Investigating the Effect of Duration, Page Size and Frequency on Next Page Recommendation with Page Rank Algorithm

Banu Deniz GUNEL
Middle East Technical University
Computer Engineering Department
Ankara, TURKEY
deniz.yanik@ceng.metu.edu.tr

Pinar SENKUL
Middle East Technical University
Computer Engineering Department
Ankara, TURKEY
senkul@ceng.metu.edu.tr

ABSTRACT
In this paper, we extend the use of page rank algorithm for next page prediction with several navigational attributes, which are size of the page, duration time of the page and duration of transition (two page visits sequentially), frequency of page and transition. In our model, we define popularity of transitions and pages by using duration information and use it in relation to page size and visit frequency factors. By using the popularity value of pages we bias conventional Page Rank algorithm and model a next page prediction system that produces page predictions under given top-n value. Actually we devise Duration Based Rank (DPR), which focuses on page duration with size proportion and Popularity Based Page Rank (PPR) ranking model, which focuses on both page duration with size proportion and frequency value of page visits. In addition to this, we investigate the effect of global and local ranking on PPR and DPR.

Categories and Subject Descriptors
H.3.4 [Information Storage and Retrieval]: Systems and Software—World Wide Web (WWW); H.3.1 [Software Engineering]: Information Search and Retrieval—selection process, search process

General Terms
Algorithms, Measurement, Performance, Experimentation, Design, Verification

Keywords
Recommendation, Next Page Prediction, Web Usage Mining, Page Rank Algorithm, Directed Graph, Markov Model

1. INTRODUCTION
Internet users on the World Wide Web (WWW) has increased by the rate of 400% by 2011 [7]. In addition to this number of web pages that are indexed on the Internet is over 50 billion [12]. According to the studies, the size is doubling itself every six to ten months. Web includes a high volume of data that can be described as a bulk of data, which is unfortunately in a raw format. Since gathering information is an indispensable process in our lives, it is necessary to transform this raw data into information. Web usage mining is one of the most common approaches for extracting information that is hidden in the web. Web usage mining can be defined as data mining that is applied on web page visit specific data. In our research we are focused on combination of web usage mining and structural information of web sites, which can be seen as a hybrid web mining approach, depending on web server logs.

Next page prediction on a web site is a widespread and promising research area. Especially for recommendation systems, navigations of users in the web site are used for recommending them new pages. These recommendations are usually specialized in predicting the next page of user. This can be applied on various domains. For instance, in shopping web sites, movie or music web sites such information is very useful for recommending new items by analyzing similar behavior of other users in the web site.

In the literature various techniques have been used in order to analyze the web logs [9, 13, 3, 5] for next page prediction. Data mining techniques are heavily used used for this purpose. Clustering, sequence mining, associative rule mining and probability models are some of the popular techniques for predicting the next page of user [8, 4, 14].

Markov models are one of the approaches that is used for calculating the probability of a sequence [16]. They have been studied for random processes and it has been shown that they are well suited for predicting next page of user [1, 2, 14]. In Markov a model using longer sequence of navigation for predicting next page leads to more precise results. On the other hand, using longer navigation sequence increases space complexity. This is the main limitation behind Markov models.

Another preferred approach for predicting user’s next page navigation is using Page Rank, which is the algorithm behind Google’s search engine [15]. Next page will be the page that has the highest rank on these kinds of systems. The main idea behind Page Rank algorithm is that if one page is popular and it points to another page, the page that is pointed by a popular page is more popular than the pointing page. Therefore, in-links of a page’s popularity determines the popularity of that page. At this point, popularity can be
defined in many different ways. Despite the fact that Page Rank is a promising method for labeling pages that can be used for recommending next page, there is an important disadvantage of page ranking algorithm for this domain. The method produces popular pages in a global context, which does not include user’s historical navigation behavior. Ignoring this kind of information causes to produce always very similar results for predicting next page. As a remedy, in [6, 5], it is combined with low-level Markov model (which can also seen as a directed graph). In this work, we extend this hybrid approach with the effect of time spent on the web page, structural information of the page (size of the page) and frequency of transitions. In our work, we define popularity of page transitions and popularity of pages in terms of the frequency of transitions among pages and frequency of page clicks, respectively.

Therefore, the following factors are considered in page rank calculation; visit frequency of the page and transition, duration on the page and transition (average time spend on the page), size of the page, number of in-links and out-links of the page. These factors are investigated under two separate algorithms. Duration Based Rank (DPR) algorithm focuses on page duration and size, whereas Popularity Based Page Rank (PPR) algorithm focuses on both page duration and size proportion and frequency of pages.

Briefly, in our recommendation system, the architecture is composed of a sequence of components (see Figure 1) in offline part. In the offline process, Page Finder analyses pages and applies cleaning operations on the data. After that, Session Finder constructs sessions from web page click logs. Feature Calculator calculates duration values of pages and transitions, frequency values of pages and transitions and size of pages. Lastly, Rank Calculator calculates rank values for PPR, DPR, and UPR ranking algorithms for both local and global models. In the online part of the system Recommender recommends top-n pages related to a user’s last visited page that is given to system.

![Figure 1: General architecture of recommendation system](image)

In a nutshell, the basic contribution of this work is that while calculating the page rank value, we emphasize the time spent on a page (duration) in relation to the size of the pages together with the page and transition frequencies. In addition, we investigate the effect of top-n limits, modeling transitions globally (as a whole web site) and locally (as a synopsis of a web site).

The rest of this paper is organized as follows. Section 2 presents the related work on next page prediction domain, followed by a brief overview of Markov model and Page Rank algorithm. Section 4 presents a detailed description of proposed model including DPR and PPR calculation. In Section 5 evaluation method for our new algorithms are stated. Finally in Section 6 we conclude our work.

2. RELATED WORK

In this work, we present several studies from literature, which are similar to our work in different aspects. In [14], Mobasher et al. works on web usage mining area, focusing on producing associative rules from web server logs. In their work, they extract rules for predicting user’s next page by using Apriori algorithm.

Another method used in next page prediction is employing probabilistic reasoning methods. Especially Markov model and variations of them are used for predicting next page of user’s navigation by using historical navigation patterns of users. It depends on the idea that in a sequence of visits of a user, each probability of visiting one page and probability of the binary permutations of this sequence determines the whole sequence’s probability [16].

In Markov models, the probabilities are kept in a huge probability matrix and dimensions can be defined as the combination of pages by the order level. For this reason, several studies aim to reduce the size of Markov model with some pruning methods. The work given in [2] uses Markov model with error pruning, frequency pruning and confidence pruning. It is called selective Markov model. Another work is presented in [1], which can be defined as variable length Markov model. The model defines variable length Markov model depending on the complexity of the problem.

The Page Rank algorithm [3] uses the link structure of pages for finding the most important pages with respect to the search result. The algorithm states that if the in-links (pages that pointed to the page) of a page are important, then out-links (pages that pointed by the page) of the page also become important. Therefore the page rank algorithm distributes the rank value of itself through the pages it points to.

There are models that bias Page Rank algorithm with other type of web usage data, structural data or web contents. In [5], Usage Based Page Rank algorithm is introduced as the rank distribution of pages depending on the frequency value of transitions and pages. They model a localized version of ranking directed graph. In [9], they modify Page Rank algorithm with considering only the time spent by the user on the related page. However in their work, neither the effect of size value of pages nor frequency values of pages are considered.

3. BACKGROUND

3.1 Markov Model and Directed Graph

Whole web site or some local subset of it can be modeled as a directed graph with nodes as web pages and edges as transitions between web pages.

A Markov (chain) model is a mathematical system that undergoes transitions from one state to another, among a finite number of states [16]. It is a random process characterized as memoryless, where the next state depends only on
the current state and not on the sequence of events that preceded it. However in the kth-ordered Markov model, transition probabilities can be calculated with previous actions depending on the order of the model.

Hence web page navigations can be modeled as a directed graph which models Markov (chain) model by adding probability values to edge labels of these transactions. In our calculations we use first order Markov model due to the high number of pages that appear in the server logs.

3.2 Page Rank Algorithm

Page Rank algorithm [15] models the whole web as a directed graph that keeps nodes as web pages. They use link structure of pages for determining the importance (rank value) of pages. Google Web search engine [11] mechanism uses Page Rank algorithm for recommending relevant pages to user by ordering them through their rank values. In this algorithm, it is stated that if a page has some important in-links to it then its out-links to other pages also become important. In other words if a page is important then pages that it points to are also important. Therefore the algorithm propagates in-links of pages and if the in-links’ total is high then the rank value of it is also high.

Basic calculation of Page Rank algorithm is given in Equation 1. \( IN(v) \) represents the in-links of page \( v \), \( OUT(v) \) is the out-links of page \( v \), \( |OUT_v| \) list the number of out-links of page \( v \) and \( WS \) is the number of the web page set that includes all pages in the web site.

\[
PR(u) = \frac{(1-\epsilon)}{WS} + \epsilon \sum_{v \in IN(u)} \frac{PR(v)}{|OUT_v|} \quad (1)
\]

In page rank calculation, especially for larger systems, iterative calculation method is used. In this method, the calculation is implemented with cycles. In the first cycle all rank values may be assigned to a constant value such as 1, and with each iteration of calculation, the rank values become normalized within approximately 50 iterations under \( \epsilon = 0.85 \).

In [6] Usage Based Page Rank (UPR) is introduced. UPR is a variation of the Page Rank algorithm, based on the visit frequency data obtained from previous users’ sessions. Equation 2 is given for UPR calculation for \( n \) iterations. \( IN(p_i) \) formulates the set of in-links of page \( p_i \), \( OUT(p_i) \) formulates the set of out-links of page \( p_i \), \( w_i \) is the frequency of page \( p_i \) and similarly \( w_{i,j} \) is the frequency of page \( p_i \) visit after \( p_j \).

\[
UPR^n(p_i) = \epsilon \sum_{p_j \in IN(p_i)} \left( UPR^{n-1}(p_i) \star \frac{w_{j,i}}{\sum_{p_k \in OUT(p_j)} w_{j,k}} \right) + (1-\epsilon) \sum_{p_j \in WS} w_j \quad (2)
\]

4. PROPOSED MODEL

4.1 Defining Sessions

While speaking of user navigations, user sessions should be considered as a basis. A user session reflects user’s navigation behaviors (or transitions) in a web site. All user navigations in a web site can be modeled as a directed graph.

In Table 1 sample user sessions are shown. In transitions column of the table, \( P_i \)’s are the pages that a user visits in a session with the given order.

<table>
<thead>
<tr>
<th>Session ID</th>
<th>Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>P1→P2→P3→P4</td>
</tr>
<tr>
<td>S2</td>
<td>P2→P4</td>
</tr>
<tr>
<td>S3</td>
<td>P1→P2→P4</td>
</tr>
<tr>
<td>S4</td>
<td>P2→P3→P1→P4</td>
</tr>
</tbody>
</table>

In Figure 2 directed graph of sessions S1, S2, S3 and S4 are modeled. In this graph, node weights and edge weights are page frequencies and transition frequencies, respectively. In order to complete the graph, we add start (S) and finish (F) nodes to the graph, which is an abstraction and those are not actually mapped to a real page in sessions. We assume that every session starts with start node and finishes with finish node, respectively.

Figure 2: Directed Web Graph of Sample Sessions

Navigational behaviors on the web page can be modeled as a weighted directed graph that includes pages as nodes and edges as transitions between pages. In addition to this system, frequency of transitions and frequency of pages can be defined in navigational graph by node weights and edge weights. In the rest of this section, two proposed algorithms, Duration Based Page Rank (DPR) and Popularity Based Page Rank (PPR) are presented. In both algorithms, this directed graph is taken as the basis for calculations.

4.2 Duration Based Page Rank (DPR)

Distributing rank values of a page to the pages it points to equally is not the best solution for page rank calculation towards next page prediction. In Duration Based Page Rank (DPR) calculation, the distribution simply depends on the duration values of pages and transitions and their web page file size. Page duration can be defined as the time spent by user on the page after another page visit in a given session. Since we want to analyze general behavior of transitions and page rank values, in DPR calculation we use average duration values. On the other hand, transition duration can be defined as the time spent on two given pages’ transitions consecutively. For instance \( P_1 \rightarrow P_2 \) duration can be calculated by searching all \( P_1, P_2 \) transitions in the sessions and retrieving time that is spent on visiting \( P_2 \) after visiting \( P_1 \). Furthermore, we consider the ratio of duration to page size, since in some cases, user spends much time on a web page not for his own interest, but just due to the page size.
size. With the proportion of the two values we aim to focus the real interest of the users on the web pages by considering the file size of them.

\[
DPR_i = \epsilon \times \frac{\sum_{x_j \in IN(x_i)} DPR_j}{\text{AvgDuration}_{P \rightarrow i}} + (1 - \epsilon) \times \text{PageP}_i
\]

(3)

General page rank calculation approach adds a random surfer jumping factor to the rank value of the page, which means that user may jump to another page that is not a linked navigation [3]. For example, user may write on the internet browser the URL that she wants to go. Hence we can define normal user navigation (sequential) behavior as an edge visit on the graph and jumping behavior as a node visit on the graph.

DPR calculation uses page duration for random surfing behavior and transition duration for regular visiting behavior of users. The calculation for DPR is given in Equation 3.

4.3 Popularity Based Page Rank (PPR)

Popularity Based Page Rank (PPR) calculation, given in Equation 4, is modeled in terms of transition popularity and page popularity of the pages that point to (in-links of) the page that is under consideration. In the equation, \(IN(x_i)\) is the set that keeps the in-links of that page.

\[
PPR_i = \epsilon \times \sum_{x_j \in IN(x_i)} [PPR_j \times \text{TransitionP}_{j \rightarrow i}] + (1 - \epsilon) \times \text{PageP}_i
\]

(4)

In the equation above, rank distribution of pages in our model depends on the popularity of a page (\(\text{PageP}\)) and transitions (\(\text{TransitionP}\)) that point to that page.

In our model, we define popularity in two dimensions. The first one is page dimension and second one is transition dimension. For both dimensions we define popularity in terms of time user spends on page, size of page and visit frequency of page. Our calculation model is constructed by using coefficients in a different form for assigning rank values to pages than traditional page rank distribution that assigns equal rank values to all in-links of a page.

In popularity calculation, page and transition popularity are calculated separately but in a similar way. Page popularity is needed for calculating random surfer jumping behavior of the user and transition popularity is needed for calculating the normal navigating behavior of the user. However the main idea is common for finding popularity for nodes and edges. The calculations for transition and page popularity is given in Equation 5 and Equation 6, respectively.

\[
\text{TransitionP}_{j \rightarrow i} = \text{Frequency}_{j \rightarrow i} \times \text{AvgDuration}_{j \rightarrow i}
\]

(5)

\[
\text{PageP}_j = \text{Frequency}_j \times \text{AvgDuration}_j
\]

(6)

The main difference between transition popularity and page popularity can be seen as the focus of their calculation. We start with explaining page dimension and continue with transition dimension. In Equation 7, frequency of page calculation can be found. \(w_j\) is the frequency of visiting \(p_j\) page.

\[
\text{Frequency}_i = \frac{w_j}{\sum_{p_j \in WS} w_j}
\]

(7)

Average duration calculation that also uses the size of value of pages can be found in Equation 8. In this equation, \(d_i\) is the time spend on that page visit until next navigation and \(s_i\) is the size of the page.

\[
\text{AverageDuration}_i = \frac{d_i/s_i}{\max (d_m/s_m)} \text{ where } p_m \in WS
\]

(8)

Finally the open form of page popularity formula can be found in Equation 9.

\[
\text{PageP}_i = \frac{w_i}{\sum_{p_j \in WS} w_j} \times \frac{d_i/s_i}{\max (d_m/s_m)} \text{ where } p_m \in WS
\]

(9)

Equation 10 gives the formula for transition frequency calculation. In this equation, \(\text{OUT}(p_j)\) can be described as the frequency of the transaction. Hence it can be seen as the number of the visits in which \(p_i\) comes after page \(p_j\). In addition, \(\text{OUT}(p_j)\) is the set of pages that point to \(p_j\).

\[
\text{Frequency}_{j \rightarrow i} = \frac{w_{j \rightarrow i}}{\sum_{p_k \in \text{OUT}(p_j)} w_{j \rightarrow k}} \times \frac{d_{j \rightarrow i}/s_i}{\max (d_m/s_m)}
\]

(10)

In Equation 11, \(d_{j \rightarrow i}\) is duration of the transaction, and \(s_i\) is the size of the transition’s result page. \(WS\) is the web page set that includes all pages in the web site. Duration size proportion is inspired from [13] which uses this proportion in a different concept.

\[
\text{AvgDuration}_{j \rightarrow i} = \frac{d_{j \rightarrow i}/s_i}{\max (d_m/s_m)} \text{ where } p_m, p_n \in WS
\]

(11)

In Equation 12, transition popularity is defined in terms of transition frequency and duration.

\[
\text{TransitionP}_{j \rightarrow i} = \frac{w_{j \rightarrow i}}{\sum_{p_k \in \text{OUT}(p_j)} w_{j \rightarrow k}} \times \frac{d_{j \rightarrow i}/s_i}{\max (d_m/s_m)}
\]

(12)

4.4 PPR and DPR Calculations In Detail

In this section, we present how the given equations are used in the proposed algorithms on a sample case. Since PPR includes both frequency, time and page size factors, we present only PPR calculations. In Table 2, page id, page size, average duration and frequency values of pages for the sample case are listed.

**Table 2: Page Properties in Sample Sessions**

<table>
<thead>
<tr>
<th>Page Id</th>
<th>Page (byte)</th>
<th>Size (Avg. Dura. (ms))</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1216</td>
<td>297000</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>8103</td>
<td>231000</td>
<td>2</td>
</tr>
<tr>
<td>P3</td>
<td>303537</td>
<td>97000</td>
<td>2</td>
</tr>
<tr>
<td>P4</td>
<td>9039</td>
<td>10500</td>
<td>3</td>
</tr>
</tbody>
</table>

In Table 3, transitions and average transition durations for the sample case are given. This is a synthetic data that is produced for illustration purpose. Since defining Start(S) and Finish(F) nodes is an abstraction for completing the directed graph, transaction times related to these navigations are not calculated from server logs. In our proposed model, we assigned these transitions the average value of transaction durations. Assumed values in the sample case
are higher than real values, however it should be pointed out that in real data set, these values are radically less than the values calculated for the sample sessions\(^3\).

Table 3: Avg. Duration Table for Sample Sessions

<table>
<thead>
<tr>
<th>Transition</th>
<th>Calculated Avg. Duration (ms)</th>
<th>Final Avg. Duration (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S→P1</td>
<td>NA</td>
<td>77000</td>
</tr>
<tr>
<td>S→P2</td>
<td>NA</td>
<td>77000</td>
</tr>
<tr>
<td>P1→P2</td>
<td>123500</td>
<td>123500</td>
</tr>
<tr>
<td>P2→P3</td>
<td>97000</td>
<td>97000</td>
</tr>
<tr>
<td>P3→P4</td>
<td>10500</td>
<td>10500</td>
</tr>
<tr>
<td>P2→F</td>
<td>NA</td>
<td>77000</td>
</tr>
<tr>
<td>P4→P3</td>
<td>NA</td>
<td>77000</td>
</tr>
</tbody>
</table>

According to these values, popularity rank values of pages can be calculated easily. We show one calculation in detail and give the other pages rank values in Table 4. Let us calculate popularity value of page \(P2\) step by step. From the page popularity equation, popularity of \(P2\) is calculated as 0.023.

\[
\text{Page}_{P2} = \frac{w_2}{\sum_{j \in WS} w_j} \frac{d_{Q_2}^*}{d_{Q_2}} \cdot \frac{\sum_{j \in WS} w_j}{\text{max} (ws/s, sm)} = \frac{2}{(3+2+2+3) \cdot 28.51} = 0.023
\]

The values calculated for all transitions in the sample case are given in Table 4.

Table 4: Transition Popularity for Sample Sessions

<table>
<thead>
<tr>
<th>Transitions</th>
<th>Frequency</th>
<th>(\gamma_j)</th>
<th>TransitionP</th>
</tr>
</thead>
<tbody>
<tr>
<td>S→P1</td>
<td>2</td>
<td>63.32237</td>
<td>0.50000</td>
</tr>
<tr>
<td>S→P2</td>
<td>2</td>
<td>9.50265</td>
<td>0.07503</td>
</tr>
<tr>
<td>P1→P2</td>
<td>3</td>
<td>15.24127</td>
<td>0.24069</td>
</tr>
<tr>
<td>P2→P3</td>
<td>2</td>
<td>0.31957</td>
<td>0.00336</td>
</tr>
<tr>
<td>P3→P4</td>
<td>1</td>
<td>1.16667</td>
<td>0.00143</td>
</tr>
<tr>
<td>P2→F</td>
<td>1</td>
<td>1.16667</td>
<td>0.00143</td>
</tr>
<tr>
<td>P4→P3</td>
<td>3</td>
<td>1.16667</td>
<td>0.00143</td>
</tr>
</tbody>
</table>

At the end of the first iteration under \(\epsilon = 0.85\), rank values for our sample session are given in Table 5. Although we just show the results for one iteration for demonstration purpose, while making next page recommendations, the stability of rank values will be important. This is provided by normalization through further iterations.

In Table 5, Popular Page Rank values of pages that are calculated for single iteration are listed.

\(^1\)In our duration calculations we make 2 iterations. In the first one we calculate exact values of durations and averages of them. In the second iteration we update NA values with whole average value of durations.

\(^2\)Commonly in rank calculation experiments \(\epsilon\) value is set to 0.85 and iteration number is set to 50. In our experiments we applied these constants.

\(^3\)As a base for the calculation, we assumed that average values of file size and duration are acceptable for start (S) page of the sessions. So we calculated popularity of S in this table.

4.5 Predicting Next Page

For predicting the next page, a recommendation set is constructed under the proposed algorithms. The main idea behind predicting next page is to produce recommendations from directed graph that is designed from sessions in web server logs. In the directed graph, for a certain depth, pages are listed and sorted in descending order by calculated rank values. Hence the next page prediction method can be seen as Markov model that is supported by rank values of pages instead of probabilities. This model can be seen as 1st order Markov model that has a page rank value base.

Consider the example navigation graph given in Figure 3. If a user visits page P1, the recommendation set for depth 2 includes P2 and P3 pages, and they will be sorted in descending order with respect to rank values. Therefore, the recommendation set will be \(R\{P2, \ P3\}\) sorted by PPR value.

In order to make rank calculations faster, we record intermediate steps of our calculations to database. Intermediate step values related to rank calculations are, average duration value of pages, average duration values of transitions, page size, frequency value of pages, frequency value of transitions. After defining sessions and relating them to pages, we calculate average duration values of pages that can be inferred from transition durations that are already recorded. Since duration values may not be calculated for pages that appear at the end of sessions, we assign them average duration values of pages. In addition to this, while analyzing sessions, we calculate transition durations and the size of pages from web server logs. Moreover while analyzing sessions we record page frequencies and calculate transition frequencies. We calculate rank related intermediate values concurrently. Therefore, in our model we recommend a set of pages that are sorted by the rank value calculated in the model in descending order on the basis of the pages visited before.
5. Evaluation

In the experiments, we analyze and compare the accuracy performance of UPR, DPR and PPR methods. In our experiments basically we use the evaluation method employed in [5]. In addition, we consider extra evaluation methods. In our experimental evaluations, we use METU’s web server logs from 29/May/2010 to 18/Feb/2011. In the raw data there are 5,168,361 unique page URLs. We partition training data and test data randomly from this data set. We separate approximately 1/3 of data for test data and 2/3 of data for training.

Frequency pruning [2] is used for cleaning the data. Working with frequent pages produce more stable results in general. For this reason, in our experiments we set minimum frequency limit to 10. After cleaning less frequent data in training set we have approximately 6000 pages in total and in test data we have 2000 pages.

In the experiments, we apply 4-fold cross validation. We also perform evaluation under local ranking [5] and global ranking models. In addition to frequency pruning, we apply several more steps for pruning. In order to construct sessions, we group page visits (transitions) that are done by the same IP and client name (Internet browser) occurring within 30 minutes. Moreover, we define a rule that controls the duration of transitions such that the time spent on a page should not exceed 10 minutes. Otherwise, a new session starts beginning with the last transition.

After preprocessing, the formulas given in Section 4 are applied under \( \epsilon = 0.85 \), which results in 0.15 jumping factor and 50 iterations. Rank values are calculated for all three algorithms (Usage Based, Duration Based and Popularity Based Ranking algorithms). It should be pointed out that, these calculations are performed for both global and local model with depth 2.

After the pruning, test data set contains approximately 110 sessions and training data set contains about 225 sessions. In these sessions, we produce a directed web graph of test data in order to produce real transition values and to compare with the predictions. In every evaluation, we pick one page in the directed graph that have 2 or more nodes that it points to. Then for that page, every algorithm produces recommendation sets.

In comparing the predictions with the real page visits, we use two similarity algorithms that are commonly preferred for finding similarities of two sets. The first one is called Osim [10] algorithm, which calculates the similarity of two sets without considering the ordering of the elements in the set. It focuses on the number of common elements of two sets with a limit value. The limit value can be seen as the top-n next page recommendations for a visited page. The equation of Osim algorithm is defined in Equation 13, where \( A \) and \( B \) are the sets to be compared, having the same length and \( n \) is the top-n value of comparison. The similarity value range is [0-1] and 1 denotes maximum similarity.

\[
Osim(A, B) = \frac{A \cap B}{n}
\]  
(13)

As the second similarity metric we use Ksim similarity algorithm, which concerns Kendall Tau Distance [10, 5] for measuring the similarity of next page prediction set produced by training data set and real page visit set on the test data. Kendall Tau Distance is the number of pairwise incompatibility between two sets. It is also titled as bubble sort distance since it is equivalent to the number of swaps for making the two lists in the same order by using bubble sort algorithm. In this similarity metric, as the distance increases, similarity decreases. Ksim similarity calculation is given in Equation 14. Sometimes the compared sets may have different lengths. The lengths of the sets are equalized, by utilizing the union set, as shown in Equation 14.

\[
d_1 = A \cup B - A \quad \text{and} \quad d_2 = A \cup B - B
\]

\[
A' = A \quad \text{followed by} \quad d_1 \quad \quad \text{and} \quad \quad B' = B \quad \text{followed by} \quad d_2
\]

\[
Ksim(A, B) = 1 - \frac{\tau \text{ distance}(d_1, d_2)}{|A \cup B| * (|A \cup B| - 1)}
\]  
(14)

\( \tau \) distance comes from the Kendall Tau distance algorithm mentioned before. In our experiment setup, we make separate experiments with top-3, top-5 and top-10 recommendation comparisons that are measured by Ksim and Osim with global and local ranking methods. The results of the experiments for the next page prediction accuracy for three different page ranking algorithm under Ksim and Osim similarity metrics and local and global models are given in Table 7.

<table>
<thead>
<tr>
<th>Model</th>
<th>Local Model</th>
<th>Global Model</th>
<th>Local Model</th>
<th>Global Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 3</td>
<td>UPR</td>
<td>0.69</td>
<td>0.68</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>0.79</td>
<td>0.79</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>PPR</td>
<td>0.83</td>
<td>0.85</td>
<td>0.94</td>
</tr>
<tr>
<td>Top 5</td>
<td>UPR</td>
<td>0.73</td>
<td>0.72</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>0.84</td>
<td>0.84</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>PPR</td>
<td>0.87</td>
<td>0.87</td>
<td>0.70</td>
</tr>
<tr>
<td>Top 10</td>
<td>UPR</td>
<td>0.75</td>
<td>0.77</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>0.87</td>
<td>0.87</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>PPR</td>
<td>0.89</td>
<td>0.96</td>
<td>0.42</td>
</tr>
</tbody>
</table>

When we analyze the effect of global versus local model, it is seen that, in general, under different similarity metrics, the effect of the graph model is very limited. The only effective case is for where global modeling with Ksim leads to approximately 8% increase when compared to local models.

As seen in the results, global ranking model has advantage, although little, at certain experiments. Therefore in the rest of the experiments, we work only under global model. In [5], the basic drawback of global model is reported as its inefficiency. However, in this work, we decrease the global model calculation time by storing the intermediate results. By this way, we can benefit from the global model without increasing the time cost.

In order to evaluate the effect of recommendation set, we made experiments with top-3, top-5 and top-10 predictions respectively. Since in real recommendation systems, the user would not expect a long list of recommendations, we limit the maximum recommendation level with 10. In the values that we obtained, we deduce that by using Osim measure becomes disadvantageous as the size of the recommendation set increases. For both models we observed that top-3 is the best solution.

In addition, by using Osim similarity measure, we have observed that PPR modeling produces the most similar values to real visit data by accuracy value of 0.94, which is followed by DPR modeling value of 0.92 and lastly UPR.
modeling with value of 0.70. We also can infer that by using higher prediction count limits, the accuracy is decreasing and the accuracy values of PPR and DPR are getting closer to the UPR. In other words PPR is effective on on smaller recommendation page sets with proportion of 34%.

In Figure 5 we summarize the effect of prediction set size under Ksim metrics for three algorithms. As seen in the figure, as the set size increases, accuracy of all algorithms increases. In addition to this we observed that for every limit value there is a difference of UPR and PPR with approximately proportion of 19%.

Finally, in Figure 6, we present the average values of measurements for global modeling and top-3 data limit for both Ksim and Osim.

The detailed results for each fold of 4-fold cross validation given in Figure 7 and Figure 8.

5.1 Deciding Local and Global Modeling Effectiveness

For evaluating the effect of the local and global modeling on the accuracy of next page predictions, we apply hypothesis t-test with global and local models. For each fold values, we evaluate t-test results with confidence interval 99%. In our t-tests, since we know that each value is the same measurement, we use one tailed pairwise t-tests. We apply all of the method’s results in each iteration for Ksim and Osim similarity measures.
In these t-tests, we define $h_0$ hypothesis as "There is no statistical difference between global and local modeling while calculating the accuracy of similarity measures," and we choose $\alpha = 0.01$. If the $p$-value < $\alpha$ then we can say that the local and global modeling is statistically significant in calculating the accuracy of similarity measures in 99% confidence interval. In our calculations none of the results is smaller than the $\alpha$ value which means the $h_0$ hypothesis is accepted with 99% confidence.

6. CONCLUSION

Page rank algorithms are commonly used for both next page prediction and web searching. In addition, duration of page visits retrieved from transitions can be considered as well [9]. However the duration of page, which can be directly related to page size is not modeled for page ranking algorithm. For example if the user is waiting for the download of a long page including large objects such as images, obviously it would take more time than a page which includes really small amount of data. Although just the size information of page cannot produce information for popularity of a page, the proportion of duration and size can produce information for popularity of pages. We model this situation as Duration Based Page Rank (DPR), which concerns duration vs. size proportion. In addition, we model another hybrid approach that concerns both duration size proportion and frequency of pages and transitions which is called Population Based Page Rank (PPR) algorithm.

We conducted a set of algorithms in order to evaluate the success of DPR and PPR. We observed that PPR and DPR models for next page prediction is a promising approach with higher accuracy than that of previous similar models. In formula variations, using sum of all durations instead of maximum duration in average duration calculation, can be elaborated more. We have tried variations for average duration and page popularity formulas however the current form gives the optimal results. As a future work, semantic information that is related to web pages can be considered and the popularity factor may depend on the concept of pages. Labeling or concept grouping of pages may be added to popularity value of pages, which add relativity to popularity concept in our model with semantics.

7. ACKNOWLEDGEMENT

This work is supported by grant number TUBITAK-109E239, TUBITAK.

8. REFERENCES


Table 7: P-Values of Local and Global Modeling

<table>
<thead>
<tr>
<th></th>
<th>Top-2</th>
<th>Top-4</th>
<th>Top-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ksim</td>
<td>0.017</td>
<td>0.0143</td>
<td>0.022</td>
</tr>
<tr>
<td>Osim</td>
<td>0.0145</td>
<td>0.035</td>
<td>0.022</td>
</tr>
<tr>
<td>3-Fold</td>
<td>0.0141</td>
<td>0.0141</td>
<td>0.0141</td>
</tr>
</tbody>
</table>