Using Domain Ontology for Semantic Web Usage Mining and Next Page Prediction

Nizar R. Mabroukeh and Christie I. Ezeife*
School of Computer Science
University of Windsor
401 Sunset Ave.
Windsor, Ontario N9B 3P4
mabrouk@uwindsor.ca

ABSTRACT

This paper proposes the integration of semantic information drawn from a web application's domain knowledge into all phases of the web usage mining process (preprocessing, pattern discovery, and recommendation/prediction). The goal is to have an intelligent semantics-aware web usage mining framework. This is accomplished by using semantic information in the sequential pattern mining algorithm to prune the search space and partially relieve the algorithm from support counting. In addition, semantic information is used in the prediction phase with low order Markov models, for less space complexity and accurate prediction, that will help solve ambiguous predictions problem.

Experimental results show that semantics-aware sequential pattern mining algorithms can perform 4 times faster than regular non-semantics-aware algorithms with only 26% of the memory requirement.

Categories and Subject Descriptors: H.2.8 [Database Management]: data mining; H.4.2 [Information Systems Applications]: decision support; J.1 [Administrative Data Processing]: Marketing

General Terms: Algorithms

Keywords: Association Rules, Domain Ontology, Markov Model, Semantic Relatedness, Semantic Web, Sequential Pattern Mining, Web Usage Mining.

1. INTRODUCTION

Web usage mining is concerned with finding user navigational patterns on the world wide web by extracting knowledge from web logs. Finding frequent user's web access sequences is done by applying sequential pattern mining techniques on the web log [1]. Its best characteristic is that it fits the problem of mining the web log directly. On the other hand, current sequential pattern mining techniques suffer from a number of drawbacks [4], some of which include: (1) Support counting has to be maintained at all times during mining, which adds to the memory size required, (2) the sequence data base is scanned on nearly every pass of the algorithm or a large data structure has to be maintained in memory all the time, and, (3) most importantly they do not incorporate semantic information into the mining process and do not provide a way for predicting future user access patterns or, at least, user's next page request, as a direct result of mining. Predicting user's next page request usually takes place as an addistional phase after mining the web log.

2. CONTRIBUTIONS AND OUTLINE

This paper proposes to integrate semantic information, in the form of domain ontology from an e-Commerce application (e.g., *eMart* online catalogue) into the pattern discovery and prediction phases of web usage mining, for intelligent and better performing web usage mining.

This paper contributes to research as follows:

1. It provides a complete generic framework (called *SemAware*) that utilizes an underlying domain ontology available at web applications (e.g. Amazon.com[1], eBay[2]), on which any sequential pattern mining algorithm can fit. The feasibility of this integration is characterizied by the fact that the domain ontology is separated from the mining process.

2. It proposes to incorporate semantic information in the heart of the mining algorithm. Such integration allows more pruning of the search space in sequential pattern mining of the web log.

3. It introduces a novel method for enriching the Markov transition probability matrix with semantic information to solve the problem of tradeoff between acuracy and complexity in Markov models [6][7] used for prediction, as well as the problem of ambiguous predictions.

Section 3 surveys related work. The integration of semantic information into the second phase of web usage mining is

*This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an operating grant (OGP-0194134) and a University of Windsor grant.

CIKM'09, November 2–6, 2009, Hong Kong, China.

[1]http://www.wsmo.org/TR/d3/d3.4/v0.2/#ontology
[2]www.ebay.com

described in Section 4. In Section 5, semantic-aware next page request prediction is introduced, then a combination of both systems into one framework in *SemAware* is provided in Section 6. Section 7 describes experimental results. Finally, future work and conclusions are given in Section 8.

3. RELATED WORK

Pirolli and Pitkow's research in [5], in addition to Sarukkai in [7], lead to the use of higher order Markov models for link prediction. The order of a Markov model corresponds to the number of prior events used in predicting a future event. So, a k^{th}-order Markov model predicts the probability of the next event by looking at the past k events.

Using Markov models for prediction suffers from a number of drawbacks. As the order of the Markov model increases, so does the number of states and the model complexity. On the other hand, reducing the number of states leads to inaccurate transition probability matrix and lower coverage, thus less predictive power. As a solution to this tradeoff problem, the All-Kth-Order Markov model [6] was proposed, such that if the kth-order Markov model cannot make the prediction then the (k-1)th-order Markov model is tried and so on. The problem with this model is the large number of states. Selective Markov models SMM [2], that only store some of the states within the model, have been proposed as a better solution to the mentioned tradeoff problem. This proposed solution may not be feasible when it comes to very large data sets. In order to overcome the problems associated with All-Kth-order and SMM, Khalil et al. [3] combine lower order all-Kth Markov models with association rules to give more predictive power for a Markov model while at the same time retain small space complexity. In case prediction is ambiguous (i.e., two or more predictive pages having the same conditional probability), then association rules are constructed and consulted to resolve the ambiguity. In our proposed model semantic information is associated with the Markov model, during its creation to provide informed prediction without unjustified contradictions. To our knowledge semantic distance, or domain knowledge in general, has never been used to prune states in a Markov model or prune the search space in sequential pattern mining algorithms.

4. SEMANTICS-AWARE SEQUENTIAL PATTERN MINING

SemAware integrates semantic information into sequential pattern mining, this information is used during the pruning process to reduce the search space and minimize the number of candidate frequent sequences, minimizing as well the number of database scans and support counting processes.

Assume an e-Commerce application web site similar to Amazon.com, as an example web site to mine its server-side web log, call it *eMart*. Assume also that domain knowledge is available in the form of domain ontology provided by the ontology engineer during the design of the web site. A *core ontology* with *axioms* is defined by Stumme et al. [9] as a structure $\mathcal{O} := (\mathcal{C}, \leq_{\mathcal{C}}, \mathcal{R}, \sigma, \leq_{\mathcal{R}}, \mathcal{A})$ consisting of:

- two disjoint sets $\mathcal{C}$ and $\mathcal{R}$ whose elements are called *concept identifiers* and *relation identifiers*, respectively,
- a partial order $\leq_{\mathcal{C}}$ on $\mathcal{C}$, called concept hierarchy or taxonomy,
- a function $\sigma : \mathcal{R} \rightarrow \mathcal{C}^+$ called *signature* (where $\mathcal{C}^+$ is the set of all finite tuples of elements in $\mathcal{C}$),
- a partial order $\leq_{\mathcal{R}}$ on $\mathcal{R}$, called *relation hierarchy*, and
- a set $\mathcal{A}$ of logical axioms in some logical language $\mathcal{L}$.

Objects representing products in *eMart*, and dealt with in the mining process, are instances of concepts (also called *classes*) represented formally in the underlying domain ontology using a standard ontology framework, and an ontology representation language like OWL[3]. Each web page in *eMart* is annotated with semantic information, during the development of the website, thus showing what ontology class it is an instance of.

Definition 1. A *semantic object* o_i is represented as a tuple $<pg, ins_i>$, where pg represents the web page which contains the object/product, usually an URL address of the page, and ins_i is an instance of a class c $\in \mathcal{C}$, from the provided ontology $\mathcal{O}$, that represents the product being referenced, where i is an index for an enumeration of the objects in the sequence, from the web access sequence database being mined. □

During preprocessing, a simple parser goes through the web log and extracts all the ontology instances represented by web pages in the log, converting the web log to a sequence of semantic objects.

Definition 2. The *Semantic Distance* M_{o_i,o_j} is a measure of the distance in the ontology $\mathcal{O}$ between the two classes of which o_i and o_j are instances. □

In other words, it is the measure in units of semantic relatedness between any two objects o_i and o_j. In this paper, semantic distance is achieved during preprocessing by computing the topological distance, in separating edges, between the two classes in the ontology.

Definition 3. A *Semantic Distance Matrix* M is an $n \times n$ matrix of all the semantic distances between the n objects represented by web pages in the sequence database. □

M is not necessarily symmetric, as the semantic distance between two ontology concepts (e.g., *Digital Camera* and *Batteries*) is not always the same from both directions.

Definition 4. Maximum Semantic Distance η is a value which represents the maximum allowed semantic distance between any two semantic objects. □

Maximum semantic distance can be user-specified (i.e., a user with enough knowledge of the used ontology can specify this value) or it can be automatically calculated from the minimum support value specified for the mining algorithm, by applying it as a restriction on the number of edges in the ontology graph, $\eta = min\_sup \times |\mathcal{R}|$.

Given the above definitions, we propose *SemAwareSPM* in Algorithm 1 for semantics-aware sequential pattern mining, and *SemApJoin* as a replacement generate-and-test procedure [4], that uses semantic distance to prune candidate sequences, such that if the semantic distance between the two (*k-1*)-sequences is more than an allowed maximum semantic distance η, then the candidate k-sequence is pruned

Algorithm 1 *Semantics-aware SPM*

SemAwareSPM(M, S, η, min_sup)
Input: sequence database S,
semantic distance matrix M,
maximum semantic distance η
minimum support min_sup
Output: Semantic-rich frequent sequences
Algorithm:

```
1: Scan database S to find the set of frequent 1-sequences, L_1 = {s_1, s_2, ..., s_n}.
2: k=1,
3: C_1 = L_1
   {Apply any apriori-based sequential pattern mining algorithm using η to prune the search space, as follows.}
4: repeat
5:    k++
6:    for L_{k-1}⋈L_{k-1} do
7:       ∀ s_i, s_j such that s_i, s_j ∈ L_{k-1}
8:       C_k ← C_k ⋃ SemJoin(s_i, s_j)
9:    end for
10:   L_k={c∈C_k|support(c)≥min_sup}
11: until L_{k-1}=φ
12: return ⋃_k L_k
    end
```

Function SemJoin() implementation is a variation of the join procdure of the sequential pattern mining algorithm adopted in SemAware for a specific application. An example is SemApJoin() in Figure 1.

from the search space without the need for support counting. Figure 1 shows the details of *SemApJoin* which replaces Apriori-generate function in *AprioriAll-sem* (a semantics-aware variation of AprioriAll [1]). It uses semantic distance for pruning candidate sequences, such that a semantic object is not affixed to the sequence if its semantic distance from the last object in the current sequence is more than η.

insert into C_k
select p.litemset$_1$, ..., p.litemset$_{k-1}$, q.litemset$_{k-1}$
from L_{k-1} p, L_{k-1} q
where (p.litemset$_1$=q.litemset$_1$, ...,
p.litemset$_{k-2}$=q.litemset$_{k-2}$)
AND
$M_{p.litemset_{k-1}, q.litemset_{k-1}} \leq \eta$

Figure 1: *SemApJoin* procedure.

5. SEMANTICS-AWARE NEXT PAGE REQUEST PREDICTION

A Markov process can be used to model the transitions between different web pages [7], or semantic objects in the sequence database. All transition probabilities are stored in an $n \times n$ transition probability matrix P, where n is the number of states in the model. Semantic information can be used in a Markov model as a proposed solution to provide semantically meaningful and accurate predictions without using complicated All-Kth-order or SMM. The semantic distance matrix M is directly combined with the transition matrix P of a Markov model of the given sequence database, into a weight matrix W. This weight matrix is consulted by the predictor software, instead of P, to determine future page view transitions for caching or prefetching.

[3]http://www.w3.org/TR/owl-features/

$$P = \begin{bmatrix} & a & b & c & d & e \\ a & 0 & 0.13 & 0.34 & 0.34 & 0.28 \\ b & 0.5 & 0 & 0.125 & 0.125 & 0.25 \\ c & 0 & 1 & 0 & 0 & 0 \\ d & 0 & 0 & 0 & 0 & 0 \\ e & 0 & 0.25 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2: Example transition probability matrix for a 1^{st}-order Markov model.

Definition 5. The *Weight Matrix* W is an $n \times n$ matrix, which is the result of combining the semantic distance matrix M with the Markov transition probability matrix P, as follows,

$$W_{o_i,o_j} = P_{S_i,S_j} + \begin{cases} 1 - \dfrac{M_{o_i,o_j}}{\sum_{k=1}^{j} M_{o_i,o_k}} & , \ M_{o_i,o_j} > 0 \\ 0 & , \ M_{o_i,o_j} = 0 \end{cases} \quad (1)$$

□

Consider the transition probability matrix P in Figure 2. Assume that the user went through this sequence of page views <*beac*>, there is a 100% chance that the user will next view page b, because $P_{S_3=c,S_2=b} = 1$.

A problem with using Markov models is *ambiguous predictions*, that is when the system reaches a contradiction, such that there is a 50-50 chance of moving from the current state to any of the next two states. For example, notice, in Figure 2 that $Pr(c|a) = Pr(d|a)$, which means that there is an equal probability a user will view page c or d after viewing page a. Thus, the prediction capability of the system will not be accurate in terms of which is more relevant to predict after page a, and the prediction will be ambiguous. The proposed solution utilizes the semantic distance matrix to solve this problem. The transition matrix can be combined with the semantic distance matrix, resulting with W matrix according to equation (1). The resulting matrix provides weights for moving from one state to another, that can be used in place of transition probabilities, with no ambiguous predictions.

6. SEMANTICS-AWARE PREDICTION-ASSISTING SEQUENTIAL PATTERN MINING

In light of the two proposed systems —semantics-aware sequential pattern mining and semantics-aware next page request prediction— a third system can be introduced, that combines sequential pattern mining and Markov models, in *SemAware* architecture as described by Algorithm 2. This combination is supposed to save overall mining time while running *SemAware*. *Semantic-rich association rules* are rules that carry semantic information in them, such that the recommendation engine can make better informed decisions. Such rules are used to provide more accurate recommendation than regular association rules, by overcoming ambiguous predictions problem. For example, consider the following two semantic-rich association rules.

$$o_3 o_2 \rightarrow o_4$$
$$o_3 o_2 \rightarrow o_5$$

Algorithm 2 *SemAware Framework*

Input: clean web log WL=$\{w_1, w_2, \ldots, w_m\}$,
Domain Ontology $\mathcal{O}$,
maximum semantic distance η,
minimum support *min_sup*
Assumptions: Web pages in WL are annotated with semantic information
Output: (1) frequent semantic objects,
(2) semantics-aware association rule,
(3) semantics-aware Markov weights matrix W
Algorithm:
1: $\forall t_i \in WL$, to find semantic-rich user transactions
2: $j \leftarrow 0$
3: $SemW = \{\}$, semantic web log
4: **for** i=1 to m **do**
5: **while** w_i contains semantic objects **do**
6: $j = j + 1$
7: $SemW \leftarrow SemW \bigcup \{< o_j, t_i >\}$
8: **end while**
9: **end for**
10: **for** i=1 to j **do**
11: **for** k=1 to j **do**
12: $M_{o_i,o_j} \leftarrow |r| \in \mathcal{R}$, number of edges to reach from o_i to o_j
13: **end for**
14: **end for**
15: **output** $fo = SemAwareSPM(M, SemW, \eta, min\_sup)$, from Algorithm 1
16: Find Markov transition matrix P while executing step 1 in *SemAwareSPM*
17: **output** semantics-rich association rules by using fo
18: **output** W using eq. (1)
end

Such that $M_{o_2,o_5} < M_{o_2,o_4}$, meaning that o_5 is semantically closer to o_2 than o_4 is. Then, the recommendation engine will prefer o_5 over o_4 and the page(s) representing product o_5 will be recommended.

Such association rules can also be used for more intelligent user behavior analysis, a capability that is not provided by regular association rules. An example of such capability is the generalization "users who rent a movie will also buy a snack", which is a taxonomic abstraction resulting from mapping the representative frequent sequence to the ontology, and looking at higher levels in the concept hierarchy for generalization. This is referred to as *concept generalization*, and it allows the decision maker to make generalizations from frequent user sequences, within the limits of the domain ontology available.

7. EXPERIMENTATION AND ANALYSIS

GSP-sem and *AprioriAll-sem*, semantics-aware variations of GSP [8] and AprioriAll [1] sequential pattern mining algorithms, were tested on two synthetic data sets. A medium sized data set, described as C10T6N40S4D50K [1], and a large sized data set described as C15T8N100S4D100K. These are mined at low minimum support of 1%, while the maximum semantic distance is fixed at η=10. Semantic distances are entered as random numbers into the semantic distance matrix. Experimentation was made for CPU execution time and physical memory usage. It was found that semantic-aware algorithms, namely, *GSP-sem* and *AprioriAll-sem*, require on the average only 26% of the search space, although the semantic distance matrix is stored in the form of a direct access 2-dimensional array. A good increase in mining speed was also noticed. *GSP-sem* and *AprioriAll-sem* are 3-4 times faster than the other algorithms.

To test the scalability of the semantic algorithms against different values for η, a sparse synthetic data set is used, C8T5S4N100D200K. The results showed enhanced performance at smaller values for η, as expected, the result of pruning more candidate sequences during mining. To find the optimal value for η, that will produce mining results similar to non-semantic-aware algorithms, a real web log was constructed to resemble a web log of *eMart*, with 50,000 transactions and 100 unique web pages. The semantic distance matrix was produced manually, from a given ontology, and fed to *GSP-sem* to mine the data set. It was found that values for η between 3 and 4 allow *GSP-sem* to produce same frequent sequences as GSP, and yet still use 38% less memory, and run 2.8 times faster than GSP.

8. CONCLUSIONS AND FUTURE WORK

SemAware is introduced as a comprehensive generic framework that integrates semantic information into all phases of web usage mining. Semantic information can be integrated into the pattern discovery phase, such that a semantic distance matrix is used in the adopted sequential pattern mining algorithm to prune the search space and partially relieve the algorithm from support counting. A 1^{st}-order Markov model is also built during the mining process and enriched with semantic information, to be used for next page request prediction, as a solution to ambiguous predictions problem and providing an informed lower order Markov model without the need for complex higher order Markov models.

Future work includes (1) applying semantic-aware techniques introduced in this paper to pattern-growth sequential pattern mining algorithms [4]. (2) Using the semantic distance matrix as a measure for pruning states in SMM [2]. (3) Investigating more into concept generalization, and the effect of semantics inclusion on answering more complex pattern queries with improved accuracy.

9. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the 11th Int'l Conference on Data Engineering (ICDE-95)*, pages 3–14, March 1995.

[2] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *Transactions on Internet Technology*, 4(2):163–184, 2004.

[3] F. Khalil, J. Li, and H. Wang. A framework for combining markov model with association rules for predicting web page accesses. In *Proceedings of the Fifth Australasian Data Mining Conference (AusDM2006)*, pages 177–184, 2006.

[4] N. R. Mabroukeh and C. I. Ezeife. A taxonomy of sequential and web pattern mining algorithms. *ACM Computing Surveys*, 2010. To appear.

[5] P. Pirolli and J. E. Pitkow. Distributions of surfers' paths through the world wide web: Empirical characterization. *World Wide Web*, 1:1–17, 1999.

[6] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict www surfing. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems 2*, pages 13–21, October 1999.

[7] R. R. Sarukkai. Link prediction and path analysis using markov chains. In *Proceedings of the 9th Intl. World Wide Web Conf. (WWW'00)*, pages 377–386, 2000.

[8] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th Int'l Conference on Extending Database Technology: Advances in Database Technology*, pages 3–17, 1996.

[9] G. Stumme, A. Hotho, and B. Berendt. Semantic web mining: State of the art and future directions. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 4(2):124–143, 2006.