A Time Influence Analysis for Web Page Recommender System

Chen-Chung Chi, Nai-Lung Tsao, Chin-Hwa Kuo
Department of Computer Science and Information Engineering,
Tamkang University
ryanjih@mail.tku.edu.tw, beaktsao@gmail.com, chkuo@mail.tku.edu.tw

Abstract

In this paper, we propose a time weighting model to generate a webpage recommendation list in a single website for the visitors. We use this proposed model to analyze the trend of users' behavior, then generates a webpage recommendation list depend on it. The time weighting model provides a solution for a web page recommender system, find a better time interval to generate a webpage recommendation list, therefore enhance the recommendation precision based on a given time interval. The experimental result shows if we choose an appropriate time interval, the recommendation list generated from the specified time interval can obtain a better time weight, therefore the recommender system can make a better recommendations based on choosing an appropriate time interval for web log data mining.

Keywords: time influence, web page recommender system, probabilistic model.

1. Introduction

Many web sites contain tons of web pages. Users get lose easily when accessing these web sites. Web page recommender systems (WPRS) help users surfing a large web site and finding deep and hidden web pages which users might be interested. Usually, WPRS predict a small set of the web pages of a specific large web site according to the user's current traversal path and assume the recommended web pages match users' interest. In many cases WPRS apply some kinds of web usage mining (WUM) techniques. WUM was introduced by Cooley et al. in 1997 [1]. WUM is the approaches to discover patterns in web server log. The patterns can be used in many applications, such as web prefetching [2] and e-Commerce [3]. It's main concept to analyze a website's access log, extract some user browsing behavior models from log, and then provide some appropriate webpage recommendation based on these browsing behavior models to fit in users' demand. The design of the WPRS not only can save the time for the users to searching some information they need in a website, but also can provide a personalized information service for individual users based on their different browsing behavior models.

Most of the WPRS use some web usage mining technique, such as association rules or its variants [4]. Association rules method can be used to extract traversal patterns from a web server log and use the traversal patterns to recommend web pages. However, association rules is frequency sensitive, which means it might ignore low frequent items, rare traversal patterns in WUM, such that these low frequent items in traversal pattern can't be applied in recommendation. So in this paper we propose a probability model to improve this drawback of WPRS. The proposed model does consider the situation of the rare web pages or traversal patterns. Meanwhile we also deal with time influence issue. To our knowledge, most of the previous research treat the whole web server log as a data resource and extract traversal patterns from it. In [5], Su and his colleagues proposed a WPRS model which considered time influence but their time weighting scheme was intuitive, not involving any computation.

Some other research with regard to time-weight method, Yoon[7] has proposed a time-weighted clustering method, and integrate this method in a personalized music recommendation system. This approach defines a time weight α, which is a decreasing rate. It's handled by system manager. If α become smaller, then latest music in a time sequence will be more important. Yolanda[8] introduces a parameterize time function in their study. This approach focused on some time functions adopted in their recommend strategy, such that linear increasing function, linear decreasing function, rectangle function, and constant function. The shape of each time function depends on the nature of each product and on some manufacturer-specified parameters. Gong[9] proposed

a data-weight of time based on user interest changes and advance the significance of then user lately access data in the recommend generative process. In spite of these researches have mentioned time-weight function to figure out all the users' behavior model, these approaches need a method to define time interval to make recommendations more precisely.

In this paper, we design a weighting model for evaluate time influence. We try to define some variant time interval to analyze a log data from a web server as a training data, and use traversal sessions extracted from various time interval to figure visitors' special browsing behaviors. Our assumption is that the older data in web server log have the trend of reduce contribution for make a webpage recommendation list, this trend is towards obvious in some specific time intervals. Consequently on the basis of the concept from smoothing and weight method to acquire weights of different time stages, then the WPRS can generate a web page recommendation list based on the higher weight mapping a specified time interval.

This paper is organized as follows. In section 2, we describe the proposed probabilistic model for WPRS. Section 3 we show time influence for recommendation and how to design a model for time weights. Experimental result and conclusion are presented in Section 4 and Section 5, respectively.

2. Probabilistic Model for WPRS

In WPRS, the recommendation list is usually generated by ranking all web pages except the current and previous browsing pages which has been accessed by visitors. The recommendation list can be predicted by $P(u_i|U_c)$. where u_i means the recommended candidates and U_c is the current traversal path which is the set of the current and previous browsing pages in a session. Obviously U_c is rare in web server log if the number of the web pages in U_c is large. This phenomenon makes difficulty to extract patterns by the association rules because that $P(u_i|U_c)$ might be difficult to get from prior knowledge. In order to tackle the above issue, we apply a Naïve Bayes probabilistic model to acquire $P(u_i|U_c)$. The calculation of $P(u_i|U_c)$ can be transferred by Bayesian theorem and Naïve Bayes assumption as follow:

$$P(u_i|U_c) = \frac{P(U_c|u_i)P(u_i)}{P(U_c)} = \frac{\prod_{u_j \in U_c} P(u_j|u_i)P(u_i)}{P(U_c)}$$

Because our goal is to get $\arg\max_{topN(u_i)} P(u_i|U_c)$ $P(U_c)$, can be ignored. The final equation is as follow:

$$\arg\max_{topN(u_i)} P(u_i|U_c) = \arg\max_{topN(u_i)} \prod_{u_j \in U_c} P(u_j|u_i)P(u_i) \quad \text{.....(1)}$$

Then the system performs the prior knowledge acquisition based on this model.

3. Time Influence

The prior knowledge $P(u_j|u_i)$ can be acquired from the web server log. As mentioned in section I, we assume that the web server log in different time stages might contribute different credits for predicting probability. A simple significance of correlation experiment that proves our assumption is shown in Table 1. We use month as the time unit and generate the recommendation list by the model mentioned in section II. The prior knowledge is acquired based on each previous month. Time stage 1 means the previous month, time stage 2 means the second previous month and so on. The precision is predicted by the prior knowledge of each time stage. The trend is obvious and as assumption, there is a significant negative correlation between the time and the precision.(correlation = -0.835, n=6, df=4, p=0.038591, two tailed)

Table 1. Relation between time stages and precision

Time stages	Precision rate
1	70
2	47
3	40
4	43
5	36
6	36

Because the performance of considering time influence is significant, we then consider how to integrate time feature into our probabilistic model. We make use of the smoothing techniques to calculate weights for different time stage. We utilize $P(u_j|u_i)$ as a sample to describe the model. $P(u_j|u_i)$ can be estimated as the linear interpolation of the time stages as follow:

$$P(u_j|u_i) = \sum_t \lambda_t P_t(u_j|u_i) \quad \text{.....(2)}$$

where $P_t(u_j|u_i)$ is the conditional probability at time stage t and λ_t is the weight of time stage t. then we set $\lambda_t = 1 / |T|$, where T is the set of all the time stages. Finally, we calculate the degree to which each estimate predicts the suggested u_i at time stage t:

$$\beta_t = \frac{\sum_{i,j} \lambda_t P_t(u_j|u_i)}{\sum_{k \in T} \sum_{i,j} \lambda_k P_k(u_j|u_i)} \quad \ldots\ldots(3)$$

We can assume the trend of time influence is the same between all the prior knowledge. Therefore $P(u_i)$ in (1) in section 2 can use the same result of λ trained by $P(u_j|u_i)$.

4. System approaches and experimental result

The proposed method as shown is applied in a web site in our university. Most of the web server logs from popular web servers, like Apache or IIS, contain some basic fields, such as remote IP, query url, and access time. We do not use special configuration of log format so only the three fields mentioned above are used. The web server log is divided into small logs based on specified time intervals.

We use the log information from May to November in 2007. Table 1 in Section 3 is the result of predicting the recommendation list for November. We use the logs from May to October as training data and November as testing data. In this paper, we use the log data of November to be the testing data and evaluate the performance of recommendation list. All the log data is preprocessed by following procedure:

Table 2. Log size from the analyzed website

Data type	Data size
Log records	2,333,355
Log records after cleaning	2,290,622
Users count	18,957
Session count	25,647
Atomic session count	16,889

① *Data cleaning*

This procedure is used to discard unnecessary log data from web log, such as log information about icon file request and some other kind of unnecessary log data. We just keep HTML access log (accessed file log which file extension like *.htm, *.html, *.aspx, *.php…etc.) to enhance the performance in log data analysis speed.

② *Session segmentation*

We use IP address and time stamp to define a session. Under the same IP, we can recognize individual log for each user. In a session, user's requests between two adjacent time stamps should within thirty minutes, or should treat it as another new session.

③ *Traversal path analysis*

Assume query url in one session is in the form $\{U: u_0, u_1, u_2, u_3, \ldots\ldots, u_n\}$ and u_i is one of the query url in U, $0<i<n$. Therefore we define $\{U_p: u_0, u_1, u_2, u_3, \ldots\ldots, u_{i-2}\}$ is the current traversal path of the user in one session and u_i is the recommended candidate for any query url in U_p. The reason that u_{i-1} is ignored because there usually exists a hyperlink between u_{i-1} and u_i so recommending ui to u_{i-1} is not necessary. Then the prior knowledge $P(u_j|u_i)$ (described in section 2) can be acquired from calculating the count of ui and $uj \in Up$ and $P(u_i)$ can be easily acquired from web server log.

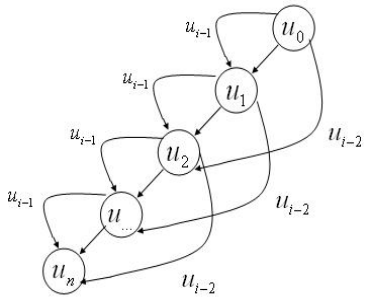

Figure 1. Traversal Path in Session U

④ *Prediction Evaluation*

We evaluate the proposed approaches by performing our method in a specific web server log, the log information of November as mentioned above. Assuming that $\{Ur: u_0, u_1, \ldots, u_n\}$ is the most frequent n pages after browsing Uc. Then the precision rate of predicting recommendation list for Uc is equal to the following equation:

$$\frac{the\ number\ of\ the\ recommended\ pages\ in\ U_r}{the\ number\ of\ the\ recommendation\ list} \quad \ldots(4)$$

We use the web log during November as testing data, and using different kind of time interval to estimate time weights. In our evaluation, we use variant time interval such as 10 days, 20 days, 30 days, 40 days, 50 days and 60 days for time interval, to analysis the training data during May 2007 to October 2007, which have 180 days log data from the training data. If we set time interval as 30 days, then we should get six time weights after evaluation (180/30 = 6). The precision acquired by using the time weights to predicting

the recommendation list is shown in Table 1.

In order to evaluate the time weight in each time interval, at first we use association method to find out popular web pages which have been viewed by visitors, afterwards we use a probabilistic modeling method shown in equation (1) and webpage recommend ratio weight method to generate time weight in each time interval. The second method calculates weights by webpage recommend order position both in training data and recommended webpage list, then we get a weight score from each time stage as follows:

$$ScoreWeight(t) = \frac{\sum_{u \in Uc} |N| + 1 - pos(t,u)}{\sum_{u \in Uc} pos(s,u)} \quad \ldots\ldots(5)$$

In the equation (5), N is the count of web pages recommended in the recommend list *U, pos(t,u)* is the position of the recommend webpage u appears in the generated testing data, *pos(s,u)* is the position of the recommend webpage u appears in the generated training data.

For example, webpage "A" occurred in the recommend list generated by testing data, the recommended order is the 1st order, and there are 30 web pages in the recommended list. Supposed that the webpage "A" is also appear in a recommended list generated by training data which is belong to a time interval, and the recommended order is 2nd, then we use equation (5) to get a weight by (30+1-1) / 2 = 15. Afterwards we get a sum total by count each score generated by web page appears in recommendation list Uc. Finally we get a weight *r* by equation (6) as follows:

$$r_t = \frac{ScoreWeight(t)}{\sum_{t \in T} ScoreWeight(t)} \quad \ldots\ldots(6)$$

⑤ *Behavior modeling*

We use a correlation co-efficient formula to measure the trend for the precision in each time interval. The correlation value represent the trend of user behavior which has influnced by time factor. The bigger of the value, the faster of the weight increasing speed. It represents the trend to the user behavior influenced by time- weighting is more obvious. These evaluation results represent obviously that taking time influence issue into account gets more significant result.

$$Correlation(r) = N\sum XY - (\sum X)(\sum Y) / \sqrt{[N\sum X^2 - (\sum X)^2][N\sum Y^2 - (\sum Y)^2]} \quad \ldots\ldots(7)$$

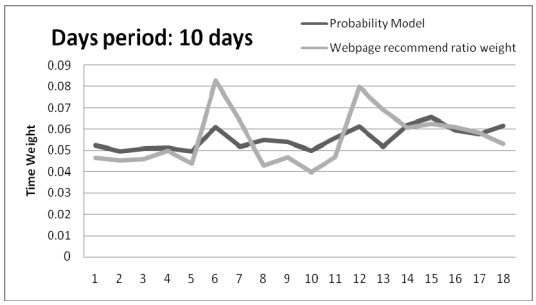

Figure 2. Time weight evaluation in 10 days

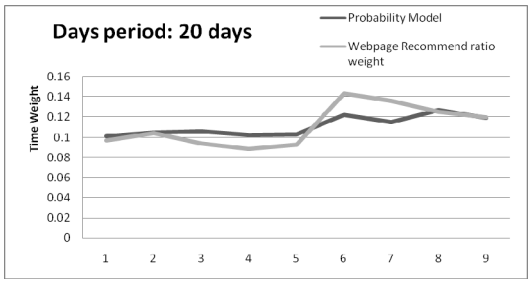

Figure 3. Time weight evaluation in 20 days

In figure 2, we observed that the curve belong to probability model is smoother than webpage recommend ratio weight evaluation method, and in figure 2 to figure 5, vertical axis represent time stage, smaller number of vertical axis means older time stage from now. We observed that there is a trend that in figure 2 to figure 5, older time stage have less time weight commonly. This trend can obviously figured out by calculating a correlation value by equation (7), and the result shown in table 3 and figure 6:

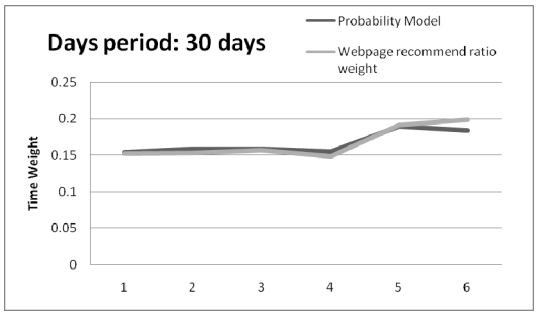

Figure 4. Time weight evaluation in 30 days

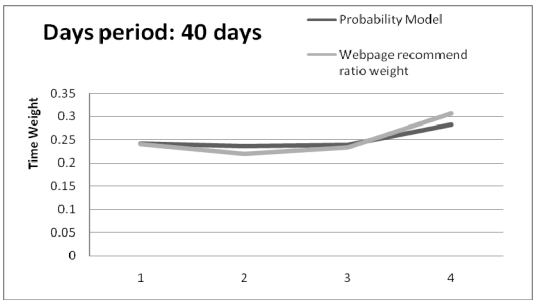

Figure 5. Time weight evaluation in 40 days

Table 3. Correlation measure

Time Interval	10	20	30	40	50	60
Probability Model	0.6913	0.8312	0.8034	0.7396	0.8727	0.8441
Weight Ratio	0.3396	0.6582	0.8166	0.7064	0.7589	0.7886

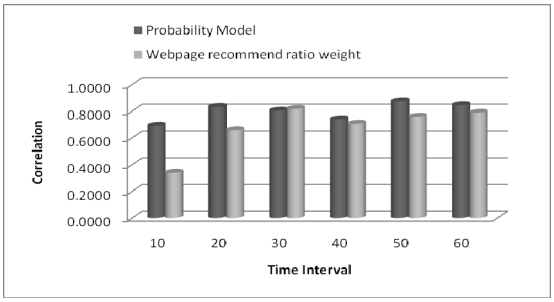

Figure 6. Correlation measure
using prob. model and recommend ratio weight

5. Conclusion

In this paper, we propose a probabilistic model and webpage recommended ratio weight method to evaluate a time weighting model for web users. These methods provides a solution for time weighting model contributes a modeling user behavior trend if we pick an appropriate time interval for finding user's behavior model. Experiment shows that user's behavior influenced by time interval obviously, older data in web server log have the trend of reduce contribution for make a webpage recommendation list. This is an important feature for a webpage recommendation system for making recommendation. In spite of not every website log represents a trend similar to our experiment result, the proposed methods still can find out the trend represents in a web log, and then use this to make better recommendation. Our next step is to consider integrating the features of the semantic content of web pages into our current model and hope to provide the users a better web page recommender system.

References:

[1] R. Cooley, J. Srivastava, and B. Mobasher, "Web mining: Information and pattern discovery on the world wide web", In 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.

[2] Q. Yang, H. H. Zhang and I. T. Y. Li, "Mining web logs for prediction models in WWW caching and prefetching", in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 473--478, 2001

[3]C.-H Yun and M.-S. Chen, ``Mining Web Transaction Patterns in an Electronic Commerce Environment", in Proceedings of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD-00), pp. 216-219, April 18-20, 2000.

[4] Mathias Géry, Hatem Haddad, "Evaluation of web usage mining approaches for user's next request prediction", in Proceedings of the 5th ACM international workshop on Web information and data management , 74-81, 2003

[5] Yi-Jen Su, Hewijin Christine Jiau, Shang-Rong Tsai, "Using the moving average rule in a dynamic web recommendation system", International Journal of Intelligent Systems, Volume 22 Issue 6 , Pages 621 – 639, April 25, 2007
Andrew McCallum , Ronald Rosenfeld , Tom M.

[6] Mitchell, Andrew Y. Ng, "Improving Text Classification by Shrinkage in a Hierarchy of Classes", in Proceedings of the Fifteenth International Conference on Machine Learning, p.359-367, July 24-27, 1998

[7] Taebok Yoon, Seunghoon Lee, Kwang ho Yoon, Dongmoon Kim, Jee-Hyong Lee, "A personalized music recommendation system with a time-weighted clustering", Intelligent Systems, 2008. IS '08. 4th International IEEE Conference, Volume 2, Page(s):10-48 - 10-52, Septemper 6-8, 2008

[8] Blanco-Fernández, Yolanda; López-Nores, Martín; Pazos-Arias, Jose J.; Gil-Solla, Alberto; Ramos-Cabrer,Manuel; **"**Personalizing e-Commerce by Semantics-Enhanced Strategies and Time-Aware Recommendations"**,** Third International Workshop on Semantic Media Adaptation and Personalization, Page(s):193 – 198, December 15-16, 2008

[9] SongJie Gong, GuangHua Cheng, “Mining User Interest Change for Improving Collaborative Filtering”, Second International Symposium on Intelligent Information Technology Application, Volume 3, Page(s):24 – 27, December 20-22, 2008