Web Page Prediction Model Based on Clustering Technique

Rajni Pamnani, Pramila Chawan

Department of computer technology,
VJTI University, Mumbai
rajni_as@yahoo.com, pmchawan@vjti.org.in

Abstract -- Users are very often overwhelmed by the huge amount of information and are faced with a big challenge to find the most relevant information in the right time. Prediction systems aim at pruning this information space and directing users toward the items that best meet their needs and interests. Predicting the next request of a user as she visits Web pages has gained importance as Web-based activity increases. Nowadays, Prediction systems are definitely a necessity in the websites not just an auxiliary feature, especially for commercial websites and web sites with large information services. Previously proposed methods for recommendation use data collected over time in order to extract usage patterns. However, these patterns may change over time, because each day new log entries are added to the database and old entries are deleted. Thus, over time it is highly desirable to perform the update of the recommendation model incrementally. In this paper, we propose a new model for modeling and predicting web user sessions which attempt to reduce the online recommendation time while retaining predictive accuracy. Since it is very easy to modify the model, it is updated during the recommendation process.

1 INTRODUCTION

Recommender systems have been used in various applications ranging from predicting the products a customer is likely to buy, movies, music or news that might interest the user and web pages that the user is likely to seek, which is also the focus of this paper. The amount of information available on-line is increasing rapidly with the explosive growth of the World Wide Web and the advent of e-Commerce. Although this surely provides users with more options, at the same time makes it more difficult to find the “right” or “interesting” information from this great pool of information, the problem commonly known as information overload. To address these problems, recommender systems have been introduced. They can be defined as the personalized information technology used to predict a user evaluation of a particular item or more generally as any system that guides users toward interesting or useful objects in a large space of possible options.

Web page recommendation is considered a user modeling or web personalization task. One research area that has recently contributed greatly to this problem is web mining. Most of the systems developed in this field are based on web usage mining which is the process of applying data mining techniques to the discovery of usage patterns form web data. These systems are mainly concerned with analyzing web usage logs, discovering patterns from this data and making recommendations based on the extracted knowledge. One important characteristic of these systems is that unlike traditional recommender systems, which mainly base their decisions on user ratings on different items or other explicit feedbacks provided by the user these techniques discover user preferences from their implicit feedbacks, namely the web pages they have visited. More recently, systems that take advantage of a combination of content, usage and even structure information of the web have been introduced and shown superior results in the web page recommendation problem.

2 MOTIVATION AND RELATED WORK

Today, millions of visitors interact daily with Web sites around the world and massive amounts of data about these interactions are generated. We believe that this information could be very precious in order to understand the user’s behavior. Web Usage Mining is achieved first by reporting visitors traffic information based on Web server log files. For example, if various users repeatedly access the same series of pages, a corresponding series of log entries will appear in the Web log file, and this series can be considered as a Web access pattern.

In the recent years, there has been an increasing number of research works done in Web Usage Mining and their developments. The main motivation of these studies is to get a better understanding of the reactions and motivations of users navigation. Some studies also apply the mining results to improve the design of Web sites, analyze system performances and network communications or even build adaptive Web sites. We can distinguish three Web Mining approaches that exploit Web logs: association rules (AR), frequent sequences and frequent generalized sequences. Algorithms for the three approaches were developed but few experiments have been done with real Web log data. In this paper, we compare results provided from the three approaches using the same Web log data.

3 CONSIDERATIONS AND DEFINITIONS

With the aim to study Web usage mining, we present in this section our definition of user and navigation.

3.1 User and navigation

A user is identified as a real person or an automated software, such as a Web Crawler (i.e. a spider), accessing files from different servers over WWW. The simplest way to identify users is to consider that one IP address corresponds to one distinct user. A click-stream is defined as a time-ordered list of page views. User's click-stream over the entire Web is called the user session. Whereas, the server session is defined to be the subset of clicks over a particular server by a user, which is also known as a visit. Catledge has studied user page view time over WWW and recommended 25.5 minutes for maximal session length [6]. An episode is defined as a set of sequentially or semantically related clicks.

3.2 Web log files

The easiest way to find information about the users navigation is to explore the Web server logs. The server access log records all requests processed by the server. Server log L is a list of log entries each containing timestamp, host identifier, URL request (including URL stem and query), referrer, agent, etc. Every log entry conforming to the Common Log Format (CLF) contains some of these fields: client IP address or hostname, access time, HTTP request method used, path of the accessed resource on the Web server (identifying the URL), protocol used (HTTP/1.0, HTTP/1.1), status code, number of bytes transmitted, referrer, user-agent, etc. The referrer field gives the URL from which the user has navigated to the requested page. The user agent is the software used to access pages. It can be a spider (ex.: GoogleBot, openbot, scooter, etc.) or a browser (Mozilla, Internet Explorer, Opera, etc.).

4 WEB PAGE PREDICTION MODEL

Following the data cleaning and preprocessing steps, we use similarity metric in the second step for calculating the similarities between all pairs of session. In the third step, the sessions are clustered based on this similarity metric using the graph partitioning algorithm and each cluster is represented by a User Access Stream Tree Model. In order to produce the recommendation set, we first select the cluster and then select the path from the UAST of the cluster that best matches the active user session.

4.1 User Access Stream Tree Model

This model uses both sequence of visiting pages and time spent on each page. In this Model, the sessions are clustered based on this similarity metric using the graph partitioning algorithm and each cluster is represented by a User Access Tree Model. The model we propose has two characteristics:

1. Preservation of whole path of a user session
2. Preservation of time information of visited pages

We generate a click-stream-tree for each cluster. Each user access stream tree has a root node, which is labeled as \null". Each node except the root node of the user access tree consists of three fields: data, count and next node. Data field consists of page number and the normalized time information of that page. Count field registers the number of sessions represented by the portion of the path arriving to that node. Next node links to the next node in the user access stream tree that has the same data field or null if there is any node with the same data field. Each user access stream tree has a data table, which consists of two fields: data field and first node that links to the first node in the user access stream tree that has the data field. The tree for each cluster is constructed by applying the algorithm given in Figure 1.

The children of each node in the click-stream tree is ordered in the count-descending order such that a child node with bigger count is closer to its parent node. The resulting user access stream trees are then used for recommendation.

```
Create a root node of a click-stream tree, and label it as “null"
index ← 0
while index ≤ number of Sessions in the cluster do
Active Session ← tindex
m ←0
Current Node ← root node of the click-stream tree
while m ≤ Active Session length do
```

```
8: Active Data ← $\{p^{m}_{t_{index}}\}$-$\{T^{m}_{p_{t_{index}}}\}$
9: if there is a Child of Current_Node with the same data field then
    10: Child.count + +
    11: Current_Node ← Child
12: else
    13: create a child node of the Current_Node
    14: Child.data = Active_Data
    15: Child.count = 1
    16: Current_Node ← Child
17: end if
18: m + +
19: end while
20: index + +
21: end while
```

Figure 1: User Access Stream Tree Algorithm

4.2 Prediction System

The Prediction System is the real time component of the model that selects the best path for predicting the next request of the active user session. The task of the recommendation engine is to compute a *recommendation set* for the current (active) user session, consisting of the objects.

In general there are several design factors that can be taken into account in determining the recommendation set. These factors may include:

• A short-term history depth for the current user representing the portion of the user's activity history that should be considered relevant for the purpose of making recommendations;

• The mechanism used for matching aggregate usage profiles and the active session; and

• A measure of significance for each recommendation (in addition to its prediction value), which may be based on prior domain knowledge or structural characteristics of the site.

Maintaining a history depth is important because most users navigate several paths leading to independent pieces of information within a session. In many cases these *episodes* have a length of no more than two or three references. In such a situation, it may not be appropriate to use references a user made in a previous episode to make recommendations during the current episode. It is possible to capture the user history depth within a sliding window over the current session.

The recommendation engine must be an online process, providing results quickly enough to avoid any perceived delay by the users (beyond what is considered normal for a given Web site and connection speed).There is a trade-off between the prediction accuracy of the next request and the time spent for recommendation. The speed of the recommendation engine is of great importance in on-line recommendation systems. Thus, we propose the clustering of user sessions in order to reduce the search space and represent each cluster by a click-stream tree. Given the time of the last visited page of the active user session, the model recommends three pages. The most recent visited page of the active user session contains the most important information. The click-stream tree enables us to insert the entire session of a user without any information loss. We not only store the frequent patterns in the tree but also the whole path that a user follows during her session. Besides this, the tree has a compact structure. If a path occurs more than once, only the count of its nodes is incremented.

Based on the construction of the click-stream tree, a path (p1, p2… pk), (Tp1, Tp2… Tpk) occurs in the tree dk.count times, where dk is the data field formed by merging the page request $p^{k}_{t_i}$ and corresponding normalized time value $T_{p^{k}_{t_i}}$ of the path.

4.3 Optimal Path Algorithm

Figure 2 presents the algorithm for finding the path that best matches the active user sessions. For the first two pages of the active user session all clusters are searched to select the best path (line 3). After the second request of the active user top-N clusters that have higher recommendation scores among other clusters are selected (line 29-31) for producing further recommendation sets (line 5). To select the best path we use a backward stepping algorithm. The last visited page and normalized time of that page of the active user session are merged together to build the data field (line 10). We find from the data table of the click-stream tree of a cluster the first node that has the same data field (line 11). We start with that node and go back until the root node (or until the active user session has no more pages to compare) to calculate the similarity of that path to the active user session (line 16-19). We calculate the similarity of the optimal alignment.

To obtain the recommendation score of a path, the similarity is multiplied by the relative frequency of that path, which we define as the count of the path divided by the total number of paths (S[cl]) in the tree (line 20). Starting from the first node of the data field and following the next node, the recommendation score is calculated for the paths that contain the data field in the cluster (line 26). The path that has the highest recommendation score is selected as the best path for generating the recommendation set for that cluster (line 21-24). The first three children nodes of the last node of the best path are used for producing the recommendation set. The pages of these child nodes are recommended to the active user.

Optimal Path Algorithm

```
1: $t_a$ ← Active User Session
2: if $t_a$ .length ≤ 2 then
3:     Clusters = All Clusters
4: else
5:     Clusters = Top-N Clusters
6: end if
7: for i = 0 to NumberOfClusters do
8:     cl = Clusters[i]
9:     Sim[cl] = 0
10:    $d_a$ ← $\{p^m_{t_a}\}-\{T_{p^m_{t_a}}\}$
11:    Node ← data table[cl]($d_a$ ).first_node
12:    path = null
13:    while Node ≠ null do
14:        path = {path} + {Node.data}
15:        Parent Node ← Node.Parent
16:        while Parent Node ≠ null do
17:            path = {path} + {Parent Node.Data}
18:            Parent_Node ← Parent_Node.Parent
19:        end while
20:        Sim(path) = sim($t_a$ , path)* Node.count / S[cl]
21:        if Sim(path) > Sim[cl] then
22:            Sim[cl] ← Sim(path)
23:            BestPath[cl] ←path
24:        end if
25:        path = null
26:        Node ← Node.next _node
27:    end while
28: end for
29: if $t_a$ .length = 2 then
30:     Top-N Clusters ← N Clusters with highest Sim[cl] values
31: end if
```

5 CONCLUSION

One of the goals of Web Usage Mining is to guide the web users to discover useful knowledge and to support them for decision making. In that context, predicting the needs of a web user as he visits web sites has gained importance. The requirement for predicting user needs in order to guide the user in a web site and improve the usability of web site can be addressed by recommending pages to the user that are related to the interest of the user at that time. This paper proposed a model for discovering and modeling of the user's interest in a single session. This model uses both sequence of visiting pages and time spent on each page. In this Model, the sessions are clustered based on this similarity metric using the graph partitioning algorithm and each cluster is represented by a User Access Tree Model.

REFERENCES

[1] Renata Ivancsy, István Vajk "Frequent Pattern Mining in Web Log Data",

[2] Murat Ali Bayir , Ismail Hakki Toroslu "Smart Miner: A New Framework for Mining Large Scale Web Usage Data",

[3] Mathias G´ ery, Hatem Haddad, "Evaluation of Web Usage Mining Approaches for User's Next Request Prediction",

[4] Wen-Chen Hu, Xuli Zong, "World Wide Web Usage Mining Systems and Technologies",

[5] Sule Gunduz, M. Tamer Ozsu, "A Web Page Prediction Model Based on ClickStream Tree Representation of User Behavior"

[6] Sule Gunduz, M. Tamer Ozsu , "Incremental Click-Stream Tree Model: Learning From New Users for Web Page Prediction",

[7] Natheer Khasawneh, Chien-Chung Chan, " Active User-Based and Ontology-Based Web Log Data Preprocessing for Web Usage Mining"