A Web Mining Model for Real-time Webpage Personalization

SHEN Hui-zhang[1], ZHAO Ji-di[2], YANG Zhong-zhi[2]
1 Institute of Systen Engineering, Shanghai Jiaotong University, P.R.China, 200052
2 Aetna School of Management, Shanghai Jiaotong University, P.R.China, 200052

Abstract: Determining the size of the World Wide Web is extremely difficult. The Web can be viewed as the largest data source available and presents a challenging task for effective design and access. One proposed Web mining approach to handling the problem of effective design and access is personalization. With personalization, Web access or the contents of a Web page are modified to better fit the desires of the user. This may involve dynamically creating Web pages that are unique per user or using the desires of a user to determine what Web documents to retrieve. This paper presents a Web mining model based on dynamic clustering and hidden Markov model. The output of the model is some information for dynamically creating a web page which can best meet the user's desires. The assumption of the dynamic clustering is that if a group of users who have the same interest trend, those pages they have visited are probably related. We propose that human should be the authority to judge the correlation of two pages. First, the model statistic a user's web browsing records in the log file; find a group of users who have the same interest trend with the user; collect all the pages in which this group of users are interested; calculate the correlation between pages; and cluster the pages into several categories according to a predetermined threshold. Each web page category is considered as a stochastic state variable. In the second phase, our model based on hidden Markov model is further constructed to mine the latent desires of a user given an observed sequence of Web pages that the user have browsed. In order to get the optimal parameters (transition probability matrix, the conditional probability and the initial state) in the model, we applied the Baum-Welch parameter estimation method in EM algorithm to train the model on the data set. Experimental results show that the model is practicable and efficient.

Keywords: Hidden Markov model, Dynamic clustering, Web mining

1 Introduction

Determining the size of the World Wide Web is extremely difficult. In 1999 it was estimated to contain over 350 million pages with growth at the rate of about 1 million pages a day [1]. The Web can be viewed as the largest data source available and presents a challenging task for effective design and access. Each visitor of a web site leaves records in the log file of the pages that he or she has even visited. The visitor is characterized by different sessions (different sequences) and each of these sequences varies in length. Analysis of these clickstreams (sequences) can provide the maintainer of the site with abundant information on how to reorganize the site, which web pages to be connected and how to personalize the web service according to a specific user's desires.

Several attempts have been made to learn user clickstreams. Cadez et al. clustered individuals based on their observed Web browsing behavior[2][3]. Most notably, some researches have focused on probabilistic clustering of individuals with Markov model[2][3], hidden Markov model[4][5], even mixtures of Markov chains[6][7]. In this paper, we are interested in the problem of personalization. We presented our model to analyze observed user request sequences, predict user potential desires and personalize user web pages. This paper is organized as follows. In section 2, we first gave two definitions used later. We propose that human should be the authority to judge the correlation of two pages. Then the process of web pages clustering in our model is illustrated in detail. In section 3, we further constructed our model based on hidden Markov model to mine the latent desires of a user. In order to get the optimal parameters in the model, we applied the Baum-Welch parameter estimation method in EM algorithm to train on the data set. In section4, we used some experiments to illustrate our model and analyzed the results. We concluded the paper with a brief discussion.

2 Web pages clustering process

First, we describe two definitions that will be used in this paper.

2.1 Interestingness of a Web page β_{ij}

Billsus and Pazzani use the interestingness of a document to determine if a user is interested in it[6]. The interestingness is based on the similarity between the document and that of what the user wishes. Similarity is measured by the co-occurrence of words in the documents and a user profile created for the user. Here in this paper, the interestingness is also defined as the degree of a user' interest in a web page. But the measurement is different from Billsus and Pazzanis'.

Let β_{ij} denote the i^{th} user's interestingness in web page j.

$$\beta_{ij} = t_{ij} \Big/ \sum_{j \in P_i} t_{ij} \qquad (1)$$

Where t_{ij} denotes the times that the i^{th} user has browsed web page j. All the observation sequences with

web page j included, generated by the i^{th} user, was collected. P_i denotes the set of web pages in these observation sequences. The lager β_{ij} is, the more interest the i^{th} user has in web page j.

2.2 Correlation between Web pages $d(p_i, p_j)$

Let $d(p_i, p_j)$ be the correlation between web pages p_i and p_j. Suppose we have a group of users who have ever browsed web pages p_i and p_j. Let $G_k = \{U_1, U_2, \cdots, U_n\}$ be the group of users. The computation of $d(p_i, p_j)$ is based on the user group's overall interestingness.

Suppose $B(p_i) = (\beta_{1p_i}, \beta_{2p_i}, \beta_{3p_i}, \cdots \beta_{np_i})$ represents the interestingness vector of Group G_k in web page p_i. $B(p_j) = (\beta_{1p_j}, \beta_{2p_j}, \beta_{3p_j}, \cdots \beta_{np_j})$ represents the interestingness vector of Group G_k in web page p_j. There are several algorithms for computing the correlation.

(1)Euclidean distance.

$$d(p_i, p_j) = \sqrt{\left|\beta_{1p_i} - \beta_{1p_j}\right|^2 + \left|\beta_{2p_i} - \beta_{2p_j}\right|^2 + \cdots \left|\beta_{np_i} - \beta_{np_j}\right|^2} \quad (2)$$

(2)Manhattan distance.

$$d(p_i, p_j) = \left|\beta_{1p_i} - \beta_{1p_j}\right| + \left|\beta_{2p_i} - \beta_{2p_j}\right| + \cdots \left|\beta_{np_i} - \beta_{np_j}\right| \quad (3)$$

(3)Minkowski distance.

$$d(p_i, p_j) = \left(\left|\beta_{1p_i} - \beta_{1p_j}\right|^p + \left|\beta_{2p_i} - \beta_{2p_j}\right|^p + \cdots \left|\beta_{np_i} - \beta_{np_j}\right|^p\right)^{1/p} \quad (4)$$

Where p is a positive integer.

(4)Pearson Correlation Coefficient

$$d(p_i, p_j) = \frac{\sum_{l=1}^{n} (\beta_{lp_i} - \overline{\beta}_{p_i}) \cdot (\beta_{lp_j} - \overline{\beta}_{p_j})}{\sqrt{\sum_{l=1}^{n} (\beta_{lp_i} - \overline{\beta}_{p_i})^2 \cdot \sum_{l=1}^{n} (\beta_{lp_j} - \overline{\beta}_{p_j})^2}} \quad (5)$$

Where $\overline{\beta}_{p_i} = \frac{1}{n}\sum_{l=1}^{n} \beta_{lp_i}$ and $\overline{\beta}_{p_j} = \frac{1}{n}\sum_{l=1}^{n} \beta_{lp_j}$

For the first three algorithms, the smaller $d(p_i, p_j)$ is, the larger the correlation between the two pages is. But for the last algorithm, the lager $d(p_i, p_j)$ is, the larger the correlation between the two pages is.

2.3 Clustering process

An individual with observation sequences (sessions) $O = \{O_1, O_2, \cdots O_{n_i}\}$. Each sequence in O consists of page requests categorized into K categories. Note that each sequence here can be of a different length from others. That is, there maybe 10 web pages in a sequence (session) and 25 web pages in another sequence (session). These K categories are automatically categorized, based on the nature of the web site.

We propose that human should be the authority to judge the correlation of two pages. In this paper, a group of users' interestingness in web pages is used to cluster web pages. The assumption of the dynamics clustering is that if a group of users who have the same interest trend, those pages they have visited are probably related.

The following steps are used to cluster the web pages according to user's interestingness.

Step 1: Let $k = 1$

Step 2: Find P_k. P_k is the web page in category k $(1 \le k \le K)$ that has been browsed most by the individual.

Step 3: Let page P_k be the prototype in the clustering. Part goal of the clustering process is to find out those web pages that are most correlated with web page P_k under the meaning of interestingness. This step includes the following subordinate steps.

①Find out the user group that has high interest in page P_k. This is done by checking over the users who have visited the web page P_k for more times than a predefined threshold. In order to put it simply, the times the specific individual browsed the page P_k is chosen as the threshold. Thus the user group is collected. $G_k = \{U_1, U_2, \cdots, U_n\}$ implies that there are n users that has high interest in page P_k.

②For every person in the user group G_k, find out the web pages in his or her interest. Collect all the observation sequences with web page P_k included, generated by the user U_i. Collect all the web pages in these sequences. Let the user U_i's interest set of web pages on Web page category k be P_{U_i}. Repeat this subordinate step until all the users' interest sets are checked over. Find the union of all these web pages according to equation 6. Thus a web page set $\mathrm{P}(category = k)$ is got.

$$\mathrm{P}(category = k) = P_{U_1} \cap P_{U_2} \cap \cdots \cap P_{U_n} \cup P_{U_0} \quad (6)$$

Where P_{U_o} is the set of web pages on category k $(1 \le k \le K)$ that have even been browsed by the specific individual.

Note that here the union of web pages sets is used. The aim of it is to cross out most of the web pages that

are included because of the group users' other interests than the specific individual's interest.

③Calculate the correlation between the web page P_k and each page in the web page set $\mathrm{P}(category = k)$ (using equation 2, 3, 4 or 5). Filter the web page set with a predefined threshold. The threshold is predefined based on experience and the chosen correlation computation algorithm. All the web pages with a correlation over the threshold (if one of the first three algorithms is chosen) are crossed out from the web page set $\mathrm{P}(category = k)$. All the web pages with a correlation under the threshold (if the last algorithm is chosen) are crossed out from the web page set $\mathrm{P}(category = k)$.

Step 4. If $k < K$ then $\{k = k + 1$; Go to Step 2;$\}$ else end the clustering process.

After the clustering process, we get K categories of Web pages. Note that a particular Web page maybe assigned to one or a few of the K categories.

3 HMM construction and Baum-Welch parameter estimation algorithm

In this paper, discrete HMM is proposed to model clickstreams of web users and Baum-Welch parameter estimation method in EM algorithm is used to train the HMM. A HMM Model is specified by a set of states $S = \{s_1, s_2, \cdots, s_N\}$, and a set of parameters $\Theta = \{\pi, A, B\}$ [8][9][10]. These parameters are listed as follows.

①The prior probabilities $\pi_i = P(q_1 = s_i)$ are the probabilities of s_i being the first state of a state sequence. Collected in a vector π.

②The transition probabilities are the probabilities to go from state i to state j: $a_{i,j} = P(q_{n+1} = s_j \mid q_n = s_i)$. They are collected in the matrix A.

③ The emission probabilities characterize the likelihood of a certain observation o, if the model is in state s_i. For discrete observations, $o_n \in \{v_1, \ldots, v_k\}$: $b_{i,k} = P(o_n = v_k \mid q_n = s_i)$, the probabilities to observe v_k if the current state is $q_n = s_i$. The numbers $b_{i,k}$ can be collected in a matrix B.

In this paper, the probability that a user jumps from page category i to category j: $a_{i,j} = P(q_{t+1} = j \mid q_t = i)$ is represented by elements in the transition matrix. The categorization of a web page is determined by the probability that a certain web page $o_t = l$ is observed in a clickstream at time t, given that the page category at time t is $q_t = i$. This is represented by elements in the observation matrix B, $b_{i,k} = P(o_t = l \mid q_t = i)$. Here l denotes an observed page id. The prior probabilities $\pi_i = P(q_1 = i)$ are the probabilities of category i being the first category in a hidden category sequence. The hidden Markov model of web user clickstream is shown in Fig. 1.

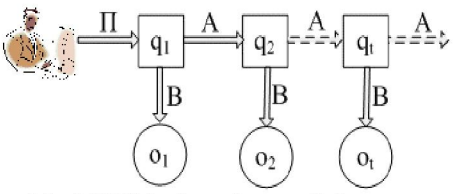

Fig.1 HMM of a web user clickstream

Then the likelihood of an observation sequence (a specific click stream) $o = \{o_t\}$ with respect to a HMM with parameters $\Theta = \{\pi, A, B\}$ is $L(q_1, \ldots, q_t, o_1, o_2, \ldots, o_t)$.

$$\begin{aligned} & L(q_1, \ldots, q_t, o_1, \ldots o_t) \\ & = P(o_1, \ldots, o_t \mid q_1, \ldots q_t) P(q_1, \ldots q_t) \\ & = \pi(q_1) \times \prod_{i=1}^{t} P(o_i \mid q_i) P(q_i \mid q_{i-1}) \end{aligned} \tag{7}$$

The likelihood of being in category $k \quad (1 \le k \le K)$ at time t given observation sequence $o = \{o_t\}$ with respect to a HMM with parameters $\Theta = \{\pi, A, B\}$ is $L(q_t = k, o_1, o_2, \ldots, o_t \mid \Theta)$.

$$\begin{aligned} & L(q_t = k, o_1, o_2, \ldots, o_t \mid \Theta) \\ & = \sum_{K} L(q_1, \ldots, q_t = k, o_1, \ldots o_t) \end{aligned} \tag{8}$$

Where $q_1, q_2, \cdots, q_{t-1}$ is evaluated $\{1,2,3, \cdots K\}$ respectively.

Once the likelihood is defined, the problem is to optimize the parameters $\Theta = \{\pi, A, B\}$ such that $P(o \mid \Theta)$ maximal. Using Baum-Welch Parameter Estimation Method, we can get the optimal solution of these parameters. The implementation of the method is depicted as follows:

/*Beginning of Baum-Welch algorithm*/

Initialization: Θ_0, ε

/* ε is the value of allowable error*/

Calculation: $\Theta = \{\overline{\pi}, \overline{A}, \overline{B}\}$

$\alpha_t(i) = P(o_1 o_2 \cdots o_t, q_t = i \mid \Theta)$

/* $\alpha_t(i)$ is the Forward variable*/

$\beta_t(i) = P(o_{t+1} o_{t+2} \cdots o_T \mid q_t = i, \Theta)$

/* $\beta_t(i)$ is the Backward variable*/

$$\zeta_t(i,j) = P(q_t = i, q_{t+1} = j \mid o, \Theta)$$
$$= \frac{P(q_t = i, q_{t+1} = j, o \mid \Theta)}{P(o \mid \Theta)}$$
$$= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}$$

/* $\zeta_t(i,j)$ is the conditional probability, at time t, from state i to state j */

$$\gamma_t(i) = P(q_t = i \mid o, \Theta) = \sum_{j=1}^{n} \zeta_t(i,j)$$

/* $\gamma_t(i)$ is the conditional probability of state t at time j */

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \zeta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \overline{b}_{i,k} = \frac{\sum_{t=1}^{T-1} \gamma_t(j) \mathrm{I}(o_t = k)}{\sum_{t=1}^{T-1} \gamma_t(j)}$$

/*where $\mathrm{I}(o_t = k)$ is an indicator function which equals to 1 if $o_t = k$ and equals to 0 otherwise.*/ $\overline{\pi}_i = \gamma_1(i)$

If $\left|\log P(o \mid \Theta) - \log P(o \mid \Theta_0)\right| < \varepsilon$ end;

Otherwise { let $\Theta_0 = \Theta$, goto calculation}

/*End of Baum-Welch algorithm*/

After training our model with the training data, we can get the parameters $\Theta = \{\pi, A, B\}$.

The next step is computing the likelihood function (equation 8) given an observation sequence. Category or categories with likelihoods above the predefined threshold are the output of the model based on hidden Markov model. Thus we get the Web page category or categories the Web user will most likely transit to. According to the web page categories we acquire in the above clustering process, we will get some information of Web page links to dynamically create the next Web page presented to the user.

4 Experiments

In the following section we report various experimental results. There are two main points should be presented first.

First, the raw data source is log files from a large Intranet web site of a company. There are several reasons for choosing this web site. One is that it is one of our customers from which we can get the first-hand real information. The other is that there are certain users in the system and each user has accumulated large amounts of log files records. These features avoid unnecessary "noises" in the raw data set and help to proof-test the performance of our Web mining model. Details of the data are proprietary. However, we have been authorized to presented part of the experiment results.

Second, sessions are defined as any set of page requests where the gap between successive requests does not exceed 30 minutes. Thus the raw entries in the file were 'sessionized' into observation sequences and irrelevant entries (like directories, gifs, and CGI requests) were removed.

Now, choose one of the users from the raw data set arbitrarily. The 47 sessions of this user can be automatically categorized into six categories. These sessions (observation sequences) include 53 independent web pages (page requests). We used all the four algorithms listed in session 2 to cluster web pages for a specific web page and got four different web pages sets. Whether those collected pages are correlated with the specific web page or not are determined manually afterwards. Thus, we got the shoot rates as shown in Fig. 2. In Fig. 2, "Eu" stands for Euclidean distance algorithm, "Ma" for Manhattan distance algorithm, "Mi" for Minkowski distance algorithm (p=3) and "Pe" for Pearson Correlation Coefficient algorithm.

From Fig. 2, we know that the Pearson Correlation Coefficient algorithm has the highest shoot rate in our data set. Therefore, we choose it as the correlation algorithm in the following experiment.

The web pages clustering results are shown in Tab. 1.

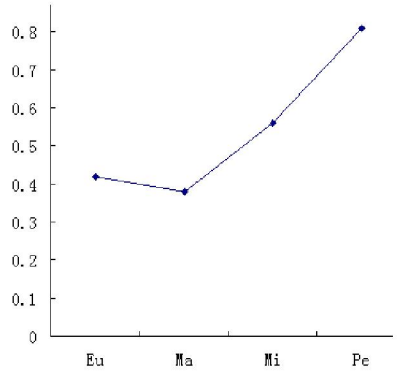

Fig.2 Shoot rate of different correlation algorithms

Tab.1 Results of clustering web pages process

Category k	Users in Group G_k	Thresholds	Pages in the Category k	No. of sessions (sequences)
1	11	0	46	146
2	8	0	37	102
3	13	0	23	297
4	17	0	56	373
5	9	0	35	165
6	3	0	8	37

After training our Web mining model on the training data set ($\varepsilon = 0.01$), we got the observation matrix B, the page categories transition probability matrix A and the prior probabilities vector π. The parameters are displayed in Fig. 3.

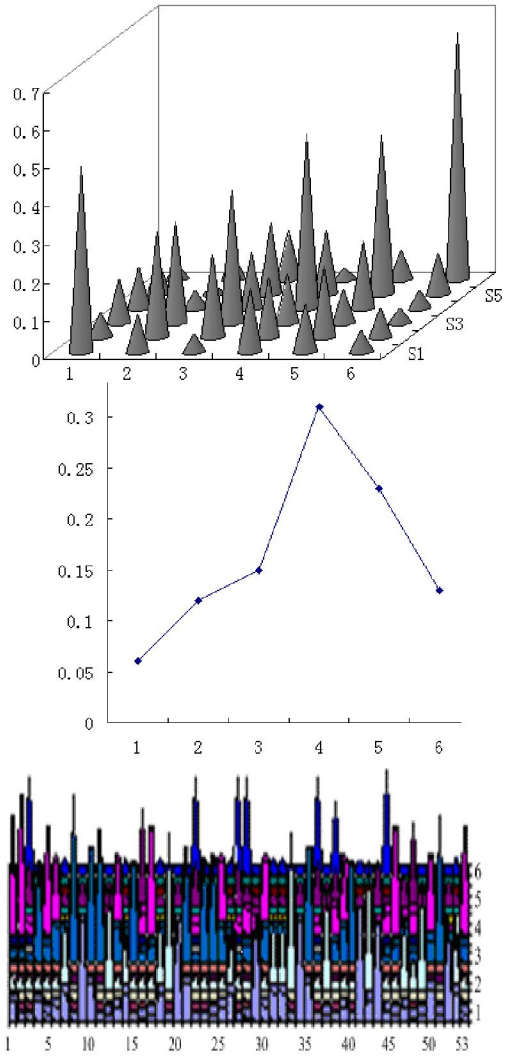

Fig.3 a. Learned 6-state transition matrix A
b. The prior probabilities π;
c. The observation matrix B

For example, we suppose the user input the following Web page requests: www.xxx.com, www.xxx.com/securityproducts.htm; www.xxx.com/itbook/security.htm.

The outputs of our Web mining model are listed as follows:

www.xxx.com/products/firewallspro.htm
www.xxx.com/itbook/security/book.asp?id=135
www.xxx.com/itbook/security/book.asp?id=21
www.xxx.com/crm/customers?id=304
www.xxx.com/netinfo/prolink.htm
……

There are 16 results according to the predefined thresholds. About 31% of these Web pages can be observed from the user's log records. The rest 69% pages recommended by our model have never been browsed by the user. These recommendations are quite useful according to the user.

5 Discussions and conclusion

In this paper, we presented a Web mining model to personalize user's Web pages. This model is based on dynamic clustering and hidden Markov model. We illustrated the clustering process and the predicting process based on HMM and implemented the empirical case study. Because the technology of dynamic creating web pages is still in its infancy, we cannot proof-test the performance of our model in depth by now. In future research, there is still a lot of work to do to improve and optimize the model.

References

[1]Soumen Chakrabarti, Martin van den Berg, Byron Dom: Focused crawling: a new approach to topic-specific web resource discovery. Proceedings of the WWW8 Conference, 1999.

[2]Cadez, I., Gaffney, S., Smyth, P.: A general probabilistic framework for clustering individuals. Technical report, Univ. Calif., Irvine, March 2000.

[3]Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Visualization of navigation patterns on a Web site using model-based clustering, in Proceedings of ACM SIGKDD 2000, New York, NY: ACM Press.

[4]Smyth, P.: Clustering sequences with hidden markov models. In M.C. Mozer, M.I. Jordan, and T. Petsche, editors, Advances in NIPS 9, 1997.

[5]Smyth, P.: Probabilistic model-based clustering of multivariate and sequential data. In Proc. of 7th Int. Workshop AI and Statistics, 1999, 299–304.

[6]Billsus, D., and Pazzani, M.: A hybrid user model for news story classification. Proceedings of the Seventh International Conference on User Modeling, 1999.

[7]Sarukkai, R.R.: Link prediction and path analysis using Markov chains. In Proceedings of the Ninth International World Wide Web Conference, Amsterdam, 2000.

[8]Barbara Resch. Hidden Markov Models, a Tutorial for the Course Computational Intelligence. http://www.igi.tugraz.at/lehre/CI, 2001.

[9]Premaratne HL, Jarpe E, Bigun J. Lexicon and hidden Markov model-based optimisation of the recognised Sinhala script. Pattern Recognition Letters 2006, 27 (6), 696-705.

[10]Ge HW, Liang YC. A hidden Markov model and immune particle swarm optimization-based algorithm for multiple sequence alignment. Lecture Notes in Artificial Intelligence 2005, 3809, 756-765.