Chapter 76
A Web Content Recommendation Method Based on Data Provenance Tracing and Forecasting

Zuopeng Liang and Yongli Wang

Abstract How to choose an appropriate releasing strategy for site content, and which one caters to user's habits, have become the main challenges. This article provides a provenance-aware model to design the content of the website. Based on the user's browsing history data, it constructs timed automaton that can trace the provenance of the data to find what the user may be interested in, and it establishes a Markov chain model to determine the content of the link relationship. Experiments show this model not only meets the dynamic needs of users when they browses the site, but also gives certain options to the administrator of site content. It provides recommending result efficiently and should have a bright application prospect.

Keywords Content recommendation · Data provenance · Timed automation · Markov chain

76.1 Introduction

With the advances in Web technology, the amount of information inherent in the Internet became more. How to provide targeted, appropriate information to user presents many challenges for the field of information retrieval (IR).

Z. Liang (✉)
Department of Economics, Nanjing University, Nanjing, China
e-mail: 896073265@qq.com

Y. Wang
School of Computer Science and Engineering,
Nanjing University of Science and Technology,
Nanjing, China
e-mail: yongliwang@mail.njust.edu.cn

W. E. Wong and T. Ma (eds.), *Emerging Technologies for Information Systems, Computing, and Management*, Lecture Notes in Electrical Engineering 236, DOI: 10.1007/978-1-4614-7010-6_76,

The site administrators not only need to obtain reliable data in a complex, interconnected network, but also need to explore all kinds of network users' needs. Web-based development has changed the traditional development method such as data flow, storage, and statistics. Firstly, it is extremely easy to access data and copy data on the network environment, which causes the reliability of data is difficult to be guaranteed; secondly, the pages in the browser are evolving and expanding, and the relationship between the pages is relatively unstable.

In recent years, Web applications are booming, and the study on the Web personalized recommendation are quietly rising [1]. However, the defect of these algorithms is that it cannot meet the needs of most users and only provide a recommendation for the specified user. A data provenance consists of the entire processing history of the data, which includes its source and all subsequent processing steps [2]. If we regard the click history or browsing log as the data provenance of certain user, we can use provenance workflow to trace the habit of the user. There are two approaches to calculate the data provenance: query inversion mode ("lazy" approach), and labeling mode ("eager" approach) [3]. This article uses "eager" approach to calculate the data provenance.

There are some existing methods, such as timing diagram, provenance diagram, XML DTD (XML Schema), to realize a labeling mode based on workflow provenance [4, 5]. In the forecast, policy makers always expect subjective judgments as much as possible close to the objective judgments [6]. Markov chain algorithm provides us with the scientific method to resolve these problems [7]. However, the existing prediction models only process the basic data structures and do not fully mine inter- relationship of access log.

We annotate these data to establish the state of the automaton, and the Content Manager can achieve the content semantics that users concern. On this basis, we use the extended Markov chain model to predict the order of the browsing the web content by user.

76.2 Definition of Timed Automata Model

The proposed recommendation model supports time constraints about accessing Web network, the Web Workflow is a real-time workflow.

Definition 1 (Web real-time workflow): Web real-time workflow consists of activities, participants and dependencies between activities. An activity refers to a separate step in the business processes; it can be viewed web content that users browsed. A participant mainly refers to the user. A dependency determines the execution order of activities and data flow between activities, which is the conversion between the web content.

A timed automaton is widely used in the modeling and verification of real-time systems. Constructing a timed automaton for Web accessing can record timing constraints relationship between which the user browses the web pages.

Definition 2 (Time automata): timed automata A is a seven-tuple $(S, S_0, \Sigma, X, I, E, F)$, where S is a set of finite states, which indicates all state of web content. $S_0 \subseteq S$ is the initial state set, which denotes the first web page content. Σ is a finite event set, which includes clicking on the link, closing the page. X is a finite clock set. I is a mapping, which assigns a timed constraint in clock constraints set $\Phi(X)$ for each state s in S.

When the clock of the page state does not satisfy the timing constraint, the automata must be able to perform the migration to leave the web page content. $E \subseteq S \times S \times \Sigma \times X \times \Phi$ is a collection of content conversion. A conversion $< s, s', \alpha, \lambda, \delta >$ denotes that when an event α occurs, the web content converse from s into s', where $\lambda \subseteq X$ is the resetting zero value of the clock collection while the conversion occurs; δ is a time constraint in X, which specifies the time constraint to met the conversion condition. We denote the time constraints of users browsing in specified contents as $s \xrightarrow{\alpha,\lambda,\delta} s'$, where α, λ and δ can by default. $F \subseteq S$ is the set of states of termination of the web content.

We construct a four-tuple (t, ip, s, s') for a specific website. This data structure represents that a user, whose IP address is *ip*, transferred from the content *s* to the content *s*' during time *t*. For example, Fig. 76.1 shows a timed automaton based on 3 web pages content.

While the state transition occurs, timed automata record and transfer the identity of the relevant web content. The timed automaton can constantly update data set to reflect the latest status of the user. Thus, it provides considerable flexibility and scalability.

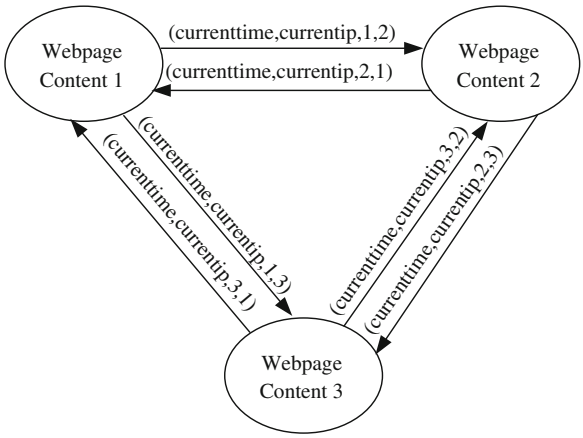

Fig. 76.1 Three web pages of timed automata

76.3 Forecasting the Content of Links Based on Markov Chain

When users are browsing the web, it is difficult to identify the inherent regularity to discover clicking on which link and selecting the link in what order in the complex environment. We propose a method to predict the linking content that the user select based on time automata.

76.3.1 The Markov Chain for Web Content

Definition 3 (Web Markov chain): Suppose that $\{X(n), n = 0, 1, 2, \ldots\}$ is a random sequence and Q is a discrete state space, if for any m non-negative integers $n_1, n_2, \ldots, n_m (0 \leq n_1 < n_2 < \ldots < n_m)$, any natural number k, and arbitrary $i_1, i_2, \ldots, i_m, j \in Q$ meet:

$$\begin{aligned} &\mathrm{P}\{X(n_{m+k}) = j | X(n_1) = i_1, X(n_2) = i_2, X(n_m) = i_m\} \\ &= \mathrm{P}\{X(n_{m+k}) = j | X(n_m) = i_m\}. \end{aligned} \tag{76.1}$$

We call $\{X(n), n = 0, 1, 2, \ldots\}$as a Markov chain. If n_m represents the present moment, $n_1, n_2, \ldots, n_{m-1}$ represents the last moment, and n_{m+k} represents the future moment. Eq. (76.1) shows that the webpage content j in the future moment n_{m+k} only depends on the webpage content in the present moment n_m, In the other word, the webpage content j in the future moment n_{m+k} is independent of the webpage content in m-1past moments $n_1, n_2, \ldots, n_{m-1}$. This reflects the characteristic of Markov process.

Markov chain is a particular case of the Markov process. Markov chain model of the Web content describes that the state of webpage content change from the past to the present, and from the present into the future, which changes one by one. It like a chain and has no aftereffect. The Markov chain reflects the randomicity of user's browsing behavior.

We denote the data to be forecasted as an instance of seven-tuple from timed automata. The transition probability matrix can be updated dynamically, and the calculation process can be executed according to the recurrence relation. As long as the initial web content that the transition matrix obtained is accurate, the future of predicted link results has certain credibility.

A random sequence with the characteristics of the Markov chain can be divided into m states, for example, $i_1, i_2 \ldots i_m$ in Eq. (76.1)… and j. In this paper, the interlinked webpage content can be seen as the different status. The state space is $Q \subseteq S$, which represents the collection of webpage content. For example, $Q = \{1, 2, 3, 4, 5, 6\}$, each element corresponds to the contents 1 to the contents 6.

76.3.2 State Transition Matrix

The basic idea of Markov prediction is to obtain the state transition matrix of sequence using the original data sequence. The goal of Markov prediction is to estimate the future development trend according to the state transition matrix.

Definition 4 (Conditional probability): The condition probability $P\{X(n_{m+k}) = j|X(n_m) = i\} = P_{ij}(m,k)$, we call $P_{ij}(m,k)$ as k-step transition probability at moment n_m. After k-step transition, Web content i inevitably reach one webpage content in set Q, and only to reach one webpage content. Thus, k-step transition probability meets the following conditions:

$$P_{ij}(m,k) \geq 0, i,j \in Q; \tag{76.2}$$

$$\sum_{j \in E} P_{ij}(m,k) = 1, i,j \in Q. \tag{76.3}$$

Assume that the transition probability P_{ij} (m, k) of Webpage content does not depend on the Markov chain of m, we call P_{ij} (m, k) as homogeneous Markov chain. The status of the webpage is relevant to the starting content i, transfer step number k and the reaching content j. It is not relevant to m. At this point, we denote k step transition probabilities as P_{ij} (k), namely:

$$P_{ij}(k) = P_{ij}(m,k) \tag{76.4}$$

We use a transition probability matrix to represent the changed probability during transferring from one state to another state in Markov chain. For Webpage content space $Q = \{1, 2, 3, 4, 5, 6\}$, the corresponding one step state transition matrix is as following:

$$P(1) = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{16} \\ P_{21} & P_{22} & \cdots & P_{26} \\ \cdots & \cdots & \cdots & \cdots \\ P_{61} & P_{62} & \cdots & P_{66} \end{bmatrix} \tag{76.5}$$

The i-th row, j-th column element P_{ij} in P (1) represents the one-step transition probability while the Markov chain model transferred from the webpage content i to webpage content j.

76.3.3 Calculation of Transition Probability Matrix

In timed automata model which annotates browsing workflow, when the transition of state occurs, the timed automata record two webpage content that corresponds

to the transferring, and denote it as the four-tuple (t, ip, s, s'). We let N_{ij} represents the number of the transition during the t period while transfer page i to page j. During t statistics period, the number of four-tuple meet the $s = i$ and $s' = j$ is N_{ij}.

$$P_{ij} = \frac{N_{ij}}{\sum_{j=1}^{6} N_{ij}} (1 \leq i, j \leq 6) \tag{76.6}$$

The k-step transition probability is $P_{ij}(k)$, we can obtain the recurrence relations using C-K equation:

$$P\ (k)\ = P\ (1)\ P\ (k-1)\ = P\ (k-1)\ P\ (1) \tag{76.7}$$

Thus,

$$P\ (k)\ = P\ (1)\ P\ (1)\ldots P\ (1) = P\ (1)^k \tag{76.8}$$

It is vital to keep the one-step-state transition matrix correctness, which ensures the forecast close to the true value of the k-step transition of webpage content.

76.4 Experimental Analyses

Experiments establish on the basis of a small website. We use the Visual Studio 2005 to implement all the algorithms. The base station server is an IBM compatible computer (CPU Intel(R) Xeon(R) E5620 2.40 GHz and RAM 12 GB), and the OS is Windows 7.

We found that the state transition matrix approach stable when $k = 5$. For this experiment, the one-step transition will be able to reflect the user's browsing habits. According to one-step state transition matrix, we use the web page, which was visited with the maximum probability, to speculate the user's browsing order. For example, we should push the webpage content in the following order for the above matrix P: webpage content 1 → webpage content 4 → webpage content 5 → webpage content 6 → webpage content 3…

76.4.1 Verification of Reliability

The algorithm calculates the transferring probability of webpage content based on the data provenance of timed automata. In order to verify the reliability of the probability, we set the order 1 → 4 → 5 → 6 → 3…, in which the user's actual browsing the webpage within a period as a standard. And we explore the variance between the probability matrix generated from timed automata and the actual probability matrix.

We sample the running example and achieve the statistical transferring sequence as follows:

(1) Start from the webpage content 1 and go to webpage content 2, 3, 4, 5, 6, probability: 5.1, 36.7, 48.2, 3.5, 6.5 %;
(2) Start from the webpage content 2 and go to webpage content 1, 3, 4, 5, 6, probability: 18.6, 14.7, 9.7, 10.5, 46.5 %;
(3) Start from the webpage content 3 and go to webpage content 1, 2, 4, 5, 6, probability: 15.5, 8.7, 57.2, 12.5, 6.1 %;
(4) Start from the webpage content 4 and go to webpage content 1, 2, 3, 5, 6, probability: 11.1, 1.7, 17.2, 23.5, 46.5 %;
(5) Start from the webpage content 5 and go to webpage content 1, 2, 3, 4, 6, probability: 14.6, 12.2, 24.7, 12.0, 36.5 %;
(6) Start from the webpage content 6 and go to webpage content 1, 2, 3, 4, 5, probability: 25.1, 16.3, 32.1, 4.2, 22.3 %;

The probability variances in one-step state transferring matrix are 0.068, 2.468, 1.07, 0.402, 3.102, 2.088. Variance range is less than 3 basically, which is within the acceptable range. Figure 76.2 shows the trend of the variance of state transition probability.

We construct the proposed algorithm on the basis of the access log and click historical data, and we fully take into account the actual user's browsing habits. Thus, the data set that algorithm obtained is authentic. One-step state transition matrix is been calculated statistically by analyzing the history of user's clicking on a link. Therefore, we believe the state transition probability that generated from timed automata is credible.

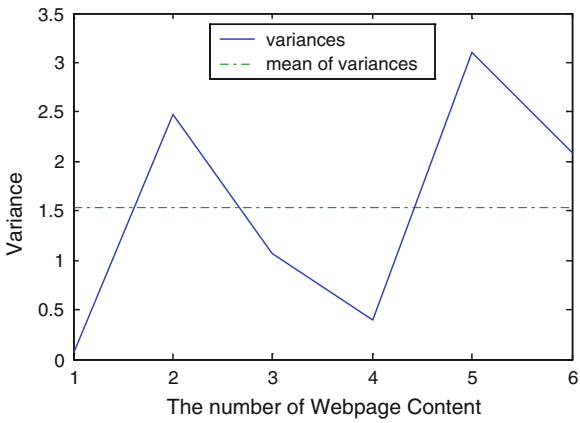

Fig. 76.2 The variance of the state transition probability distribution

76.4.2 Validation of the Recommended Quality

In order to verify the performance of the algorithm, we compare the proposed algorithm and the collaborative filtering algorithm that mentioned in the related work. Collaborative filtering algorithm is the most widely used personalized content recommendation algorithm.

In order to measure the quality of the recommended content intuitively, we use the average absolute deviation (MAE) [8] as referral service quality standards. Support that the forecasted set that consists of N user's scores is $\{m_1, m_2 \ldots, m_N\}$, the actual score set of user's ratings is $\{ n_1, n_2 \ldots, n_N \}$, the MAE defined as following:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |m_i - n_i|}{N} \tag{76.8}$$

Figure 76.3 shows the MAE comparisons of the collaborative filtering algorithm and the proposed in this paper. We can conclude that the accuracy of the automata-based Webpage content recommendation method is higher than one of the collaborative filtering algorithms.

Collaborative filtering algorithm uses the similarity between the users to filter information. However, the main drawback of this algorithm is that the similarity bears sparsity problems and scalability problems, and the similarity of the user calculated by this algorithm has a certain deviation. The proposed algorithm in this paper directly establishes in the habits of user's accessing content; thus the MAE is relatively low. In addition, we can update the recommended strategy and dynamically know the user's new interest; thus the proposed algorithm is high flexibility and wide applicability.

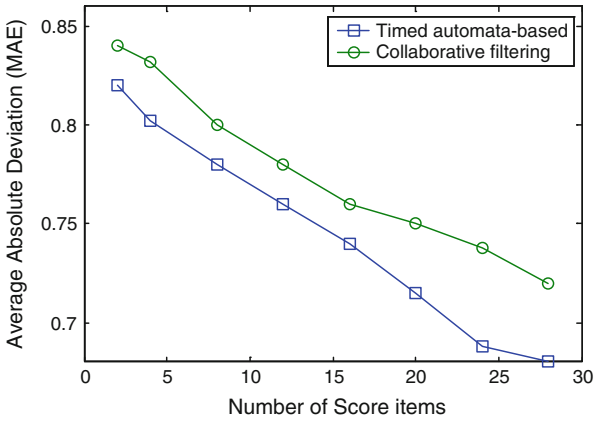

Fig. 76.3 MAE comparisons of two algorithms

76.5 Conclusion

In recent years, Network resources have increasingly become an indispensable part of people's lives, which brings a golden opportunity for businesses recommendation. Researchers creatively use timed automata, which is constructed by a labeling workflow method, to find Webpage content that is welcomed by users in this paper. Researchers employ Markov chain principle to establish the content link model. Based on this method, the website content administrator can design web pages that users are most interested in. This model is not only convenient for the user to view, but also improves the efficiency and quality of the user's view.

Acknowledgments This work is supported in part by China Postdoctoral Science Foundation (2012M511227), Jiangsu Province Postdoctoral Science Research Fund (1101073C), National Natural Science Foundation of China (61170035), Natural Science Foundation of Jiangsu (BK2011022, BK2011702).

References

1. Wang, J., Tang, X.: Personalized recommendation algorithm research based on content in social network. Applica. Res. Comput. **24**(8), 1248–1250 (2011)
2. Woodruff, A., Stonebraker, M.: Supporting fine-grained data lineage in a database visualization environment. In: Proceedings of the International Conference on Data Engineering (IEEE ICDE), pp. 91–102 (1997)
3. Liu, X., Wan, C.: Research on data provenance an overview. Sci. Mosaic **1**, 47–52 (2005)
4. Kiran-Kumar, M., Uri, B., David, A.H., Peter M., Diana M., Daniel M., Margo S., Robin S.: Layering in Provenance Systems. USENIX Annual technical conference (2009)
5. Zoé, L, Christophe, L., Spyro, M.: Storing Scientific Workflows in a Database. ACM (2009)
6. Geoffrey, R.G., David R. S.: Probability and Random Processes. Oxford University Press, USA; 3 edition (2001)
7. Deng, M.: Research on the top three places in men's modern pentathlon of olympic using gray markov chain prediction model. Mathemat. Pract. Theory **41**(2), 134–137 (2011)
8. Herrlocker, J., Konstan, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)