

Usage-Aware Average Clicks

Ramya Rangarajan
University of Minnesota
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455
1-612-625-6597
ramya@cs.umn.edu

Kalyan Beemanapalli
University of Minnesota
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455
1-612-625-6597
kalyan@cs.umn.edu

Jaideep Srivastava
University of Minnesota
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455
1-612-625-6597
srivasta@cs.umn.edu

ABSTRACT

A number of methods exists that measure the distance between two web pages. Average-Clicks [18] is a new measure of distance between web pages which fits user's intuition of distance better than the traditional measure of clicks between two pages. Average-Clicks however assumes that the probability of the user following any link on a web page is the same and gives equal weights to each of the out-going links. In our method "Usage Aware Average-Clicks" we have taken the user's browsing behavior into account and assigned different weights to different links on a particular page based on how frequently users follow a particular link. Thus, Usage Aware Average-Clicks is an extension to the Average-Clicks Algorithm where the static web link structure graph is combined with the dynamic Usage Graph (built using the information available from the web logs) to assign different weights to links on a web page and hence capture the user's intuition of distance more accurately. This method has been used as a new metric to calculate the page similarities in a recommendation engine to improve its predictive power.

Categories and Subject Descriptors

I.5.3 [Computing Methodologies]: Algorithms, Similarity Measures.

General Terms

Algorithms, Measurement, Human Factors, Experimentation.

Keywords

Web Mining, Link Analysis, Web Usage Analysis, Recommendation Engine

1. INTRODUCTION

The World Wide Web is an ever growing collection of web

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WEBKDD'06, August 20, 2006, Philadelphia, Pennsylvania, USA

Copyright 2006 ACM 1-59593-444-8...\$5.00.

pages. Thousands of web sites and millions of pages are being added to this repository every year. The web pages vary widely in their content and format and are very diverse in nature. A lot of useful information is available on the World Wide Web (WWW) and Search Engines help us find what we are looking for. For doing this, Search Engines make extensive use of the Link Structure Graph and a lot of research is going on to ensure that the best set of links are returned to the user based on the search query.

The link structure graph of the web is a digraph where the nodes represent the pages and a directed edge from node A to node B implies that page A has a link to Page B. There are important algorithms in the literature like Page Rank [11], HITS [10, 2] and Average-Clicks [18] which use the link structure graph as their basis. These algorithms differ in the way they use link structure graph to assign different weights to the nodes. The Google [7, 6, 14] search engine uses PageRank algorithm to rank web pages. PageRank is a global ranking algorithm which ranks web pages based solely on the position of the page in the web graph. HITS is another popular algorithm which ranks the web search results. HITS classifies the web pages as Hubs and Authorities. Average-Clicks uses Link Analysis to measure the distance between two pages on the WWW. One inherent problem with all these methods is that all of them are heavily dependent on the link structure graph and hence are static. The dynamic nature of user behavior is not taken into consideration when assigning weights to nodes.

In the Intranet Domain, useful information is available in the form of web logs which record the user sessions. User Sessions track the sequence of web pages visited by the user in addition to a lot of other information like the time spent on each page etc. The user session information is used to alter these algorithms so that they are not biased by the link structure graph and make more accurate weight calculations. Extensions to PageRank and HITS have been proposed in [4] and [9] respectively and they take into consideration the factors from the usage graph as well. These algorithms will be explained in Section 2. Having understood the importance of Usage Behavior data in link analysis, we propose an extension to the Average-Clicks [18] algorithm which measures the distance between two web pages on the WWW.

Our experiments were conducted in the intranet domain where complete and accurate usage data is easily available. On the internet it is difficult to get the usage data as it needs to be heuristically gathered from various external tools like Google Toolbar, Yahoo Toolbar, etc. Data from these sources is incomplete and skewed and needs to be refined using sophisticated algorithms before it can be used. Moreover, the

work has been motivated by a recommendation engine that is being built for the intranet. Hence, we confine ourselves to the intranet domain.

The organization of this paper is as follows: Section 2 gives a brief description of some of the algorithms which use Link Analysis and their extensions using Usage Information. Section 3 talks about the Average-Clicks algorithm which forms the basis of our proposed method. Section 4 describes our approach to incorporating usage information into the Average-Clicks algorithm. Section 5 presents experimental results obtained by running the modified algorithm on the cs.umn.edu website. A comparison of the distances obtained by running the new method with that obtained by running the original Average-Clicks algorithm is also made here. Section 6 talks about test cases and evaluation methodologies. Section 7 presents conclusions and potential future work.

2. Related Work

In this section we give a brief description of the PageRank and the HITS algorithms and also their extensions using Usage Information. Our idea of incorporating usage data into the Average-Clicks algorithm has been drawn from these methods. Google uses the PageRank algorithm to rank the web pages. A web page gets a high page rank if it has a large number of backlinks (a lot of pages pointing to it) or if it has backlinks from popular pages (pages that have very high page ranks) [11]. The page rank of a page is the sum of the weights of each of its incoming links and the PageRank of a page is equally distributed among its out links. Figure 1, reproduced from [11] gives an overview of PageRank calculation.

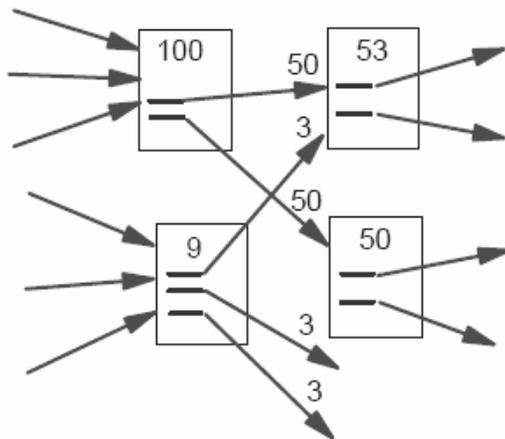


Figure 1: Simplified PageRank Calculation

In Usage Aware PageRank (UPR) [4] an extension to the PageRank Algorithm using usage statistics to improve the Rank calculation is suggested. The simple approach taken to incorporate the usage data into Page Ranks algorithm is to use the counts obtained from the web logs. Thus, the weight assigned to each page is based on the page popularity and it has been found that this method, in general, out performs the original PageRank algorithm. In [13, 16] the authors improve on the PageRank algorithm by using Content information of the web pages. They

are probabilistic models of the relevance of the page to a search query.

Kleinberg’s HITS is another algorithm which uses Link Structure and HITS ranks the web search results. HITS classifies the web pages as Hubs and Authorities. Good Hubs are pages which have a number of inlinks from good Authorities and good Authorities are the pages which have links to a number of good Hubs for the particular search topic [10]. HITS emphasizes on having a number of incoming links from related pages (good Authorities) rather than just having a large number of inlinks from unrelated pages. Figure 3, reproduced from [10] explains the concept of Hubs and Authorities.

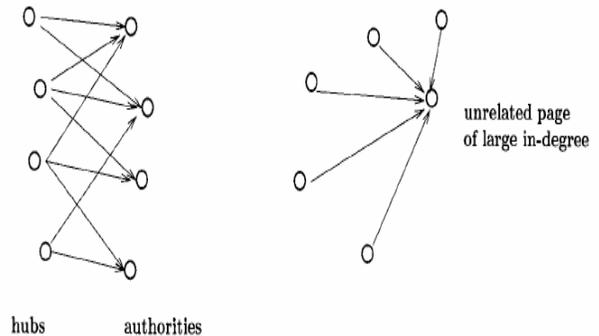


Figure 2: HITS Algorithm – example

In [9] an extension to the Kleinberg’s algorithm using Matrix Exponentiation and Web log records is proposed. The key idea of this approach is to replace the adjacency link matrix used by the original algorithm by an exponential matrix using Taylor’s Series. The usage graph is combined with this adjacency graph to assign new weights and the preliminary results show that this approach works well and gives improved results

3. Background

In this section we introduce the basic terms used in this report and also give a brief description of the Average-Clicks method which forms the basis of our approach.

As already mentioned, the WWW can be represented as a digraph [8, 1] where the nodes represent the pages and there is a directed edge from node A to node B if the page corresponding to node A has a link to the page represented by node B. The number of edges that point to a node (web page) is known as its in-degree (back links) and the number of edges that go out of a node is known as its out-degree (forward links) [11]. For example, in Figure 4, the in-degree of node B is 1 and that of node D is 2 and the out-degree of node B is 1 and that of node D is 2

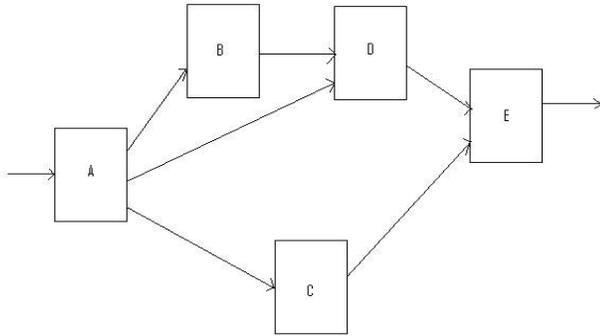


Figure 3: Sample Link Structure of a Web Graph

3.1 Average- Clicks

The Average-Clicks algorithm calculates the length of the link in a page p as

$$-\log_n(\alpha/OutDegree(p)).$$

Here the probability of clicking each link in page p is a constant given by $\alpha/OutDegree(p)$, where α is the damping factor. Negative log is taken to transform the multiplications while calculating the distances between web pages to additions. One average-click is defined as one click among n links on a page [18]. Extending the definition, one average-click from two pages with n links each is one click among n^2 links. Figure 5 reproduced from [18] shows a sample link graph along with the calculations of Average-Clicks distances.

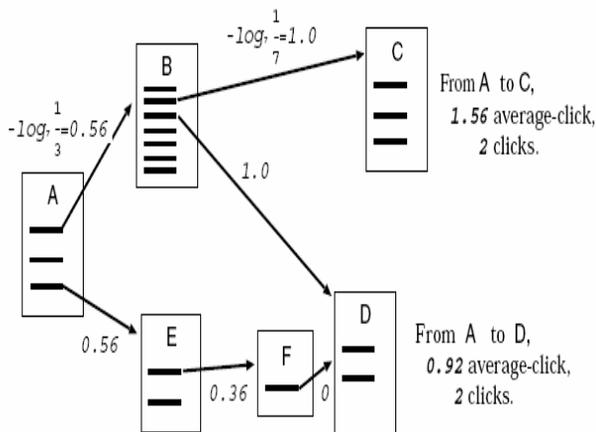


Figure 4: Average-Clicks and Clicks - example

The distance between two web pages p and q is defined as the shortest path between the nodes representing the pages in the graph. The shortest path can be calculated using any All-Pairs Shortest Path algorithm. Detailed description of the algorithm used in Average-Clicks and the results obtained can be found in [18].

4. Usage Aware Average-Clicks

In this section we give a detailed description of the approach taken by us to incorporate the usage information into the Average-Clicks Algorithm. The usage graph analysis gives the popularity of the pages with the users. The link graph analysis provides the importance of the web pages as designated by the creator of the pages. For example, if there is a link from the Yahoo main page then the link can be considered very important. Similarly, a page might be highly accessed by users even though it is not referenced by important pages in the web domain. Hence, it is important to consider both kinds of information and combine them effectively, so that we are able to determine pages that are popular and also important

4.1 Usage Graph

As the first step, the usage-graph (U) [15, 5] is constructed from the information available from the web logs. Each node in the graph U represents a page and an edge from node p to node q implies that page q has been accessed from page p . Every edge is assigned a weight and this value is a measure of the co-occurrence pattern of the two pages in the web logs (corresponding to the number of times page q was accessed from page p). Each node is also assigned a numerical value which indicates the number of times the page corresponding to that node occurred in the web logs. Using this graph, an $N \times N$ matrix C (holding the conditional probability $P(q \rightarrow p)$) is calculated where N is the number of nodes in the graph. Any value $C(i,j)$ indicates the probability of accessing page j from page i . The weight of the edge between each pair of pages p, q is calculated as follows:

$$C(p, q) = \frac{\text{Number of co- occurrences of } p, q}{\text{Number of occurrences of } p}$$

By co-occurrence of p, q we mean that page q should be accessed immediately after accessing page p . In terms of the usage graph this can be written as follows:

$$C(p, q) = \frac{\text{Weight of the edge from } p \text{ to } q}{\text{Weight assigned to node } p}$$

4.2 Link Graph

Next, a link graph is constructed using a web crawler. The Web crawler is run on the website used for testing and a directed graph is generated from the information obtained. Each node is a web page and an edge from node p to node q implies that page p holds a link to page q . Each node is assigned a value which is based only on the number of outgoing links from that page. An $N \times N$ link matrix D is calculated where N is the total number of pages in the website. Any value $D(i, j)$ gives the distance of page j from page i . The value of $D(i,j)$ is calculated as follows:

$$D(i, j) = (1/\text{Outdegree}(\text{page } i)) \text{ if there is a link from page } i \text{ to } j$$

$$\infty \text{ otherwise}$$

We then combine the Link matrix and the Usage matrix to define the new distance between 2 pages as follows:

$$Distance(p, q) = C(p, q) * \left(-\log_n \left(\frac{\alpha}{\text{Out degree}(p)} \right) \right)$$

where n^1 is the average number of links on a page and α^2 is the damping factor. Using the above distance matrix D, the matrix containing the shortest paths between pairs of pages is calculated using the Floyd Warshall's Algorithm.

4.3 Distance Measure using Floyd Warshall's Algorithm

Given the web link structure, the shortest distance between any pair of pages can be calculated using any All-Pairs Shortest Path algorithm. The all-pairs shortest-path problem involves finding the shortest path between all pairs of vertices in a graph. Algorithms like Dijkstra's algorithm can be run N times (once for each of the N pages), or Floyd Warshall's algorithm can be used. In our approach we used the Floyd Warshall's algorithm to construct the final NxN distance matrix. Floyd Warshall's Algorithm is very efficient as it uses the concept of Dynamic Programming and hence doesn't make any unnecessary computations.

4.4 Implementation Issues

The Floyd Warshall's algorithm uses an NxN matrix for distance calculation. As the number of web pages increases, the amount of memory needed to hold the NxN distance matrix increases drastically. Also, the Computation Cost increases exponentially. Thus, this algorithm has poor scalability. To overcome this issue and to make our program highly scalable and memory efficient, we have taken the following approach:

Each page is given a unique page id (starting from 0) and the set of links on a web page is stored as a linked list. The head of each of the linked list is stored in a vector called PageDetail. Thus PageDetail[0] points to the head of the linked list which stores the set of links on page 0. Each node in the linked list for page p stores the PageId of the page q to which it is connected, C(q->p), Average Clicks distance, Usage Score and the Usage aware

¹ For the World Wide Web the value of n has been identified as 7. A better approach uses 1 external link and 4 internal links.

² The damping factor for the World Wide Web is 0.85. For the intranet domain, this can be calculated from the usage data

Average-Clicks distance between page p and page q. Hence to get the distance between page p and page q, we have to search the list stored at PageDetail[p] for node q. This implementation is highly scalable as adding a new page to a vector is easy and does not require resizing an array each time a new page is added to the list. Also, it is very memory efficient as instead of storing N nodes for each page, we only store a very small number of pages equal to the number of links on that page. These are very important issues because as the number of pages in the domain increases, it is not possible to match the memory requirements of the algorithm.

5. Experimental Results

In this section we provide some preliminary results and also provide a comparative study of the distances obtained from the original Average-Clicks Algorithm and our approach.

5.1 Test Data

We have run our experiments on the CS website which is the Computer Science Department website of the University of Minnesota. The website can be accessed at www.cs.umn.edu [17]. The usage data has been collected over a period of 2 weeks in Apr 2006. The data set has been reduced to about 100,000 user sessions by refining and filtering the data. Noise data such as one page sessions, broken sessions etc have been removed to reduce the negative impact on the algorithm. We have implemented a web crawler to spider the website and collect the link information. The crawler has been programmed in such a way that URL's outside of the domain we are interested in are not considered. Also, self links are ignored as it does not make sense to recommend to the user, the same page he is already on.

5.2 Example Distances

Table I shows the distances as measured by the original Average-Clicks algorithm and also by Usage Aware Average-Clicks Algorithm between the graduate admissions index page <http://www.cs.umn.edu/admissions/graduate/index.php> and the links present on the page.

It can be seen that our approach, unlike the Average-Clicks algorithm, gives different weights to different links based on link access frequency. Thus, using the traditional Average-Clicks measure, we will declare that from www.cs.umn.edu/admissions/graduate/index.php it is equally probable to go to any link on the page, whereas, using Usage Aware Average-Clicks measure we can say that users on page www.cs.umn.edu/admissions/graduate/index.php are more likely to go to pages <http://www.cs.umn.edu/index.php>, <http://www.cs.umn.edu/admissions/graduate/checklist.php>, <http://www.cs.umn.edu/about/contact.php> or <http://www.cs.umn.edu/admissions/index.php> than any other page. Similarly, it is also possible to find out the set of pages to which the users are least likely to go to. These results can be very helpful in making recommendations to users. From the user's perspective, the links that have high Usage Aware Average-Clicks scores are nearer to the index page than those that have lower scores. Such results are significant in link analysis.

6. Evaluation Methodologies

There are a number of ways of analyzing the results obtained from our method. The significance of the distance between pages can be tested against the Domain Expert's Views or the User's Views. The Domain Expert's view can be obtained by designing test cases that capture the distances between randomly sampled pages. The expert can then be asked to evaluate the distances obtained by using both the approaches. The idea is to be able to verify that the distances obtained by using the Usage Aware Average-Clicks method, match his view of distances (or similarity between pages) more closely than those obtained from the Average-Clicks method.

Distance from

<http://www.cs.umn.edu/admissions/graduate/index.php>³

Table 1: Comparison of results from Average-Clicks and Usage Aware Average-Clicks

Destination Page	Average-Clicks	Usage aware Average-Clicks
http://www.cs.umn.edu/index.php	0.0566667	0.000612
http://www.cs.umn.edu/admissions/graduate/evaluation.php	0.0566667	0.002460
http://www.cs.umn.edu/admissions/graduate/procedure.php	0.0566667	0.002460
http://www.cs.umn.edu/admissions/graduate/hecklist.php	0.0566667	0.000612
http://www.cs.umn.edu/admissions/graduate/fellowships.php	0.0566667	0.002460
http://www.cs.umn.edu/admissions/graduate/transfers.php	0.0566667	0.056666
http://www.cs.umn.edu/admissions/graduate/application.php	0.0566667	0.003690
http://www.cs.umn.edu/admissions/graduate/faculty.php	0.0566667	0.001228
http://www.cs.umn.edu/about/contact.php	0.0566667	0.000612

³ Please note that in the tables, the distance values in the two columns are on different scale. Hence comparing columns doesn't make sense. The relative distances within the column can be compared for analysis

http://www.cs.umn.edu/admissions/index.php	0.0566667	0.000612
http://www.cs.umn.edu/degrees/grad/index.php	0.0566667	0.001228

Two different approaches can be used to evaluate the User Views. In the first approach, we can automate and verify the distances calculated against user logs which are different from the logs used for calculating the distance values. In the second approach, we can evaluate the results by distributing questionnaires to users. This is similar to the approach proposed in the original paper [18]. The idea is to randomly sample web pages from the website and segregate them into groups based on their context. We can then calculate the distances between each of these pages. The users of the website can be asked to rate the pages in a particular context. The two results can be compared for further analysis.

In this report we use its predicting power as a measure of its capability to measure distances accurately. The idea behind using Usage Aware Average-Click scores in a recommender system is that pages that are close (smaller Usage Aware Average-Clicks Dist) to each other are given higher similarity scores than pages that are farther apart (larger Usage Aware Average-Clicks Dist). Hence to recommend a page P_i from $\langle P_2 \dots P_n \rangle$ to a user who is on page P_1 , the Usage Aware Average-Click distance of P_i from P_1 should be the minimum among that from each of $\langle P_2 \dots P_n \rangle$. We test its predicting capability by incorporating it into a Recommendation Engine which uses both usage pattern and link structure to make recommendations [3] and measure the quality of recommendations made. Detailed description of the Architecture and working of the recommendation can be found in [3].

We followed a similar approach to testing as done in [3]. The performance of this model was compared against the model that uses a '2,-1' scoring model [3] which gives a score of 2 for a match and -1 for a mismatch and the results of the various experiments are shown in the graphs that follow. The only way in which the models differ is in the way similarity scores are calculated between web pages and all the other parameters of the Recommendation Engine remain a constant.

Web logs from the CS server were filtered to get meaningful sessions. A part of the session data was used to train the model and then the model was tested on the remaining sessions. The next page that will be accessed was predicted for the test sessions and if the predicted page was actually accessed later on in the session, it was considered a hit.

The definitions of the various measures used to measure the effectiveness of these models as taken from [3] are restated below:

- **Hit Ratio (HR):** Percentage of hits. If a recommended page is actually requested later in the session, we declare a hit. The hit ratio is thus a measure of how good the model is in making recommendations.
- **Click Reduction (CR):** Average percentage click reduction. For a test session $(p_1, p_2, \dots, p_i, \dots, p_j, \dots, p_n)$, if p_j is recommended at page p_i , and p_j is subsequently accessed in the session, then the click reduction due to this recommendation is,

$$\text{Click reduction} = \frac{j-i}{i}$$

A High the hit ratio indicates good quality recommendations

6.1 Comparison of Results

In the following figures, we refer to the ‘2,-1’ model as Session Similarity Model (SSM) and our model as Link Aware Similarity Model (LASM).

The following box-plots and graphs compare the two models based on the Hit Ratio. The performance of both the models was recorded when the number of required recommendations was set to 3, 5 and 10. Both the models were trained on 1000 sessions and the Clustered Sessions [3] are represented as ClickStreams into 10, 15 and 20 clusters. Once the models were trained, they were tested on a different set of user sessions. Each of these experiments was repeated 5 times (using a different set of training sessions) to check the consistency of the results. Also, a t-test was done in each of the case to show that the results of the two experiments were statistically different. The t-test is a statistical test which computes the probability (p) that two groups of a single parameter are members of the same population. A small (p) value means that the two results are statistically different.

The above procedure was repeated for 3000 training sessions as well.

The X-axis of both the box plots and the line graphs shows the number of recommendations made and the Y-axis shows the Hit Ratio corresponding to the number of recommendations made. The box plot shows the distribution of Hit ratio values for different input data. The average value of Hit Ratio across the different experimental runs is used to plot the line graph.

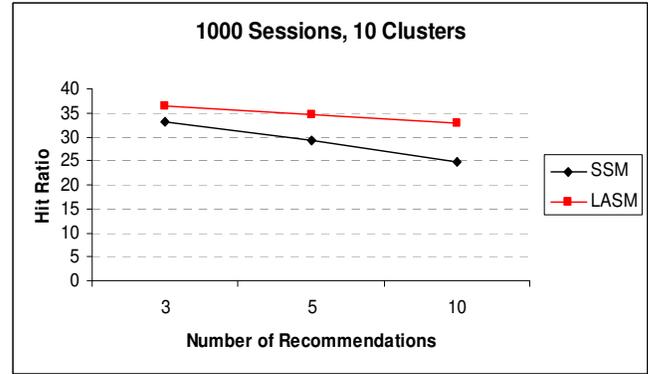
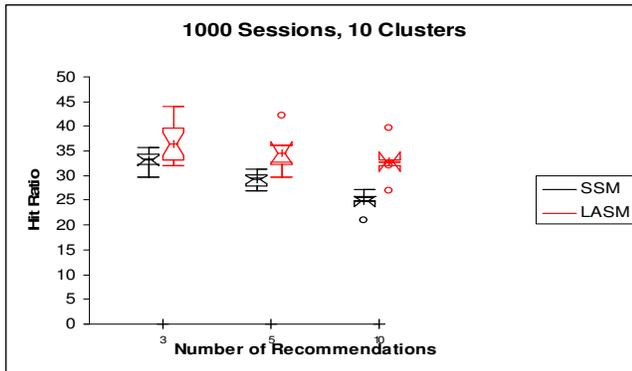


Figure 5: Hit Ratio Vs No. of Recommendations for 1000 sessions, 10 clusters

Table 2: t-test scores for 1000 sessions, 10 clusters

Recommendation	3	5	10
p value	0.123242	0.030262	0.006292

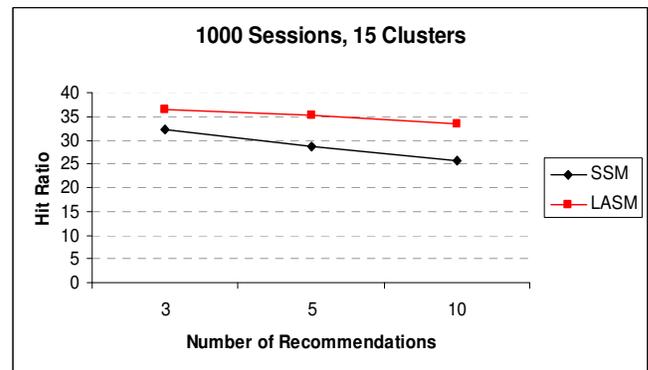
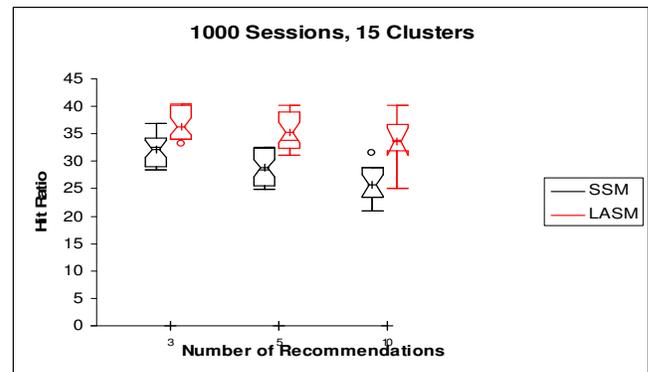


Figure 6: Hit Ratio Vs No. of Recommendations for 1000 sessions, 15 clusters

Table 3: t-test scores for 1000 sessions, 15 clusters

Recommendation	3	5	10
p value	0.053543	0.014464	0.020082

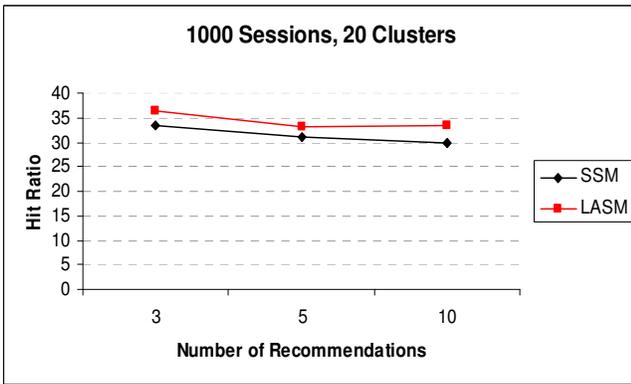
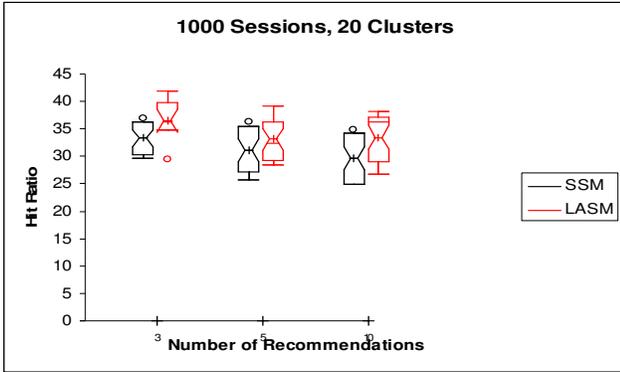


Figure 7: Hit Ratio Vs No. of Recommendations for 1000 sessions, 20 clusters

Table 4: t-test scores for 1000 sessions, 20 clusters

Recommendation	3	5	10
p value	0.04985	0.224891	0.125186

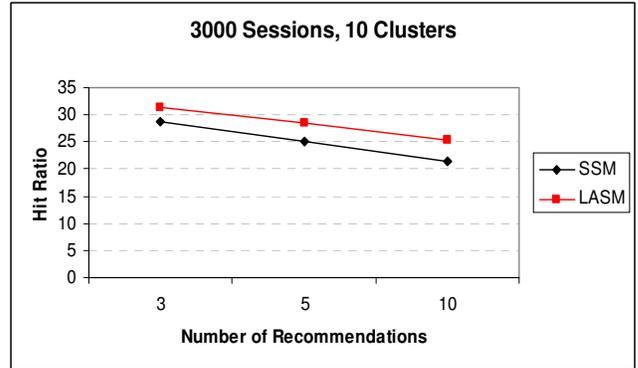
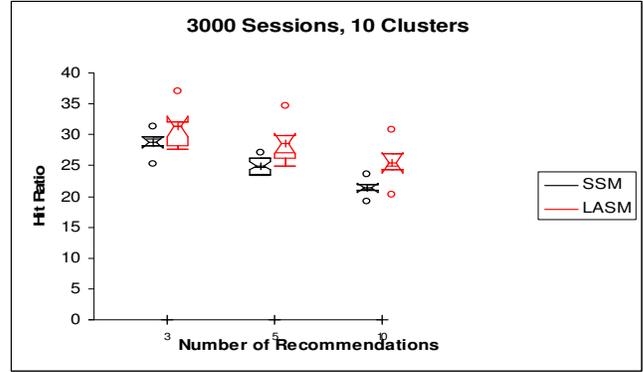


Figure 8: Hit Ratio Vs No. of Recommendations for 3000 sessions, 10 clusters

Table 5: t-test scores for 3000 sessions, 10 clusters

Recommendation	3	5	10
p value	0.122187	0.055483	0.039619

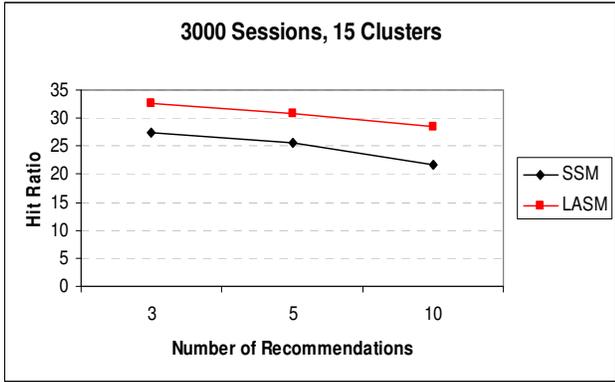
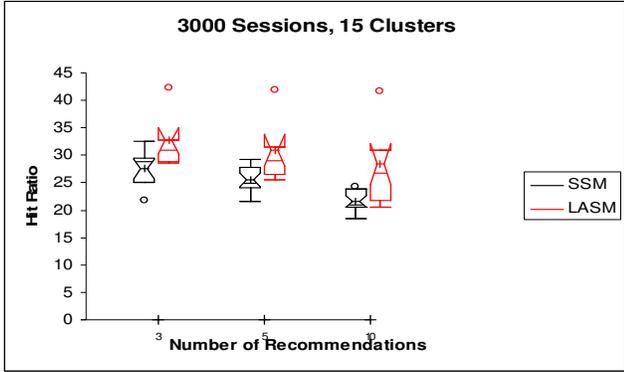


Figure 9: Hit Ratio Vs No. of Recommendations for 3000 sessions, 15 clusters

Table 6: t-test scores for 3000 sessions, 15 clusters

Recommendation	3	5	10
p value	0.070035	0.076632	0.078475

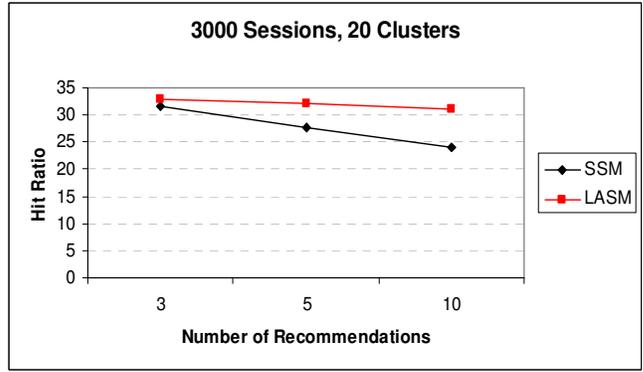
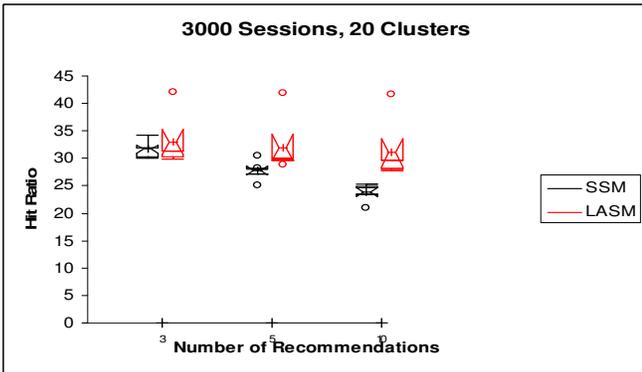


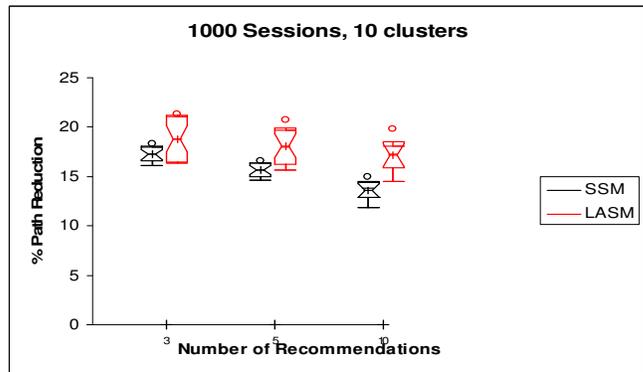
Figure 10: Hit Ratio Vs No. of Recommendations for 3000 sessions, 20 clusters

Table 7: t-test scores for 3000 sessions, 20 clusters

Recommendation	3	5	10
p value	0.300533	0.082899	0.02874

From the graphs and the t-test results it is evident that our model performs better in all the cases. While the ‘2,-1’ method attains a hit ratio of 25% to 30%, the hit ratio obtained for our method is about 40% on an average. This improvement is significant considering the fact, on 100% scale, this is an improvement by 20-25%. Hence, not only our model gives better recommendations but also proves that domain information like link graph is very important in performing usage analysis.

Next, we give the improvement obtained for the measure “Path Reduction Ratio”. Figures 11 through 13 depict the results. The X-axis corresponds to the number of recommendations made and the Y-axis corresponds to the % Path Reduction. The Average % Path Reduction across the different runs is used to plot the line graphs. Here again we see about a 20% improvement on a relative basis. The box-plots and graphs for the experiments using 1000 training sessions are shown below:



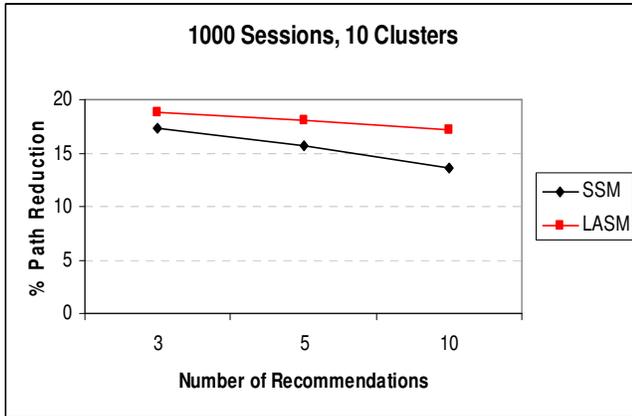


Figure 11: % Path Reduction Vs No. of Recommendations for 1000 sessions, 10 clusters

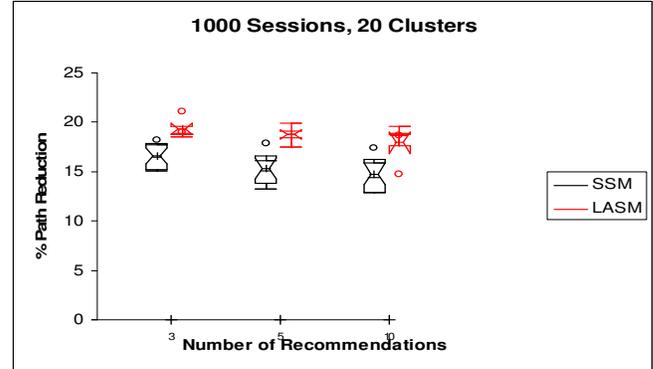


Figure 13: % Path Reduction Vs No. of Recommendations for 1000 sessions, 20 clusters

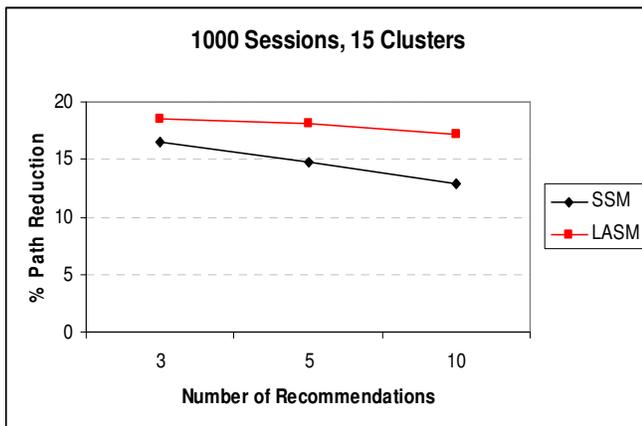
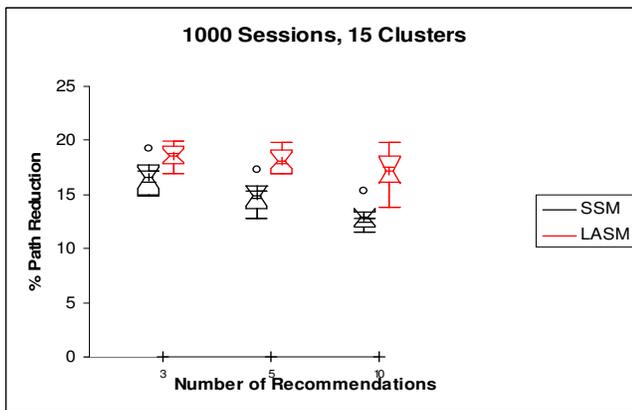


Figure 12: % Path Reduction Vs No. of Recommendations for 1000 sessions, 15 clusters

7. Conclusions and Future Work

In this paper we have proposed an extension to the Average-Clicks Algorithm which uses the Usage Data obtained from the web logs to assign appropriate weights to links on a page. The experiments show that popular links are given higher weights compared to less popular ones rather than just assigning equal static weights to all the links on a page. We used this algorithm in a Recommendation Engine for the Intranet Domain and found that recommendations made using this method were much superior to those made using the '2,-1' scoring method for similarity between web pages.

In the future we plan to verify the accuracy of the results by distributing questionnaires to the web users as well as to the domain experts and evaluating the answers. Verifying with the Domain Experts/User's view is a good idea because the usage logs might not be very representative at all times and the Domain Expert/User's judgment will be better.

Also, we plan to compare the distances obtained from this method with those obtained from an algorithm based on concept hierarchy [12,3] and usage information. We would also like to fine tune the Recommendation Model to consider domain knowledge in addition to usage information to get higher Hit Ratios.

8. References

- [1] A. BRODER, R.KUMAR, F.MAGHOUL, P.RAGHAVAN, S.RAJAGOPALAN, R.STATA, A.TOMKINS, AND J.WIENER - Graph structure in the web. In Proc, 9th WWW Conf., 2000
 - [2] ALAN BORODIN, GARETH O. ROBERTS, JEFFREY S.ROSENTHAL, AND PANAYIOTIS TSAPARAS - Finding authorities and hubs from link structures on the world wide web. In World Wide Web, pages 415-429, 2001
 - [3] AMIT BOSE, KALYAN BEEMANAPALLI, JAIDEEP SRIVASTAVA, AND SIGAL SAHAR. Incorporating Concept Hierarchies into Usage Mining. March 20, 2006 http://www.cs.umn.edu/tech_reports_upload/tr2006/06-009.pdf
- Accepted for full presentation at *WEBKDD'06*, August 20, 2006, Philadelphia, Pennsylvania, USA
- [4] BU OZTEKIN, L ERTOZ, V KUMAR, J SRIVASTAVA - Usage Aware PageRank - World Wide Web Conference, 2003 - www2003.org

- [5] COOLEY, R., SRIVASTAVA, J., AND MOBASHER, B. WEB MINING - Information and pattern discovery on the world-wide web. In 9th IEEE International Conference on Tools with Artificial Intelligence (November 1997)
- [6] ERIC WARD, How Search Engines Use Link Analysis - A special report from the Search Engine Strategies 2001 Conference, November 14-15, Dallas, TX. Dec 2001
- [7] GOOGLE <http://www.google.com/>
- [8] J.M. KLEINBERG, R.KUMAR, P.RAGHAVAN, S. RAJAGOPALAN, AND A.S. TOMKINS - The web as a graph: measurements, models, and methods. In Proc. Of the International Conference on Combinatorics and Computing 1999
- [9] JOEL C. MILLER, FRED SCHAEFER, GREGORY RAE - Modifications of Kleinberg's HITSAlgorithm Using Matrix Exponentiation and Web Log Records. In ACM SIGIR Conference, pages 444--445, September 2001
- [10] JON M. KLEINBERG 1999: Authoritative sources in a hyperlinked environment (Journal of the ACM (JACM), Volume 46, Issue 5, pages: 604 - 632)
- [11] LAWRENCE PAGE, SERGEY BRIN, RAJEEV MOTWANI, TERRY WINOGRAD 1998: The PageRank Citation Ranking: Bringing Order to the Web. (Stanford Digital Library Technologies Project)
- [12] LEE, J. H., KIM, M. H., AND LEE, Y. J - Information retrieval based on conceptual distance in IS-A hierarchies. Journal of Documentation (1993), 49 (2), 188-207
- [13] M. RICHARDSON AND P. DOMINGOS - The intelligent surfer: Probabilistic combination of link and content information in pagerank. In Advances in Neural Information Processing Systems, volume 14. MIT Press, Cambridge, MA, 2002
- [14] SERGEY BRIN AND LAWRENCE PAGE - The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7):107-117, 1998
- [15] SRIVASTAVA, J., COOLEY, R., DESHPANDE, M., AND TAN, P. N - Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations (2000), 1(2):12- 23
- [16] T. HAVELIWALA - Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE Transactions on Knowledge and Data Engineering, 2003
- [17] University of Minnesota, Computer Science Department website. <http://www.cs.umn.edu>
- [18] Y MATSUO, Y OHSAWA, M ISHIZUKA - Average-Clicks: A New Measure of Distance on the World Wide Web- Journal of Intelligent Information Systems, 2003