Work-in-Progress
A Punishment/Reward Based Approach to Ranking

Pedram Ghodsnia
University of Tehran
Karegar Shomali, Tehran, Iran
+982166521356
pedram@pedram.ir

Ali Mohammad Zareh Bidoki
University of Tehran
Karegar Shomali, Tehran, Iran
+982166946927
zare_b@ece.ut.ac.ir

Nasser Yazdani
University of Tehran
Karegar Shomali, Tehran, Iran
++982161114177
yazdani@ut.ac.ir

ABSTRACT
One of the important challenges in current search engines is dealing with the "rich get richer" problem. In popularity-based ranking algorithms like PageRank, due to considering the structure of the web as the measure for ranking the pages, newly-created but highly-qualified pages are effectively disregarded shoot out, and can take a very long time before becoming popular. In this paper we present a new punishment/reward based approach that adds a new dimension to the PageRank model for reducing the effect of the rich get richer problem using implicit feedback of visitors. In this approach, in addition to considering the structure of links as a page-creator's point of view, we use the page-visitor's view as an important parameter to improve the accuracy of the PageRank algorithm.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *relevance feedback, search process, selection process.*

General Terms
Algorithms, Measurement, Experimentation, Human Factors

Keywords
Ranking, PageRank, Relevance Feedback, Search Engine

1. INTRODUCTION
Interacting with search engines has became a common daily task for million of internet users. Users send a query related to the information they need to a favorable search engine and follow some of the links in the results list and sometimes reformulate their queries and repeat the searching process in a new way.
They click on the links of the results list according to their judgment about the relative quality and relevance of links. This judgment can be considered as implicit feedback and this implicit feedback can serve as a valuable source of information for improving the accuracy of ranking results. Recent studies exploring implicit feedback in controlled environments have shown the value of incorporating this new concept into the ranking process. Most of the previous studies on incorporating implicit feedback or clickthrough data into the ranking process are based on learning methods. For example in [2] a support vector machine and in [8] a neural net tuning algorithm called RankNet has been used.

In this paper we introduce a new approach to incorporating implicit feedback into ranking process. We analyze different aspects of this new approach and show its effect in simulated controlled environments. A distinct advantage of our approach compared to prior methods is reducing the effects of the "rich get richer" problem [6]. Current ranking algorithms like PageRank [11] used in Google [4], HITS [10] and OPIC [1] consider the structure of links in the web as the most important measure for ranking pages, which results in popular web pages receiving more popularity in time. In [6], a solution has been proposed that modifies the PageRank formula. The modification applies the difference between two snapshots of PageRank values to the PageRank algorithm. In [7], two models have been introduced to represent how users discover new web pages: the Random-Surfer model and the Search-Dominant model. In Random-Surfer, a user finds new pages only by surfing randomly, while in the Search-Dominant model, users find new pages using only a search engine. They have also found that it takes 60 times longer for a new page to become popular under the Search-Dominant model compared to Random-Surfer. Our motivation for this work is to understand how a new punishment/reward based method can provide the chance for newly-created but highly-qualified pages to climb up to the top results and be displayed to users and become popular. In contrast to previous works, in this paper we propose a method in which implicit feedback from users indicate the worthiness of highly-qualified pages and helps them become popular faster than before.

In this paper, after a short discussion about punishment/reward ranking, we will discuss the details of our idea. Then to clarify our approach, we introduce a basic punishment/reward based algorithm and discuss the effectiveness of it. We next talk about the issues regarding this basic algorithm and improve our algorithm to overcome those issues. Afterwards, we provide a new method for combining the PageRank metric with any other desired ranking metric. Using this method, we combine the punishment/reward ranking metric with PageRankl. Then, we describe the results of the simulation of this new model. Finally, we introduce some suitable areas for future study and summarize our work.

INFOSCALE 2007, June 6-8, Suzhou, China

DOI 10.4108/infoscale.2007.85

2. PUNISHMENT/REWARD CONCEPT

One of the most common ways to distribute power in human societies is democracy. The idea of democracy results the participation of entire population in electing a subset of people as a suitable group for having the authority and power to control and lead the whole society. In the same manner, in the web society, there is a similar pattern but in different context. It seems logical and fair to evaluate the quality of web pages based on the preferences of visitors. Roughly speaking, we can look at a visitor's preferences among search results as votes to those pages. We will continue our discussion regarding this idea by introducing a punishment/reward-based ranking method in which we consider users' preferences as punishment or reward for ranking search results.

According to recent studies [3], users in real web environments only consider the first 10 items of search engine results and eventually click on 2.5 links out of those 10 on average.
The items on the results page displayed to the user can be divided to the following three categories:

- Clicked results, the items the user clicked on.
- Not clicked results, the items the user saw but didn't click on.
- Ignored results, the items that were completely ignored by the user.

Figure 1 illustrates these tree categories of items.
According to a recent eye tracking experiment [3] most users view the result list from top to bottom. When the user reaches some suitable links in the list, after clicking on them, she might ignore the next links and not even look at them at all. It means that the user prefers some pages over others and from her point of view, clicked pages have more quality than not clicked pages. In punishment/reward approach we give a reward to clicked pages and a punishment to not clicked pages. We don't give any reward or punishment to ignored pages. We consider the page following the last clicked link as a not clicked page too.
Now, using the awards and punishments that are assigned to pages based on users' interactions, we will establish a simple algorithm for ranking search results called Basic Punishment/Reward Ranking or BPRR. The main goal of this algorithm is to dynamically decrease the rank of pages with low-quality and consequently push pages of high quality to the top of the list. For simplicity we assume that all users request a specific query q and click on the some links from the returned result list based on their preferences.

Definition 1 (Score) We define the score parameter of each page as the difference between the page's total punishment and total reward.

Definition 2 (Search query process) We define a search query process as the act of submitting a query by a user to the search engine, then sending back a list of pages from the search engine to the user, and finally, having the user click on one or more of the result links.

In the BPRR algorithm after each search query process we increase the score of clicked pages and decrease the score of not clicked pages and then sort the whole list. The scores of ignored pages are not changed. The ordering of the result list converges to a state in which the pages with more quality are positioned at the top. In the other words, pages which the majority of users are interested in will be at the top of the list after an appropriate number of votes by users.

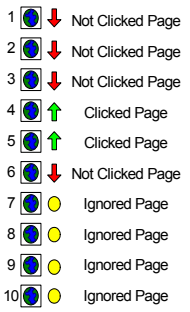

Figure 1

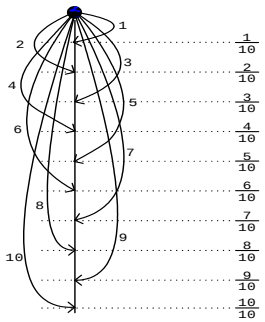

Figure 2

3. PUNISHMENT/REWARD ORDERING

Although BPRR has an interesting mechanism for pushing high quality pages to the top of the list, there are some problems that reduce the practicality of this method for use in a real environment. In this section we will introduce an enhanced version of this method with improved properties, called Punishment/Reward Ordering or PRO.

While in BPRR we had to deal with and manage score values assigned to each page, in PRO, in place of score values, we consider a page's current position on the list. So instead of decreasing a page's score when we punish it, the punishment is applied by dropping the page to a position lower than its current position on the list. Now we want to present this algorithm in details.
Assume that we have an ordered list of n pages called W. During the search query process, the search engine sends back to the user a subset of W called TR (Top Results). Just like BPRR, after each search query process we have to punish a subset of TR called PS (Punishment Subset) and should reward another subset of TR called RS (Reward Subset).
For every member of PR, if it's the i-th time that this member is punished, and $i < c$, the algorithm drops it to position $(i \times n)/c$ of W. Constant c is the drop factor, i.e. it determines the relative position a punished member will drop to.
For example if it's the first time that a member is punished, the algorithm drops it to position $(1 \times n)/c$ of W. If that member is punished for second time, the algorithm drops it to position $(1 \times n)/c$ of W, and so on. For every member of RS if the number of punishments of the member is greater than zero, the algorithm decreases that number by one. Figure 2 shows a graphical representation of the PRO algorithm. In this figure the constant value c is 10. As you can see, if a page is within the top 10, the first punishment drops it to the position corresponding to 1/10 of the size of the entire list. After 10 continuous punishments, the page is placed at end of the list.
The following properties make this algorithm more useful and general than BPRR:

- PRO algorithm converges faster than BPRR.
- Complexity of PRO is less than BPRR because in BPRR we need to sort the whole list after each step, but in PRO we only have some displacements after each step.

- PRO sorts the entire collection based on the quality of the pages, but BPRR only identifies a limited number of top pages.
- The only free parameter in PRO algorithm is constant factor c, but in BPRR we have the reward factor, punishment factor, reward function and punishment function.

In PRO, a small c leads to faster convergence of algorithm to its ideal state, but just like BPRR, faster convergence results in more sudden fluctuations in the ordering of results, and more vulnerability to the misbehaviors of users.

4. COMBINING PRO AND PAGERANK

To clarify our combination approach, first we will introduce an interesting method for combining any ranking metric with the PageRank metric and then we will modify this method for PRO in our experiment in next section.

Assume that we have a web graph structure W as in figure 3. As you can see in figure 4 we have added a virtual page corresponding to each page in the graph and a link from the virtual page to its corresponding page. The resulting graph is called the virtual graph of W.

Definition 3 (Page memory of a page) We define the page memory (PM) of page p as a virtual page corresponding to page p that doesn't exist in the real web graph and has an outgoing link to page p.

Assume that we have a specific metric M other than PageRank for ranking pages. We define vector V such that each element of this vector corresponds to the weight of a page from the web collection based on metric M. The goal of our combination method is combining this weight vector with PageRank.

To do this, first we scale the elements of vector V to the range of PageRank values.

Next, we assign the weight of each page from vector V to its PM in the web graph. Finally, using the following formula, we run the PageRank algorithm.

$$r(i) = d_1 \cdot \sum_{j \in B(i)} \frac{r(j)}{N(j)} + d_2 * V(i) + d_3 . \frac{1}{m} \qquad \text{(Equation 1)}$$

(d1+d2+d3=1 & d3<<d2+d1)

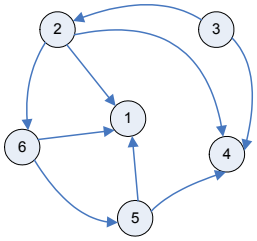

Figure 3

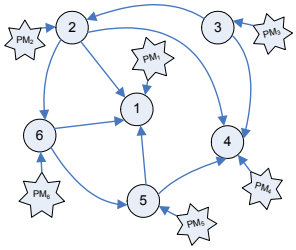

Figure4

In this equation, the term d2*V(i) has been added to the original PageRank formula. d2 is the weight factor of the new metric in the combined algorithm. The novelty of this method is that it propagates the effect of metric M in the web graph.

5. EXPERIMENTAL RESULTS

In this experiment we want to show that our new method reduces the effect of the rich get richer problem. It is equivalent to show that in our new combinational method, high quality new born pages climb up and reach the top of the list in a shorter time than the original PageRank. To do this, first we construct a random web graph in which high quality pages are initially at the end of the ranking list, and after a period of time they climb up to the top of the list because of their high quality. We measure the needed time for these pages to reach the top of the list for both the original PageRank and our new method, and compare the results to show the difference.

Our experiment contains the following initial steps:

- Constructing a random graph structure
- Computing the rank value of each page of the graph based on the original PageRank (PageRank values are scaled between 0 and 1)
- Assigning a quality value to each page in a reverse compared to the PageRank values, In other words, if R(p) is the Page Rank value of page p and Q(p) is the quality of page p then Q(p) = 1- R(p)

After these three steps we have a web graph structure in which the pages with high PageRank values have low quality values and vice versa.

First, we applied the original PageRank algorithm on this web graph structure periodically and at the end of each period, we computed the Kendall's τ between the current ordering of pages and an ideal ordering in which the entire collection is sorted based on the quality of pages. In our experiment each period consists of a specific number of queries that have been sent by virtual users to the search engine. After each query a random set of 10 to 1000 pages will be selected from the collection and sorted based on their PageRank values, and the top 10 pages will be sent back to the virtual user. The user then clicks on some of these pages with a probability equal to their **quality values**. Obviously high quality pages will be clicked with a higher probability than low quality pages. When a user has clicked on a page there is a reasonable probability that she is interested in this page and will create a link from another page of web to this page. So after every fifty clicks on a page (by fifty different users) we create a link from a random page in the collection to this page. In our simulation each period consists of 10000 virtual queries with the above conditions. At the end of each period we apply the PageRank algorithm again. After each period, the ordering of the collection based on PageRank values will be closer to the ideal state in which the entire collection is sorted based on quality values. In the ideal state the PageRank value of each page will be proportional to its quality value. Because the clicks of users on pages are based on their quality, and because the incoming link count of these pages are increasing, after each period, applying the PageRank algorithm leads to more similarity between the current ordering and the ideal ordering. As a result, the value of Kendall's τ will be greater after each period and we can say that the ordering of PageRank values will converge to the ordering of quality values. After applying the PageRank algorithm at the end of each period, we recorded the values of the computed Kendall's τ and used these values to illustrate the graph of figure 5. You can see in this graph the convergence of Kendall's τ to 0.6.

Then we repeat our simulation using our new method instead of the original PageRank. The differences are:

1- After each period, we applied our new method instead of the original PageRank
2- In each period, after each click by a user on a page we reordered that page in the list based on the PRO algorithm and before applying the new method at the end of the period, the positions of the pages were converted to their corresponding value in the weighting vector based on equation "$W(P_i) = 1 - (i/n)$".

The values we used for d1, d2 and d3 factors were 0.7, 0.2 and 0.1 respectively. Like the previous experiment, after applying the algorithm at the end of each period, we recorded the value of computed Kendall's τ and used these values to draw the graph of figure 5.

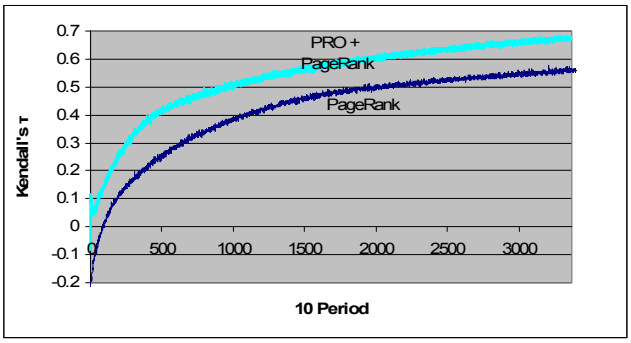

Figure 5

You can clearly see the difference of the speed of convergence of the two methods in the graph. As you see, the time needed to reach a Kendall's τ of 0.6 in the original PageRank is about 3 times more than our new method, meaning we have achieved a significant improvement in reducing the effect of the rich get richer problem. We considered the worst case as our initial state, so our results can be generalized.

6. FUTURE WORKS

There are several opportunities for future research. Investigating the effect of values d1, d2 and d3 in the algorithm and reaching an optimum combination, working on different ways for reordering pages in the PRO algorithm and analyzing the results, study of the combination of other metrics like "traffic information of websites" with the PageRank metric and finally, finding a mathematical model for analyzing the new ranking model and measuring the effect of the free parameters of our method can be some areas of interest for future studies.

7. CONCLUSION

In this paper we introduced a new punishment/reward-based concept for ranking the results of search engines. This idea is originated from the concept of democracy in real human societies. We defined a method for extracting user preferences from page clicks in the top list of the search results and considered these preferences as punishments and rewards for pages and interpreted them as user votes for selecting the best pages for the top list in a democratic web environment. Based on these punishments and rewards, we defined a reordering method for ranking the pages and showed in our experiments that this approach, in combination with PageRank, is a great improvement in reducing the effect of the rich get richer problem. To combine our approach with PageRank, we introduced an interesting method that can be used for the combination of other metrics with the PageRank metric. There are some drawbacks to this approach as well. Any democratic method in a virtual environment can be affected by the misbehavior of users. For example, our approach can be biased by spamming activities causing certain pages to move up on the result list. The study of the effects of spamming in our approach can be considered for future work. This approach can be considered as an improvement of the accuracy of the PageRank metric. Its main contribution to PageRank is considering the opinion of page visitors about the quality of pages in addition to PageRank's own consideration of page creators' opinions.

8. ACKNOWLEDGEMENT

We would like to thank our friends Mohammad Monibi, Naghmeh Ildarjaleh, Farid Amirghiasvand, Pooya Tavakoly and Somayeh Zareh for their contributions to our work.

9. REFERENCES

[1] Abiteboul, S., Preda, M. and Cobena, G. *Adaptive on-line page importance computation*. In Proceedings of the twelfth international conference on World Wide Web, 2003.

[2] Agichtein, E., Brill, E., Dumais, S. *Improving Web Search Ranking by IncorporatingUser Behavior Information*. In proceeding of SIGIR'06, Seattle, Washington, USA.

[3] Agichtein, E., Brill E., Dumais, S. Ragno, R. *Learning user interaction models for predicting web search result preferences*. Proceedings of ACM SIGIR 2006 conference, Seattle, Washington, USA, pages 3-10.

[4] Brin, S. and Page, L. *The anatomy of a large-scale hypertextual web search engine*. In Proceedings of 7th World Wide Web Conference.

[5] Cho, J. a-Molina, H.G. and Page, L. *Efficient crawling through URL ordering*. In Proceedings of the seventh conference on World Wide Web, Brisbane, Australia, 1998.

[6] Cho, J., Roy, S., Adams, R.E. *Page Quality: In Search of an Unbiased Web Ranking*. In Proceedings of 2005 ACM International Conference on Management of Data.

[7] Cho, J., Roy, S. *Impact Of Search Engines On Page Popularity*. In Proceeding of WWW2004, New York, New York, USA.

[8] Joachims, T. *Optimizing Search Engines using Clickthrough Data*. In proceeding of SIGKDD, Edmonton, Alberta, Canada, 2002.

[9] Kendall, M.G. *Rank Correlation Methods*. Griffin, London, England, 1970.

[10] Kleinberg, J. M. *Authoritative sources in a hyperlinked environment*. Journal of the ACM, 46(5):604–632, 1999.

[11] Page, L., Brin, S., Motwani, R., and Winograd, T. *The pagerank citation ranking:Bringing order to the web*. Technical report, Computer Science Department, Stanford University, 1998.