Pre-Fetching Web Pages
Through Data Mining Based Prediction

Weifeng Zhang and Baowen Xu♣
Southeast University, China

William Song
Hong Kong University, Hong Kong

Hongji Yang ♥
De Montfort University, England

Abstract

The speed of fetching web pages to users is getting lower because the rapid expansion of Internet use, the inherited character of delay in the network and the Request/Response working mode of WWW, and this is becoming a serious concern for web surfers. In order to speed up fetching web pages, this paper presents an intelligent technique of web pre-fetching. We use a simplified WWW data model to represent the data in the cache of web browser to mine the association rules. We store these rules in a knowledge base so as to predict the user’s actions. Intelligent agents are responsible for mining the users’ interest and pre-fetching web pages, based on the interest association repository. In this way, browsing time has been reduced transparently.

Keywords: WWW, Internet, Data Model, Data Mining, Pre-fetching, Cache

1. Introduction

With the rapid development of Internet, WWW is popular for its multimedia transmission and friendly interactivity. Although the speed of network has been improved considerably in recent years, the rapid expansion of the Internet, the inherited character of delay in the network and the Request/Response working mode of WWW [10] still make the internet traffic very slow and give no guarantee on the Quality of Service. Because HTTP has no states, the web server cannot know the users’ demand and the users’ requests cannot be predicted. We believe

♣ This work was supported in part by the National Natural Science Foundation of China (NNSFC).
♥ Correspondence author: Dr. Hongji Yang, Computer Science Department, De Montfort University, Leicester, LE1 9BH, England, hyang@dmu.ac.uk.
that taking advantage of a cache mechanism and the time locality of WWW access, the browser can preserve the documents ever accessed in the local machine. By this means, for the documents in the local cache, the browser does not need to send the requests to the remote server or to receive the whole response from the remote one. This is regarded as pre-fetching. Pre-fetching uses the space locality of access. First, the users’ access requests are predicted according to the users’ current request. Second, the expected pages are fetched into the local cache when the user is browsing the current page [6]. Finally, the users can access these pages downloaded from the local cache. This can reduce the access delay to some degrees. Pre-fetching is one kind of active caches that can cache the pages which are still not requested by the user. The application of pre-fetching technique in the web can greatly reduce the waiting time after users have sent their requests.

Pre-fetching technique is based on prediction algorithms. Data mining is a technique, with which a lot of unknown, implicit but potentially useful information, such as knowledge, rules and regularities, for decision-making are mined. Based on the historic data and the current data accessed by the user, user activities in the future are predicted through data mining and relevant web pages are pre-fetched. The data in the users’ local cache can be taken as the data sources of data mining.

This paper concentrates on the above issues and discusses the application of data mining techniques in the pre-fetching of web pages. First, a WWW data model is given for describing web pages and their interrelations. Second, we put forward an interest association repository and a predicting method. Then an algorithm of transforming the simple WWW data model to the interest association repository is introduced. Finally, we present an agent-based web pre-fetching system.

2. A Simple WWW Data Model

Most web pages are written in HTML language. By using MIME, a web page can contain different types of elements. A hyper media system [1] is made up of web pages linked by hyper-links between each other. To predict a user’s activities, a data model is needed to describe the interest association rules in the web pages. Several data models [3, 4, 5, 7, 17, 12] have been proposed to describe the WWW. A general data model for describing the WWW usually includes the definitions of many elements, such as page nodes, inter-node hyper-links, internal structure of page nodes, and even metadata definition like DTD (data type definition) [16]. In order to meet the demand of data mining and represent the characteristic of the cache, this paper proposes a simple WWW data model. Some concepts and definitions of this model are presented below.

**Definition 1:** Let triple \((Id, P, time)\) be a page node, where each page node is associated with an \(Id\), uniquely identifying a page node, \(P\), a collection of properties, \(P = \{ p_i | i = 1, 2, \ldots \}\), and \(time\), the latest time
when the page node is accessed. A property \( p_i \) can be a relative URL, a type, the collection of linking nodes, the container, the content, the updated time and so on.

According to whether links between web pages exist [2] and the pointing direction of the links in the web pages, the page nodes can be classified into isolated page nodes, source page nodes and target page nodes. A page node, which does not contain any links, is called an isolated page node. A page node, which contains links, is called a source page node of such links. While a page node, which is pointed to by other links, is called a target page node of such links. Apparently, according to different links, one page node can be a source page node or a target page node of such links.

**Definition 2:** Let triple \((Id, string, target_node_id)\) be a linking node, where each \(Id\) identifies only one linking node; \(string\) describes the presenting information of this linking node; and \(target_node_id\) is the id of the target page node.

**Definition 3:** Let triple \((source_node, L, target_node)\) be a link, where \(source_node\) is the source page node; \(L\) is the linking node; and \(target_node\) is the target page node; \(L \in\) the collection of links in the \(source_node\), \(L.target_node_id = target_node\).

Figure 1 shows an instance of the proposed simple WWW data model. In this instance, there are five nodes and seven relations between the nodes. Each node represents one web page; nodes are inter-linked by the links that clearly represent the association of these web pages.

Nowadays web browsers normally use the cache technique to preserve the latest web pages accessed by users. When users want to access these web pages again, the browser first examines whether these web pages have been cached locally. If so, the browser will then check whether the corresponding web pages have been updated. If so, these web pages have to be fetched from the corresponding web servers. Otherwise, the pages can just be picked up from the local cache. That is to say, it is a passive mechanism in current browsers. The historic data in the cache reflect the users’ interest in browsing information and the associating information among the users’ interest is useful to foresee the users’ activities. The associations among the web pages in the cache can be described easily in the simple WWW data model as in Figure 1. However, such a model, unfortunately, cannot visually represent the associating information of users’ interests. In order to predict users’ activities and implement an active cache, certain measures must be taken to transform the data in the cache from the simple WWW data model to another data model, so that they become more suitable for prediction. This approach will be introduced in the data mining section below.

2. Interest Association Repository and User Action Prediction

When a user is accessing a web page, normally she or he will follow the links in the web page to access other
web pages. Based on the current web page, we can predict the links to be accessed by the user, then pre-fetch these web pages pointed out by the predicted links, and hence speed up browsing. It is supposed that the browser can pre-fetch all the pages that are linked by the current web page. This approach works better when there are only a few links pointing to different web servers. However, the approach is not advisable. A reason is that it is generally not possible that the user accesses all the links in the web page. Another reason is that, if a fee is charged by network stream, the user may pay for what is not needed. Further more, the streams of the Internet system should be balanced. Therefore, in this paper, we propose to sort the links in the web page according to certain policies and only pre-fetch a small number of web pages, which are more probable to be accessed. Considering the response time of the system and the difficulty level in the realisation of the different systems, we apply simple association rules to predict the users’ activities. The interest association repository is made up of some interest association rules. The interest set (dictionary) $T$ is represented as $\{t_1, t_2, \ldots, t_m\}$, where $t_1, t_2, \ldots, t_m$ are lemmas. Here are more definitions.

**Definition 4:** The interest node is defined as a pair: $\text{Node} (t) = (t, \text{weight})$, where $t$ is the lemma in the interest set $T$ and weight is the weight of $t$.

**Definition 5:** An interest association rule is a triple, $\text{Rule} (\text{Node}(t_s), \text{Node}(t_t)) = (\text{Node}(t_s), p, \text{Node}(t_t))$, where $p$ is the probability from the interest node $\text{Node}(t_s)$ linking to the interest node $\text{Node}(t_t)$, $0 < \text{weight} < 1$.

**Definition 6:** The interest association repository is a set of interest association rules, where $\text{RULE} = \{\text{Rule} (\text{Node}(t_s), \text{Node}(t_t)) \mid t_s, t_t \in T\}$. The interest association rules in the interest association repository must satisfy:

$$\sum_{\text{Rule}(\text{Node}(t_s), \text{Node}(t_t))} \text{weight} = 1,$$

where $\text{P}(\text{Node}(t_s))$ is the set $\{t_t \mid t_s, t_t \in T, \text{Rule}(\text{Node}(t_s), \text{Node}(t_t)) \in \text{RULE}\}$.

The interest association rules in the interest association repository point out the probability from one lemma (interest) to another. From the interest association repository, the user’s browsing tracks and the current web page, we can predict the most probable accessed links in the current web page. The interest association rules in the interest association repository are captured based on the analysis of great amount of historic data.

We now introduce the construction of the interest association repository in details. The repository can be constructed as the web browsing process is going on. When a user browses a web page, quite often she or he only accesses some web pages of her or his current interests. Such web pages are more valuable than the historic data. Considering the response speed of accessing the web pages, we cannot reconstruct the interest association repository completely. But considering the user’s browsing tracks, we use a method that can modify the interest association repository step by step. Therefore irrelevant data in pre-fetching period is avoided.

Users’ browsing tracks are made up of the users’ current accessing web pages. If all $n$ previous pages must be
processed when the user accesses the \( n+1 \)th web page, the response time cannot be assured when \( n \) is big. So an incremental algorithm which modifies the interest association repository every time works well when the user accesses only one web page.

**Algorithm 1 (Incremental algorithm of adjusting the interest association repository)** Assume that a user is accessing the \( n \)th web page \( Y_n \). Two steps are needed to modify the interest association repository.

**Step 1** - The frequency (number of occurrences in the certain period) of lemmas of \( T \) in the web page \( Y_n \) (the slicing, stemming and thesaurus technologies [17]) is calculated and a set \( K(Y_n) = \{(t_i', f_i) \mid t_i \in T\} \) is obtained, where \( f_i \) is the frequency of \( t_i \) appearing in \( Y_n \), \( f_i > 0, i \in \mathbb{N} \).

**Step 2** - Each Node \((t)\) in the interest association repository is adjusted as follows:

\[
\text{if Node}(t).t = t_i' \text{ and } (t_i', f_i) \in K(Y_n) \text{ then } \text{Node}(t).weight = \text{Node}(t).weight + f_i \times F(n),
\]

Where, \( F(n) \), a monotonous and non-degressive function, representing the “freshness”.

It is easily proved that function \( F(n) \) ensures that the more recently the web pages are accessed, the more favorable to predict these web pages. \( F(n) \) can be \( n, n^2 \) and so on. During this process, the weight of the node in the interest association repository increases gradually according to the user’s accessing sequence, so that the sequence of the user’s accessing tracks has been embodied in the interest association repository. We can use the following steps to predict the links that will be chosen by the user.

**Algorithm 2 (Predicting user activities using the interest association repository)** Assume that a user is accessing the \( n \)th web page \( Y_n \). Four steps are needed:

**Step 1** - Use Algorithm 1 to process the current web page.

**Step 2** - Let \( L = \{l_1, l_2 \ldots l_k\} \) be the all links that exist in the current web page.

\[
\text{for each } l_i \text{ in } L \text{ do }
\]

Slice the lemmas in the string in the \( l_i \) and get

\[
C(l_i.string) = \{t_j' \mid t_j' \text{ is in } l_i.string, j \in \mathbb{N}\}.
\]

**Step 3** - Score all the links in \( L \)

\[
\text{Value}(l_i) = \sum_Q \text{Node}(t_k').weight \times \text{Rule}(\text{Node}(t_k'), \text{Node}(t_j')).weight,
\]

Where \( Q = \{(t_k', t_j') \mid (t_k', f_k) \in K(Y_n), t_j' \in C(l_i.string), \text{ Rule}(\text{Node}(t_k'), \text{Node}(t_j')) \in \text{RULE}\} \).

**Step 4** - Sort \( \text{Value}(l_i) \) in a descending order. The more advanced the links are, the more probable that users will access them.

Select \( s \) foregoing links (If \( s > k \) then let \( s = k \). Generally \( s \) is not more than 5). If web pages pointed by such
links do not exist in the local cache, they will be pre-fetched.

The reliability is an important factor in pre-fetching technology. From the Step 3 above, we can get the reliability of link $l_i$:

$$l_i = \frac{\sum_{Q} \text{Node}(t_k).weight \times \text{Rule}(\text{Node}(t_k^r), \text{Node}(t_j^r)).weight}{\sum_{Q} \text{Node}(t_k).weight},$$

where $Q = \{(t_k^r, t_j^r) \mid (t_k^r, f_k) \in K(Y_n), t_j^r \in C(l_i\cdot \text{string}) \land \text{Rule}(\text{Node}(t_k^r), \text{Node}(t_j^r)) \in \text{RULE}\}$.

Now, we can get the reliability of the pre-fetching algorithm:

$$\frac{\sum_{S} \text{Node}(t_k^r).weight \times \text{Rule}(\text{Node}(t_k^r), \text{Node}(t_j^r)).weight}{\sum_{S} \text{Node}(t_k^r).weight},$$

where $S = \{(t_k^r, t_j^r) \mid (t_k^r, f_k) \in K(Y_n), t_j^r \in \bigcup_{i \in L} C(l_i\cdot \text{string}) \land \text{Rule}(\text{Node}(t_k^r), \text{Node}(t_j^r)) \in \text{RULE}\}$.

4. Proposed Data Mining Technique

Data mining has been well known as a database technology developed in recent years [18, 4]. It can mine implicit, unknown and potentially useful knowledge and rules for prediction [20]. Using these rules, we can predict the user’s impending activities with the algorithms in Section 3.

The interest association rules directly represent the relations of reasoning between interests. But the data in the cache shown in the simple WWW data model directly represent the link relations in the web pages, which cannot directly reflect the association degree between interests. Therefore certain measures are needed to transform the data in the cache shown in the simple WWW data model. The associations, the sequence patterns and discovering the same time sequence can benefit from the association analysis using the data mining technology. On the basis that generality is not lost, we will not consider the transferable relations among the rules when we build the interest association rules. Hence it is suitable to apply the method of discovering relations by the model of the simple interest association rules.

Because mining information from abundant amount of historical data may take more time, it is not proper to predict user’s activities online. The solution is to use genetic algorithm to mine the user’s browsing interest association rules every certain time, based on the historical information in the cache (see Algorithm 1). Now we
Algorithm 3 (Mining interest association rules from the data in the cache) The pseudo code for the algorithm is as follows:

```plaintext
for each page \( Y_i \) in the set \( C \) do  /* for1*/
    for each link \( l_{i,r} \) in the set \( L(Y_i) \) do  /* for2*/
        assume \( l_{i,r}.target_{node} = Y_j; \)
        if \( Y_j \in C \) then  /* for3*/
            for each lemma \( (i_p, f_p) \) in the set \( K(Y_i) \) do  /* for4*/
                for each lemma \( (i_q, f_q) \) in the set \( K(Y_j) \) do  /* for4*/
                    Rule \( (Node(i_p), Node(i_q)).weight \leftarrow Rule(\Node(i_p), \Node(i_q)).weight + g(f_p, f_q, Y_i.time, Y_j.time) \)
                    \( (i_p, f_p) \in K(Y_i), (i_q, f_q) \in K(Y_j); \)
                end for  /* for4*/
            end for  /* for3*/
        else  /* for5*/
            for each lemma \( (i_p, f_p) \) in the set \( K(Y_i) \) do  /* for6*/
                for each lemma \( i_q \) in the set \( Q(l_{i,r}.string) \) do  /* for6*/
                    Rule \( (Node(i_p), Node(i_q)).weight \leftarrow Rule(\Node(i_p), \Node(i_q)).weight + h(f_p, Y_i.time) \)
                    \( (i_p, f_p) \in Y_i, i_q \in Q(l_{i,r}.string); \)
                end for  /* for6*/
            end for  /* for5*/
        end if  /* for2*/
    end for  /* for1*/
```

\[ \text{Rule}(\Node(t_i), \Node(t_j)).weight = \frac{\text{Rule}(\Node(t_i), \Node(t_j)).weight}{\sum_{Q(\Node(t_i))} \text{Rule}(\Node(t_i), \Node(t_j)).weight} \]

Where, \( Q(\Node(t_i)) = \{ t_j \mid t_j \in T, \text{Rule}(\Node(t_i), \Node(t_j)) \in \text{RULE} \} \) /* normalising the weight of interest association rules */.

In loop for4 of the above algorithm, the influence of the web pages in the cache, the links and linked web pages on the interest association rules in the interest association repository are used to define function \( g(f_p, f_q, \)
\( Y_i \cdot time, Y_j \cdot time \), where \( f_p \) and \( f_q \) are the appearance frequency of lemma \( i'_p \) in the page \( Y_i \) and lemma \( i'_q \) in the \( Y_j \), respectively, and \( Y_i \cdot time \) and \( Y_j \cdot time \) are the accessed time of pages \( Y_i \) and \( Y_j \), respectively. In loop \( for 6 \), the influence of the web pages in the cache and their links on the interest association are used to define function \( h ( f_p, Y_i \cdot time ) \), where \( i'_p \) is the appearance frequency of lemma \( i'_p \) in the page \( Y_i \) and \( Y_i \cdot time \) is the accessing time of page \( Y_i \).

The supporting rate is a very important guideline in data mining. The \( Rule (Node(t_i), Node(t_j)). weight \) from the Algorithm 3 is the supporting rate of the interest association rule, \( Rule (Node(t_i), Node(t_j)) \).

5. An Agent-Based Web Pre-fetching System

An agent [9, 8, 11, 15] based web pre-fetching system implementing the approach described in the previous sections is shown in Figure 2. In this system, the Interest Association Repository, the Mining Agent and the Decision Agent [14, 19] are attached to the Browser side. The Interest Association Repository uses SQL Server7.0 to save the pre-fetching rules. The Mining Agent and the Decision Agent are implemented by WIRPL [18]. A mining agent runs at the client side and updates the interest association rules periodically (the refresh rate can be set by users). According to Algorithm 3, the data in the local cache represented by the simple WWW data model above are transformed into the interest association rules which are then stored in the Interest Association Repository. The Decision Agent running at the client side can detect the user’s activities in real time and can provide the abilities of predicting the user’s activities online. It should take charge of the tasks such as getting the current web page in the browser, analysing the current web page using Algorithm 2, interacting with the interest association repository, pre-fetching the web pages and saving them into the local cache. The interest association repository, the mining agent and the decision agent are transparent to users. Therefore the users can use browsers as usual.
6. Experiment

We conducted a series of experiments while we were developing the proposed approach. In this paper, we are presenting a systematic experiment to demonstrate and prove our proposed approach. In this experiment, the user interest model is based on 326 web pages, which are selected from the cache of one user’s computer. Using the proposed user’s interest model, the agent-based web pre-fetching system can trace and predict the user’s actions. The data shown in the table 1 is recorded when the user was visiting 10 web sites randomly.

In the table, the link number of visited pages (vis_num) is the average link number of the user’s visited pages in the web site. The number of the clicked links (click_num) is the average number of the clicked links of the user’s visited pages in the web site. The number of the pre-fetched pages (pre_num) is the average number of the pre-fetched web pages for each of the user’s visited web page. The number of the selected web pages (sel_num) is the number of web pages, which are selected from the pre-fetched web pages when the user browses them. The hit ratio is \( \frac{pre_{num}}{click_{num}} \times 100\% \). The availability of the pre-fetched web pages is \( \frac{sel_{num}}{vis_{num}} \times 100\% \). The pre-fetching ratio is \( \frac{pre_{num}}{vis_{num}} \times 100\% \). The less pre-fetching ratio and the more hit ratio and availability of the pre-fetched web pages, the better is the optimisation of the system. The experimental data shown in the table 1 are closely associated with the web sites selected by the user, the user’s current interest and the user’s visiting activities. The benefit of tracing the user’s current interest is provided in the agent-based web pre-fetching system, which can commendably adapt the user’s current interest. The whole evaluations in this experiment show that the average hit ratio is 50.0% and that the average pre-fetching ratio is 23.4%. We can conclude that the agent-based web pre-fetching system optimises the course of the user’s visiting web pages.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Decision Agent</th>
<th>Local Cache</th>
<th>Mining Agent</th>
<th>WWW Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest Association Repository</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: An agent based web pre-fetching system

Table 1. Analysis of the experimental data
7. Conclusion

This paper first presents a simple WWW data model, and then introduces relevant techniques with which the data in the cache of the browser can be mined. Based the proposed data model, a web page pre-fetching approach is developed, i.e., the knowledge through analysing available data is saved into the interest association repository and the users’ impending activities are to be predicted. Integrated with data mining technique, agent technique and web technique, the proposed approach provides better quality for web service users. The design ideas of the agent based web pre-fetching reflect the development tendency of intelligent web browsers. Experiments carried out during our study strongly back up the proposed approach. Work is being carried out to integrate the approach described in the paper to a commercial product.

8. References


