www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 9 September, 2014 Page No. 8341-8346

Combination of Markov and MultiDamping Techniques for Web Page Ranking

Sandeep Nagpure,
Student,RKDFIST,Bhopal, Prof. Srikant Lade, RKDFIST,Bhopal

Abstract— Web page Re-ranking has been widely used to reduce the access latency problem of the Internet. However, if most prefetched web pages are not visited by the users in their subsequent accesses, the limited network bandwidth and server resources will not be used efficiently and may worsen the access delay problem. Therefore, it is critical that we have an accurate re-ranking method during prefetching. The technique like Markov models have been widely used to represent and analyze user's navigational behavior. One more technique is multidamping which utilize the stochastic matrix for the ranking of the pages. This paper give new algorithm for the page ranking different featurea combination of the markov as well as the multidamping method.

Index Terms— Information *Extraction, page prediction, web re-ranking, web mining.*

INTRODUCTION

With the increase in the use of the internet on daily basis. Importance of the web world is high. As large amount of work is done on net for the transparency and quick. As the importance introduce load in the sites for work and with limited sources one has to manage things in available resource. So other way of optimizing sites is to learn the user behavior pattern for presenting the next page on the other side of the server that is client end. So to improve web access time or in other words in order to reduce latency time for displaying page web recommendation model is built. As the access time of the web decreases then number of visitors also increase and the popularity of that web increases automatically which is the basic requirement of most of the sites

Web mining: For above requirement few steps need to be done in web mining that is of pre-processing which is required to control the result quality as the input data for processing should be fine enough to get desired output. Then the mining algorithm should be generating that is a steps for performing a web optimization which generate patterns. Once number of patterns are collected then analysis of these is done as it will actual output [4, 7].

In order to work for mining some features need to be find in the similar fashion web mining also require features like content, web logs and structure. So learning is done from these features evaluation where each feature contain special web information of the site such as in web log it contain users behavior when it visit on the sites store information like page sequence for accessing, time date, protocol, etc. In short store all the visitors information on the site with date and time. Second feature is the web content this is nothing accept the content present on the page in form of text. As different page have different information so each page is important. Then the third feature is the structure of the site it can be understand as the linking of the different pages of the page to one another. All the feature play important role in mining of the web as they give patterning and reasons for taking the decision.

In Web prediction, main challenges are in both preprocessing and prediction. Preprocessing challenges include handling large amount of data that cannot fit in the computer memory, choosing optimum sliding window size, identifying sessions, and seeking/extracting domain knowledge. Prediction challenges include long training/prediction time, low prediction accuracy, and memory limitation.

I. Related Work

As the information available for the prediction of the web or for optimizing mining the classification of the algorithms are done there are two main category first is the prediction of the future which is analyzed by the previous web access pattern [2]. First category is algorithms that use markov for this and second is algorithms that use data mining techniques such as clustering, association rule, etc.[6]. Large number of prediction algorithms based on Markov models are found in the literature and some of them provide high precision predictions but at the cost of extreme computation and lot of memory consumption. The data mining based algorithms consume the resources even.

One more category of feature is web content that is also use for ranking so this group of algorithm makes use of the web content to make ranking. Then in combination of both the feature is done where web content and web logs both are use for ranking [3]. This is done by the use of the web content as the keywords then web logs as the user behavior [1]. Many paper are done on this concept as well which produce
Good results.

As the presentation of a group for any kind of search is done by the ranks given to the participants [4]. In few works when the search results produce ambiguous situation then those results need to be rectify to a particular kind of cluster or group base on the similarity.

One more approach is done in the [5] where depend on the user interest and behavior ranking is done as depend on the query pass by the user to search on the web . One more class is depend on the how user move on the web kind of pattern it follow of opening different pages. Other works focus on tagging queries with some predefined concepts to improve feature representation of queries. However, since what users care about varies a lot for different queries, finding suitable predefined search goal classes is very difficult and impractical.

In [6] different methods of the query optimization is done by adding new keywords to the typed query. Here a query manager need to generate that read the query and update it base on the database it have. Then categorization of the query is done as it will improve results quality and processing time get reduce.

Computing HITS Algorithm [15]: Two weights are assigned to each page P : a non-negative authority weight and a non-negative hub weight. The invariant that the weights of each type are normalized so their squares sum to 1. The pages with a larger ά value are ”better” authorities, as the pages with a larger authority weight are ”better” hubs.

II. Baground

MARKOV MODEL

In [7] Different actions performed by the user while surfing the web so maintain this information in the backgroung is done by the web log feature, where the user or visitor transition are store in form of name of the pages it visit as a pattern. Here web logs are utilize for the page importance by the markov modal where it select a sequence as the order of the model the specify the target which is base in the previous action performed by the user obtain from the web log feature of the web.

In case of the single action done by the user is taken under consideration is term as the First order markov modal. If the last two action are performed by the visitor or user is taken under consideration then this is Second Order markov Modal. In the similar fashion Kth markov modal is done.

The most commonly used approach is to use a *training* set of action-sequences and estimate each *tji* entry

based on the frequency of the event that action *ai* follows the state *sj*. For example consider the *web-session WS*2 (*P*3; *P*5; *P*2; *P*1; *P*4) shown in Figure 2. If they are using *first-order Markov model* then each state is made up of a single page, so the first page *P*3 corresponds to the state *s*3. Since page *p*5 follows the state *s*3 the entry *t*35 in the TPM will be updated. Similarly, the next state will be *s*5 and the entry *t*52 will be updated in the TPM. In the case of higher-order model each state will be made up of more than one actions, so for a second-order model the first state for the *web-session WS*2 consists of pages {*P*3; *P*5} and since the page *P*2 follows the state {*P*3; *P*5} in the web session the TPM entry corresponding to the state {*P*3; *P*5} and page *P*2 will be updated. Once the transition probability matrix is built making prediction for web sessions is straight forward. For example, consider a user that has accessed pages {*P*1; *P*5; *P*4}. If they want to predict the page that will be accessed by the user next, using a first-order model, we will first identify the state *s*4 that is associated with page *P*4 and look up the TPM to find the page *pi* that has the highest probability and predict it. In the case of our example the prediction would be page *P*5.

Web Sessions:

*WS*1: {*P*3; *P*2; *P*1}

*WS*2: {*P*3; *P*5; *P*2; *P*1; *P*4}

*WS*3: {*P*4; *P*5; *P*2; *P*1; *P*5; *P*4}

*WS*4: {*P*3; *P*4; *P*5; *P*2; *P*1}

*WS*5: {*P*1; *P*4; *P*2; *P*5; *P*4}

1st Order	P1	P2	P3	P4	P5
S1={P1}	0	0	0	2	1
S2={P2}	4	0	0	0	1
S3={P3}	0	1	0	1	1
S4={P4}	0	1	0	0	2
S5={P5}	0	3	0	2	0

2nd Order	P1	P2	P3	P4	P5
{P1;P4}	0	1	0	0	0
{P1;P5}	0	0	0	1	0
{P2;P1}	0	0	0	1	1
{P2;P5}	0	0	0	1	0
{P3;P2}	1	0	0	0	0

2nd Order	P1	P2	P3	P4	P5
{P2;P5]	0	1	0	0	0
{P2;P4}	0	0	0	0	1
{P4;P5}	0	2	0	0	0
{P5;P2}	3	0	0	0	0
{P3;P4}	0	0	0	0	1

Figure 1. Representing the markov modal for paging

Multi-Damping Method:

In [9] Let Yij is an adjacency matrix for the graph of nodes. Where i represent the node after which j node is choosen by the surfers with probability p'.

P' = (Vj / Vi_total) = (number of logs contain j node after i node / total number of logs which contain i node)

In this algorithm first Zk is calculate which is the damping coefficient & G(μ) is the google matrix. stochastic matrix *S* := *P* + Y. For a random web surfer about to visit the next page, the damping factor $\mu \in [0, 1]$ is the probability of choosing a link-accessible page. Alternately, with probability $1 - \mu$, the random surfer makes a transition to a node selected from among all nodes based on the conditional probabilities in vector *v*. As an example, for the case of LinearRank for $k = 3$, the damping coefficients are $\xi 0 = 2/5 = 1 - 3/5$, $\xi 1 = 2/4*3/5 = 3/55$ (1 – 2/4), $\xi 2 = 2/4*2/5 = 3/5*2/4$ (1 – 1/3)and $\xi 3 = 2/4*1/5 = 1/3*2/4*3/5$. This clearly identifies $\mu 1 = 1/3$, $\mu 2 = 2/4$ and $\mu 3 = 3/5$ as the corresponding damping factors. M is damping factor = ($\mu 1$, ..., μk).

Proposed Algorithm

Input: Web_log, step

Output: Rank

1. Web_logs←preprocess(Web_logs)
2. M←Markov_modal(Web_logs)
3. P' = (Vj / Vi_total)
4. S=P+w+M // w is random weight
5. Loop 1:step
6. Require: Zk := {ξj ≥ 0, j = 0, ..., k} finite set of coefficients defining or approximating the functional ranking.
7. Normalize: If $\Sigma_{j=0}^{k} = \xi_j < 1$ then

$$\texttt{add\_cor}(\mathcal{Z}_k) \quad := \quad (\xi_0, \ldots, \xi_{k-1}, \xi_k + 1 - s)$$

   Zk ← add cor(Zk)
   end if
8. Encode: Generate damping factors Mk, e.g. using recurrence.

$$\mu_j \quad = \quad 1 - \frac{1}{1 + \frac{\rho_{k-j+1}}{1-\mu_{j-1}}}, j = 1, ..., k,$$

Where $\rho_k \quad = \quad \frac{\xi_k}{\xi_{k-1}}$

9. Rank = $G(\mu_{k-j+1}) = (\xi_k S^k + P_{k-1}(S))M_k$
10. w = M_k
11. EndLoop

III. Evaluation Parameter

All algorithms and utility measures were implemented using the MATLAB tool. The tests were performed on an 2.27 GHz Intel Core i3 machine, equipped with 4 GB of RAM, and running under Windows 7 Professional. Experiment done on the datasets are available from http://law.dsi.unimi.it/datasets.php dataset which have collection of web logs with page names.

In order to evaluate this work there are different parameter present for the different techniques. The best parameter which suit this work Kendall Correlation between the rank geneate by the iteration and the Actual Rank. Kendall's rank correlation provides a distribution free test of independence and a measure of the strength of dependence between two variables. Spearman's rank correlation is satisfactory for testing a null hypothesis of independence between two variables but it is difficult to interpret when the null hypothesis is rejected. Kendall's rank correlation improves upon this by reflecting the strength of the dependence between the variables being compared.

Consider two samples, x and y, each of size n. The total number of possible pairings of x with y observations is n(n-1)/2. Now consider ordering the pairs by the x values and then by the y values. If x3 > y3 when ordered on both x and y then the third pair is concordant, otherwise the third pair is discordant. S is the difference between the number of concordant (ordered in the same way, nc) and discordant (ordered differently, nd) pairs.

Tau (t) is related to S by:

$$\tau = \frac{n_c - n_d}{n(n-1)/2}$$

For Comparing proposed work it is compare with other ranking method such as Toatal Rank, Linear Rank, Generalized Hyperbolic Ranking from [9].

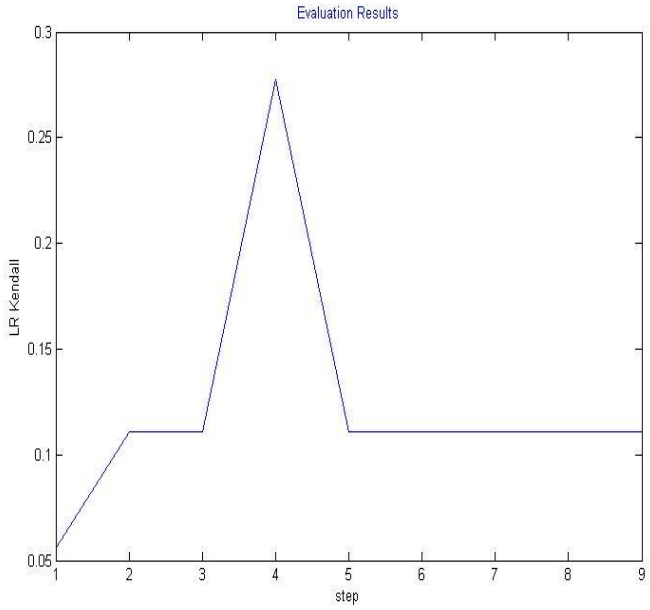

Fig. 2. LR: KendallTau versus iteration step for top-9 ranked nodes

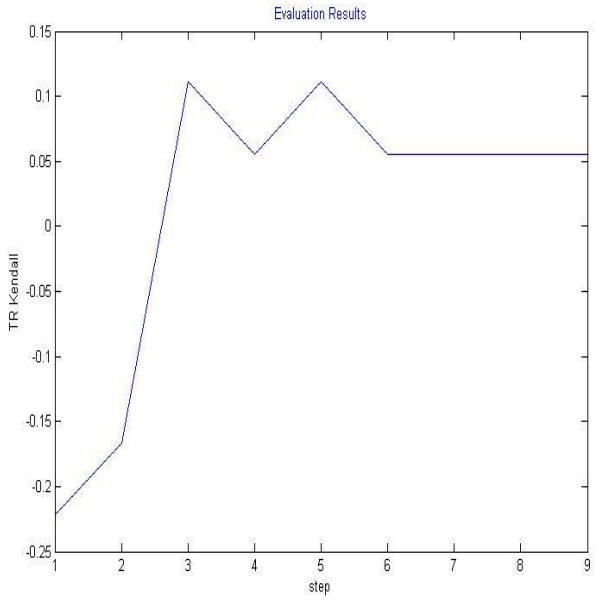

Fig. 2. TR: KendallTau versus iteration step for top-*9* ranked nodes

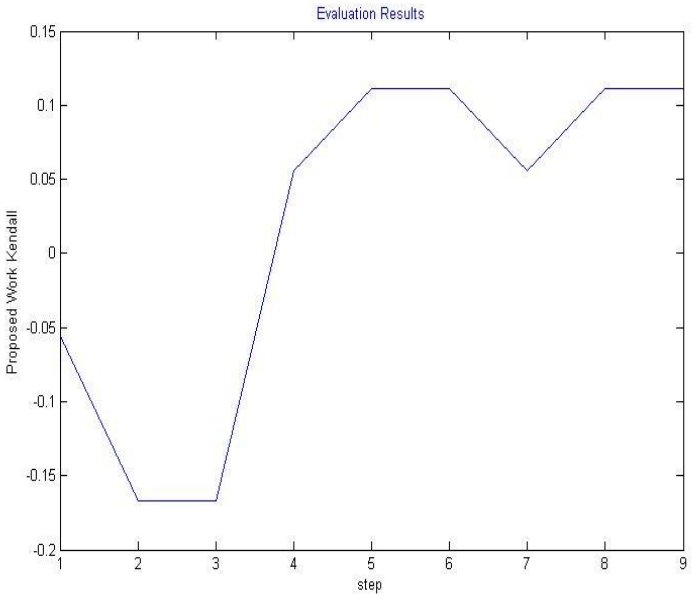

Fig. 2. Proposed Work : KendallTau versus iteration step for top-*9* ranked nodes

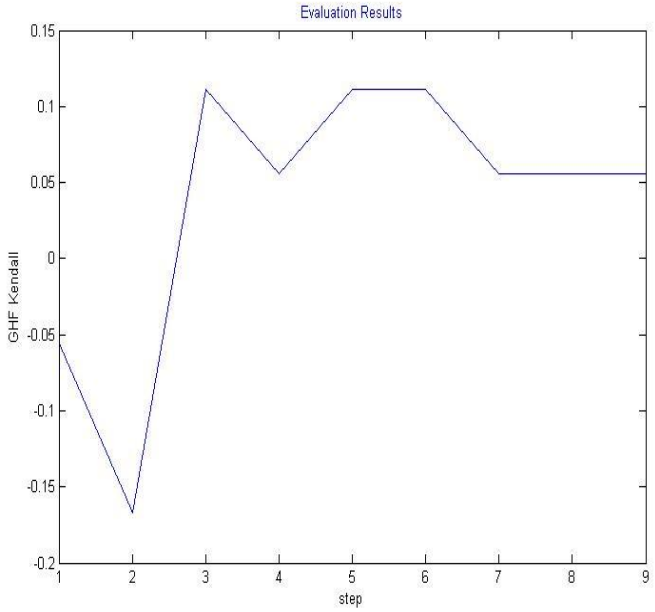

Fig. 2. GHR: KendallTau versus iteration step for top-*9* ranked nodes

Above graph shows that as the method of makov modal introduce in the proposed work of the rank generation stability achive quickly and in more stability with the other methods. As seen in the Linear rank method maximum value of the kandallTau is 0.3 while in the case of the Total rank this method is raise upto the 1 but stability is not achive then in the similar fashion GHR is also implement and result of kendalltau is quit acceptable as it give value of the 1 with small stability but in the case of the Proposed work kendalltau value not only reaches at 1 but higly stable as the iteration go further.

IV. Conclusions

With the increase of the internet user day by day it is necessary for the server to adopt some method which focus on this work. Here Ranking make the work efficient for fetching the page, as the work use pattern from the weblogs obtain that make the ranking better than the previous work, by the use of markov concept results are much better and can be work on any site.

References

1. Brian D.Davison, "A Web Caching Primer" IEEE INTERNET COMPUTING 2001
2. J. Dom_enech, J. Sahuquillo, J. A. Gil & A. Pont. The Impact of the Web Pre-fetching Architecture on the Limits of Reducing

User's Perceived Latency. Proc. of the International Conference on Web Intelligence, 2006.

3. D. Duchamp. Pre-fetching Hyperlinks. Proc. of the 2nd USENIX Symposium on Internet Technologies and Systems, 1999.