Full-Coverage Web Prediction
based on Web Usage Mining and Site Topology

Diamanto Oikonomopoulou[1], Maria Rigou[1,2], Spiros Sirmakessis[2], Athanasios Tsakalidis[1,2]
[1]*Computer Engineering and Informatics Department, University of Patras, GR-26504*
[2]*Research Academic Computer Technology Institute, 61 Riga Feraiou str., Patras, GR-26221*
oikonomo@ceid.upatras.gr, rigou@cti.gr, syrma@cti.gr, tsak@cti.gr

Abstract

Understanding and modeling user online behavior, as well as predicting future requests remain an open challenge for researchers, analysts and marketers. In this paper, we propose an efficient prediction schema based on the extraction of sequential navigation patterns from server log files, combined with web site topology. Traversed paths are monitored, internally recorded and cleaned before being completed with cashed page views. After session and episode identification follows the construction of n-grams. Prediction is based upon a 5+ n-gram schema with all lower level n-grams participating, a procedure that resembles the construction of an All 5th-order Markov Model. The schema achieves full coverage while maintaining competitive prediction precision.

1. Introduction

People display strong regularities in their cognitive behavioral model, and therefore in the actions they perform. In the web environment more specifically, there exist strong statistical regularities among the surfing patterns of a user ([1], [2]). Prediction aims to identify proper mechanisms that take advantage of the large volume of data users leave behind while navigating the web. Prediction models may be approached from a *data mining* or *distributed systems* perspective.

In the first case, prediction models can be further categorized as based on *classification*, based on *frequent itemsets and association rules* or based on *clustering*. Classification [2] is a two-phase procedure related to the appropriate association of an object, into one of pre-determined classes of common attributes (i.e. common expected behavior). Association rule mining discovers interesting associations or correlations among a large dataset, proceeding in a co-occurrence based prediction. Yang et al. [3], propose an association-based prediction model for caching and prefetching. In some cases, the two aforementioned techniques cooperate for the formation of Class Association Rules [4].

In the distributed systems approach, the key idea relies on the construction of a predictive model that suggests future events based on past experience (web access patterns). The difference from the data mining approach is that future actions are predicted without the concern for interactivity or immediate benefit [5]. A problem-solving framework, used for web document prediction and retrieval, is Case-based Reasoning (CBR). Yang et al. [3], propose a server-side Bayesian networks [6]. CBR application, aiming at the improvement of system performance during prefetching. In some cases, CBR techniques are combined with

In statistical natural language processing, an ordered sequence of *n* items is defined as an *n-gram*. In the area of web usage mining, *n-grams* are correlated with time-ordered sequences of user accesses, thus proper subsets of user sessions. There exist two types of *n-gram* based prediction models; *point-based* and *path-based*. The first ones rely on the currently observed action (user request), and thus suffer from low accuracy. Path-based models ([7]) take into account the last *n* recorded accesses to form a prediction for an ensuing request, capturing both the temporal and the sequential way web accesses are generated. Path-based models may demonstrate low applicability, due to the rarity of long-length patterns. Experimental results have shown that for $n \geq 4$, precision upper bound does not improve tremendously, while applicability decreases dramatically. *Markov models (MM)* have been extensively used in the field of stochastic processes [8], and are capable of tracking the likelihood of varying *n-grams*, in a state space encoding. The disadvantages of higher-order Markov Models summarize in non-negligible complexity, demanding storage requirements, insufficient coverage and in some cases poor predictive accuracy –compared to lower MM's ([5],[9]). As an alternative, a hybrid All-Kth order MM may be used, that combines different order MM's in a way that the resulting model has a low state complexity, improved prediction accuracy, while retaining the coverage of its components [9].

3. The Proposed Approach to Prediction

The prediction schema proposed in this paper is based on the extraction of sequential navigational patterns from recorded server log data, integrating categorization techniques. Our goal is to encapsulate the internal motivations and ultimate objectives of varying user profiles, into corresponding access patterns that will allow for reliable predictions. The mining process uses as input page access requests sent from anonymous, non authenticated users that the system has no information regarding individual goals or characteristics. This approach fits the real life situations where commercial sites and portals choose not to make users go through registration (a requirement that jeopardizes privacy or drives them away).

Apart from sequential pattern discovery, an effective categorization technique that integrates semantic information on web site content has also been implemented and has proved quite effective. More specifically, web site's page files are categorized according to their content (in our setup it is a manual process performed by the site administrator or owner). Next, based on the assumption that interrelation of page files through links implies content interrelation, we construct a connected graph that represents the site's internal hyperlink structure. The proposed algorithm, outputs a *site-map*, where page files are inter-connected through links and with the home page as the starter node. This semantic information is valuable in cases where mining does not return any matching patterns to be used for prediction.

Pattern extraction starts with data cleaning and log file parsing. Data cleaning regards stripping out requests for graphics files, as well as page misses. URI fields for all remaining page requests are normalized using a common formatting. Following executes the data processing phase comprising four successive steps: (1) *identification* and *extraction of user sessions* from the dataset, (2) their proper *completion* with cached page views, (3) *extraction of episodes* from the completed user sessions and (4) *formation of n-grams*. Formally, the term session refers to a delimited, time-ordered set of user clicks. Session identification is based on either time or structure related criteria [10]. Our working assumption is that each different agent type for an IP address represents, at least one, discrete session (as in [11] and [12]). In addition, we assume that the time interval spent on a single page file should not exceed a given threshold (in which case a new session has started). In the case of multiple candidate sessions, the assignment of requests with the same IP address and agent to one of the sessions is determined by measuring the distance between the request and each session. *Distance* between a request r and a session S, is defined as the number of links needed to be traversed from the last recorded page view of S, in order to obtain the referrer field of r as a request in the same session [11]. A request is assigned to the session of minimum such distance.

The identified sessions must be complemented with cached page views that although traversed, are not recorded. Cooley in [11] tracks cashed page accesses by keeping a certain number of most recent page requests in a stack and in the case of a stack miss he resorts to a full history search. In this work we decided to perform full history search and skip the intermediate stack access, since experimental results showed that the full history search does not cause noticeable performance reduction. Recall that, although Cooley's assumption supposes that cached page views are the result of user's backtracking - using the back button in the browser window- we should not neglect site's topology, i.e. hyperlinks that allow user's traversal between non sequential pages. Thus in many cases, full session history search is required, in order to obtain the desirable page file while it does not reduce performance, as sessions are not expected to be long. Using this technique we managed to have more accurate results without significant penalty in complexity.

Session completion is followed by episode identification. Episodes should be regarded as navigational subsets of significant semantic value that depict a well-formed snapshot of user's orientation and desire. Episodes are identified according to the *maximal forward reference* method (as described in [13]), where an episode is defined as a time-ordered click sequence up to the request before a backward reference is made (assuming that forward references may be used as reliable indications of what the user is looking for, while backtracking can be considered as 'noise' to the semantic interpretation of user traversals).

The second phase of data processing proceeds with the formation of *n*-grams based on the extracted episodes. The focus is mainly set on 3rd to 5th order *n*-grams and their suffices. These lengths were chosen as an equilibrium factor between precision and applicability, since long *n*-grams tend to increase prediction accuracy with a considerable loss in applicability. This is also backed up by the observation that in most real-world implementation scenarios, long traversal sequences repeat less frequently than shorter ones. First-order *n*-grams are also included in order to sustain high scores in coverage. The developed algorithm generates an index table T that lists all distinct *n*-gram couples and their subsequent request, as observed in the dataset. Each record in the index table includes an *n*-gram, its suffices and a corresponding support value (indicating the specific *n-gram*'s occurrence frequency). An *n*-gram consists of page identifiers separated by a proper delimiter. After extracting all possible *n*-grams, overall mean support is computed in order to prune out of the index table rows with support below average. *N*-grams whose lower order proper subsets present a quite higher support value (still over the minimum support threshold), are also removed. As a consequence, a matching (n-1)-gram will be preferable than a matching *n*-gram, in case it demonstrates better support. Even though in most cases, higher *n*-grams perform better in terms of precision, the prediction is based on a lower order *n*-gram, if the prerequisite (significantly higher support) stands. Obviously, 1-grams are not affected from this last step, which is legitimate so that coverage maintains a satisfactory level. No *n*-grams are pruned out of the dataset due to their unary suffice, mostly because 1-grams rely their decision on a single observed request.

The remaining set of *n*-grams provides the final patterns for obtaining prediction, concluding to a prediction table *P*. The prediction phase is based upon a 5+ *n*-gram schema with all remaining *n*-grams (of length 5 down to 1) participating in it. This procedure resembles the construction of an *All 5th-order Markov Model*. Generally, higher order *n*-grams receive higher priority by the prediction algorithm. In case that an observed sequence cannot be matched with some recorded *n*-gram, the prediction algorithm searches for matching (*n*-1)-grams, taking into account only the *n-1* last requests of the sequence. If attempts for matching *n*-grams (*n*>1) are unsuccessful, the algorithm searches the prediction table *P* for the matching 1-gram (corresponding to the last recorded page file in the sequence).

In case that there exist no matching *n*-gram in the table *P* (*n*=1,2,3,4,5), the prediction algorithm proceeds with examining the web site topology through the constructed site-map. Given a sequence *seq* that cannot be matched with any *n*-gram in *P* and the last observed request from *seq* is *l*, the prediction algorithm searches for all *l*'s outgoing links stored pair-wise in the site-map, in the form of (*l, p)*, and outputs as prediction the page with the higher support value in the training dataset, that also belongs in *l*'s category. The rationale is based on the fact that in absence of a suitable matching pattern, we should search upon all potential single step transitions of *l* -according to website structural constraints- and choose the most frequently observed, biased by the assumption that the user will keep on navigating through pages of the same category. This way, we are able to produce predictions even for cases that there are no recorded user patterns in the training dataset, concluding to a full coverage (100%).

4. Experimental results: a case study

The case study considers single action prediction, i.e. prediction of the page file that will be requested immediately after a recorded sequence of requests. This approach accommodates both for evaluating prediction schemas with varying lengths and reaching comparative conclusions. The log data were recorded while monitoring users navigating a multi-topic electronic magazine. The magazine web site comprised 143 pages, categorized in several topic sections. An arbitrary set of the log data was used as a training set and the remaining as test data. We attempted single action prediction using an all 3rd, 4th and 5th order *n*-gram model. When existing n-grams could not provide a prediction the decision was based on the site map table, as described in the previous section.

Comparisons in the performance of different order models were based on the precision (or accuracy) and applicability values of each schema. *Precision* is defined as the number of correct predictions P^+ divided by the number of feasible predictions P^++P^- (where P^- refers to the number of unsuccessful predictions). *Applicability* is defined as the number of feasible predictions divided by all cases *R* that are used as input to the prediction process. Note that a case in *R* that the algorithm failed to produce a prediction for is not assigned to neither P^+ nor P^-.

Figure 1 plots precision and applicability for the different models under the hypothesis that the site-map was not used in the prediction process, while figure 2 presents the resulting values when the site-map was also taken into account. As depicted in figure 2, in the latter case prediction achieves full coverage. In the big majority of cases, the pruning step resulted in removing from the data set the same n-grams for both *all 3rd* and *all 4th order* models, which in turn led to the calculation of similar prediction tables *P* and therefore precision values.

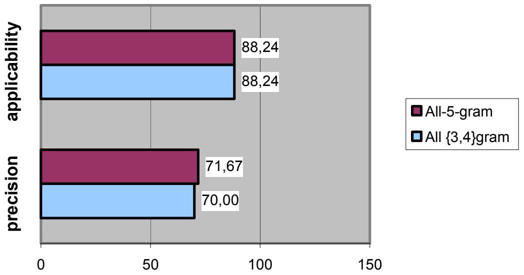

Figure 1. Precision and applicability for pure all nth order n-gram model

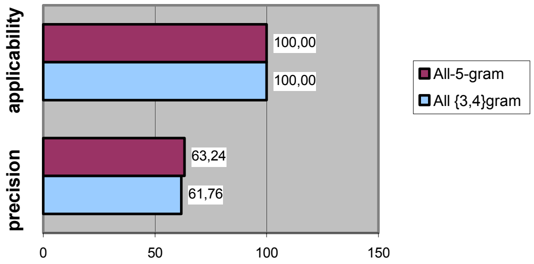

Figure 2. Precision and applicability for hybrid all nth order n-gram model

Precision upper bounds for the *all 5th order model* appear higher than lower order models, because they rely their prediction decisions upon the n-grams of the lower order models, alongside with a number of additional ones. To picture that, if we restrict out prediction to using 3-grams, 4-grams and 5-grams only, precision drops by approximately 20%. Another significant observation is the downfall in precision and applicability values in the hybrid approach (figure 2). This fall is justified by the requirement for full coverage. More specifically, applicability increases as its numerator $P^{+}+P^{-}$ increases, which also results in a simultaneous decrease in precision. A final remark is that the overhead caused by taking the *all 5th order model* approach is practically restricted in additional space requirements.

The proposed schema requires linear retrieval times regarding prediction decisions, while keeping space and memory requirements low through proper data tabulation. The prediction table is easily updated as the training dataset increases, allowing constant learning on the part of the algorithm. Besides, the schema is flexible enough to support prediction for more than one action with minor tuning. Furthermore, in cases where the schema is used for producing recommendations or applying web prefetching, the fact that it assures full-coverage is a significant asset. Precision's upper bound reached 71,67% overall prediction attempts, a quite competitive percentage when compared to other prediction techniques. For example, the *all 3-gram* based model described in [7] achieves best case prediction accuracy near 63%, while applicability drops by 40%. Deshpande and Karypis [9] follow an approach similar to ours using a 5th order Markov-based prediction model that demonstrates accuracy around 50% when tested on log files coming from e-commerce sites. Davison and Hirsh in [2] propose a machine learning algorithm that accomplishes 40% precision in predicting future requests. Yang et al. [3] propose a CBR technique for web object prediction that offers a prediction accuracy upper bounded by 40%.

5. Conclusions and future work

In this paper we have presented an efficient schema for predicting future web requests on a single site, based on the extraction of sequential navigation patterns from already recorded log data, combined with the site's existing topology in terms of topic categories.

One way to improve the proposed prediction schema is to deploy a more complex categorization (or even classification) method instead of the current one that treats all page files similarly, regardless of whether they are media or auxiliary ones. Auxiliary pages present greater support than media pages and thus the algorithm assigns them higher priority when applying site-map based prediction, undermining the related media pages. An alternative approach may lead to increased precision upper bounds. Another refinement can be achieved during episode extraction by a using linear complexity algorithm, as in [13].

6. References

[1] Liu, Jiming and Zhang, S. W., "Characterizing Web usage regularities with information foraging agents," in *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, No. 4, 2004.

[2] B.D. Davison and H. Hirsh, "Predicting Sequences of User Actions", Presented at the AAAI-98/ICML'98 Workshop on Predicting the Future: AI Approaches to Time Series Analysis, Madison, WI, July 27, 1998, and published in Predicting the Future: AI Approaches to Time Series Problems, Technical Report WS-98-07, pp. 5-12, AAAI Press.

[3] Q. Yang, I.T.Y. Li and H.H. Zhang, "Mining High-Quality cases for hypertext prediction and prefetching", in *Proceedings of the 2001 International Conference on Case Based Reasoning, ICCBR-2001,* Vancouver BC, Canada, July 2001.

[4] B. Liu, W. Hsu and Y. Ma, "Integrating Classification and Association Rule Mining", in *Proceedings of KDD-98, New York,* 1998.

[5] B.D. Davison, "The Design and Evaluation of Web Prefetching and Caching Techniques", *PhD thesis submitted to the Graduate School of New Brunswick Rutgers in the state University of New Jersey*, 2002.

[6] S.N. Schiaffino and A. Amandi, "User Profiling with Case-Based Reasoning and Bayesian Networks", in *Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI, IBERAMIA-SBIA 2000*, Atibaia, SP, Brazil, November 19-22, 2000.

[7] Z. Su, Q. Yang, Y. Lu and H. Zhang, "What next: A prediction System for Web Requests Using N-gram Sequence Models", in *p*roceedings of the First International Conference on Web Information Systems and Engineering Conference, , Hong Kong, June 2000, pp 200-207.

[8] A. Papoulis, "Probability, Random Variables and Stochastic Processes", Mc Graw Hill, 1991.

[9] M. Deshpande and G. Karypis, "Selective Markov Models for Predicting Web-Page Accesses", in *Proceedings of the 1st SIAM Data Mining Conference,* 2000.

[10] B. Berendt, B. Mobasher, M. Spiliopoulou, J. Wiltswire Honghua Dai, T. Luo and M. Nakagawa, "Measuring the Accuracy of Sessionizers for Web Usage Analysis", in *Proceedings of the Web Mining Workshop at the First SIAM International Conference on Data Mining*, April 2001. Chicago, IL pp. 7-14.

[11] R.W. Cooley, "Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data", *PhD thesis submitted to faculty of the graduate school of the University of Minnesota*, 2000.

[12] P. Pirolli, J. Pitkow and R. Rao, "Silk from a sow's ear : Extracting Usable Structures from the Web", in *CHI-96*, Vancouver, 1996, pp. 118-125.

[13] Z. Chen, R. Fowler and A. Wai-Chee Fu, "Linear Time Algorithms for Finding Maximal Forward Reference", in *Proceedings of the 2003 IEEE Intl Conference On Info Tech: Coding and Computing (ITCC03*), April 28 - 30, 2003**,** Las Vegas, Nevada, pp. 160-164.