**	49	**50**
# of States	2,351	**67**	115	**67**
KENT data (7,134 pages) $\gamma = 0.1, \sigma_{min} = 0.01$				
# of Freq. Pages	NA	**175**	210	**175**
# of States	13,205	**193**	415	**193**

Table 4. Number of Frequent Pages and State

In general, any variant of the FP-DM WUM that uses Tree-based, SPADE and GSP can achieve more complexity reduction than Borges's model. Moreover, the FP-DM WUM (tree-based) approach is the most preferable system with lowest processing time and memory required for both NASA and Kent datasets.

4.3 Discussion

From our experiments, *FP-DM WUM (Tree-Based)* and *FP-DM WUM (GSP)* performed equally well with respect to the number of derived frequently visited pages and the resultant Markov states. However, the tree-based method ran faster and required less memory than the GSP-based method. It was also observed that conventional Markov-based WUM system generally generated much more states compared with our approach. As a result, the processing time and memory of such a system dramatically increases, making the state-space of the Markov model too large to be practical. Unlike other existing WUM systems which merely implement Markov modeling, our system can derive the frequently navigated pages. These characteristics all together make our system relatively efficient compared with others.

The outcome of the Predictive Module is a graph visualizing the weighted transitions between frequently visited pages. Figure 2 displays such a graph for the case of FP-DM WUM (tree-based) with 50 frequent pages extracted from the NASA dataset. This graph is then used to make recommendation for a web user. By entering his/her current page number into the tool interface, a recommended navigation step is generated by the Recommendation Module as in Figure 3. In this graph, the two previous states are located at the top, associated with second-order probabilities and the number of hits. The current state is in the middle of the graph which is linked to three recommended states below it. For each recommended link, a weight

is attached that represents the likelihood of a page to be visited in the next step. By analyzing web navigation patterns, administrators can identify redundant materials or reorganize the hypertext structure to recommend to the web users the most interesting pages.

	Borges's WUM system	**FP-DM WUM (Tree-Based)**	**FP-DM WUM (SPADE)**	**FP-DMWUM (GSP)**
NASA data (1,446 pages) $\gamma = 0.1, \sigma_{min} = 0.01$				
Processing Time (secs)	55	**18**	32	62
Memory Requirement (MB)	356	**117**	203	170
KENT data (7,134 pages) $\gamma = 0.1, \sigma_{min} = 0.01$				
Processing Time (secs)	198	**39**	150	156
Memory Requirement (MB)	421	**165**	228	263

Table 5. Processing Time and Memory Requirement

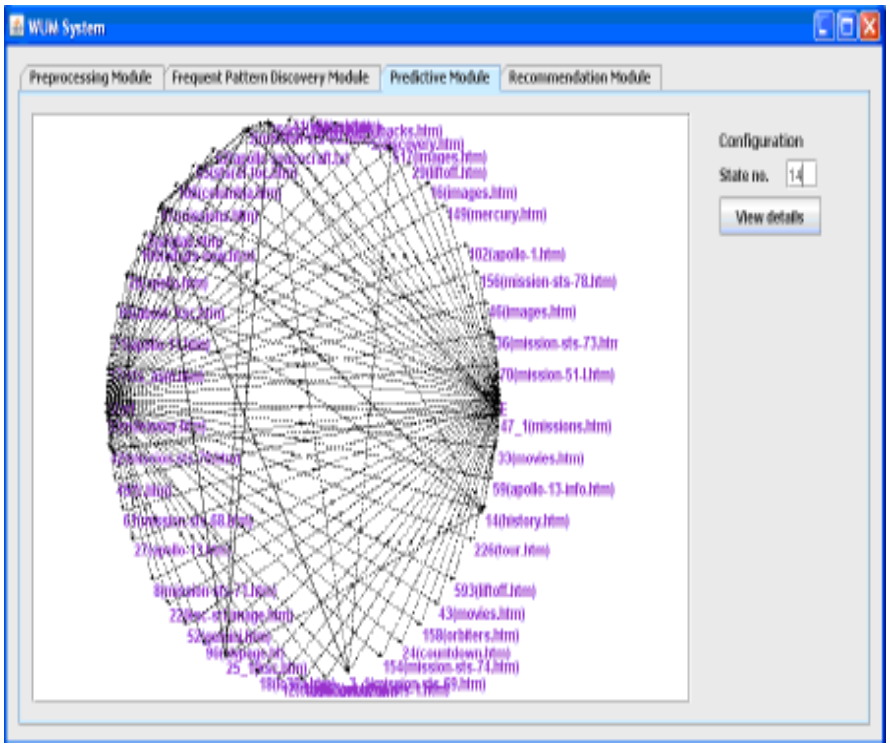

Figure 2. Weighted Navigational Graph

Compared with the earlier studies of Web prediction using Markov models or association rules [14, 24, 25], the above prediction process can refine the visited Web pages, so the search space of Web recommendation is significantly saved. Moreover, the proposed system easily allows the extension of the modules in the future. A better sequence mining algorithm can be integrated into the Frequent Pattern Discovery Module for the performance optimization of data mining. Novel

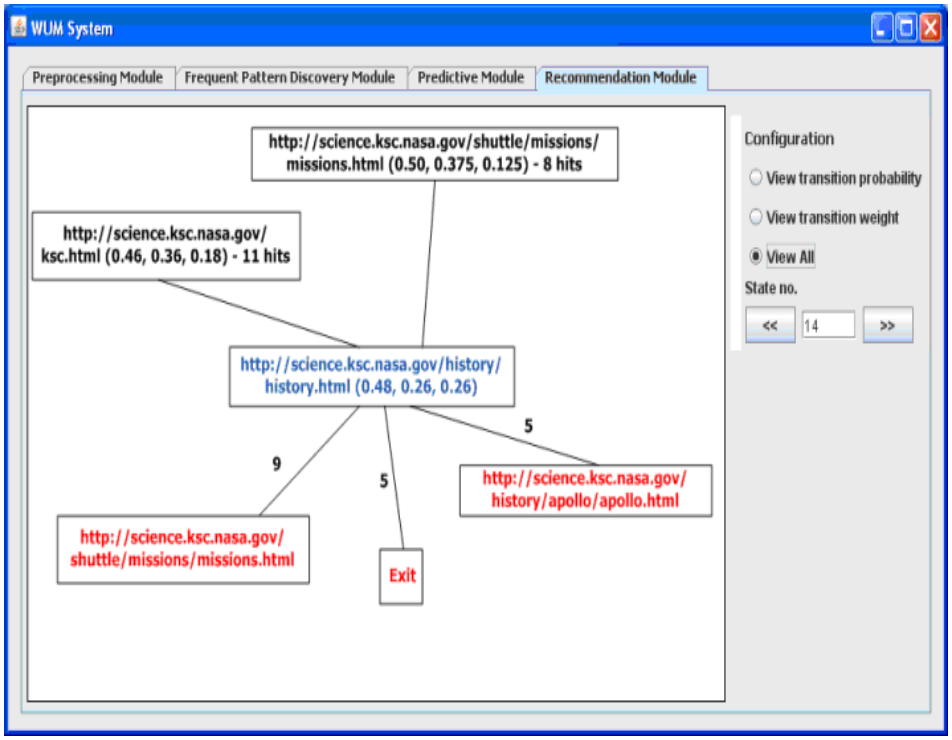

Figure 3. Recommended Next Navigation Step

Markov models, such as semantic-rich Markov models [26] or hybrid Markov models, can be applied to the Predictive Module. Semantic Web technology not only enhances predictive models with semantic information, but also improves the prediction accuracy and search space. The accuracy of Web prediction will be evaluated in our future works to verify these promising potentials.

5. Conclusions

Web mining, in general, is very challenging compared with traditional Data Mining due to its great amount of unstructured, rapidly and frequently changing data. There is also a necessity for online processing capability which requires the mining algorithms to minimize computation overheads while performing accurately with unseen data. This paper introduces a novel approach to mining web usage data that integrates a tree-based Temporal Sequence Mining technique and a dynamic Markov model. The proposed method is shown to significantly reduce complexity of original Markov models while retaining their superior predictive accuracy. The observed efficiency improvement of our model can be explained by the effect of Frequent Pattern Discovery module on the execution of the Markov-based Predictive Module. That is, the algorithm predicts web navigation patterns using the frequent web access sequences extracted from the web logs rather than all pages. Our Web-Page Recommender System also identifies frequent web navigations which are impossible in conventional systems merely using Markov models.

In the future, we plan to test our system on a live production web server and conduct both quantitative and qualitative analysis about the impacts of the generated recommendations on user navigation behaviors. Another interesting direction is to explore the use of prior domain knowledge such as information about relational structure of the site for mining process guidance and improvement. Finally, we seek to enhance the effectiveness of our recommendations with greater coherence by incorporating web usage knowledge and domain concepts that can be semantically represented by ontology.

References

[1] Scheffer, T., Decomain, C., Wrobel1, S. (2001). *Minin*g the WebWith Active Hidden Markov Models, in IEEE International Conference on Data Mining (ICDM).

[2] Kim, D.-H., et al. (2004). A clickstream-based collaborative filtering personalization model: towards a better performance, in ACM international workshop on Web information and data management.
[3] Cadez, I., et al. (2003). Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. Data Mining and Knowledge Discovery. 7(4) 399-424.
[4] Eirinaki, M., . Vazirgiannis, M., Kapogiannis, D. (2005). Web Path Recommendations based on Page Ranking and Markov Models, in ACM international workshop on Web information and data management.
[5] Nasraoui, O., et al., eds (2002). Automatic web user profiling and personalization using robust fuzzy relational clustering. ECommerce and Intelligent Methods, ed. J. Kacprzyk. Springer.
[6] Chen, L., Bhowmick, S.S., Li, J (2006). *COWES:* Clustering Web Users Based on Historical Web Sessions, *In:* Database Systems for Advanced Applications, Springer Berlin / Heidelberg. p. 541-556.
[7] Zhu, J., Hong, J., Hughes, J.G. (2004). PageCluster: Mining Conceptual Link Hierarchies from Web Log Files for Adaptive Web Site Navigation. ACM Transactions on Internet Technology, 4. p. 185-208.
[8] Ezeife, C.I., Lu, Y. (2005). Mining Web Log Sequential Patterns with Position Coded Pre-Order Linked WAP-Tree. Data Mining and Knowledge Discovery. 10 (1) 5-38.
[9] Liu, Y., Huang, X.,. An, A. (2007). Personalized Recommendation with Adaptive Mixture of Markov Models. The American Society for Information Science and Technology, 58 (12) 1851–1870.
[10] Yan, T.W., et al. (1996). From user access patterns to dynamic hypertext linking. *In*: International World Wide Web Conference.
[11] Joachims, T., Freitag, D., Mitchell,T. (1997). Webwatcher: A tour guide for the world wide web. *In:* International Joint Conference on Artificial Intelligence.
[12] Mobasher, B. (2007). Data Mining for Web Personalization, in The Adaptive Web. Springer Berlin / Heidelberg. p. 90-135.
[13] Nakagawa, M., Mobasher, B. (2003). A Hybrid Web Personalization Model Based on Site Connectivity, *In*: Proceedings of the 2003 WebKDD Workshop. KDD'2003, Washington, DC.
[14] Yu, Y., et al (2006). Mining Interest Navigation Patterns Based on Hybrid Markov Model. Springer Berlin / Heidelberg. p. 470-478.
[15] Srikant, R., Agrawal, R (1996). Mining Sequential patterns: Generalizations and Performance Improvements, *In*: *I*nternational Conference on Extending Database Technology.
[16] Zaki, M.J., *SPADE:* An Efficient Algorithm for Mining Frequent Sequences. Machine Learning, 2001. 42 (1) 31-60.
[17] Ren, J., Zhang, X., Peng, H. (2006). MFTPM: Maximum Frequent Traversal Pattern Mining with Bidirectional Constraints. *Journal of Computer Science*, 2 (9) 704 - 709.
[18] Romero, C., et al. (2009). Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems. Computers & Education, 53 (3) 828-840.
[19] Deshpande, M., Karypis, G (2001). Selective Markov Models for Predicting Web-Page Accesses, *In*: SIAM International Conference on Data Mining.
[20] Sen, R., Hansen, M, (2003). Predicting a Web user's next access based on log data, *Journal of Computational Graphics and Statistics*, 12 (1) 143-155.
[21] Markov, Z., Larose, D.t. (2007). Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage, New Britain: John Wiley & Sons, Inc.
[22] Bhaumik, R., Burke, R., Mobasher, B.(2007). Effectiveness of Crawling Attacks Against Web-based Recommender Systems, *In:* Proceedings of the 5th workshop on intelligent techniques for web personalization (ITWP-07). Center for Web Intelligence, School of Computer Science, Telecommunication and Information Systems, DePaul University, Chicago, Illinois.
[23] Srikant, R. , Agrawal, R (1996). Mining Sequential Patterns: Generalizations and Performance Improvements, *In*: International Conference Extending Database Technology. p. 3–17.
[24] Borges, J., Levene, M. (2004). A Dynamic Clustering-Based Markov Model for Web Usage Mining. Available online at http://xxx.arxiv.org/abs/cs.IR/0406032.
[25] Sobh, T., et al.,(2006). Using Association Rules and Markov Model for Predit Next Access on Web Usage Mining, in Advances in Systems, Computing Sciences and Software Engineering. Springer Netherlands. p. 371-376.
[26] Mabroukeh, N.R., . Ezeife, C.I (2009). Semantic-Rich Markov Models for Web Prefetching, *In*: 2009 IEEE International Conference on Data Mining Workshops. Miami, Florida, USA. p. 465-470.