Data Mining of User Navigation Patterns

José Borges and Mark Levene
Department of Computer Science
University College London
Gower Street
London WC1E 6BT, U.K.
Email:{j.borges, mlevene}@cs.ucl.ac.uk

Abstract

The continuous growth in the size and use of the World Wide Web is creating difficulties in both the design of web sites to suit a variety of different users and in the navigation through very large web structures of pages and links. We propose a data mining model that captures the user navigation behaviour patterns. The set of user navigation sessions, which characterise the interaction with the web pages visited, are modelled as a hypertext probabilistic grammar, whose higher probability generated strings correspond to the user's preferred trails. An algorithm to efficiently mine such trails is given. We make use of the N-grammar model which assumes that, when the user is browsing a given page, the last N pages browsed affect the probability of the next page to be visited. The model is based on the well established theory of probabilistic grammars providing it with a sound theoretical foundation for future enhancements. Moreover, we propose the use of entropy as an estimator of the statistical properties of the grammar. Extensive experiments with both real and random data were conducted and the results show that, in practice, our algorithm runs in linear time in the size of the grammar. Our experiments also show that the entropy of the grammar is a good estimator of the number of mined trails and the results from the experiments with the real data confirm the effectiveness of our model.

Keywords. Web data mining, hypertext, trails, probabilistic grammars.

1 Introduction

Data Mining and Knowledge Discovery is an active research discipline involving the study of techniques which search for patterns in large collections of data. Meanwhile, the explosive growth of the World Wide Web (known as the web) in recent years has turned it into the largest source of available online data. Therefore, the application of data mining techniques to the web, called *web data mining*, was the natural subsequent step and one of its research directions, which is being followed by an increasing number of researchers, is mining for user navigation patterns. This research field focuses on techniques to study the user behaviour when navigating within a web site. Understanding the visitors navigation preferences is an essential step both in the process of customising and adapting the site's interface for the individual user, [PE97], and in improving the site's static structure of the underlying hypertext system [RM98].

When web users interact with a site, data recording their behaviour is stored in web server logs which in a medium size site can amount of several megabytes per day. Moreover, since the log data is collected in a raw format it is an ideal target for being analysed by automated tools. There have so far been two main approaches to mining for user navigation patterns from log data. In the first approach log data is mapped into relational tables and an adapted version of standard data mining techniques, such as mining association rules, are invoked, see for example [CPY98]. In the second approach techniques have been developed which can be invoked directly on the log data, see for example [BL98] or [SFW99].

In this paper we propose a new model for handling the problem which directly captures the semantics of the user navigation sessions. We model the user navigation records as a hypertext probabilistic grammar whose higher probability generated strings correspond to the user's preferred trails. Section 2 presents the underlying hypertext model, while Section 3 presents the results of the experiments. We refer the reader to [BL99] for more detail.

2 Hypertext Probabilistic Grammars

A log file can be seen as a per-user ordered set of web page requests from which it is possible to infer the user navigation sessions. In this work we simply define a *user navigation session* as a sequence of page requests such that no two consecutive requests are separated by more than X minutes, where X is a parameter. We note however, that the more advanced data preparation techniques described in [CMS99] could be used in a data pre-processing stage to fully take advantage of all the information available in the log files.

The user navigation sessions inferred from the log data are modelled as a *hypertext probabilistic language* generated by a *hypertext probabilistic grammar* (or simply HPG) which is a proper subclass of probabilistic regular grammars [LL99]. A HPG is a probabilistic regular grammar which has a one-to-one mapping between the set of non-terminal symbols and the set of terminal symbols. Each non-terminal symbol corresponds to a web page and a production rule corresponds to a link between pages. Moreover, there are two additional states, S and F, which represent the start and finish states of the navigation sessions.

From the set of user sessions we obtain the number of times a page was requested, the number of times it was the first state in a session, and the number of times it was the last state in a session. The number of times a sequence of two pages appears in the sessions gives the number of times the corresponding link was traversed.

The probability of a production from the start state is proportional to the number of times the corresponding state was visited, implying that the destination node of a production with higher probability corresponds to a state that was visited more often. Moreover, α is a parameter that attaches the desired weight to a state being the first in a user navigation session. If $\alpha = 0$ only states which were the first in a session have probability greater than zero of being in a production from the start state, on the other hand if $\alpha = 1$ all state visits are given proportionate weight. The probabilities of the productions from the start state correspond to the vector of initial probabilities, π, and the probability of a production is assigned in such a way that it is proportional to the frequency with which the corresponding link was traversed. Note that when $\alpha > 0$ every grammar state has an initial probability greater than zero. In the example of Figure 1 we have 6 user sessions with a total of 24 page requests, wherein state A_1 was visited 4 times, 2 of which are the first state in a user session, therefore, since $\alpha = 0.5$ we have $\pi(A_1) = \frac{0.5 \cdot 4}{24} + \frac{0.5 \cdot 2}{6} = 0.25$.

ID	Trail
1	$A_1 \to A_2 \to A_3 \to A_4$
2	$A_1 \to A_5 \to A_3 \to A_4 \to A_1$
3	$A_5 \to A_2 \to A_4 \to A_6$
4	$A_5 \to A_2 \to A_3$
5	$A_5 \to A_2 \to A_3 \to A_6$
6	$A_4 \to A_1 \to A_5 \to A_3$

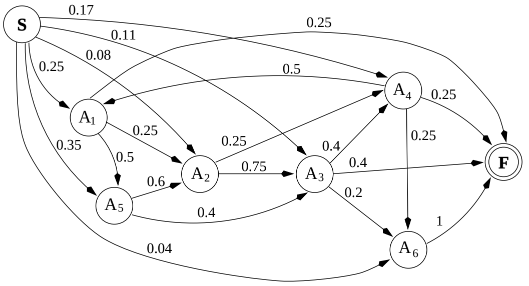

Figure 1: A set of trails and the corresponding hypertext grammar for $N = 1$ and $\alpha = 0.5$.

In a HPG the probability of the first derivation step is defined as the *support threshold*, θ, and is not factored into the derivation probability. Thus, the probabilities of the productions from the start state are used to prune out the strings which would otherwise have high probability but correspond to a subset of the hypertext system rarely visited. Moreover, a string is included in the grammar's language if its derivation probability is above the *cut-point*, λ, where the cut-point corresponds to the grammar *confidence* threshold. The values of the support and confidence thresholds give the user control over the quantity and quality of the trails to be included in the rule set. The strings generated by the grammar correspond to the user navigation trails, and the aim is to identify the subset of these strings that best characterise the user behaviour when visiting the site. Parameters such as *confidence* and *support* are defined as preference measures which rank the generated strings.

Moreover, the concept of an N-grammar [Cha96], where $N, N \geq 1$, is called the history depth, is used to determine the assumed user memory when navigating within the site. For a given N it is assumed that only the N previously visited pages influence the link the user will choose to follow next. The intuition is that the user has a limited memory of the previously browsed pages and that the next choice depends only on the last N pages browsed. In an N-grammar each of the states of the HPG correspond to a sequence of N pages visited. Figure 1 shows the grammar inferred from the given set of trails for $N = 1$ and $\alpha = 0.5$. (We freely utilise in our figures the duality between grammars and automata [HJ79].) In Figure 2 we show the grammar rules given two different confidence thresholds.

$\lambda = 0.2$ and $\theta = 0.1$				$\lambda = 0.3$ and $\theta = 0.1$	
String	Confidence	String	Confidence	String	Confidence
A_1A_2	0.25	$A_4A_1A_5$	0.25	$A_1A_5A_2$	0.3
$A_1A_5A_3$	0.2	A_4A_6	0.25	A_3A_4	0.4
$A_1A_5A_2A_3$	0.23	A_5A_3	0.4	A_4A_1	0.5
$A_3A_4A_1$	0.2	$A_5A_2A_3$	0.45	A_5A_3	0.4
A_3A_6	0.2			$A_5A_2A_3$	0.45

Figure 2: The rules obtained given two different confidence thresholds.

We propose to use the *entropy* of the HPG [CT91] as an estimator of the statistical properties of the grammar. The entropy is a measure of the uncertainty in the outcome of a random variable and in this context the sample space is the set of all strings generated by the grammar. Assuming a transition with probability one from F to S a HPG corresponds to an irreducible and aperiodic Markov chain with a stationary distribution vector μ and transition matrix A. Thus, we can estimate the entropy with the following expression $H(G) = H(\pi) + (-\sum_{ij} \pi_i A_{ij} log\ A_{ij})$ where $H(\pi)$ is included to take into account the randomness of the choice of the initial page, see [CT91] for detail on the entropy of a Markov chain. Note that we use the vector of initial probabilities π as an estimator of the stationary vector μ, since it is proportional to the number of times each state was visited. The intuition behind the estimator is that if the entropy of a probabilistic grammar is close to zero there should be a small set of strings with high probability and if the entropy is high then there should be a large number of strings with similar and low probability. Such a measure which is an estimator of the statistical distribution of the grammar string probabilities can be useful in helping the user in the specification of the support and confidence thresholds.

The algorithm used to mine rules having confidence and support above the specified thresholds is a special case of a directed graph Depth-First Search which performs an exhaustive search of all the strings with the required characteristics, cf. [BL98].

3 Experimental Evaluation

To assess the performance and the effectiveness of the proposed model experiments with both random and real data were conducted. Tests with random data provide the means of evaluating many different topologies and configurations of a HPG and tests with real data allow us to verify whether or not the model is potentially useful in practice.

In the experiments with random data we had two main objectives: (i) to evaluate the algorithm performance and scalability and (ii) to evaluate how the grammar entropy could be used a an estimator of rule characteristics. The experiments were conducted for various grammar configurations where the size, n, varied between 100 and 4000 states, the confidence threshold varied between 0.1 and 0.5, and the support was fixed to $\frac{1}{n}$ for each grammar size. For each configuration 150 runs were performed. The results show that for a given confidence the average number of iterations (for the 150 runs) varies linearly with the number of grammar states (the grammar size), see the left-hand side of Figure 3; the

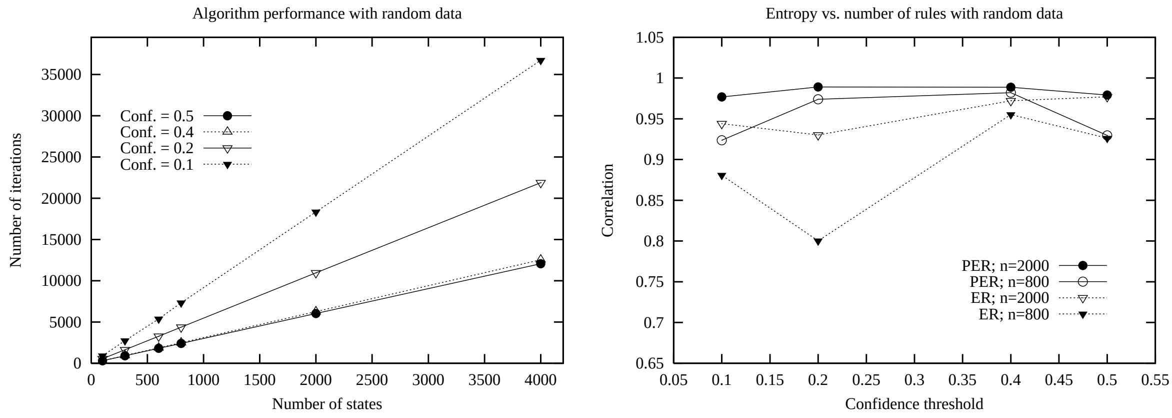

Figure 3: Results with random data.

CPU time follows a similar trend. (Note that the grammar size depends only on the number of pages in the hypertext system and not on the number of user sessions in the log data.)

In order to verify the utility of the grammar's entropy as an estimator of the properties of the rules mined we calculated for each grammar size and confidence threshold the correlation between: (i) the entropy and the number of rules, (ii) the entropy and the number of iterations and (iii) the entropy and the average rule length. In order to test to what extent both the confidence and support thresholds were affecting the entropy of the results we measured the entropy of a grammar inferred from the set of mined rules, called the *posterior grammar*. The right-hand side of Figure 3 shows the results for grammars with 800 and 2000 states where PER is the entropy of the posterior grammar and ER is the entropy of the original grammar but using the vector of initial probabilities of the posterior grammar as the estimator for the stationary vector in the original grammar. The results show that PER is a better estimator and the results regarding ER suggest that the original grammar is a good estimator for the number of rules to mine provided we find a good estimator for the stationary vector. The experimental results also show that the entropy is not a good estimator of the number of iterations or of the average rule length.

The real log files were obtained from the authors of [PE98] and contain two months of usage from the site `http://www.hyperreal.org/music/machines/`. We divided each month into four subsets, each corresponding to a week, and for each subset we built the corresponding HPG for several values of the history depth. The right-hand side of Figure 4 shows the variation of the number of states (grammar size) with the history depth where it can be seen that the size of the N-grammar model

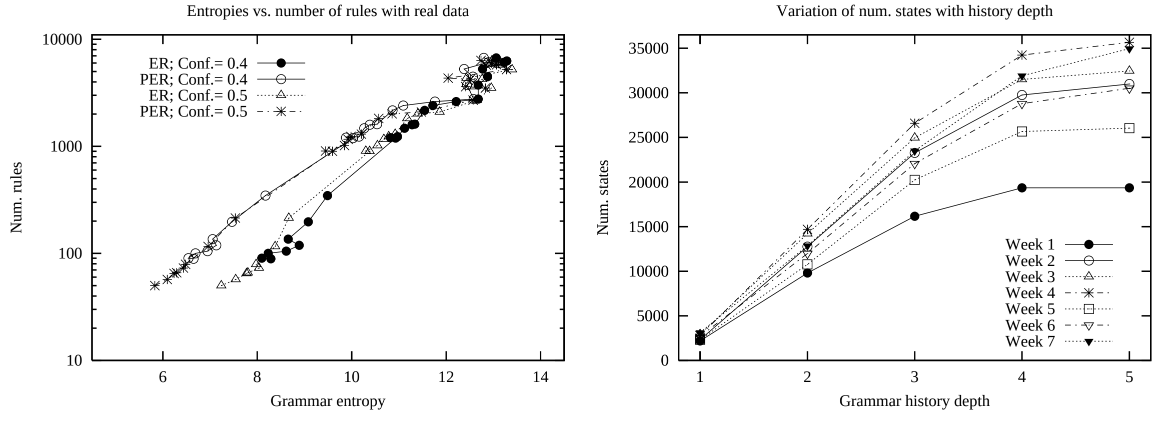

Figure 4: Results with real data.

increases at a rate much slower than the worst case, stabilising for history values of order 5, this is due to the sparseness of the data. The performance analysis with real data showed results similar to those obtained with random data. The left-hand side Figure 4 shows how the number of mined rules in the real data relates to the entropy in the posterior grammar, note that the logarithm of the number of rules was taken; a similar pattern was obtained for the random data. Note that while in the random data plot each point represents the correlation between the grammar entropy and the number of rules for a specific grammar topology (for 150 runs), the plot of real data gives the overall variation of the number of rules with the grammar entropy and each point corresponds to a single run for a specific topology. Our experiments provide strong evidence that while the overall variation presents a regular *log*-linear trend, for a specific grammar topology there is a strong linear correlation between the number of rules and the grammar entropy. Figure 5 gives some examples of the rules mined from the real data as well as the top-ten pages in the web site, which correspond to the states with the highest initial probability. A more complete presentation of the results can be found in [BL99].

The Top-Ten pages		
Rk.	URL	Prob.
1	/	0.088
2	/manufacturers/	0.050
3	/samples.html	0.026
4	/samples.html?MMAgent	0.023
5	/manufacturers/Roland/	0.020
6	/links/	0.018
7	/Analogue-Heaven/	0.017
8	/categories/software/Windows/	0.016
9	/search.html	0.015
10	/categories/software/	0.014

Rule 1	Conf.= 0.43	Avg. No. Traversals = 44.5	4 pages
/ /ecards/ /ecards/cards/ /ecards/cards/96301			

Rule 2	Conf.= 0.53	Avg. No. Traversals = 11	5 pages
/ /manufacturers/ /manufacturers/Music-and-More/ /manufacturers/Music-and-More/VF11/ /manufacturers/Music-and-More/VF11/info/VF11.review			

Rule 3	Conf.= 0.56	Avg. No. Traversals = 64	5 pages
/categories/drum-machines/samples/ /categories/drum-machines/samples/deepsky_kicks/ /categories/drum-machines/samples/deepsky_kicks/README/ /categories/drum-machines/samples/deepsky_kicks/ /categories/drum-machines/samples/			

Figure 5: Example of rules mined from the real data.

4 Related Work

The use of data mining techniques to analyse log data was first proposed by [CPY98] and [YJGMD96]. In [CPY98] the log data is converted into a form amenable by existing association rules techniques and two algorithms are given to mine the rules in this context, in which the items in a transaction must be consecutive. [YJGMD96] propose a method which stores each session in a vector that contains the number of visits to each page and an algorithm is given to find clusters of similar vectors.

In [PE97] the authors challenged the AI community to use the the log data to create adaptive web sites and in [PE98] they present a technique which automatically creates index pages from the log data, i.e., pages containing collections of links which the user navigation behaviour suggests are related. In our previous work, [BL98], we proposed to model the log data as a directed graph with the arcs weights interpreted as probabilities that reflect the user interaction with the site, and we generalised the association rule concept. [SKS98] proposes the use of log data to predict the next URL to be requested so the server can generate in advance web pages with dynamic content. A tree which contains the user paths is generated from the log data and an algorithm is proposed to predict the next request given the tree and the current user session. In [SF98] the authors propose a log data mining system composed of an aggregation module and a data mining module. The aggregation module infers a tree structure from the data in which the mining is performed by a human expert using a mining query language. Finally, [ZXH98] propose the integration of data warehousing and data mining techniques to analyse web records, and [CMS99] study cleaning and preparation techniques which convert log data into user navigation sessions in a form amenable to processing by the existing data mining techniques.

5 Concluding Remarks and Future Work

We have proposed a model of hypertext to capture user preferences when navigating through the web. We claim that our model presents the advantage of being compact, self contained, coherent, and based on the well established work in probabilistic grammars providing it with a sound foundation for future enhancements of the model such as the study of its statistical properties. In fact the size of the model depends only on the size of the web site being analysed and not on the amount of data collected.

The set of user navigation sessions is modelled as a hypertext probabilistic grammar, and the set of strings which are generated with higher probability correspond to the navigation trails preferred by the user. An algorithm to efficiently mine these strings is given. Extensive experiments with both real and random data were conducted and the results show that, in practice, the algorithm runs in linear time in the size of the grammar. Moreover, the entropy of the posterior grammar is shown to be a good estimator of the number of rules output from our algorithm, and the experiments with real data confirm the effectiveness of our model. Our model has potential use both in helping the web site designer to understand the preferences of the site's visitors, and in helping individual users to better understand their own navigation patterns and increase their knowledge of web's content.

We are currently experimenting with randomised algorithms for mining rules as an efficient alternative to the DFS algorithm, and we intend to further study the use of information theoretic measures and their relation to the model properties. We are also planning to incorporate relevance measures to web pages in order to assist the user in locating useful information.

References

[BL98] J. Borges and M. Levene. Mining association rules in hypertext databases. In *Proc. of the fourth Int. Conf. on Knowledge Discovery and Data Mining*, pages 149–153, August 1998.

[BL99] J. Borges and M. Levene. Mining navigation patterns with hypertext probabilistic grammars. Research Note RN/99/08, Department of Computer Science, University College London, Gower Street, London, UK, February 1999.

[Cha96] E. Charniak. *Statistical Language Learning.* The MIT Press, 1996.

[CMS99] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1), February 1999.

[CPY98] M.-S. Chen, J. S. Park, and P. S. Yu. Efficient data mining for traversal patterns. *IEEE Trans. on Knowledge and Data Eng.*, 10(2):209–221, March/April 1998.

[CT91] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[HJ79] J. Hopcroft and J.Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[LL99] M. Levene and G. Loizou. A probabilistic approach to navigation in hypertext. *Information Sciences*, 114:165–186, 1999.

[PE97] M. Perkowitz and O. Etzioni. Adaptive web sites: an AI challenge. In *Proc. of Int. Joint Conf. on Artificial Intelligence*, 1997.

[PE98] M. Perkowitz and O. Etzioni. Adaptive sites: Automatically synthesizing web pages. In *Proc. of the 15th National Conf. on Artificial Intelligence*, Madison, Wisconsin, July 1998.

[RM98] L. Rosenfeld and P. Morville. *Information Architecture for the World Wide Web.* O'Reilly, 1998.

[SF98] M. Spiliopoulou and L. C. Faulstich. WUM: A web utilization miner. Int. Workshop on the Web and Databases, March 1998.

[SFW99] M. Spiliopoulou, L. C. Faulstich, and K. Wilkler. A data miner analyzing the navigational behaviour of web users. In *Proc. of the Workshop on Machine Learning in User Modelling of the ACAI99*, Greece, July 1999.

[SKS98] S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict http requests. *Computer Networks and ISDN Systems*, 30:457–467, 1998.

[YJGMD96] T. W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proc. of the 5th Int. World Wide Web Conference*, pages 1007–1014, 1996.

[ZXH98] O. R. Zaïane, M. Xin, and J. Han. Discovering web access patterns and trends by applying olap and data mining technology on web logs. In *Proc. Advances in Digital Libraries Conf.*, pages 12–29, Santa Barbara, CA, April 1998.