A COMPARISON OF SCORING METRICS FOR PREDICTING THE NEXT NAVIGATION STEP

JOSÉ BORGES

School of Engineering, University of Porto,
R. Dr. Roberto Frias, 4200 - Porto, Portugal

`jlborges@fe.up.pt`

MARK LEVENE

School of Computer Science and Information Systems,
Birkbeck, University of London,
Malet Street, London WC1E 7HX, U.K.

`mark@dcs.bbk.ac.uk`

The problem of predicting the next request during a user's navigation session has been extensively studied. In this context, higher-order Markov models have been widely used to model navigation sessions and for predicting the next navigation step, while prediction accuracy has been mainly evaluated with the hit and miss score. We claim that this score, although useful, is not sufficient for evaluating next link prediction models with the aim of finding a sufficient order of the model, the size of a recommendation set and assessing the impact of unexpected events on the prediction accuracy. Herein, we make use of a variable length Markov model to compare the usefulness of three alternatives to the hit and miss score: the Mean Absolute Error, the Ignorance Score and the Brier score. We present an extensive evaluation of the methods on real data sets and a comprehensive comparison of the scoring methods.

Key words: Web usage mining; variable length Markov model; sequential prediction; scoring metrics

1. Introduction

In the context of web site personalisation web usage mining techniques have been utilised to take advantage of the data collected as a result of users' interactions with the web site (Mobasher, 2007). Herein, we focus on the problem of building models to represent past users behaviour, that are able to predict the most likely links a user will request when viewing a page. Such links can then be, for example, provided as a set of navigation recommendations to the user.

When users click on a link in a web page, submit a query to a search engine or access a wireless network they leave a trace behind them that is stored in a log file. The information stored in the log file for each user click will include items such as a time-stamp, identification of the user (for example, an IP address, a cookie or a tag), the user's location, query terms entered and further clickstream data, where appropriate. (We use the term 'click' generically to mean a click on a link, a query submission or an access to a network.) Thus the log file contains an entry for each click and can be preprocessed into time-ordered sessions of sequential clicks. In (Spiliopoulou et al., 2003) the authors present a study to evaluate heuristics to reconstruct sessions from server log data, known as sessionising. They show that sessions can be accurately inferred for web

2

sites with embedded session identification mechanisms, that time-based reconstruction heuristics are acceptable when cookie identifiers are available, and that referrer-based heuristics should be used when cookie identifiers are not available.

Higher-order Markov models have been widely used for modeling user records and predicting next page requests. For the task, we have proposed a Variable Length Markov Chain (VLMC) method (Borges and Levene, 2000; Borges and Levene, 2005), which is an extension of a Markov chain that allows variable length history to be captured (Bejerano, 2004). We note that, we have previously proposed (i) a complementary method to evaluate the predictive power of a model that takes into account a variable length history when estimating the probability of the user's next link choice given his/her navigation trail (Borges and Levene, 2007b), (ii) a method to evaluate the summarisation ability of a VLMC model (i.e., the accuracy with which the model represents a collection of sessions) and (iii) have shown how summarisation ability is related to model prediction accuracy (Borges and Levene, 2007a).

In this work we concentrate on the prediction problem, i.e. given a trail, it is the problem of how well can we predict the next link a user followed to complete the trail based on the model built from the navigation records within the site for a group of users. In particular, we focus on studying scoring metrics for evaluating the prediction accuracy of methods for solving the prediction problem.

As shown in Section 2 predicting user behavior on a web site has been extensively studied in the literature, especially in the context of web page personalization and web cache prefetching. Several methods have been proposed for modeling web usage data for predicting web page requests, however, most of these methods make use of the hit and miss score (often referred to as the hit ratio) to measure prediction accuracy. While some authors consider a prediction to be accurate if the requested page is the highest ranked prediction, other authors provide a set of predictions and let the hit and miss score measure the number of times the requested page is among the set of provided predictions. The latter case corresponds to a relaxation of the hit and miss score, being adequate for web cache prefetching, in which a set of predicted pages is pushed to the cache.

The hit and miss score is widely used but, in our opinion, is insufficient to deal with the specific requirements of the next step prediction problem. In particular, when using a Markov model to represent user sessions, the hit and miss score is not always sufficient (i) to assess the adequate order of the model; (ii) to determined the adequate size for a recommendation set of links; and (iii) to evaluate the model confidence on the predictions provided. Herein, we argue that although the hit and miss score provides a useful evaluation score, other metrics can be used to provide additional insight on how well a model is able to predict user behaviour. Thus, in addition to the Hit and Miss score, we investigate three other scoring metrics for evaluating web next page request prediction algorithms: the Mean Absolute Error (Witten and Frank, 2005), the Ignorance Score (also known as the information score) (Roulston and Smith, 2002) and the Brier Score (Roulston and Smith, 2002). Experimental results in Section 5 provide evidence of the usefulness of the three alternative scores in the prediction context.

We are also interested in assessing unexpected or surprising events (McGarry, 2005; Geng and Hamilton, 2006), since such rare events are not predictable. Although not predictable, unexpected clicks can be detected, by the fact that their probability of occurrence is low. The approach we take is to label a click as unexpected if its probability is less or equal to some threshold. Equivalently we can say that an event is expected if its probability of occurrence is greater than the threshold. We observe that if the threshold is zero then unexpected events are the ones that have not yet occurred in the log data. A significant application for the detection of unexpected events is that of patrolling the web for security purposes (Chen, 2006).

The rest of the paper is organised as follows. In Section 2 we review related work and in Section 3 we introduce the variable length Markov chain model we make use of. In Section 4 we describe the prediction problem and present four different methods of

evaluating the prediction results. In Section 5 we present an experimental evaluation of the prediction algorithm and of the four scoring metrics, and, finally, in Section 6 we give our concluding remarks.

2. Related work

Several methods have been proposed to model user sessions for predicting the next web page request. We will now review recent work on such methods, stressing that authors mainly use the hit ratio (or equivalently the hit and miss score) to evaluate the prediction precision. For methods that provide a set of predictions rather than a single one, the hit ratio is normally relaxed to measure the proportion of times the requested page is among the set of predicted pages. In some cases, authors make use of additional metrics to measure the proportion of times the method was able to provide a prediction, however, we note that the aim of such additional metrics is not to compete with the hit ratio.

In (Su et al., 2000) a system to predict web requests is proposed, which organises multiple high-order n-gram models in a step-wise manner; the precision of the system is measured by the hit ratio and a score called the applicability score is used to measure the proportion of times the system was able to provide a prediction. In (Frias-Martinez and Karamcheti, 2002) the authors propose a method for predicting sequences of user accesses that induces association rules in a way that captures the sequential and temporal manner in which web pages are visited. The hit ratio is used to measure the proportion of times the requested page was among the set of predictions provided. In (Wu and Chen, 2002) the authors propose to organise the user sessions into a tree index structure that is used to predict user requests in a proxy prefetching mechanism. In the prediction performance evaluation, the hit ratio score is used to measure the proportion of successful predictions, a score called the service rate score is used to measure the proportion of times the method was able to provide a prediction, and a score called the contribution score is used to measure the ratio between the number of successful predictions and the test set size. Dongshan and Junyi (Dongshan and Juni, 2002) propose a hybrid-order tree-like Markov model, which provides good scalability and high coverage of the state-space, and is used to predict the next page access. The hit ratio is, again, used to measure prediction accuracy. In (Sen and Hansen, 2003) the authors evaluate several methods for predicting the user next access page by measuring the proportion of times the page that was ultimately requested was included in the list of predictions, which is a relaxed version of the hit ratio.

Chen and Zhang (Chen and Zhang, 2003) utilise a Prediction by Partial Match (PPM) forest that restricts the roots to popular nodes; assuming that most user sessions start at popular pages, the branches having a non-popular page as their root are pruned. The hit ratio is used to measure the number of times the requested page is in the cache. In (Gündüz and Özsu, 2003) a tree based model for web page prediction is given that uses both the sequence of visiting pages and the time spent on pages. A set of three pages is recommended and a hit is declared if the requested page is among the three. The hit ratio measures the proportion of hits and a score called the click-soon ratio is used to measure the proportion of times a prediction is requested during the entire active session. In (Géry and Haddad, 2003) a framework for a next page request recommender system is presented, as well as a comparison of association rules, sequential rules and generalised sequential rules methods in the context of prediction. Accuracy is evaluated by the proportion of times the requested page is the top ranked prediction (hit ratio), and by the proportion of times it is among a set of provided predictions (a relaxed version of the hit ratio). In (Bonino et al., 2003) the authors use an evolutionary algorithm to evolve a prediction model for web page requests from a population of finite-state machines. To evaluate the algorithm they measure the frequency with which the system is able to provide a prediction and the frequency with which these predictions are correct, i.e. the hit ratio.

Davison, (Davison, 2004), review several Markov-based methods used for predicting

4

web request patterns. Prediction accuracy is evaluated as the fraction of requests that were predicted correctly (the hit ratio) while varying the size of the set of predictions provided. Deshpande and Karypis (Deshpande and Karypis, 2004) propose a technique that builds $k^{th}-order$ Markov models and combines them to include the highest order model that is able to provide a prediction for each state; a technique to reduce the model complexity is also proposed therein. The accuracy of the model is defined as the number of correct predictions divided by the total number of predictions, i.e. the hit ratio is again used.

In (Abe, 2005) the author proposes a Markov chain method, which uses more than one transition matrix to takes into account higher-order probabilities. A model is built from the data corresponding to the users who visited the same pages as the viewer whose trail is being handled, and the hit ratio is used to measure the proportion of correct predictions.

Recently, in (Berka and Labský, 2007) an experimental evaluation of two methods for next web page prediction is presented. The first method builds a set of interpolated n-gram Markov models and the second uses rule a learning algorithm that maintains the order of page requests within the user sessions. For evaluating, the authors compute the number of correct guesses divided by the total number of guesses, which, again, corresponds to the hit ratio score. In (Makris et al., 2007), the authors propose to cluster similar user sessions into groups and to build a generalised weighted suffix tree to represent each cluster. As a prediction for a given trail, the method gives all pages on the outgoing edges of the matching suffix (limited to at most twenty predictions) and the evaluation measures the hit ratio and the click-soon ratio described above. In (Baraglia and Silvestri, 2007) the authors propose a solution for implementing web personalization as a single online module that provides a list of suggestions based on classification of the user session being processed. The evaluation measures the capacity of anticipating further users requests that will be made in the future. A user session is split into two halves and the provided score measures the intersection of the set of pages in the second half of the trail and the set of pages provided as predictions for the first half of the trail; this can be viewed as a session-based hit ratio score. Finally, (Mukhopadhyay et al., 2007) propose an agent based method for web page prediction that clusters related pages into different categories based on access patterns. A parameter sets the number of pages sent to the client cache, and the proportion of pages sent to the cache that were clicked by the user is computed as a measure for the prediction accuracy, that is, the relaxed hit ratio is used.

3. Web Mining with Variable Length Markov Chains

In previous work we have proposed the use of Markov models to represent a collection of user sessions. A first-order Markov model (Kemeny and Snell, 1960) provides a compact way of representing a collection of sessions but, in most cases, its accuracy is low. A VLMC is a Markov model extension that allows variable length navigation history to be captured. We have proposed in (Borges and Levene, 2005) a method that transforms a first-order model into a VLMC so that each transition probability between two states takes into account the path a user followed to reach the anchor state.

We now briefly review our VLMC construction method. Consider the collection of sessions and the corresponding first-order model given in Figure 1. Each web page corresponds to a state in the model (from now on we use the terms state and page interchangeably). In addition, there is an artificial start state (S) and an artificial final state (F) appended to every session. The first-order model is incrementally built by processing each sequence of page requests. A transition probability is estimated by the ratio of the number of times the transition was traversed and the number of times the anchor state was visited. Next to a link, the first number gives the number of times the link was traversed and the number in parentheses gives its estimated probability.

The model accuracy can be assessed by comparing a transition probability with the

corresponding higher-order probability estimated by the n-gram frequency counts. For example, according to the input data the conditional probability of going to state A_3 after following link (A_1, A_2) is given by the number of times users followed (A_1, A_2, A_3) divided by the number of times users followed (A_1, A_2), that is $3/6 = 0.5$. The first-order model is not accurate since $p(A_2, A_3) = 0.30$.

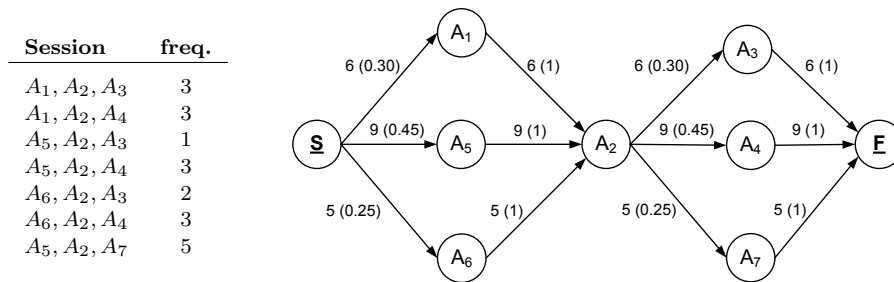| Session	freq.
A_1, A_2, A_3	3
A_1, A_2, A_4	3
A_5, A_2, A_3	1
A_5, A_2, A_4	3
A_6, A_2, A_3	2
A_6, A_2, A_4	3
A_5, A_2, A_7	5

Fig. 1. A collection of user navigation sessions and the corresponding first-order model

Our model extension that incorporates higher-order probabilities is based on a cloning operation that duplicates states whose outlink transition probabilities are not accurate. In the example, state A_2 is cloned in order to separate in-paths to it that induce distinct conditional probabilities. In order to identify in-paths inducing similar conditional probabilities a clustering method is used, and there is also a parameter used to set the intended accuracy. Figure 2 shows the resulting second-order model when a 5% deviation is allowed between the conditional probability induced by the n-gram counts and the transition probability given by the model. For example, as said above according to the input data, the probability of going to state A_3 after following link (A_1, A_2) is 0.5 and according to the second-order model such probability is 0.45. Thus the model represents such probability with an error of 5%, in case higher precision is required we need to increase the number of clones. Since the conditional probabilities induced from the paths A_1 and A_6 are closer, they were assigned to the same clone. The transition counts of the updated model are computed in a way that reflects the number of times each path was followed. Although the example corresponds to the evaluation of a second-order model, the method was generalised for higher-orders; see (Borges and Levene, 2005) for the detail.

Modelling a collection of sessions in a VLMC has the advantage of providing a platform that (i) can identify the most popular trails, which are defined as the higher probability ones, and (ii) can predict the user's next navigation step after following a given trail.

4. Prediction and Scoring Rules

Given a trail, the prediction problem is the task of predicting the next link a user will follow given the previous viewed pages. That is, by observing the clicks that lead to the given page on the trail, i.e. the clickstream history, the aim is to predict the next page the user will most likely visit. The prediction algorithm we use is simply to choose the highest probability link, given that the user has inspected n web pages (states) before reaching the current link on the trail, where $n \geq 1$; this prediction method is known as maximum likelihood.

We now give a simple example that will be used to illustrate the computation of the scores. Consider the first-order model given in Figure 1 and two test trails, (A_1, A_2, A_4)
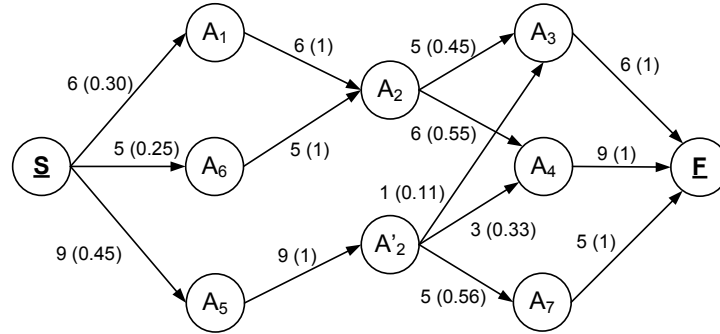
6

Fig. 2. The second order model corresponding to the model given in Figure 1

and (A_5, A_2, A_7). For the first trail, the task is to predict the next navigation step after having followed (A_1, A_2) and for the second test trail to predict the choice after having followed (A_5, A_2). Since a first-order model only takes into consideration the last page viewed, i.e. A_2, it gives the same transition probability estimates for the states following the two test trails, i.e. $p_3 = 0.30$, $p_4 = 0.45$ and $p_7 = 0.25$; thus, the maximum likelihood estimate will be A_4 in both cases. Now, consider the second-order model given in Figure 2 that predicts reaching states A_3 and A_4, respectively, with probabilities $p_3 = 0.45$ and $p_4 = 0.55$ for the first trail, and states A_3, A_4 and A_7, respectively, with probabilities $p_3 = 0.11$, $p_4 = 0.33$ and $p_7 = 0.56$ for the second trail. Thus the maximum likelihood would predict A_4 in the first case and A_7 in the second.

In the following subsections we define the four different scoring metrics we utilise for evaluating the prediction algorithm and illustrate the computation of the scores using the above example.

4.1. *Hit and Miss*

The first method is the <u>hit and miss</u> scoring rule (HM) (Witten and Frank, 2005), which counts a correct prediction, i.e. a hit, as 1 and an incorrect prediction, i.e. a miss, as 0. HM can be interpreted as the probability of guessing that the link followed was the one with the maximum probability, and is thus equal to the expected maximum likelihood probability of the next link on the trail; we denote this expected probability by MTP. We note that, as discussed in Section 2, some researchers make use of a relaxed version of this score in which a hit is counted when the requested page is among the top-n predictions.

In the above example, for the first test trail HM records a correct prediction in both the first and second order models, since A_4 is the top ranked destination state. For the second trail HM records a miss for the first-order model and a hit for the second-order one. Thus, the resulting first-order HM score is $(1 + 0)/2 = 0.50$ and the resulting second-order HM score is $(1 + 1)/2 = 1.00$.

4.2. *Mean Absolute Error*

The second method is the <u>mean absolute error</u> (MAE) (Witten and Frank, 2005), which ranks the links from 1 to r, where the rth link was the one that was followed, and records the MAE score for an individual prediction as $r - 1$. MAE can be interpreted as the expected rank, minus one, of the link on the trail that was followed. We note that as opposed to HM, a lower MAE score correspond to a more accurate prediction. The

MAE metric is commonly used to measure the accuracy of a recommender system, and in particular the average absolute deviation between the predicted rating and the user's true rating, see (Herlocker et al., 2004).

So, for example, if on average, the user clicks on the link that was ranked 3rd the MAE will be 2. This score is useful for determining the size of a recommendation set to be presented to the user, which aims to achieve high accuracy in matching users interests.

In the above example, the first-order MAE score is $(0 + 2)/2 = 1.00$, since the last link of the first trail is the top ranked among the possible predictions, and the last link of the second trail is only the 3rd in the ranking. The second-order MAE score is $(0 + 0)/2 = 0.00$, since the last link of both trails is the top ranked among the possible second-order predictions.

4.3. *Ignorance Score*

The third method is the ignorance score (IS) (Roulston and Smith, 2002), which records the score as $-\log_2(p)$, where p is the probability of the link that was followed by the user. The ignorance score has an information-theoretic interpretation as the entropy of an event (Roulston and Smith, 2002), and $p = 1/(2^{IS})$ can be interpreted as the expected probability of the link on the trail that was followed by the user; we denote this expected probability by ATP. As opposed to HM and MAE, IS is a non-linear scoring function, ranging from 0, when $p = 1$, to infinity, when $p = 0$ (when $p = 0.5$, the IS is equal to 1). It follows that the ignorance score results in a large penalty when the user follows a link whose probability of occurrence is very low. We note that as opposed to HM but in similarity to MAE, a lower IS score correspond to a more accurate prediction.

In difference to the first two scores, IS takes into account the strength of a prediction, measured by the estimated probability. In fact, although the first test trail corresponds to a hit in both models, the second-order model provides a higher probability estimate for the last trail link. Thus, while the first-order model gives IS=$-log_2(0.45) = 1.15$ the second-order model gives IS=$-log_2(0.55) = 0.86$. For the second test trail the first-order prediction gives IS=$-log_2(0.25) = 2.00$, while the second-order prediction gives IS=$-log_2(0.56) = 0.84$.

4.4. *Brier Score*

Finally, the fourth scoring method we use is the Brier score (BS) (Roulston and Smith, 2002), which is defined as the sum of the squared deviation between the predicted probabilities and the observed outcome. For a prediction having m possible link choices BS is given by,

$$\frac{1}{m} \sum_{i=1}^{m} (p_i - \delta_i)^2,$$

where p_i is the estimated probability for link i, $\delta_i = 1$ if i was the link followed and $\delta_i = 0$ otherwise.

We observe that BS expresses the difference between the predicted probabilities and the observed event. Not only does it measure the deviation from 1 of the probability assigned to the observed outcome, but it also takes into account the way the remaining probability is distributed among the other possible outcomes. For example, for an event with three possible outcomes in which the observed outcome is o_1) the prediction ($o_1 = 0.5, o_2 = 0.25, o_3 = 0.25$) gives a better score than the prediction ($o_1 = 0.5, o_2 = 0.49, o_3 = 0.01$). In general, a good BS score is obtained when a relatively high probability is assigned to the observed outcome and the other possible outcomes are not close competitors for the top ranked place. We note that in comparison IS, defined Section 4.3, takes into account only the probability estimate of the observed outcome. Moreover, as opposed to HM but in similarity to MAE and IS, a lower BS score correspond to a more accurate prediction.

8

In order to facilitate its interpretation, we normalise BS by dividing it by the corresponding worst case value, that corresponds to the situation in which the last link on the trail is not among the m choices. The reason for this normalisation is that the predictions to be evaluated may have a high variation with respect to the number of outcomes. Thus, we consider an additional link to represent the worst case, whose predicted probability is $p_{m+1} = 0$. Thus the normalised version of BS is given by, $\sum_{i=1}^{m}(p_i - \delta_i)^2/(\sum_{i=1}^{m} p_i^2 + 1)$.

In the above example, the computation of this score for the first trail is given by $((0.30 - 0)^2 + (0.45 - 1)^2 + (0.25 - 0)^2) = 0.46$ divided by the corresponding worst case $((0.30)^2 + (0.45)^2 + (0.25)^2 + 1) = 1.36$, which results in BS= $0.46/1.36 = 0.34$. On the other hand, the second-order model gives $((0.55 - 1)^2 + (0.45 - 0)^2)$ divided by $((0.55)^2 + (0.45)^2 + 1)$, which yields BS=$0.41/1.41 = 0.29$. For the second test trail the first-order BS is $0.86/1.36 = 0.63$ and the second-order score yields BS=$0.31/1.43 = 0.22$.

4.5. *Interpretation of the Scores*

In this subsection we will discuss the interpretation of the four scores and the utility of having a set of complementary measures for assessing prediction accuracy.

We would like to stress that, while the first two scores (HM and MAE) take into consideration the rank among the possible outcomes of the followed link, the latter two scores (IS and BS) measure the strength of the prediction by the assessing the actual probability assigned to the observed outcome. Moreover, while we wish to maximise HM, the other scores should be minimised.

The HM score provides a simple and intuitive measure of accuracy. For a system providing a rank ordered set of link suggestions on each web page, the HM corresponds to the probability of users following the top ranked prediction. In Table 1 we summarise the results for the above example, where it can be seen that the HM score reveals that, overall, the second order model is more accurate.

One limitation of the HM score when assessing recommender systems is the restrictive way in which it evaluates user choices. When browsing the web, although there are paths that are more likely to be followed than others, it is not always that case that a single link dominates users' choices when navigating. Thus, we do not expect the top ranked prediction to dominate and always be followed by the users. The MAE score, on the other hand, measures the average rank of the observed users' choices among the suggested web pages. Such a score is very useful for determining the size of a recommendation set to be presented to users aiming to achieve a high probability of including the pages that users are actually interested in. Thus, the MAE score provides this additional information relative to the HM score.

In the above example, for the first-order predictions, HM tells us that on average 50% of the highest probability links were hits, while the MAE tells us that on average we should recommend to the user the top-2 ranked links rather than just the highest ranked one. For the second-order predictions, in this particular case, the two scores provide identical insight, since HM = 1 corresponds to MAE = 0.

Instead of evaluating the rank of the user's choice, the IS measures the accuracy as the logarithm of the probability assigned to the user's choice. Thus, the score depends only on the probability assigned to the occurring outcome. The IS is non-linear and strongly penalises a model that assigns a very low probability to the actual outcome. Intuitively, this means that assigning a high probability to an event, which in reality, has a very low probability implies low predictive power of the model for this event.

In the above example, it is interesting to note that for the first-order predictions the IS score is much higher (75% higher) for the second trail while the estimated probability for the target only differs by 10%. Moreover, the two previous scores did not detect the improvement in probability in the prediction for the first trail from 0.45 to 0.55, as does IS. Thus, IS provides additional insight on the quality of the prediction.

Finally, BS takes into account not only the probability assigned to the occurring outcome but also the way in which the remaining probability is distributed among the

other possible outcomes. For the first trail it is interesting to note that although the probability of the target increases from the first to the second-order model, the probability of the second ranked outcome has also increased, and thus the improvement of the score is not significant. For the second trail there is a higher gain when moving from the first to the second order model, since not only does the probability of the target increase but also the target is more distinct from the other outcomes. Thus, the BS score is useful for detecting situations in which there are one or more competing links for the top rank prediction.

Trail	Target	FO		SO		HM		MAE		IS		BS	
		page	prob	page	prob	FO	SO	FO	SO	FO	SO	FO	SO
$A_1, A2$	A_4	A_3	0.30	A_3	0.45	1	1	0	0	1.15	0.86	0.34	0.29
		$\underline{A_4}$	0.45	$\underline{A_4}$	0.55								
		A_7	0.25										
A_5, A_2	A_7	A_3	0.30	A_3	0.11	0	1	2	0	2	0.84	0.63	0.22
		A_4	0.45	A_4	0.33								
		$\underline{A_7}$	0.35	$\underline{A_7}$	0.56								
						0.5	1	1	0	1.57	0.85	0.48	0.25

Table 1. Scores for the example

4.6. *Unexpected Events*

If the user follows a link whose probability of occurrence is very low, or even zero when it is followed for the first time, it may be considered as being <u>unexpected</u> from the point of view of the Markov model. Such unexpected or surprising events (McGarry, 2005; Geng and Hamilton, 2006) are not predictable, since they are unlikely to occur according to the constructed VLMC model. However, although not predictable, unexpected user clicks can be detected, by the fact that their probability of occurrence is low.

The approach we take here is to label a click as unexpected if its probability is less or equal to some threshold, α. Equivalently we can say that an event is expected if its probability of occurrence is greater than the threshold, α. We observe that if the threshold is less than zero, then there are no unexpected events (in the following we use `all` to denote a value of α which is less than 0). We also observe that when $\alpha = 0$, the only unexpected events are the ones whose probability is zero, i.e. representing links that do not occur in the Markov model.

When $\alpha = $ `all`, i.e. $\alpha < 0$ we apply a form of Laplace smoothing (Zhai and Lafferty, 2004) to the model. In particular, for a each state, we set the probability, p_i, of the ith outlink from the state as

$$p_i = \frac{w_i + d}{W + md},$$

where m is the number of links that can be followed from the state, w_i is the number of times the ith outlink was traversed, $W = \sum_{i=1}^m w_i$, and d is an initial small positive weight assigned to each outlink; the value of $d = 0.001$ used in our experiments was determined by means of a pilot test. We note than when $d = 0$ Laplace smoothing is turned off.

5. Experimental Evaluation

5.1. *Description of Data Sets*

For the experimental evaluation we make use of three distinct data sets. The first data set (LTM) represents four months of usage from the London Transport Museum between

10

November 2002 and February 2003. Erroneous requests, image requests and all requests from IP addresses that correspond to identified web spiders were eliminated. A session was defined as a sequence of requests from the same IP address with a time limit of 30 minutes between consecutive requests. After sessionising, sessions with a single request were removed. For testing, we decided to use a temporal-based natural split, thus, the training set is composed from the first three months of data and the test set from the last month of data.

The second data set (PKDD) corresponds to the ECML/PKDD 2005 challenge and contains server sessions from seven e-commerce vendors (Berka, 2005). The data contains a session generated ID, and a session was thus defined as a sequence of requests with the same ID for a given vendor. For testing, we randomly split the induced sessions into a 70/30 training set and test set split.

The third data set (MSWEB) was obtained from the UCI KDD archive (`http://kdd.ics.uci.edu/databases/msweb/msweb.html`) and records the areas within `www.microsoft.com` that users visited in a one-week time frame during February 1998. Two separate data sets are provided, a training set and a test set.

Table 2 summarises the characteristics of the data sets. For each data set we indicate the number of pages occurring in the log file, the total number of requests recorded, the total number of sessions derived from each data set and the corresponding average session length (ASL). One question often raised is whether the contents of a test set is representative of the corresponding training set. The results show that the average session length in the training sets is very close to the corresponding values in the test sets. We note that not all pages in a given training set are in the corresponding test set, meaning that test sets do not cover all the web pages that are present in the training sets. A finer analysis reveals that there are 242 pages (12%) in the LTM test set that do no occur in the corresponding training set; for the PKDD data set there are 31 (12%) such pages, and for the MSWEB there is just one such page (0.3%). This suggests that taking into account concept drift (Koychev, 2004) when building a model may be beneficial in cases were the estimated probabilities are not stationary.

	Training set				Test Set			
	Pages	Requests	Sessions	ASL	Pages	Requests	Sessions	ASL
LTM	2438	792886	31953	13.6	2048	338322	13695	13.8
PKDD	316	456786	60288	5.4	263	188922	25837	5.3
MSWEB	285	98654	32711	3.0	236	15191	5000	3.0

Table 2. Summary characteristics of the three real data sets used (we let ASL be average session length)

To further compare the training and test sets, we computed, for each data set, the n-grams from both the training and the test sets. For a given n, we ordered the resulting n-grams by their frequency of occurrence in the corresponding set and compared the two rankings by means of the Spearman footrule with a location parameter (Fagin et al., 2003), which measures the distance between two top-m lists. The footrule metric is defined as follows. Given two top-m lists, L_1 and L_2, each with m elements, we let L be the union of the two lists and the location parameter be $m + 1$. In addition, we let $f(i)$ be a function that returns the ranking of any element $i \in L$ in L_1, and when $i \notin L_i$ we let $f(i) = m + 1$; we let $g(i)$ be the equivalent function for L_2. The footrule metric is now defined as:

$$F(L_1, L_2) = \frac{\sum_{i \in L} |f(i) - g(i)|}{MAX},$$

where MAX is a normalisation constant, which for a top-m list is $m(m+1)$ corresponding to the case when there is no overlap between the two lists.

Table 3 presents the results of the Spearman footrule analysis; the last column corresponds to the rankings including n-grams with $n = 2, \ldots, 5$. The results show that frequent n-grams in the MSWEB test set differ from the frequent n-grams in the corresponding training set, thus, suggesting that prediction will be harder for that data set. Also, for MSWEB the footrule values increase sharply for higher values of n, indicating that prediction from higher order models do not seem promising. This fact is consistent with values given in Table 2 showing that on average the trails from the MSWEB site are short (ASL=3.0). The variation of the footrule for PKDD data set is the smallest, and the footrule for the PKDD and LTM data sets are very close when the rankings include n-grams of varying length.

	2-grams	3-grams	4-grams	5-grams	(2-5)-grams
LTM	0.097	0.111	0.148	0.207	0.069
PKDD	0.145	0.131	0.156	0.178	0.070
MSWEB	0.234	0.335	0.515	0.739	0.202

Table 3. The Spearman footrule as a measure of the distance between the top-2000 n-grams induced from each of the training and test sets.

As mentioned in Section 4, a model is induced from the training set and the trails in the test set are then used to assess the model's predictive ability, i.e. for a given test trail its clickstream history is used to predict the trail's last state. When predicting, an n-order VLMC model makes use of the $n-1$ links prior to the last link in the clickstream history. For example, a first order model provides predictions based only on the last page visited. Thus, on the one hand, for very long trails some of the navigation information is not used, and on the other hand, it is not expected that very long trails are traversed very often by users. The trail length distribution for the data sets used is shown in Figure 3. While the LTM data set has the highest number of requests, it has the smallest number of sessions in the training set (see Table 2), since a significant number of sessions are long.
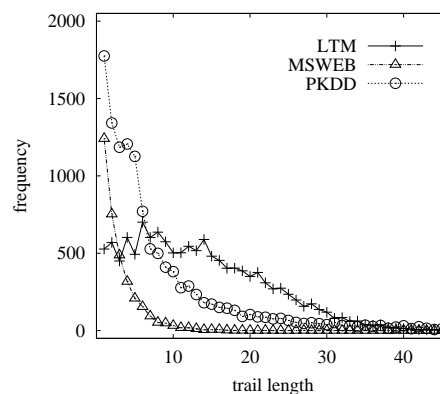

Fig. 3. The distribution of trail length in test sets for the three data sets used

Another aspect that effects the prediction ability is the number of choices from the anchor state of the trail's last link. This can be measured by the model branching factor,

12

BF. For a model having $i = 1, 2, \ldots, s$ states and each state i having u_i outlinks we define BF by

$$BF = \frac{1}{s} \sum_{i=1}^{s} u_i.$$

In addition, we define the weighted branching factor wBF to measure the average number of outlinks weighted by the number of times the anchor state was visited. Although BF is related with the model complexity, wBF gives a clearer indication of the difficulty associated with predicting a user next navigation step when viewing a given state because it takes into account the estimated probability of each state being visited. We let w_i be the number of times state i was visited, $W = \sum_{i=1}^{s} w_i$ and define wBF as follows:

$$wBF = \frac{1}{W} \sum_{i=1}^{s} w_i u_i.$$

In Figure 4 we show how BF and wBF vary with the order of the model. It can be seen that wBF is, on average, larger than BF by a factor of over 10, indicating that, although for higher order models BF is, on average, less than 5, states with a higher probability of being visited have a much larger number of outlinks and thus the choice of which link to predict appears to be quite a difficult task. It is interesting to note that while the MSWEB data set has a larger BF, the LTM data set has a larger wBF value, especially for lower order models. This means that for LTM, states with a large number of outlinks have high probability of occurring on test trails, making prediction harder. In particular, as given by the induced site topology, the average number of outlinks including repetitions from the anchor state of the test trails' last link is 198.5 for LTM, 83.9 for MSWEB and 90.7 for PKDD. (Note that these values do not correspond to the wBF since a given anchor state may occur more than once in the test set of trails while other states may not be represented in the test set.) The large values for wBF in comparison to BF indicate a skewness in the usage data implying that most user tend to visit pages with a high number of outlinks.
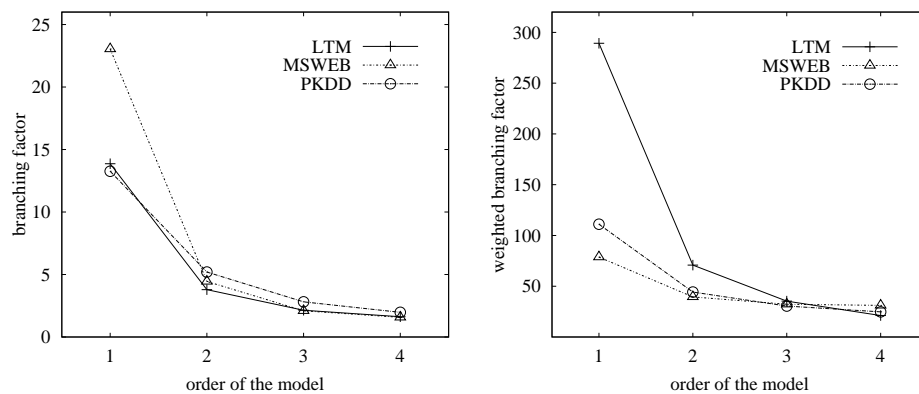

Fig. 4. Branching factor (left) and Weighted branching factor (right)

5.2. *Unexpected Events in the Data Sets*

In the scoring metrics results analysis we will take into account unexpected events as introduced in Section 4.6, where an occurring event is labelled as unexpected if its

predicted probability is below a user defined threshold, α. We would like to stress that, when a VLMC model makes a prediction that corresponds to a zero probability in the test trail's last link it implies that, according to the training data set, no other user in the past has followed the link after traversing the trail defined by the clickstream history. In case the predicted probability is above zero but below α it is implies that the chosen link occurred in the past but in a very small percentage of the trails. Since unexpected clicks are not predictable, in the following we will evaluate the scoring metrics for both cases when including and excluding these events.

Figure 5 shows an approximate linear decrease in the percentage of trails in the test sets as α is increased. (We note that although the α values on the x-axis in the figure are not proportionally spaced, when adjusting the data points appropriately, linear regression shows a good fit for the three data sets.) Overall, LTM and MSWEB have more unexpected events than PKDD, and for $\alpha > 0.01$ the increase in the number of unexpected events is sharper for MSWEB than for LTM. This could account for the links of MSWEB being less predictable than the other two data sets, as discussed below. We note that increasing α by too much is not desirable, since as seen from the trend in Figure 5, this will have an adverse effect on the size of the test set.
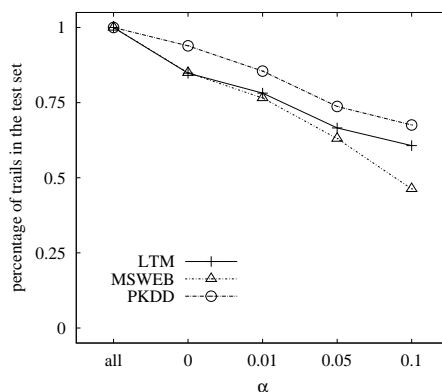

Fig. 5. Average percentage of expected events in the test data

5.3. *Scoring rules*

In Figures 6, 7, 8 and 9 we show for the three data sets how HM, MAE, IS and BS vary, respectively, with the order of the model for varying values of α. The general pattern for all scoring metrics is that the scores improve as the order of the model increases and as α increases. By increasing the order of the model we are able to make use of the history to limit the choices of clicks that users have made, and by increasing α, until it reaches a predefined level, we are able to eliminate unexpected events which make the prediction more difficult.

HM Scoring Rule

By inspecting Figure 6 we see that for LTM when α is `all`, the HM score does not increase much (from first to second-order it increases from 0.55 to 0.57). For the same data set the results improve significantly when α increases; HM is > 0.9 when $\alpha > 0.1$. Therefore, for the LTM data set there is a set of trails for which it is very hard to predict

14

the precise outcome due to the improvement in the score when these trails are filtered out.

For the other two data sets when α is `all`, the HM score increases when moving from a first to a higher order model. In addition, the HM increase with α is less evident for these two data sets than with the LTM data set (the plotted lines are less spread out in the former cases). For the PKDD data set the highest value for HM is 0.72 and for MSWEB it is 0.63 (both results are for fourth-order models with $\alpha > 0.10$).

In summary, it can be seen that for HM, the LTM data set corresponds to the best level of performance, then the PKDD data set, and the worst level of performance is for the MSWEB dataset. When α is greater than 0, and unexpected events are filtered out, we can reach a prediction level of above 0.75 for LTM, between 0.5 and 0.75 for PKDD, while for MSWEB only when $\alpha = 0.10$ can we reach a level above 0.5. (We note that 0.5 is still much better than a uniform guess as, in general, there are many more than two links to choose from.)

The hit and miss score is adequate for assessing the prediction accuracy of a model when the aim is to provide a single suggestion. In such context, and without removing unexpected events, the results indicate that for the LTM web site a first-order model gives a 55% hit ratio, performing almost as well as the corresponding higher order models. Thus a first-order model may be recommended in this case. When providing a single prediction for the PKDD web site, a second-order model may be recommended, which gives a 47% hit ratio, and for the MSWEB web site a third-order model may be recommended, which gives a 29% hit ratio. It is evident that in each of the three cases the accuracy is much higher than that expected when using a prediction method based on uniform random choice (see Figure 4 for the average number of choices).
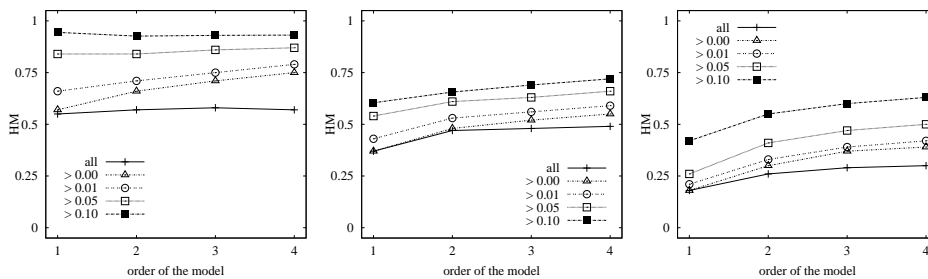

Fig. 6. Hit and miss score for LTM (left), PKDD (middle) and MSWEB (right)

MAE Scoring Rule

As mentioned in Section 4.2 the MAE score gives the average ranking of the observed outcome among the predictions provided by a model. Thus, it can be a useful method for determining the size of a recommendation set to be presented to users during a navigation session.

For the examined data sets, it can be seen in Figure 7, that the MAE score improves with the model order when α is `all` or 0, but when α is greater than 0 the distinction, although present, is less evident. Assuming that $\alpha > 0$, we can conclude that, on average, the link we are trying to predict is in the top-3 most probable links. An interesting fact to note regarding the LTM data set is that when α is `all`, HM does not improve significantly with the model order but MAE improves significantly. (Note that for HM improvement is synonymous with a higher score, while for MAE improvement implies a lower score.) This implies that, although the higher order models are not able to predict the exact

link that was followed, they are better able to place it in a higher rank relative to the other possible links.

For MAE the best level of performance is for PKDD followed by MSWEB; for $\alpha \geq 0.05$ the results are very similar. The comparatively good level of performance for MSWEB can probably be explained by the its lower wBF as seen in Figure 4.

In the context of configuring a next link prediction model, the MAE score is useful for determining the size of the recommendation set presented to users that is necessary to achieve high accuracy. If we set as a criterion, choosing the order of a model that gives high accuracy with a set of 5 recommendations, then we may suggest using a second-order model for the PKDD site ($MAE = 3.33$, which is very impressive considering that $wBF = 44$), a fourth-order model for the LTM ($MAE = 4.44$) and a third-order model for the MSWEB site ($MAE = 4.44$). If we set as the criterion that moving to higher order models is worth the computational effort if the corresponding score improves by 10% or more, then the results imply that a fourth-order model should be used for LTM ($MAE = 4.44$), and a third-order model used to both PKDD ($MAE = 2.86$) and MSWEB ($MAE = 4.44$).
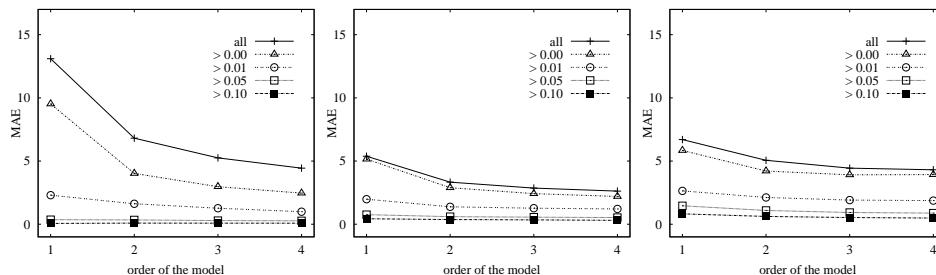

Fig. 7.   Mean absolute error for LTM (left), PKDD (middle) and MSWEB (right)

IS Scoring Rule

As shown in Figure 8, for IS when $\alpha > 0$ the results are consistent with HM, that is, the LTM data set corresponds to the best level of performance followed by PKDD and MSWEB. (Note that for IS, for clarity, we plot $p = 1/2^{IS}$ against the order of the model, so improvement is synonymous with a higher score.)

The results for the IS scoring rule show that there are significant differences between the three data sets with respect to the expected probability of the link that we are trying to predict. As we can see for LTM the probability 2^{-IS} is able to reach levels above 0.5, when $\alpha \geq 0.05$, for PKDD the probability levels are all below 0.5, while for MSWEB they barely reach 0.25. Thus, although, as seen in Figure 4, the branching factors for the data sets are comparable, the probabilities of the links we are trying to predict are generally lower in MSWEB than in PKDD, and lower in PKDD than in LTM. This is also shown in Figure 10, where it can be seen that the difference between MTP (the average of the probabilities estimates for the top prediction) and ATP (the average of the probabilities estimated for the observed outcome) is larger for MSWEB than for PKDD and larger for PKDD than for LTM.

We note that the results shown in Figure 10 have been averaged. A detailed analysis of the LTM data set reveals that when moving from a second-order model to a third-order model the estimated probability of the target state increases by 3%, on average, for 37% of the test trails, remains unchanged for 33%, while it decreases by 2%, on average, for the rest of the test trails. When moving from a third to a fourth-order model, for

16

54% of the test trails the estimated probability of the target state remains unchanged, for 25% it increases by 2%, on average, and for the rest of the test trails it decreases by 1%, on average. The overall average results reveal that, for LTM, the ATP does not increase significantly beyond the second-order model.

The IS score provides additional insight into the HM and MAE scores. Consider, for example, the LTM data set; when α is `all` and the model changes from first to second order the HM score goes from 0.55 to 0.57, corresponding to a 4% relative increase while the 2^{-IS} goes from 0.084 to 0.076, corresponding to a 10% relative improvement. This means that while the number of hits is close to constant (see Figure 6), and the average probability estimate for the observed outcomes slightly increases (see Figure 10) there are a number of predictions for which the observed outcome was among the reachable states in lower order models but were missed when the order of the model increased. This is an effect of a decrease in coverage when the order of the model increases, which is amplified by the logarithmic nature of the score. In fact, a more detailed analysis of the results reveals that the average IS score is strongly affected by the misses due to its non-linear nature; when we separate the hits from the misses the overall measure increases with the order of the model for hits and decreases for misses. When all events are included, the overall ignorance score is strongly penalised by link choices that never occurred in the past. As mentioned in Section 4.6 we apply Laplace smoothing in order to deal with predictions with $p_i = 0.0$, due to the logarithmic nature of IS such events have a very substantial impact on the score. Moreover, the number of unexpected events increases with the model's order, due to the fact that when a long clickstream history is analysed the number of choices to follow is generally much less than otherwise. We further note that, for a given test trail, the set of predictions made by a second-order model is a subset of the corresponding first-order model predictions. Thus, links that are among the first-order predictions having a low probability estimate may be left out of the second-order predictions and become unexpected events.

In summary, IS is useful for analysing the impact of misses on the overall model performance that is, it provides a measure of the model coverage. When a model assigns probability zero to an outcome it is a very strong statement implying that the such event has no chance of occurring. Thus, the IS score is complementary to the previous scores which as it provides a measure of how low are the probabilities of the misses.
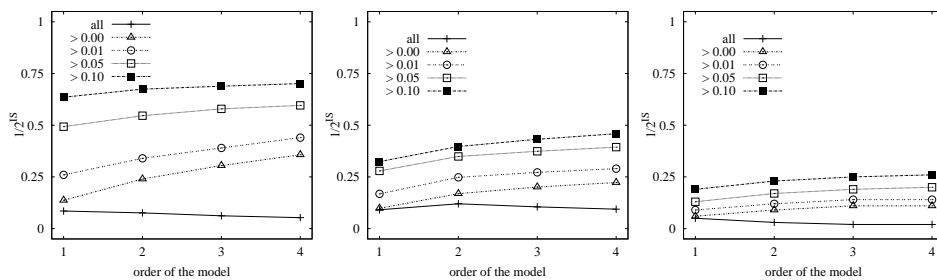

Fig. 8. Ignorance score (represented by the expected probability of the predicted link) for LTM (left), PKDD (middle) and MSWEB (right)

BS Scoring Rule

For BS the results are consistent with HM and IS. We note that BS takes into account the probability attributed to the trail's last link (as does IS) but is also affected by the probabilities assigned to the other links that may be followed. As mentioned in

Section 4.4 BS improves when the probability of the observed outcome increases while it penalises situation when there are other outcomes having relatively high probabilities. This is important since the score measures how much is an outcome distinguishable from the other possible outcomes. We also note that BS is less penalising than IS for the predictions with $p_i = 0.0$, since when α is `all` the score also improves with the order of the model, although very slightly.

When comparing the results from Figures 8 and 9 it is interesting to note that when α is `all`, the data set showing the best level of performance according to IS is PKDD but according to BS it is LTM. The interpretation of this is that although the probability estimate for the observed outcome for LTM is in general lower than for PKDD, on the LTM data set the predictions are more distinguishable, since there are fewer competing predictions. Also, for PKDD the predictions become more distinguishable with the order of the model.

In summary, BS is useful, since not only does it takes into account the probability estimates for the observed outcomes but it also measures how much the probability estimated for the observed outcome is distinguishable from the other outcomes, which can be seen as a measure of the model confidence on the provided prediction.
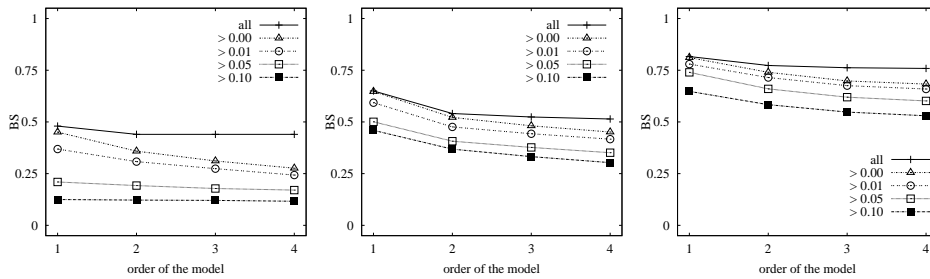

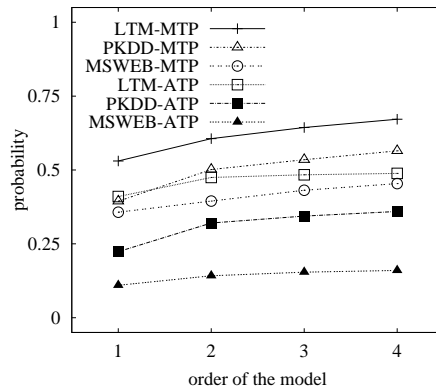Fig. 9. Brier score for LTM (left), PKDD (middle) and MSWEB (right)

Fig. 10. Average (ATP) and maximum (MTP) test trail probability

18

5.4. *Using Scoring Metrics to Configure a Prediction Model*

A model that is able to record an historical record of a user's navigation activity and make use of it to predict the user's next navigation step can be used, for example, in adaptive web site applications (Perkowitz and Etzioni, 2000) or in recommender system that provides a list of suggested navigation options that may be of interest to the user. As referred before, the hit and miss score has been the standard for the evaluation of next link prediction models. We argue that the hit and miss has limitations in the context; thus, a study of a collection of scoring metrics that could be useful in such scenarios is an important step in the requirements design of such applications.

We have presented an extensive evaluation of four scoring metrics for measuring the quality of predictions of users' next step when navigating on the web. While two of the scores measure the quality of the ranking of the predictions (HM and MAE) the other two scores (IS and BS) measure the actual probabilities assigned to the predicted event, thus enabling us to conduct a finer grained analysis of the probabilities assigned to the predictions, and to the links that were actually followed by the users. That is an important factor for determining the adequate order of a Markov model.

We illustrate the potential usefulness of a set of scores when tuning a prediction model for the LTM web site. When setting up a Markov model for predicting users' next web page request the following questions need to be addressed: (1) What should the order of the model be? (2) How many recommendations should be given to the user? (3) Is the user behaviour within the site predictable? (4) How confident is the model in the predictions provided? and (5) How is the prediction coverage of the model affected by its order?

As seen in Figure 6 the HM score is <u>not</u> significantly affected by the order of the model when α is `all`, which suggests that for this particular site a low order model would be sufficient. The MAE score provides additional insight with respect to the events in which the top ranked prediction does not correspond to the link chosen by the user. According to Figure 7 a third-order model is able to include the user link choice in the top five predictions. Thus, if the aim is to provide a set of recommendations, as opposed to a single recommendation, the MAE score indicates that a third-order model will achieve a good performance when providing a set of five recommendations to the user.

The analysis of Figure 8 provides further insight on the user behaviour within the site. In fact, the trend of decrease when α is `all` reveals that in spite of the overall better performance for higher order models according to MAE, when longer trails are considered, there are some predictions that complectly miss the target. Those predictions correspond to the unexpected (or unpredictable) events which, when identified, should prevent the system of proving the suggestion to the users. When such events are ignored the score shows that the probability estimate assigned to the link chosen by the user increases with the order of the model. Thus, while MAE reveals an increase in accuracy with the model order, the IS score reveals the decrease in prediction coverage with the model order. Finally, the analysis of Figure 9 also confirms that when the unpredictable events are discarded higher order models are able to provide stronger predictions, in the sense that the probability attached to the link chosen by the user increases relatively to the other competing links with the order of the model.

We stress that if, for the LTM data set, we restrict the analysis to the HM score a first or second order model would be considered adequate. However, if we utilise both the additional scores and the concept of unexpected events, a third order model in conjunction with a five-item recommendation set would be the adequate choice.

Overall, it seems that none of the scoring methods is definitive, but rather that their joint use as an analysis tool enables us to gain insight into users' navigation behaviour within a web site. As a summary, we provide guidelines for the use and interpretation of the four scores:

- <u>**HM**</u> measures the precision of a prediction model that provides a single prediction. It

can be used to assess the effectiveness of an <u>I am feeling lucky</u> Google-style navigation suggestion, since it determines the probability of a user following the top ranked suggested link.

- **<u>MAE</u>** measures the average rank among the predictions of the link followed by the user. It helps determine the size of a recommendation set necessary to achieve a high accuracy, for example, in a web site that provides a set of navigation suggestions in a sidebar.

- **<u>IS</u>** measures the strength of the probability estimate for the observed outcome by taking its logarithm, and can be understood in terms of the expected probability of the last link on a trail that was followed. Due to its logarithmic nature it helps us to assess the impact of misses that were assigned zero or low probability, proving the means to assess the variation of the model coverage when setting its parameters.

- **<u>BS</u>** measures the difference between the predicted probabilities and the observed outcomes. It is useful to assess the strength of the provided predictions (measured by their probability) and how much the probability of the observed outcome is distinguishable from the probabilities of the other outcomes. Thus, it provides a measure of the model confidence in the provided predictions.

6. Concluding Remarks

In this work we tackle the problem of predicting the next web page request of users' when navigating the web. Most previous research in the field has exclusively made use of the Hit and Miss score (HM) for evaluating prediction accuracy. We argue that the HM score has limitations in terms of evaluating the accuracy and therefore complementary scoring methods are necessary.

To alleviate this problem we have investigated three additional useful scoring metrics: the Mean Absolute Error (MAE), the Ignorance Score (IS) and the Brier Score (BS). As we have discussed the scoring metrics have different interpretations: (i) HM can be understood in terms of the expected maximum likelihood probability of the last link on a trail that was followed (MTP), (ii) MAE can be understood in terms of the expected rank, minus one, of the last link on a trail, (iii) IS can be understood in terms of the expected probability of the last link on a trail (ATP), and (iv) BS measures the average deviation between predicted probabilities and the outcome of following the last link on a trail.

Our experiments show that the additional scores and the concept of unexpected events provide valuable insight when setting up a model for predicting the next link choice of a user based on other users navigation preferences. Therefore, we claim that the hit and miss score, although useful, is not sufficient for evaluating next link prediction models. In addition, the experimental results confirm that the prediction accuracy increases with the order of the model, and also increases when unexpected events (or unpredictable), controlled by a parameter, α, are being detected rather than being predicted. The experiments also show that the accuracy of prediction varies for different data sets.

Future work involves a better understanding of what makes a prediction algorithm such as maximum likelihood perform better on different data sets. A preliminary investigation taking into account <u>concept drift</u> (Koychev, 2004) when building the variable length Markov model over a long period was conducted and the results reported in (Borges and Levene, 2008). Finally, we also wish to apply the prediction algorithm to data sets from different applications areas such as patrolling the web.

References

Abe, M. (2005). A prediction model for web page transition. <u>International Journal of Electronic Business</u>, 3(3/4):378–391.

20 *REFERENCES*

Baraglia, R. and Silvestri, F. (2007). Dynamic personalization of web sites without user intervention. Communications of the ACM, 50(2):63–67.

Bejerano, G. (2004). Algorithms for variable length Markov chain modelling. Bioinformatics, 20:788–789.

Berka, P. (2005). Guide to the click-stream data. In Proceedings of the ECML/PKDD Discovery Challenge.

Berka, P. and Labský, M. (2007). Predicting page occurrence in a click-stream data: statistical and rule-based approach. In Proceedings of the 7th Industrial ICDM Conference, Advances in Data Mining. Theoretical Aspects and Applications, LNAI 4597, pages 135 – 147.

Bonino, D., Corno, F., and Squillero, G. (2003). Dynamic prediction of web requests. In Proceedings of the 2003 Congress on Evolutionary Computation, volume 3, pages 2034 – 2041.

Borges, J. and Levene, M. (2000). Data mining of user navigation patterns. In Masand, B. and Spiliopoulou, M., editors, Web Usage Analysis and User Profiling, LNAI 1836, pages 92–111. Springer-Verlag, Berlin.

Borges, J. and Levene, M. (2005). Generating dynamic higher-order Markov models in web usage mining. In Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pages 34–45, Porto, Portugal.

Borges, J. and Levene, M. (2007a). Evaluating variable length Markov chain models for analysis of user web navigation sessions. IEEE Transactions on Knowledge and Data Engineering, 19(4):441–452.

Borges, J. and Levene, M. (2007b). Testing the predictive power of variable history web usage. Soft Computing, 11(8):717–727.

Borges, J. and Levene, M. (2008). Detecting concept drift in web usage mining. In Proceeding of the Workshop on Web Mining and Web Usage Analysis (WEBKDD), pages 98–110.

Chen, H. (2006). Intelligence and security informatics: information systems perspective. Decision Support Systems, 41:555–559.

Chen, X. and Zhang, X. (2003). A popularity-based prediction model for web prefetching. IEEE Computer, 36(3):63–70.

Davison, B. (2004). Web dynamics: Adapting to change in content, size, topology and use. chapter Learning web request patterns, pages 435–460.

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

Dongshan, X. and Juni, S. (2002). A new Markov model for web access prediction. IEEE Computing in Science & Engineering, 4(6):34–39.

Fagin, R., Kumar, R., and Sivakumar, D. (2003). Comparing top k lists. SIAM Journal of Discrete Mathematics, 17(1):134–160.

Frias-Martinez, E. and Karamcheti, V. (2002). A prediction model for user access sequences. In Proceedings of Web Mining for Usage Patterns and User Profiles Workshop, (WEBKDD).

Geng, L. and Hamilton, H. (2006). Interestingness measures for data mining: A survey. ACM Computing Surveys, 38(3).

Géry, M. and Haddad, H. (2003). Evaluation of web usage mining approaches for user next request prediction. In Proceedings of the 5th ACM International Workshop on Web Information and Data Management (WIDM), pages 74–81. ACM.

Gündüz, Ş. and Özsu, M. (2003). A web page prediction model based on clickstream tree representation of user behavior. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 535–540.

Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1):5–53.

Kemeny, J. and Snell, J. (1960). Finite Markov Chains. D.V. Nostrand, Princeton, NJ.

Koychev, I. (2004). Experiments with two approaches for tracking drifting concepts. Serdica Journal of Computing, 1(1):27–44.

Makris, C., Panagis, Y., Theodoridis, E., and Tsakalidis, A. (2007). A web-page usage prediction scheme using weighted suffix trees. In Proceedings of the 14th International Symposium in String Processing and Information Retrieval, LNCS 4726, pages 242–253.

McGarry, K. (2005). A survey of interestingness measures for knowledge discovery. The Knowledge Engineering Review, 20:39–61.

Mobasher, B. (2007). The adaptive web - methods and strategies of web personalization. chapter Data mining for web personalization, pages 90 – 135. Springer.

Mukhopadhyay, D., Mishra, P., and Saha, D. (2007). First kes international symposium, agent and multi-agent systems: Technologies and applications, lnai 4496. chapter An agent based method for web page prediction, pages 219–228.

Perkowitz, M. and Etzioni, O. (2000). Towards adaptive web sites: conceptual framework and case study. Artificial Intelligence, 118(1-2):245–275.

Roulston, M. and Smith, L. (2002). Evaluating probabilistic forecasts using information theory. Monthly Weather Review, 130:1653–1660.

Sen, R. and Hansen, M. (2003). Predicting web users next access based on log data. Journal of Computational and Graphical Statistics, 12(1):1–13.

Spiliopoulou, M., Mobasher, B., Berendt, B., and Nakagawa, M. (2003). A framework for the evaluation of session reconstruction heuristics in web usage analysis. INFORMS Journal on Computing, 15:171–190.

Su, Z., Yang, Q., Lu, Y., and Zhang, H.-J. (2000). Whatnext: a prediction system for web requests using n-gram sequence models. Proceedings of the First International Conference on Web Information Systems Engineering, pages 214–221.

Witten, I. and Frank, E. (2005). Data Mining, Practical Machine Learning Tools and Techniques. Morgan-Kaufmann, San Francisco, Ca., 2nd edition.

Wu, Y.-H. and Chen, A. (2002). Prediction of web page accesses by proxy server log. World Wide Web: Internet and Web Information Systems, 5(1):67–88.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems, 22(2):179–214.