A Co-Recommendation Algorithm for Web Searching

Mingjun Lan, Shui Yu, Ruth Bacher, Wanlei Zhou
School of Computing and Mathematics, Deakin University
221 Burwood Hwy, Burwood, Vic 3125, Australia
{mlan, syu, rsbacher, wanlei}@deakin.edu.au

Abstract

This paper presents an approach called the Co-Recommendation Algorithm, which consists of the features of the recommendation rule and the co-citation algorithm. The algorithm addresses some challenges that are essential for further searching and recommendation algorithms. It does not require users to provide a lot of interactive communication. Furthermore, it supports other queries, such as keyword, URL and document investigations. When the structure is compared to other algorithms, the scalability is noticeably easier. The high online performance can be obtained as well as the repository computation, which can achieve a high group-forming accuracy using only a fraction of web pages from a cluster.

1. Introduction

The World Wide Web, WWW, is growing at an exponential speed and is the most significant media source for most Internet users [1]. How to retrieve the valuable information remains of constant focus. Current researchers are concentrating on subject matter, which includes mechanisms of searching engines, query languages, query expansions, indexing, multimedia Information Retrieval (IR), content-based retrieval, and semantic thesaurus etc. These research investigations will assist in retrieving the necessary information from the WWW. In the real world of Web searching, however, some proposals, such as query expansions, are explored limitedly in commerce because of time consuming and complexity issues [14] [8]. For most WWW users, the query languages are complicated. Therefore the advanced searching options are rarely used. Then, the situation worsens when users do not know how to specify their query. This can result in a user being unable to access any relevant information, after a significant amount of time has been spent exploring the WWW.

Before the solution to this problem is examined, it is useful to understand how a Web search engine is typically organized. Figure 1. illustrates the schema of search engine [4][11].

Some recommended systems collect the navigation history of the user and insert those results into a central repository. Data mining techniques, such as *Surflens* [16], can then be applied to discover hidden information from the repository [2] [7]. Normally, there are two types of hidden knowledge that can be revealed:

a. Do any users access the same group of URLs as each other? Statistically, there must be some relationship among these URLs. They may be of similar content, or they may contain separate components of a related topic.

b. What groups of users read similar web pages? If two users have read a lot of similar pages, we can conclude these two users have similar interests. These results are then used to make further recommendations.

The original assumptions are intuitively correct, but these types of algorithms cannot address issues such as a user changing his interests. According to the evaluation results of [16], the user's navigation history does in fact reflect their interests if the number of good recommendations increases. However, in this case, the algorithm does not work when a user just wants to temporarily navigate on a non-related topic.

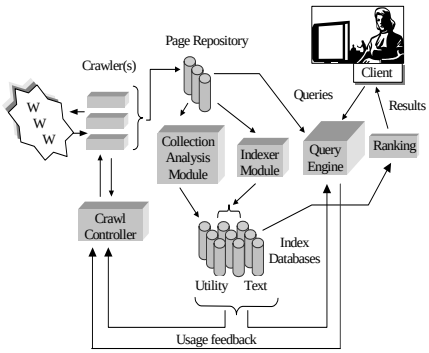

Figure 1. General architecture of search engine

The purpose of this paper is to propose an algorithm called co-recommendation, which involves in indexer and ranking modules. It combines the recommendation rule, the association rule [16], and co-citation algorithms [6], thus owning the features of these three methods.

Section 2 of this paper introduces the related works of this topic. Section 3 then provides the foundation knowledge for the new algorithm. Section 4 describes the co-recommendation algorithm and some comparisons and analysis are presented in section 5. The final section will then provide some concluding remarks and future work under consideration.

2. Related work

The *Netscape-Alexa* algorithm is based on links from web sites and its contents. Successful URL 'hits' and annotations of web pages are used to rank the sites. This algorithm also

Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'02)
0-7695-1512-6/02 $17.00 © 2002 IEEE

examines the 'surfing' paths to determine the relationship between the different sites that have been visited. Moreover, it utilizes the users usage of query results to improve its own performance for the future. This scheme can be applied in the broad Internet community. A disadvantage of the *Netscape-Alexa* algorithm is that it only focuses on the website URLs, but does not consider the specific interests of those sites[3] [6]. In the *Pandango* project**,** the algorithm depends on the bookmarks of the users. It incorporates the peer-to-peer model, thus providing a distributed search engine. Its aim is to match the users with similar preferences and suggest results for a similar query. Users preferences are evaluated by surveying the click-through data. However, the algorithm evaluates browsing preferences inadequately [13]. In the *Sinergis* project, useful knowledge is aggregated so that users with common preferences can find relevant material whilst doing similar search queries. It can present users with a customized list of pages related to their current pages. It also allows users to rank and cluster different pages based on these results. Furthermore, some methods, such as 'specific interests' and 'voted pages', can be investigated to improve the performance. [15].

HITS [10] relies on the query and examines the set of pages that point to, or are pointed by the pages in the result [14]. PageRank [12] [4] is a global ranking scheme deriving from citation ranking which is concerned with hyperlink structure. It takes the importance of a directed page into considerations. In this scheme, recursive-importance of a page does not only depend on, but indeed influences the importance of other pages. The Cocitation+ algorithm focuses on a different approach for information retrieval from the Web [6]. It uses a URL to find other URLs of related pages. Related pages are defined as pages on the same topic as the query page. The experiments show that the technique performs extremely well for finding related web pages but its performance is disappointing for finding people with similar interests.

In [5], the authors propose item-based algorithms, which integrate a collaborative filtering approach. The bottleneck in conventional collaborative filtering algorithms is the searching for potentially related neighbours among a large user population. Item-based algorithms eliminate this bottleneck by exploring the relationships among items rather than the relationships among users. *Surflen* is an information recommendation system, which recommends interesting web pages to users [16]. It captures a users navigation history and applies "association rule" of data mining to discover hidden knowledge contained in this history. Its experiments show that the more a user interacts with the system, the better its recommendations will be. It also indicates that the users browsing history becomes more indicative of their interests if the number of successful recommendations increases.

3. Foundations for the algorithm

This section describes three aspects of knowledge about the new algorithm. The Tf-idf scheme is used to compute the index term weights. Then, a Correlation-based similarity computation is conducted to assess the similarity between two web pages. The Cocitation+ algorithm incorporates this information in to the Web ranking module.

3.1. Tf-idf scheme

The new algorithm is based on the index term weighting. These term weights are ultimately used to compute the degree of similarity between stored web pages in the system and the user query. These index terms are noun groups from the web pages, because most of the semantics comprise of nouns in a sentence using natural language text. The Index term weighting can be calculated using a Tf-idf scheme [14] [9].

Let N be the total number of documents in the system and n_i be the number of documents in which the index term k_i appears. Let $freq_{i,j}$ be the raw frequency of term k_i in the document d_j (i.e, the number of times the term k_i is mentioned in the text of the document d_j) then the normalized frequency $f_{i,j}$ of terms k_i in document d_j is given by

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}} \qquad (1)$$

The maximum is computed over all terms, which are mentioned in the text of the document d_j. If the term k_i does not appear in the document d_j then $f_{i,j}$=0. Further, let idf_i, inverse document frequency for k_i, be given by

$$idf_i = \log \frac{N}{n_i} \qquad (2)$$

The best-known term-weighting schemes use weights, which are given by

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i} \qquad (3)$$

More details of the Tf-idf scheme can be seen in [14].

3.2. Correlation-based Similarity Computation

One critical step in the term-based collaborative filtering algorithm is to compute and then select the similarity between web pages. The basic idea in similarity computation between two terms t_i and t_j is to first isolate the web pages, which have rated terms using the Tf-idf scheme.

Here, a correlation-based similarity can be adopted [5]. For this method, the similarity between two terms *i* and *j* is measured by computing the Pearson-r correlation, $corr_{i,j}$. Two web pages are thought of as two vectors. Let the set of web pages that both rated *i* and *j*, be denoted by *D = {d1, d2, … dm}*. Then the correlation similarity is given by

$$sim(i,j) = \frac{\sum_{d \in D} \left(R_{d,i} - \overline{R_i}\right)\left(R_{d,i} - \overline{R_j}\right)}{\sqrt{\sum_{d \in D} \left(R_{d,i} - \overline{R_i}\right)^2} \sqrt{\sum_{d \in D} \left(R_{d,i} - \overline{R_j}\right)^2}} \qquad (4)$$

Here $R_{d,i}$ denotes the rating of web pages *d* on term *I* and $\overline{R_i}$ is the average rating of the i^{th} term.

3.3. Cocitation+ Algorithm

In [6], the author proposes the cocitation+ algorithm, which differentiates between pages that are just a collection of links, and pages that have more content than links.

The algorithm will find pages for the ***A_part*** which entails providing links to similar topics of the initial search. This part of the algorithm could be used for bookmarking and for a users interest analysis. The ***B_part*** is similar to the Cocitation algorithm. However, it is simpler and faster. The Cocitation algorithm uses the number of common parents to determine the most relevant page. In contrast, the Cocitation+ algorithm uses the number of most common siblings. Cocitation+ is used in sections 4 and 6 to complete some comparisons with the new algorithm.

4. Co-Recommendation Algorithms

Borrowing the method from [5], and using index term weighting, a correlation-based similarity computation is applied to determine the similarity, $s_{i,j}$, between web pages. This approach changes the cocitation+ algorithm in that it now has to adapt to a term-based cluster for a user. Preceded by some assumptions, the basic steps of the co-recommandation algorithm are as follows:

a. Firstly, assume that an indexer module exists in a server, thus providing an index term weighting for the new algorithm. This algorithm can then be binded to an existing searching engine. Or, an agent can be created using the Tf-idf scheme on a proxy server (e.g. a university proxy server) to calculate index terms and their weighting for a users link, by scanning the cache. This method is feasible since most proxy servers incorporate a temporal algorithm.

b. Secondly, a threshold, *P*, for clustering web pages is located, using a Metric Clusters algorithm to organize groups in the user-based web page link collection. The metric clusters algorithm can be found in [14].

List 1 shows an initial data collection of the new algorithm.

```
Initial data collection(){
  while (TRUE){
    receive(user u_k);
    assign user set U={u_1, u_2,..., u_k}, web page number N;
    assign index term set T={t_1,t_2,..t_m};
    assign weighting set W_T=(w_1, w_2, ..w_l};
    webpageSimilarity.calculate(T, W_T){
      return S={s_ij};
      N++;
    }
    similarityDecision(s_ij){
      assign threshold P;
      int x=0;
      do {
        if(s_ij>P){
          create a group of webpage g_i;
          store web page di and dj in g_i;
          Get group index term T_sup and their weights W_sup;
          x++;
        }
      }while(x<N)
    }
}
```

List 1 Initial data collection of the new algorithm.

After a users initial group clustering, list 2 shows the routine process of the new algorithm.

```
routineProcess(){
  assign another threshold P_p;
  webpageGroupSimilarity.calculate()
  {return s_g(i,j);}
  similarityDecision();
  store some users who provide high correlation with s_g(i,j) ;
}
```

List 2 Routine Process of the new algorithm

This algorithm can support queries such as keyword, link and document. It can deliver the high recommendation links and survey users with a similar preference. If the client provides two different links in one query, the algorithm can process them separately and provide the user with two preferences, which can even be further increased to present the users with even more suitable preferences. Suggesting that *G* is the parent of links for the cocitation+ algorithm, G_{sup} refers to the results, which introduce related contents from the cocitation algorithm.

A set of web page groups helped alleviate coverage and improved quality in the integrated users different preferences. The repository can own a lot of attributes (item ID, item title, frequency, URL, rated terms, and top amount of groups detail etc.) for further analysis. These analyses can then provide other useful recommendations for different fields. The biggest advantage is that this algorithm does not only support keyword searching, but it also supports URL and preferences searching. When users use keywords, it searches weighted index terms in total groups. When a user clicks a URL, this recommendation algorithm can provide top frequency links from a similar group, or the group that owns that particular URL. As a result, a much faster recommendation procedure tends to be produced.

5. Comparisons and Analysis

Normally, most recommendation algorithms require a lot of interaction in order to determine the users preferences. The experiment mentioned in [16] shows that the more a user interacts with the system, the better its recommendations will be. The new algorithm differs in that it obtains a message from the user's web page. When one value web page is accessed, the system automatically proposes some value links based on the other usage. The content-based data group then encompasses other related interests. If another users link is involved in the group, recommendations can be considered to enhance the correlation. This feature solves the underlying problem that the recommendation is related to the users interests.

A very important observation from [5] is that the high group-forming accuracy can be achieved using only a fraction of web pages. The following case demonstrates this study.

Consider set $D=\{d_1, d_2, \ldots d_n\}$, $d_i,d_j,d_k, d_m \in D$. If *sim (i,j), sim(i,k)* and *sim(i,m)* is high, then *sim (j,k), sim(j,m)* and *sim(k,m)* can be considered high too. A higher accuracy is achieved when using the same proportional web pages with

bigger dimensions. This is an important advantage for the new algorithm. We can use this observation to decrease the computation coverage.

The results [5] also show that *item-item* scheme provides better recommendations than the *user-user* (k-nearest neighbour) scheme. Here, *item* is changed to *term* and *user* is changed to *web pages.* It is worthy to note that the results from the *term-term* scheme are better than the other *link-link* collaborative filtering recommendation scheme.

On the other hand, good recommendations have been identified from experiments of the cocitation+ algorithm [6]. The cocitation+ algorithm is described in Section 3.3. The parents of a URL can be changed to related groups. For *B-part* algorithms, the number of common parents is used to determine the most relevant page. However, a lot of factors will affect the sibling relevant pages. Whereas some links may be just friendly links, the situation is different in the new scheme. Since a user visited a clustered link on the basis of its contents, the relevant probability should be higher than that of the cocitation+ algorithm. The *A-part* algorithm will find the people who have similar preferences. The results do not exceed the guessing for the cocitation+ algorithm, where people tend to focus on the similar topic by combining the number of web pages between a super-group and a user's related group. Certainly, this super-group is similar to the active user's query web pages. Since this algorithm supports a query with two or more different preferences, it can solve the problem when a user changes his interests from one subject to another.

There are two conflictive challenges for the user-based scheme; the scalability of collaborative filtering algorithms and the quality of the recommendation for the users. The less time the algorithm spends on searching for neighbours, the more scalable it will be, but the quality will appear inferior. The off-line computation and the observation of [5] do not require this algorithm to address the issue.

6. Conclusions

This paper presents the Co-Recommendation Algorithm, which consists of the features of the recommendation rule and the co-citation algorithm. Visited history links from a user are considered as recommended links for other users.

The algorithm focuses on challenges that are fundamental for other searching and recommendation algorithms. It does not require users to heavily participate in an interactive communication environment. Furthermore, it supports different queries, such as keyword, URL, and document. This algorithm also supports a query with two or more different preferences thus allowing the user to change his interests from one subject to another. If a user has several preferences, this algorithm can still find similar users for him. Compared to other algorithms, its structure affords scalability. With most of the computation being done off-line, the high online performance can be obtained. In the repository computation, it can achieve a high group-forming accuracy using only a fraction of web pages of a group.

By and large, every recommendation system using a site history is involved in the privacy problem. But if a user wants to get recommendation from other users, it is recommended that he offer his history as a resource. It is a basic sharing principle. For the initial repository, we can scan caches in the big organisations.

Since the connection between filter engines, search engines, and query facilities is quite unexplored, future investigations will implement this algorithm and subsequently evaluate its real time performance. Furthermore, a more flexible model, such as the distributed web indexer system, will need to be built for the commercial application.

References

[1] John Paul Ashenfelter, Web Database Connectivity Buzzwords, http://www.webreview.com/1999/11_05/developers

[2] Andreas Paepcke, Hector Garcia-Molina, Gerard Rodriguez-Mula, Junghoo Cho, beyong document similarity: understanding value-based search and browsing technologies, SIGMOD Records, 29(1): March 2000

[3] Alexa Inc. http://www.alexa.com

[4] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan, Searching the web, ACM Transactions on Internet Technology, Vol.1, No.1, August 2001, Pages 2-43

[5] Badrul Sarwar, George Karypis et al, item-based collaborative filtering recommendation algorithms, The tenth international World Wide Web conference on World Wide Web ,2001 , Hong Kong

[6] Al Barabasi, R. Albert, H.Jeong, and G.Bianconi, Response: power-law distribution of the World Wide Web, science 287 2115a 9in technical comments), 2000

[7] Doug Beeferman and Adam Berger., Agglomerative clustering of a search engine query log, Proceedings of the sixth ACM SIGKDD international conference on Knowledge Discovery and data mining ,2000 , Boston, Massachusetts, United States , pp.407-416

[8] Athman Bouguettaya, Boualem Henatallah,Lily hendra, Mourad Ouzzani, Supporting dynamic Interactions among Web-Based Information Sources, IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 5, page September/October 2000

[9] Venkat N. Gudivada, Vijay V. Raghavan,William I. Grosky, Rajesh Kaasanagottu, Information retrieval on the world wide web, IEEE Internet Computing, 1997, pp. 58-68

[10] Kleinberg, J. 1999. Authoritative sources in a hyperlinked environment. J. ACM 46,6(Nov.).

[11] Mei Kobayashi and Koichi Takeda, Information retrieval on the web, IBM research, ACM computing surveys, Vol.32, No.2, June 2000

[12] Page, L., Brin,S.,Motwani,R., and Winograd,T. 1998. The pagerank citation ranking:Bringing order to the web. Tech. Rep.. Computer Systems Laboratory, Stanford University, Stanford, CA.

[13] The Pandango Search Engine http://news.cnet.com/news/01-1005-200-4950537.html@stanford.edu

[14] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern information retrieval, ACM Press, 1999, ISBN 0-201-39829-X

[15] Javier Nicolas Sanchez, Rohit Singh, Stanford University, Sinergia: Improving Browsing By Exploiting Community Knowledge http://www.stanford.edu/~javiers/cs241/writeup.pdf

[16] X. Fu, J. Budzik, and K. J. Hammond. Mining navigation history for recommendation. In Proc. 2000 International Conference on Intelligent User Interfaces, New Orleans, January 2000. ACM, pp. 106-112