INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

A Brief Survey of Algorithms used by Web-Based Recommendation Systems

Ritesh Raj[1], Rohan Manoj Thakkar[2], Shantanu Mane[3]

[1]K. J. Somaiya College of Engineering, Computer Engineering department,
VidyaVihar (E), Mumbai 400077, India
riteshraj.o@somaiya.edu
[2]K. J. Somaiya College of Engineering, Computer Engineering department,
VidyaVihar (E), Mumbai 400077, India
rohan.t@somaiya.edu
[3]K. J. Somaiya College of Engineering, Computer Engineering department,
VidyaVihar (E), Mumbai 400077, India
shantanu.m@somaiya.edu

Abstract: *A recommendation system is a novel mechanism that is used to employ web mining techniques for identifying the interest areas of the users and recommending new content to them. By using the concept of web usage mining, we can easily mine the information regarding registration and other details left by user access with the user access model. This can further assist in laying a foundation to ease the process of making decisions in organizations pertaining to the category of pages they are more likely to recommend to users having similar interests. The scope of this mechanism is to develop a working system that can recommend pages to the users from the database then use various techniques to estimate users' interest in web pages and ultimately, provide users ability to give tags to a page in order to better describe it.*

Keywords: *Recommendation system, PageRank, Random search, User tagging.*

1. INTRODUCTION

Web-page recommendation estimates new pages on the Web that users will be interested in, while browsing the Web. It is this technique that can help users in order to collect more web pages without having to ask for them in an explicit manner. This mechanism has received a lot of positive reviews in the domain of Web Usage mining. Some research on Web recommendation however considers personalization that is an important feature so as to fulfill several user preferences.

2. ALGORITHMS USED

2.1 Improved Page Rank Algorithm

For overcoming the issues faced in PageRank algorithm, this paper takes into account the user feedback information, then optimizes the formula for PR value, and later adjusts the page PR value accordingly.

Usually search results are first achieved with the help of keywords. Accordingly, the users then decide whether or not have the result page satisfied their expectations – in terms of the summary, title page, and so on. [2] All they click is their needed results. Hence what we do is we make a user-based search option and retain a log of users' clicks. The logs retain a record of the time of users' clicks and all the other data pertaining to the visited URLs. For reducing the chances of occurrences of page cheating and topic shift, we attest a given weight-based parameter with a said PR value while we analyze the click information that we've collected.

2.1.1 Weight of clicks

Depending on characteristics of the PageRank algorithm, we attest a third criterion, i.e., whenever one page gets extra clicks from any possible search-result, it is essential and has a higher PR value. Whenever a user makes a click on a page in search results, it is similar to manually sorting it. This click shows whatever data supplied which has the potential to meet the requirements of a user. [3]

The higher the clicks on any particular result page, the higher is the ability to meet the requirements of other clients, and thus there is a higher importance in these pages. Also, the survey demonstrates that clicks made by a user in the web pages appearing in the search results focus on one or more particular categories of a page. An efficient web-search engine must propose highest number of hits; almost 30% of the highest ever click rate is needed for meeting a 70 ~ 80% of total needs of users. [6] In spite of that, whenever any new page is visited, it has not a single click from a user. Thus, it is important to assign some form of compensation to any new web page, to prevent a possibility of algorithm using PageRank from assigning too much emphasis to any visited page. [4]

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Clicks weight S (i):

S(i) D ln(Nc _1) _D0 …(1)

Here, Nc stands for number of clicks on a page i within a given time period. i stands for the attenuation effect coefficient, and it controls the weight assigned to clicks. It usually sets to 0.3. i denotes a compensation effect's coefficient, and corresponds to the importance attested to any new page - generally sets to one. There is a positive correlation between PR (i) and S (i). Considering the clicks of a user onto searched result pages, many a times attributed to keywords that it looks for.

Based on relative study, we differentiate result pages that a user clicks on - based on their respective keywords. [1] We then calculate the clicks on result pages that are looked for by every keyword, in contrast to summation of clicks on the resultant page, we understand that the frequency of clicks on result pages that were visited by most keywords and the resultant frequency of clicks on resultant pages that were looked by all words resemble.

Thus, result page clicks can be understood as to be identical to clicks of queries. It reduces the space resultant from collecting keywords and user clicks. Moreover, it weakens bad impact resulting from resultant pages click times due to client's randomness.

2.1.2 Weight of click time

Only taking into account the weight of clicks cannot accurately prevent emphasizing on old page. Since a previously visited page stays for much time, chances of click are higher than any newly visited page. Soon, because of uprise of new events or clicks of pages linked with famous events rise rapidly. Once they have lost their heat, owing to previous agglomeration of visits, they take pages that deserve lesser ranking because of losing heat. It will take a lead of those search results. Thus, it becomes essential to bring in weights of the clicks' times.

Prospect of recent activities of link can be believed to determine the weight of time. To see whether or not a page is passive or active lately, it can be determined by current click time. In case assumed that its normal- the important pages' clicks are 1for every time unit. In this case, the web page that is not click per time unit will lose its popularity. Thus, we attest an additional criterion: 4. In case a page isn't clicked recently, the popularity of page is lowered, and its value of PR is lowered. [5]

Also, data of new page is essential relatively, page can repeatedly refresh its data for improving its popularity. Similarly, we estimate that a normal page is clicked once a month.

The click time reset weight is:

T(i) ®1_ET Ta !1 in which T -1 Ta d1 -Tnow _Tlast ®Tnow _Tupdate Tlast NULL Tlast NULL z …(2)

Here, Ta stands for time interval of a web page i. The difference between Tnow and Tlast. Tupdate shows the time required to modify page i. The symbol ù stands for the coefficient of attenuation– it controls time interval's weight, it is set to 0.08333. [9]

There is a negative correlation between T(i) and PR(i).

2.1.3 Formula for Improved PageRank algorithm

Based on similarity of T(i), S(i) and PR(i), we attest the weights of clicks Sc and weights of click time Tc to the PageRank algorithm formula. Thus, we obtain the PR value of the new formula:

$$PR(I) \,.\, PR(i)\ u = k. \frac{Sc(i)}{Tc\,(i)} \quad \ldots(3)$$

The higher the number of clicks of pages, the greater is the PR value and the higher ahead is its position. However, they are opposite.

2.2 Tagging mechanism

The tagging mechanism consists of three parts-

2.2.1 Tag Refinement

Most web pages like Amazon have several tags. Any web page also has at least tens of tags. In reality, majority of user tags in a page are created by limited users, and such user tags don't stand for data on a page. This thing engenders an improper diversification and a lesser efficiency in retrieval of information. So it is essential to remove unimportant tags by pruning them. To be specific, by calculating the w t of each user tag in a page, we estimate their popularity. The click weight of a user tag shows whether the tag is actually useful.

W(tk, p1) = freq(tk,pq)/ ∑Tifreq(t,p1)*(1-a) + ntk/Nt1*a …(4)

Where w(tk,p1) and freq(tk,p1) respectively stand for the click weight and viewing frequency of tag k within webpage i. Ti shows all user tags associated with page i. Symbols ntk and Nt1 reflect total of tags k and the complete number of user tags in page. Few tags are not available in the data of pages even when several users may have used them. [7] There's only minimal content in a page related to videos and pictures. Also, several users choose tagging a page, i.e., tagging multiple words together with no space. Thus, different weights are attributed to tags and data. Since folksonomy is a collective classification that is dependent on users, popularity is added to user

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS…..

choice. Hence the weight of α is set to be 0.3. In case the tag's weight is less than a threshold, the tag is discarded.

2.2.2 Web Page Cluster

Because the purpose of this project is to give users several topics, a vital step is to analyze those pages as they are intricate to various keywords. Some of them stand for one single topic, however many others don't. The project aims at judging the data of web pages using cosine similarity as well as cluster- like pages. If majority of users deploy obvious words, these words are considered to reflect a web page well. Here, we use tags as a vector space. We then contrast the similarities of several web pages. It is not same as other methods. [8] Here, we set the vector space of pages as (w1, w2, w3…, wn). In case the commonness crosses a given threshold, what we do is we determine that the web pages are like and we put them together under one type. Based on earlier formulae, we observe the tag with more click weight can determine commonness better as compared to user tags with lower weight. Thus if many web pages have like tags and the weights of those pages are more, it is understandable to put them together.

To be specific, clustering follows these steps:

- Set first visited web page in first cluster.
- The web page which has not clustered should be compared with each web page in that cluster. In case average amount of similarity surpasses the threshold value, it can be considered.
- A considered page with highest similarity participates in clustering.
- In case a web page is not to be clustered with existing hub, the web page is sent to a brand new cluster.

Clustering is a process that continues as long as all the web pages aren't put into some or the other cluster. Although this method is complicated, it ensures that pages in one cluster keep high similarity. These similar pages can give users perfect tags and pages in two phases - web page search and page recommendation.

2.2.3 Tag Rank

Although major user tags are filtered in the first process, number of user tags in a cluster is huge. Several tags are duplicated. Thus, it is important to remove duplicated tags and to find out the popularity of each tag. We take the 'TF-IDF' method. [9] Frequency of inverse document is essential and it looks to filter useless words (the, a) and pick rare words to express topic. In contrast to requirements, the aim is selecting a user tag as topic user tag for cluster, and the user tag must show the cluster much better than how others do. Dissimilarity is that all user tags in cluster are pruned and most stand for typical data instead of meaningless information.

Therefore, we use 'TF-DF' to determine weight of every tag in a cluster based on relation to data of pages and the ratio of user tag in cluster.

We prioritize user tags in an order based on weights of clicks in cluster. User tag with highest weight becomes the topic tag of cluster. In case the keywords that the user input matches with the tag, the topic of cluster should be suggested as Super-tag.

2.2.4 Tag Recommendation

For recommending best tags and improving efficiency of retrieving information, two new concepts are defined – Subtag and Supertag. Sub-tag means detailed data and Super-tag implies general data.

When a client enters a keyword and searches, Supertags are recommended by the system to choose from. As the client selects dissimilar Super-tags, even dissimilar Sub-tags get suggested. Later, on submission of a particular Subtag and Super-tag, the user is able to see relevant web pages.

To be specific, tags are recommended by these steps:

Search all pages that include target tag relevant to all keywords.

Run 3 modules of the tagging mechanism to prune tags, and to cluster pages and to calculate the TF-DF values for every tag in each cluster.

For every cluster with target tag, calculate the TF-DF value of weight of the user tag and then multiply it by approximate number of users. Rank and then select top k clusters. Then return topic tags as Super-tags to client. If client picks Super-tag, it implies the clients chose a relevant topic. Then a whole set of Sub-tags shall be supplied to the client. Such Sub-tags are those that have more weights for a cluster and such tags are related to target tag. [10] In reality, these Super-tags are different topics which are relevant to input keywords. A client chooses favorite issues using Super-tags, and then their preferences for position by Sub-tags.

2.3 Random Selection

The users are recommended randomly selected URLs from the database, the categories of which fall in the users' selected categories. The main purpose of doing so is to make sure that every page gets a chance to be visited and have its rank incremented to raise its popularity. Pages which a lot of users find worth visiting will obviously have their rank incremented. The users will be able to discover new web pages that they might like.

The ORDER BY RAND() function of SQL works perfectly fine when the number of rows in the table is

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

in the order of a thousand. But this function becomes very slow when you have around 10,000 rows in your table.

In order to ensure fast responses from the database we have created a stored procedure in MySQL. We named this procedure get_rands. get_rands accepts as input parameters an integer value to determine the number of URLs that it has to randomly select and the category of the URLs. get_rands accesses the following tables during its operation :

1. urltable: The table containing all the URLs.
2. catstore: The table in which all the URLs of the required category during operation will be stored.
3. rands: The temporary table in which the result set will be stored.
4. sel_url: This table ensures that a URL is not selected twice by storing the URLs that get selected.The working of the get_rands stored procedure goes about as follows:

Input parameters: count INT, category VARCHAR

1. DROP TABLE rands if it exists.
2. CREATE TABLE rands.
3. INSERT all the URLs from urltable into catstore which match 'category'.
4. LOOP till 'count' is not equal to 0:
   i. INSERT INTO table rands using the following query:

```
INSERT INTO rands(rand_id, rand_url,
 rand_urltitle)
SELECT r1.urlid, r1.ct_url, r1.ct_urltitle
FROM catstore AS r1 JOIN
(SELECT (RAND() *
(SELECT MAX(ct_urlid)
FROM catstore)) AS id) AS r2
WHERE r1.ct_urlid >= r2.id AND
 r1.urlid NOT IN (SELECT sel_urlid
 FROM sel_url) ORDER BY r1.ct_urlid
 ASC
 LIMIT 1;
```

   ii. If a row was inserted into rands then INSERT the id of that URL corresponding to the id in url table into table sel_url. Else repeat the loop.
   iii. Decrement the value of 'count' by 1.
5. The loop goes on until the specified number of URLs (count) is not selected.

Once the procedure has finished executing, the results can be fetched from the rands table. Inserting the selected URLs into table sel_url ensures that a URL already selected randomly is not selected again. Thus, the result set of the get_rands procedure is always made up of a list of unique URLs.

2.3.1 Shares:

Users also have the ability to share pages that they like. A share button is provided for every URL that is recommended to a user. Once the 'Share' button is clicked, the share is recorded. The user's id, the URL's id and category are stored in a table. A user can see what other users have shared by visiting the 'Activity' page. The shared pages are shown according to the user's categories.

2.3.2 Most Preferred Category:

Each hit that a user makes is recorded in the database. This data along with the data about shares is later used to find out a user's most preferred category.

3. COMPARISON OF ALGORITHMS

The characteristics of Improved PageRank Algorithm are:

1. Takes user clicks into account, i.e. the number of hits that a page gets.
2. It is also based on time of the hit, so pages which are more popular during that time period will have their rank increased by a greater amount.
3. Reduces the chances of page cheating, i.e. the developers of a website cannot get a web page's rank wrongly increased.
4. The effects of link spamming are nullified by using this method.
5. Solves the problem of PageRank algorithm related to topic shift, i.e. results not being relevant to what the user wants to search.

Similarly for Tagging Mechanism:

1. It customizes the web page search results to favorite category of web pages.
2. User defined tags incorporate highest level of personalization.
3. Improves speed of browsing since only relevant content is showed.

And for Random Selection:

1. It gives every page a chance to be visited.
2. All the pages get a fair chance to have their rank incremented.
3. Users are able to discover new pages through this method.
4. Gives unique results for each instance of the selection.
5. The method used ensures a fast response from the database even when the size of the data becomes very large.

4. CONCLUSION

With the advancement of the technology software complexity and requirements has been changed in

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

past decades. To develop efficient application, software developers need to follow certain set of rules. Before developing any application developer need to do requirement analysis. If system developed as per the requirements and software development rules are strictly followed then there will be a more chance that efficient software will developed.

ACKNOLWDGEMENTS

We wish to thank K. J. Somaiya College of Engineering for providing us a good opportunity to work on creating a project and presenting our survey on topics of great importance. We would also like to thank our mentor Ms. Poonam Bhogle who teaches at the Computer Engineering Department of the college for sharing valuable information with us about this topic and guiding us in publishing this paper and in the prototype implementation.

References

[1] T. Zhang, B. Lee, S. Kang, H. Kim and J. Kim, "Collective Intelligence-Based Web Page Search: Combining Folksonomy and Link-Based Ranking Strategy," Proc. of 2009 Ninth IEEE International Conference on Computer and Information Technology, pp.166-171, 2009.

[2] Y. Liu, M. Liu, X. Chen, L. Xiang and Q. Yang, "Automatic Tag Recommendation for Weblogs," Proc. of the 2009 International Conference on Information Technology and Computer Science, pp. 546 – 549, 2009.

[3] S. Niwa, T. Doi, and S. Honiden, "Web Page Recommender System based on Folksonomy Mining for ITNGŠ06 Submissions," Proc. of the Third International Conference on Information Technology: New Generations, pp. 388–393, 2006.

[4] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu, "Exploring folksonomy for personalized search," Proc. of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 155-162, 2008.

[5] S. A. Golder, and B. A. Huberman, "Usage Patterns of Collaborative Tagging Systems," Journal of Information Science, Volume 32 Issue 2, pp. 198-208, 2006.

[6] Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, "Personalized recommendation in social tagging systems using hierarchical clustering," Proc. of the 2008 ACM conference on Recommender systems, pp. 259–266, 2008.

[7] Hotho, R. Jäschke, C. Schmitz and G. Stumme, "Information Retrieval in Folksonomies: Search and Ranking," Lecture Notes in Computer Science, Vol. 4011, pp. 411-426, 2006.

[8] Byde, H. Wan, and S. Cayzer, "Personalized tag recommendations via tagging and content-based similarity metrics," Proc. of the International Conference on Weblogs and Social Media, 2007.

[9] Aizawa, "An information-theoretic perspective of tf-idf measures," Information Processing and Management, Vol. 39, pp. 45-65, 2003.

[10] S. O. K. Lee and A. H. W. Chun. "Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid ANN semantic structures," Proc. of the 6th Conference on WSEAS International Conference on Applied Computer Science, pp. 88–93, 2007.