An Approach to Web Prefetching Agent Based on Web Ontology with Hidden Markov Model

Xin Jin

School of Information, Central University of Finance & Economics, Beijing,
100081, P.R. China
d0229jin@mail.dhu.edu.cn

Abstract. With the rapid growth of web services on the Internet, users are experiencing access delays more often than ever. Recent studies showed that web prefetching could alleviate the WWW latency to a larger extent than the traditional caching. Web prefetching is one of the most popular strategies in web mining research domain, which are proposed for reducing the perceived access delay, improving the service quality of web site and mining the user requirement information in advance. In this paper, we introduce the features of the web site model named web ontology, and build a web prefetching agent-WebAGENT based on the web ontology and the hidden Markov model. With the agent, we analyze the user access path and how to mine the latent information requirement concepts, then we could make semantic-based prefetching decisions. Experimental results show that the web prefetching scheme of the WebAGENT has better predictive mining effect and prefetching precision.

1 Introduction

Due to the rapid development of Internet technique and the exponential growth of online information, Internet has become one of the most important information sources. However, owing to the limitation of the bandwidth, users always suffer from long delay time when they access web pages. In order to solving the problem, experts proposed a lot of solutions. Web prefetching is the most prominent one [1,2,3].

Web prefetching is an active caching scheme [4]. Compared with the normal passive caching scheme, web prefetching predicts the next request for web documents based on the current request of users through analyzing the server log data, fetches them in advance and loads into the server cache. It reduces the perceived access delay in some exent and improves the service quality of web server [5,6,7].

In this paper, we propose a novel web prefetching approach based on web ontology with the Hidden Markov Model(HMM). By utilizing the HMM, we analyze the user's browsing track, capture the user's actual information requirement intention, and make semantic-based web prefetching decisions.

Remainder of the paper is organized as follows. Section 2 describes the details of our prefetching tasks and algorithms of the web fetching agent named WebAGENT. Section 3 describes the experiments designed and evaluates the performance of the scheme based on our web prefetching agent. Section 4 provides a summary of this work.

O. Castillo et al. (Eds.): Theor. Adv. and Appl. of Fuzzy Logic, ASC 42, pp. 535–543, 2007.
springerlink.com

2 Web Prefetching

In general, a user always accesses the web site with certain intention. Driven by the intention, the user follows a link on a page that he is currently visiting and browses continually until that he or she satisfied. In other words, access path contains the user's certain requirement intention. If we extract the latent information requirement concepts from the user access path, we can make accurate decisions for web prefetching based on it. For example, let us consider that a given user has accessed the following pages Pa, Pb and Pc, according to the following sequence Pa→Pb→Pc . Page Pd, Pe and Pf are cited by page Pc. If we find that the concept c is latent intention that the access path implies and page Pe contains the concept c, we can pre-fetch page Pe for the user's next request.

Obviously, mining information requirement concepts that access path implies is crucial to web prefetching. Our approach is based on the following observations:

Observation 1: The author of web page always uses hyperlink to implement the organization of server host Hyperlink establishes the relation between web pages in certain concepts.

Observation 2: Anchor text of hyperlink can provide users with sufficient information. It generalizes the major content of linked page.

The conceptualization architecture of the web site, organized by hyperlink, can always be considered as a web ontology, which can help us to describe and analyze the web site structure. Based on the two observations above, we utilized the web ontology and the HMM to implement the concept mining. The web pages that access path includes can be denoted as the states of the HMM, the concepts that web pages contain correspond to the observer symbols, respectfully. Therefore, the observation symbol sequence of the HMM is the concept sequence actually. In fact, the probability of concept sequence about a certain concept is the possibility of the latent requirement concept implied by user access path. According to the probability calculated, we can choose some concepts as the latent requirement concepts, and evaluate the web pages that the current page cited. The web pages that satisfied these concepts as more are picked out as prefetching objects.

2.1 Model Description

2.1.1 Web Ontology

Proposition: Consider web ontology as conceptualization architecture of the web site, which is a tuple structure<D, R> where D is a domain, and R is a set of relevant relations on D.

For a web site ontology, the domain D can be considered as the set of all existing web pages, denoted as D={p| p∈D}, and R is the set of hyperlinks in all web pages to link the different pages, denoted as R={l| l∈R} .

- *Definition 1*: The web page p can be denoted as p =(P_ID, P_Url), where P_ID is the unique ID of the page, P_Url is the URL of the page.
- *Definition 2*: The hyperlink can be defined as a quadruple l=(L_ID,Anchor_Text,SP_ID,EP_ID), where L_ID is the unique ID of the hyperlink,

Anchor_Text is the anchor text around the hyperlink, SP_ID is ID of the web page that contains the hyperlink , EP_ID is ID of the page which the hyperlink points to.

- *Definition 3*: User access path E is a request sequence on domain D, defined as E=<p_1,p_2,…p_i,…,p_n>, where pi is the web page that the user requests for in the i^{th} step , p_n is the current page, n is the length of access path. The candidate pages for web prefetching are all web pages that the current page cited.
- *Definition 4*: Session S is a request sequence of the user in a certain time interval, defined as S=<s_1,s_2,…,s_n>, where each $s_i \in S$ is the web page the user accesses in the ith step of the session, n is the length of the session. The page b follows page a, denoted as a→b, iff there exists s_i =a and s_i+1 =b($1 \le i < n$).
- *Definition 5*: C is the Concept set of user's information requirement, denoted as C={c_1,c_2,…,c_m} , which is fetched from the web ontology.

In order to fetch the concepts of user's requirement in the web ontology, we use the HMM to analyze the user access path E from the domain D. The HMM is well known and widely used statistical method of characterizing the spectral properties of the frames of a pattern, which was proposed firstly by Baum and his colleagues in the late 1960s .

2.1.2 Describe the HMM

- Set the HMM to be λ = (A, B,π), [7]
- N, the number of states in the model.
- Described the individual states as {1,2,…,N}
- Denoted q_t as the state at time t.
- M, the number of distinct observation symbols per state.
- Denoted the individual symbols as
  V={v_1,v_2,…,v_M}.
- A={a_{ij}}, The state transition probability distribution, where

$$a_{ij} = P(q_{t+1} = j \mid q_t = i) , \quad i,j \le N \tag{1}$$

- B={$b_j(k)$}, The observation symbol probability distribution, in which

$$b_j(k) = P(v_k(t) \mid q_t = j), \; 1 \le k \le M \tag{2}$$

defines the symbol probility distribution in state j, j =1,2,…,N.

- The initial state distribution π={π_i}, in which

$$\pi_i = P(q_1(t) = i), \; 1 \le i \le N, \; t=0 \tag{3}$$

In order to make the HMM model suitable to web prefetching from the web ontology, we let the states of the HMM corresponding to the web pages of user access path *E* respectively, and the observation symbols correspond to the concepts of the Set C respectively. That is, the state q_i corresponds to the web page p_i and the symbol v_i corresponds to the concept c_i.

2.2 Web Prefetching Agent - WebAGENT

Based on the web ontology with HMM, we build a web prefetching agent named WebAGENT, whose tasks include four-phase processing as the Fig.1: preprocess the server log data, extract the requirement concepts, analysis user access path and fetch the web pages. Before fetching web page, we will analyze the user access path, mine the latent information requirement concepts, then, based on it, we make predictive prefetching decisions. We will give a detailed introduction as follows.

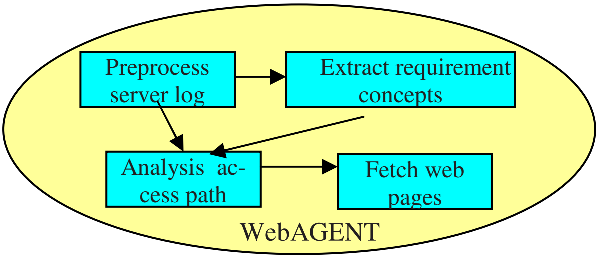

Fig. 1. The architecture of the WebAGENT. This also shows the tasks of the WebAGENT

2.2.1 Preprocess the Server Log Data for Prefetching

The web prefetching is constructed on the basis of analyzing the server log. The WebAGENT preprocess the server log file, extract user sessions and form a collection of user sessions. Algorithm 1 shows the main procedure.

Algorithm 1. Preprocessing the Server Log

1) Removing all false requests and requests for web pages such as graphic files, .cgi files.

2) Segregating the log file into the independent collections of requests according to IP address.

3) Processing each independent request collection, respectively.

- Sorting the requests by access time.
- Extracting sessions from the user request collection. For any two adjacent requests, if the time interval between its access time is smaller than the time threshold *tw*, these requests belong to the same user session.
- Gathering all sessions of the user into the user's session collection.

4) All user session collections compose the server session collection.

2.2.2 Extract the Requirement Concepts

According to Observation 1, access path can be regard as the deep search for certain concepts. Observation 2 shows that anchor text of hyperlink generalizes content of page linked. The user can judge directly whether the page linked is worth browsing or not, without having actually read the page. Therefore, Let *p* be the current page of access path, anchor texts of all hyperlinks that page *p* contains are the basis on which the user chooses the next page. So All concepts that those anchor texts contain compose the user's information requirement concept collection. We define a pseudo document, denoted as *HyperDoc* , which contains all anchor texts of hyperlinks in the current page *p*. We have:

$$HyperDoc = \bigcup_{l' \in L} l'.Anchor\_Text \tag{4}$$

where, *L* is the set of hyperlinks,

$$L = \{ l' \mid l'.SP\_ID = p.P\_ID \}$$

HyperDoc is processed for generating the requirement concept set. The main steps can be described in the following algorithm 2.

Algorithm 2. The Main Procedure of Composing the Requirement Concept Set *C*

1) Removing all html tags from *HyperDoc*.
2) If *HyperDoc* is in Chinese, word segmentation is for *HyperDoc* .
3) Removing all words that belong to a stop-word list.
4) If *HyperDoc* in English, all words are stemmed.
5) All results from above steps form the user's requirement concept set *C*.

2.2.3 Analysis User Access Path

Following the above steps, the WebAGENT calculates the probability of concept sequence of each concept in the requirement concept set. For example, Let E=<p1,p2,…,pn-1,pn> be the user path access, where pn is the current page, n is the length of access path. $o = (\underbrace{c_i c_i \ldots c_i}_{n})$ is the observation sequence, where ci ∈ C. The probability of *O* represents possibility that the concept c_i is latent requirement concept that access path *E* implies.

While responding access requests of users, the web server imports the request into corresponding user buffer. The user buffer is flushed once a time threshold *tw* is exceeded. The request sequence in the time intervals *tw* will be regarded as the current user's access path. Because the access path *E* is fixed, we only consider the access path as the fixed–state sequence. The probability of the observation sequence O is

$$P(o \mid q, \lambda) = \prod_{t=1}^{n} P(o_t \mid q_t, \lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \cdot \ldots \cdot b_{q_n}(o_n) \tag{5}$$

where, $b_{q_j}(o_i)$ is the probability of observation symbol o_i in state q_j. Transforming it into the probability of the concept observation sequence, we have

$$P(o \mid q, \lambda) = b_{p_1}(c_i) \cdot b_{p_2}(c_i) \cdot \ldots \cdot b_{p_n}(c_i) \tag{6}$$

where, $b_{p_j}(c_i)$ is the observation concept probability of the concept c_i in the page p_j.

Considering the characteristics of access path, main action of access path details certain concepts, and leads users to web pages that they satisfy. So, we calculate the page's navigation capability for these concepts as observation concept probability. For each concept in the requirement concept collection, the weight in a page formula is defined as follow,

$$W_p(c_i) = \frac{\left(\frac{tf_i}{tf_{\max}}\right)}{\sqrt{\sum_{j=1}^{m}\left(\frac{tf_j}{tf_{\max}}\right)^2}} \tag{7}$$

where, tf_i is the frequency of the concept c_i in the document p, $tf_{\max}$ is the maximum concept frequency of all concepts in the requirement concept set C.

Suppose the user's session set is T. Let $S(p)$ be the page set that contains all pages that follow the page p in the user sessions,

$$S(p) = \{ p' \mid \forall s' \in T, p' \in s', p \rightarrow p' \} \tag{8}$$

the navigation capacity of page p to the concept c_i is estimated as follows

$$Nav_p(c_i) = \sum_{p' \in S(p)} W_{p'}(c_i) \tag{9}$$

It is normalized by dividing each weight of a concept by the sequence root of the sum of the squared weights.

$$\overline{Nav_P}(c_i) = \frac{Nav_p(c_i)}{\sum_{c' \in C} Nav_p(c')} \tag{10}$$

The $b_p(c_i)$ is computed by using the following formula,

$$b_p(c_i) = \overline{Nav}_P(c_i) \tag{11}$$

The modified function $P(o \mid q, \lambda)$ is stated as follows:

$$P(o \mid q, \lambda) = \overline{Nav}_{p_1}(c_i) \cdot \overline{Nav}_{p_2}(c_i) \cdot \ldots \cdot \overline{Nav}_{p_n}(c_i) \tag{12}$$

2.2.4 Fetch the Web Pages

Based on the modified function described above, the agent sort the concept set by the probability of the observation sequence calculated and choose the first τ concepts to form the concept set η. In other word, we have $\eta = \{c_1, c_2, \ldots, c_\tau\}$, where c_i is the ith concept. In our implementation, it is reasonable to set τ to 7. The prefetching prior score of page p is estimated with the following formula.

$$Score(p) = \sum_{c' \in \eta} W_p(c') \tag{13}$$

Using this formula, the agent calculates the prior score of all web pages that the current page of the user access path cited. According to the prefetching threshold θ, the first θ pages are the prefetching pages and are load into the server cache in advance.

3 Experimental Evaluation

3.1 Experiment Design

3.1.1 Dataset

To evaluate the performance of the web prefetching scheme of the WebAGENT, we conducted experiments on the Test web server (http://test.dhu.edu.cn). The Test web server includes 865 html pages, main topics of which are the related techniques about Chinese information processing. We obtained the server log from Jun. 1, 2004 to Sept. 30, 2004, which contains 22169 user requests. Set the time threshold *tw* to be two hours. Given the preprocessing step outlined above, the characteristics of the server log are shown in the Table 1. For evaluation purpose, we divide the complete dataset into the training dataset and the testing dataset, which the training dataset contains the first 1500 sessions and the testing dataset contains the remaining sessions.

Table 1. The characteristics of the server log preprocessed

Total of user requests	22169
Total of user session collections	212
Total of Sessions.	3128
Avg. Session Length	7
Min. Sesesion Length	1
Max.Session Length	10

3.1.2 Evaluation Metrics

We define the following measures to evaluate the performance of the WebAGENT:

Definition 6: *Request Hit Ratio* is the ratio of number of pages requested that are accurately prefetched and the total pages requested.

Definition 7: *Session Hit Ration* is the ratio of number of pages requested that are accurately prefetched and the total pages requested in a session.

3.2 Experimental Results and Analysis

3.2.1 Experimental Results

Firstly, we measured *Request Hit Ratio* in the different step of the user access path. So as to do it, we set the prefetching threshold θ to be 4, regard each session in the testing set as the current user access path and calculate the *Request Hit Ratio* of the model in each step. Figure 2 shows the relation between *Request Hit Ratio* and number of the steps of the current access path(θ =4).

Secondly, we measured the avg. *session Hit Ratio* in the different prefetching threshold θ. Figure 3 shows the relationship between avg. *Session Hit Ratio* and the prefetching threshold θ.

3.2.2 Experimental Analysis

As can be seen, In figure 2, we observed that *Request Hit Ratio* of the scheme increases rapidly from 50.4% to 58.6% when number of steps increases from 2 to 7, but number of steps exceeds 7, it begins to drop and decreases from 58.6% to 55.5%. The main reason is that ongoing bias of the user's interests may occur as step exceeds the certain value, introduce noise into predictive web prefetching, and reduce the predictive precision, but the schema is always higher than the average value 50%. As Figure 3 shows, avg. *Session Hit Ratio* of the scheme increases rapidly while the prefetching threshold θ increases from 1 to 4, the rate of increase declined after θ exceeds 4. we observe that our scheme has better avg. *Session Hit Ratio* than the non-prefetching web access.

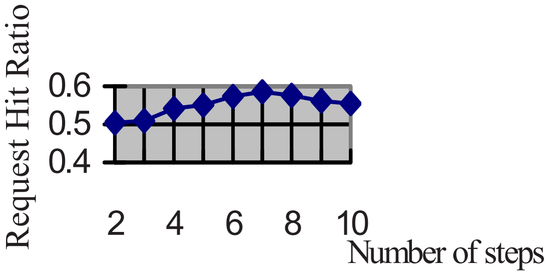

Fig. 2. Request Hit Ratio Vs Number of Steps (θ =4). This shows the relation between *Request Hit Ratio* and number of the steps of the current access path(θ =4).

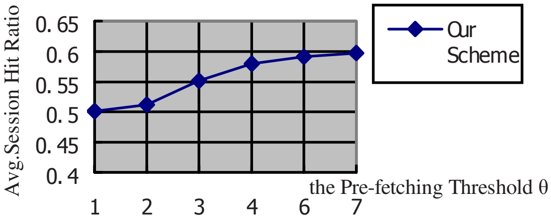

Fig. 3. Avg. Session Hit Ratio Vs the Prefetched Threshold θ. This shows the relationship between avg. *Session Hit Ratio* and the prefetching threshold θ.

4 Conclusions

Web prefetching reduces significantly the perceived latency and improves the service quality of web site, which is implemented successfully in many correlated applications. In this paper, we built the web prefetching agent-WebAGENT based on web ontology with HMM. The web prefetching scheme of the WebAGENT is based on the idea that it could make semantic-based prefetching decision in virtue of mining the latent information requirement concepts that the user access path implies. Web prefetching has common feature with other web applications that involve prediction of user access pattern. We hope this approach can be useful for reference by some relative research domains.

References

1. Evangelos P. Markatos and Catherine E.Chironaki, A Top 10 Approach for prefetching the web, *Proceedings of INET'98: Internet Global Summint* , July 1998
2. Styart Schechter, Murali Krishnan, and Michael D. Smith. Using Path profiles to predict http requests. *Proceedings of WWW7*, 1998.
3. SuYoung Yoon,Eunsook Jin, Jungmin Seo, Multimedia Technology ResearchLab, Korea, Telcom,http://www.isoc.org/inet99/proceedings/posters/106.
4. M.Deshpande, G.Karypis, Selective Markov Models for Predicting Web_Page Accesses, *Proceedings SIAM Int.Conference on Data Mining(SDM'2001)*,Apr.2001.
5. Sarukkai,Ramesh R.. Link Prediction and Path analysis Using Markov Chains, *9th World Wide Web Conference* ,May 2001.
6. XU Bao-wen,ZHANG Wei-feng, Applying Data Mining to Web Prefetching. *Chinese J.Computers,*2001,24(4):1-7.
7. Lawrence Rabiner, Riing-Hwang Juang, *Fundamentals of speech recognition*, Prentice Hall PTR. 1993:312-389.