Extracting User's Interests from Web Log Data for Implementing Adaptive Education System

Nu War Hsan, Sabai Phyu
University of Computer Studies, Yangon
nuwarhsan87@gmail.com,sabaiphyu72@gmail.com

Abstract

As World Wide Web is a repository of web pages and links, it provides not only useful information for the Internet users but also becomes delivery platform for searching and surfing day by day. Web personalization is the process of customizing a Web site to the needs of each specific user or set of users, taking advantage of the knowledge required through the analysis of the user's navigation behavior. Integration usage data with user profile data enhances the personalization process. In this paper, the adaptive educational system is developed to extract user's interests from web log data and implemented the recommender system to suggest the next links for studying next. The SPADE (Sequential Pattern Discovery using Equivalence classes) is used in finding semantic association rules to overcome the burden of repeated database scans while calculating the support of the candidates and Dynamic LCS(Longest Common Subsequence) is applied in mapping with users' current session and association rules which are generated from the SPADE algorithm. In the proposed system, the teacher and the content developer are performed their tasks to become the most accurate information for the best recommendations by using domain ontology. The main objective of this proposed system is to analyze the student's behavioral patterns to recommend the new links that best match the individual user's preferences.

Keywords: personalization, semantic web, domain ontology, web usage mining, SPADE, LCS

1. Introduction

Web usage mining is concerned with finding user navigational patterns on the World Wide Web by extracting knowledge from web logs. Modeling and analyzing Web navigation behavior is helpful in understanding the type of information online user demand. Although providing people with access to more information may not be the problem, tracing to people's navigation and find the suitable information they want is the basic fact for personalization. Personalization is the solution of these difficulties because of its objectives which is to provide users with information they want without having to search for its explicitness.

Although applying web usage mining in personalization has developed increasingly, there are still many disadvantages such as no semantic meaning of web navigation profile and new-item problem to recommend a newly added page or link to the visitors which is not in the current navigation profiles.

In this paper, a framework which is integrated with semantic information in web usage mining process is proposed to implement. Most of the conventional Web usage based recommender systems are limited in their ability to use the domain knowledge of the Web application and their focus is only on Web usage data. Recent studies have suggested that domain knowledge of the Web application in the form of ontology which can play an important role in providing smarter and more comprehensive recommender systems are needed to develop. The aim of this proposed system is to generate frequent sequential patterns with semantic information and to evaluate the quality of the generated recommendation.

2. Related Work

Mobasher et al [1] presented Web Personalizer, a system which provides dynamic recommendations as a list of hypertext links to users. The analysis is based on anonymous usage data combined with the structure formed by the hyperlinks of the site. Data mining techniques are used in the preprocessing phase in order to obtain aggregate usage profiles. Web

Personalizer is a good example of two-tier architecture for Personalization systems.

Liu and Keselj [2] proposed the automatic classification of web user navigation patterns and proposed a novel approach to classify user navigation patterns and predicting users' future requests. They used character N-grams to represent the contents of web pages, and combined them with user navigation patterns by building user navigation profiles composed of a collection of Ngrams.

Axita and Sonal [3] proposed the model which uses multiple agents which delivers personalized SERP (Search Engine Results Page) and is more suited for personalization of web pages based on learner's query expanded to manifold queries with novel concept of keyword search and discover knowledge using browser's behavior. Dejung, Paea, Shiwoong and Peishen [4] introduced their system namely OntoGrate to address the critical and challenging problem of supporting human experts in multiple domains to interactively integrate information that is heterogeneous in both structure and semantics.

Losarwar ad Joshi [5] described the importance of data processing methods and various steps involved in getting the required content effectively. Manoj and Manasi discussed personalization process and its various modules. They also discuss the recommender system which makes use of Web personalization for providing tailored recommendations to the user.

3. Theory Background

The work described in this paper is an intersection of several research areas: personalization, recommendation, semantic web mining and adaptive web based educational systems.

3.1. Personalization and Web Usage Mining

The aim of personalization based on Web Usage Mining is to recommend a set of objects to the current user as determined by matching usage patterns. This task is performed by matching the active users' actions with the usage patterns discovered through web usage mining. The recommendation engine is performed this process which is the online component of the personalization system.

The process of extracting web usage logs to implement adaptive educational system performs three main steps:

- Data collection and transformation
- Association Rule Extraction
- Recommendation

The first two steps are performed off-line and the last is on-line. The data collection and transformation steps transform web log files into transaction data which can be processed by data mining tasks. The unnecessary entries from web server logs are pruned and the navigation histories of each visitor of the web site are extracted. In second step, association rules are extracted in terms of the call's individuals for each call of the ontology. In recommendation phase, the association rules are joined with the user's navigation paths and new pages are recommended to the user.

3.1.1 Building Ontology

To include semantic information on a Web page, an ontology which defines the classes of the domain space and their properties showing the relationships among them is needed to be developed. The construction of basic ontology is built from the domain websites based on structure and content. In this proposed system, the semantic information Web ontology is written by OWL (Web Ontology Language) using the education domain.

3.1.2. Data collection and transformation

In this step, there are three main processes to perform such as pruning web server logs, extractions of transactions and mapping with ontology classes and web page addresses. Normally, web server registers the entire request made to the server in the log file including request time, request type (GET, POST and HEAD), http version, user agent information, client IP address, response status and referrer address.

Firstly, the non-responded Web requests are pruned from the status field of request log entry. The second step is to eliminate the requests made by software agents which sometimes automatically request web content from a web site. The third step is to remove the irrelevant requests from the log file such as am image request or style sheet request which are not taken into account since these files are auxiliary files for displaying web site to the user.

To sum up, the status code is used to prune non-responded requests, address fields are used to prune Web page addresses and user agent fields are used to prune the web crawlers. The next step after pruning is extracting navigation history of each session from log files. The navigation history is the set of Web objects requested by the user in his active session time. A session is established when the end-user makes the first request to the Web server and the session is torn down after a period of idle time from end-user. The allowed time is called the session timeout. In this system, thirty minutes will be assigned as session timeout. The classification of sessions is assigned according to the algorithm described in Figure 1.

The next step of pre-processing is to map between ontology individuals and the requested Web address in the Web server log. The Web server does not registers semantic information about the request in the log file, only the address of the request. Therefore, before starting the frequent sequence finding, mapping between ontology and the Web site address is carried out. To include semantic information on a Web page, an ontology which defines the classes of the domain space and their properties showing the relationships among them needs to be developed. The address and ontology individuals mapping is stored in an ontology file called the mapping file. In the mapping file, the ontology instances are mapped to the Web address. The concept ontology file, instance ontology files and mapping ontology files are created by Protege. For each class of the ontology, all sessions and their visited pages are extracted. For each visited address, its corresponding instances are extracted from mapping ontology files.

```
Procedure Extract Session from a given Web server log file where T=transaction
T=0;
Session=0;
For all log entry β ∈ L
    Do for each Ti ∈ T   //each transaction among the log entry
        If (Ti.ip= β.ip and Ti.useragent= β.useragent and Ti.lasttime= β.lasttime+session_timeout)
        then
                Session=Ti;
        Else
                {
                Session= new Session;
                T=T U Session;
                }
    End do
End for
Session=Session U β;
Session.lasttime= β.time;
```

Figure 1: Extract sessions form a given Web server log file

For each concept, a different session-visited instance transaction is constructed.

3.1.3 Association Rule Extraction Step

After preprocessing, the next step is the extraction of frequent navigation patterns. In this step, frequent sequences are extracted from the transactions. In SPADE algorithm, frequent 1-sequences, 2-sequences, and 3-sequences and longer are found.

Finding frequent 1-sequences is a relatively a straightforward step. For each item its ID-list (i.e. Session-ID, Event-ID) pairs are read from the vertical dataset where each sequence contains a list of event along with a timestamp and each event has a set of items. Then the number of distinct Session-ID is calculated and if this count is greater than the minimum support count, this item is added as frequent 1-sequence. Before performing to find frequent 2 sequences, the method of converting the vertical dataset to a horizontal one is preprocessed. This step is performed according to Figure 2.

Frequent 2-sequences are by joining all the frequent 1-sequence with themselves and counting the cardinality of the result. For example, if A and B are frequent 1-sequences, then the sequence {A→B}, {B→A} and {AB} are the candidate frequent 2-sequences. The cardinality is calculated by the temporal join operation.

```
Procedure Convert to Horizontal where T= Frequent 1- sequence, H= Horizontal dataset,
sid=Session ID, eid=Event ID

Begin;
  T = Φ;  H = Φ;
        For each item I ∈ V
                Do for each (distinct sid in <sid, eid> ∈ ID-list(I))
                        If (H contains Session_sid )
                                then  Session_sid.put(I);
                        else H= H U Session_sid;
                                Session_sid.put(I);
                        End if;
                End Do;
        End For;
End;
```

Figure 2: Convert to Horizontal Algorithm

The next step is finding 3-sequences and over by using the frequent 1-sequences and 2-sequences. The procedure is performed according to Figure 3.

Frequent sequence generation form equivalence class where F=Frequent 3-sequences, σ(R)=support count of candidate frequent sequence, R= candidate frequent sequence, min_sup=minimum support count that system defined, A_i=Frequent 1-sequences, A_j=Frequent 2-Sequences

```
For all atoms Ai ∈ S
    Ti=Φ;
    For all atoms Aj ∈ S
        L(R) =L(Ai)∩ L(Aj)
        If (σ(R)≥min_sup)
        Then
        {
            Ti=Ti U R;
            F=F U R;
        }
        End if
    End for
End for
```

Figure 3: Finding Frequent 3-Sequences

After generating frequent sequences, some of them can be uninteresting whose support count can be known from other sequences. Generally a new frequent sequence is added to the list of frequent sequences for all combinations of items. For example sequences {A→ABCDE→AB}, {A→ABCD→AB}, {A→ABC→AB} are the same since all sequences have a sub-event of the event {ABCDE} and they all have the same support count. Therefore, all subsequences except the biggest are pruned. And then A→ABF→AB and A→ABCDEF→AB are not pruned which has no same event with another.

Sequence	Support Count	Pruned
A→ABC→AB	20	yes
A→ABCD→AB	20	yes
A→ABCDE→AB	20	no
A→ABF→AB	20	no
A→ABCDEF→AB	30	no

Table 1: Sample Frequent Sequences

There are two rules to form an association rule from two frequent sequences. Let (f1, f2) be frequent sequences to be joined. The first rule is that the event count of sequence f2 should be greater than the event count of the sequence f1. The second rule is that sequence f2 start with sequence f1. If these two conditions are met, then these two sequences make a rule. After that, a minimum confidence is predefined and if the confidence of the rule is greater than the minimum confidence, then this rule is added to the association rules.

3.1.4 Recommendation

The main objective of recommendation is to generate recommendation by using certain filtering parameters like subjects, marks and interestingness and so on. To generate list of recommendation, the dynamic Longest Common Subsequence is used to recommend the next links for student to study by using the current student's session and association rules which are generated from the SPADE algorithm using historical student logs.

3.1.4.1 Dynamic Longest Common Subsequence Algorithm

A prefix of a sequence is an initial string of vales, $X_i = \{x_1, x_2 \ldots xi\}$. The idea of this algorithm is to compute the longest common subsequence for every possible pair of prefixes. Let c [i, j] is the length of the longest common subsequence of X_i and Y_j. The idea is to compute c [i,j] assuming that the values of c[i',j'] for i'≤i and j'≤ j (but not both equal). There are rules to obtain the longest common subsequence for recommendation.

Basic: c[i,0] =c[j,0]=0. If either sequence is empty then the longest common subsequence is empty.

Last characters match: Suppose that $x_i=y_j$ then c [i, j]=c[i-1,j-1]+1.

Last characters do not match: Suppose that $x_i \neq y_j$ then c[i,j]=max[c[i-1,j],c[i,j-1]].

So, the LCS algorithm will perform according to the following rule:

$$c[i,j] = f(x) = \begin{cases} 0, & if\ i = 0\ or\ j = 0 \\ c[i-1, j-1] + 1, & if\ i, j > 0\ and\ x_i = y_j \\ \max(c[i, j-1], c[i-1, j]) & if\ i, j > 0\ and\ x_i \neq y_j \end{cases}$$

4. Proposed Architecture

In this system, three different models are used: student, teacher and content developer. In the student model, the two kinds of data: static data which are not altered during the student-system interaction and dynamic data which changes according to the learning progress. In static data, the personal data about the students are recorded such as student name, student's personal activities, list of degrees and qualifications, students' interests, concentration skills, collaborative skills, relational skills, course evaluation and course navigation control. The dynamic data comprises two sets of data: the performance data and the student knowledge data. The performance data gathers

information about the student's current performance in the course sessions.

The student knowledge data describes the knowledge concepts and competences relevant for the current course that the student possesses and must possess until the end of course. This set of data also gathers information about the student's progress during the course sessions. The teacher and content developer are performed according to the results of student's performance in modifying the teaching methods and system design to get exact information for the best recommendation by using the student navigation paths using domain ontology.

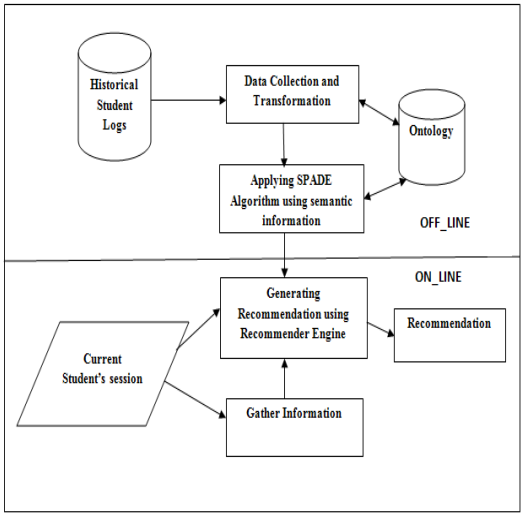

Figure 4: Proposed System Design

5. Comparison Results

In this paper, the proposed system use SPADE algorithm to generate the semantic association rules because of the following strengths compared with other algorithm. Comparative analysis of GSP (Generalized Sequential Patterns) and SPADE (Sequential Pattern Discovery using Equivalence classes) is discovered in the field of data mining. When the same datasets are used in both algorithms, SPADE is more stable and better than GSP at lower values of support threshold. The main aim of SPADE algorithm is to overcome the burden of repeated database scans while calculating the support of the candidates. SPADE not only minimizes computational costs by using efficient search schemes. The following result is described as the comparison result of general concepts between these two algorithms.

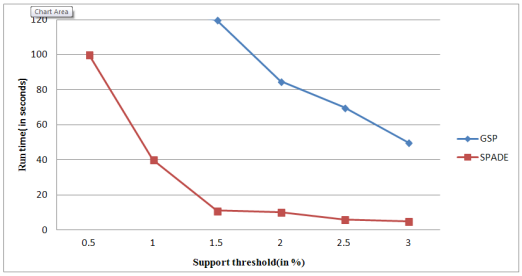

Figure 5: Performance of GSP and SPADE Algorithms

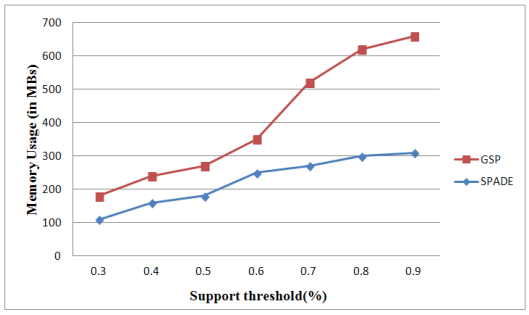

Figure 6: Memory Usage of GSP and SPADE Algorithms

6. Conclusion

In this system, the architecture recommender system utilizes Web Usage Mining to recommend the links to develop the students' results using domain ontology. This system overcomes the disadvantages of classical web usage mining such that results are in the form of web pages with no semantic meaning of common navigation profile. By introducing the semantic information, web usage mining algorithms are performed in terms of ontology individuals instead of web page addresses. In the future, there may be further experiments using other sequence mining algorithms and can use intelligent agents to do online Web mining automatically.

References

[1] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on web usage mining," *Communications of the ACM*, ACM, 2000, pp. 142–151

[2] R. Liu and V. Keselj, "Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests," *Data & Knowledge Engineering*, Elsevier, 2007, pp. 304-330.

[3] Axita Shah and Sonal Jain "An Agent based Personalized Intelligent e-learning". International Journal of Computer Application.2011.
[4] Dejing Dou, Paea LePendu, Shiwoong Kim and Peishen Qi, "Integrating Databases into Semantic Web through an Ontology-based Framework", Proceeding of the 22nd International Conference on Data Engineering Workshops,2006.
[5] Vijayashri Losarwar and Dr. Madhuri Joshi. "Data PreProcessing in Web Usage Mining" International Conference on AI and Embedded Systems,2012.
[6]Yun Xu and Jianbin Chen, "Research and Design of Web Data Mining in Personalized E-business"
[7]Mala Bharti Lodhi, Vineet Richariya and Vivek Richariya. "Design and Developing of Efficient Algorithm in Web Usage Mining for Web Personalization, International Journal Computer Science and Information Technology,2012
[8] Manoj Swamiand Prof.Manasi Kulkarni, "Understanding Web Personalization with Web Usage Mining and its Application: Recommender System", International Journal of Emerging Technology and Advanced Engineering, 2013.
[9] OWL Web Ontology Language, "http://www.w3.org/TR/owl-ref.