Refining Web Authoritative Resource by Frequent Structures *

Haofeng Zhou [1], Yubo Lou [1], Qingqing Yuan [1], Wilfred Ng [2], Wei Wang [1], Baile Shi [1]
[1] Dept. of Computing and Information Technology, Fudan University, Shanghai, China
[2] Dept. of Computer Science, Hong Kong University of Science & Technology
[1] haofzhou@eastday.com, [2] wilfred@cs.ust.uk

Abstract

The web resource is a rich collection of the dynamic information that are useful in various disciplines. There has also been much research work related to improving the quality of information searching in the web. However, most of the work is still inadequate to satisfy a diversified demand from users. In this paper, we exploit the hyperlinks in the web and propose a new approach called SFP in order to improve the quality of research results obtain from search engines. The SFP algorithm evolves from the frequent pattern mining technique which is a common data mining technique for conventional databases. The essential idea of our approach is to mine the frequent structures of links from a given web topology. By using the SFP algorithm, we extract the authoritative pages and communities from the complex web topology. We demonstrate our approach by running several experiments and show that the performance and functionalities of using the SFP in managing search results are better than other known methods such as HITS.

1 Introduction

The World Wide Web (WWW) serves as a huge, widely distributed, global information service center for different kinds of information services. It is important to note that the authoritative resources on the web, which are not explicit in the structure of web links, are able to serve as a reference to facilitate much better navigation. When a web designer/adminstrator includes a hyperlink referencing to another web site in his/her own page, he/she actually uses a hyperlink to represent a semantic relationship between the linked pages.

The work related to mining the web's link structures to recognize authoritative web pages has been known in the community for several years [6, 15]. Some algorithms were also proposed for this purpose [5, 14]. Most of them are applied in the search engines over the Web. But there are still some limitations in these methods, for example the *recall* and it precision, which indicate the quality of the search result, are not satisfactory. The previous work on frequent pattern mining, such as the well-known Apriori [2] and FP-Growth [10], faces the problem of finding the relationship between the relations of items. It is also well perceived that the web structure can be described in the form of directed-graph in a natural way, where vertices represent pages with URLs as their labels, and directed-edges for hyperlinks [8].

In this paper, we apply our earlier proposed method, called the SFP algorithm, for improving the precision of web search. The SFP algorithm has been applied in the context of simple semi-structured data [22], which shares the similar spirit of the conventional frequent pattern mining technique [2]. Using several experiments concerning keyword searching, we show that our proposed method is effective to refine the results obtained from a search engine.

The rest of this paper is organized as follows. In Section 2, some related work is reviewed and the problem of our work is stated. Then, our proposed method, SFP, is introduced in Section 3. In Section 4, a set of experiments is presented to illustrate the process of refinement in our method. Finally, the conclusion is given in Section 5.

2 Related Work

There has been much work on both the structure analysis and frequent pattern mining [2, 5, 6, 8, 10, 14, 15].

The idea of determining the importance of a document using its corresponding link information was proposed as early as the emergence of the WWW. However, the link structure of web pages has some special features which are detailed as follows: (1) many links on the web, for example those links for advertising or navigation, do not denote the authority preference; and (2) some links may be authoritative to some specific part but have no relationship with other links in the same page. Hence, some research work

* This paper was supported by the Key Program of National Natural Science Foundation of China (No. 69933010) and China National 863 High-Tech Projects (No. 2002AA4Z3430 and 2002AA231041)

has been done to study the authority of a web page, among which HITS of Kleinberg [14] is a typical web information retrieval algorithm which perform link analysis to improve the quality of search result. In [9, 16], HITS algorithm is further employed to find communities over the web. The concept of community is important, since it represents a group of authoritative web pages. However, we find that there are some drawbacks of HITS algorithm. First, HITS either treats internal links within the same site as the external ones or just ignores them completely. These two ways of handling internal links are not satisfactory in our opinions. In fact in [11], this problem has been noted and thus HITS-SW is then proposed to improve HITS algorithm. HITS-SW assigns similarity weights to all the links in a collection of web pages, and runs a connectivity-analysis considering both the links and the content of the document. Second, HITS (including HITS-SW) needs to perform iteration until reaching a convergent state and as a result the process is inefficient. Comparing to HITS, our proposed SFP algorithm, which will be introduced later on, do not need to perform iteration when generating authoritative pages.

There has also been other related work such as [20] which calculates web communities by using links. However, this class of work does not consider the features of the communities in detail, such as the mutual influence of communities, e.g, the weight or the number being linked.

Frequent pattern mining plays an important role in the field of data mining [1, 2, 4, 10, 19]. Most of them do not assume any association between basic data items. In the case of a web structure, if the links are considered as the basic items for mining, they are apparently associated. Therefore, it is necessary to provide an algorithm to deal with the mining problem on these inherently associated data items.

Recently, some research work has been done on mining structured data [3, 7, 21, 23]. In addition, [12, 17, 18] also did research on mining frequent sub-graphs. However these methods have paid too much attention on the isomorphism problem, which both considers the structrual and label mapping. But it will not happen in the a web structure graph, where URLs serve as unique labeling on each vertex. So the mentioned methods are not efficient enough for mining frequent patterns in a web structure . Besides, most of these methods are based on the well-known Apriori algorithm, in which the generation of a large number of candidate itemsets is the bottleneck. We remark that, in [22] we has proposed a new algorithm SFP for mining graph structures, which will be applied in this paper.

3 SFP: Mining the Frequent Structure from the Web

In this section, we first introduce the basic terminology and the SFP algorithm (or simply SFP). Then, we describe the modification of SFP in order to adapt it to the context of web mining.

3.1 Terms and Concepts

In SFP algorithm, all the graphs are assumed to be *simple unique labeled graphs*, which means that in such graphs, there is no loop, no multiple edges, and no two vertices assigned the same label. SFP only applies to connected graphs. In the sequel we assume all original graphs G consist the graph dataset GD. The result of running SFP on G is a frequent connected structure represened as the connected subgraph, which is only a connected part of the original graph. As a frequent pattern mining algorithm, SFP also has the similar definitions with the original ones, such as support and minimum support. A new term here is “rank” of G, which is defined as the total number of edges in a given graph. As a special case, 0-rank graph is a graph G having only one vertex.

Therefore, a web page is authoritative if it is a 0-rank frequent sub-graph, and an authoritative community demonstrates the close relationship among authoritative pages. It is one co-op consists of linked authoritative pages.

3.2 Adapting SFP for Web Mining Case

SFP extracts frequent subgraphs from a given set of simple unique labeled graphs. The idea is similar to the well-known FP-Growth, but we consider the relationship between edge items rather than the vertices. For more details about SFP, please refer to [22].

In order to adapt the SFP in the context of web structure, two parts of works are to be considered. One is related to the mapping from a web structure to a graph, and another is related to how to handle the connectivity in a directed graph.

The solution for the first part of work is straightforward. We simply use the directed-graph to represent the web structure, where the vertex denotes the web page, its label is the URL of this page, and directed edge represents the hyperlink from the one page to another. In the web structure, we should remove all the hyperlinks pointing to the pages that contain themselves, which means that there is no loop in the directed graph.

Next, in SFP, the edge is denoted as the form l_il_j, where the l_i and l_j are the labels on the two vertices related to the edge. Since the edge is undirected, it forces the edge in the form of l_il_j with $l_i < l_j$ in the lexical order, which may need some transformation between the original data and the data used in SFP. One subtle point is that the l_il_j and the l_jl_i are the same in SFP. But in a directed graph, this way of handling of links will lose the direction of the edge. The l_il_j and the l_jl_i are regarded as two different edges in the graph.

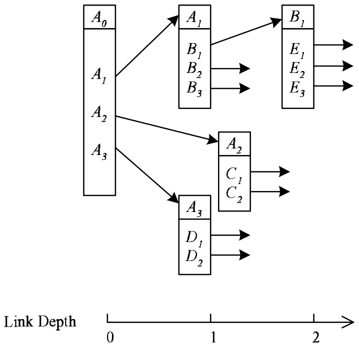


Figure 1. Mapping from Web to a SFP graph

Therefore, we need to deal with them separately. The $l_i l_j$ and the $l_j l_i$ will be two elements in the SFP, each of them has its own support or frequency.

For the second part of work, in SFP, we still need the bi-direction links which preserve the information about the connectivity of the original data. However, we should mark a tag on the bi-direction which indicates its real direction. This mark is used only for the purpose to restore the subgraphs in the results, and nothing else.

The final remark is the issue of connectivity. We adopt the definition of weak connectivity, which require that there exists a undirected path between any two vertices. The idea is also used for the SFP in [22].

The graph used in SFP is generated from the result of search engine. In this paper, we use a root page to denote each result page returned by the search engine (e.g. in Figure 1, A_0 is one result page of a certain search, and it is a root page). If we start from the root page, separate out the links contained in the page content iteratively, we can finally obtain a tree-like directed graph (Strictly speaking, this graph is not a tree because it may have cycles). Then, according to a certain search order (breadth first), we simply put all these links into a set for SFP.

For convenience, we use $Level$, $MaxPages$ and $MaxLinks$, respectively, to represent the level of linked pages (or page depth), maximum page number and maximum link number. For each page and within the limit of $MaxPages$, we analyze its content and separate out the links. The $Level$ of root page is set to be zero. Starting from the root page, we are able to obtain a directed connected graph by traversing and analyzing the link structure based on the method and constraints above, which we call such a graph a Generated Graph. Obviously, each root page has a corresponding Generated Graph and all the Generated Graphs consist the dataset GD

4 Experiment of Using SFP for Search Results

The experiments were running on a desktop PC with PentiumII 433MHz CPU, 384MB RAM and Windows 2000 Advanced Server. All the algorithms such as SFP, HITS, and a tool named pageSnagger are coded in C++.

Algorithm HITS is implemented by referencing [13]. The data is from the first 200 entries of the search results, the filter threshold is set to be 5, and the iterating turn is set to be 20. The domains which have the same name are regarded as the same domain.

The pageSnagger was used to get corresponding link structure within a certain depth between pages induced from each separate result of search engine. In pageSnagger, we can set the previously mentioned parameters: $Level$, $MaxPages$ and $MaxLinks$. In our experiment, we set page depth to be 2, since the memory resource in our system is not able to handle the number of pages and links if using higher page depth.

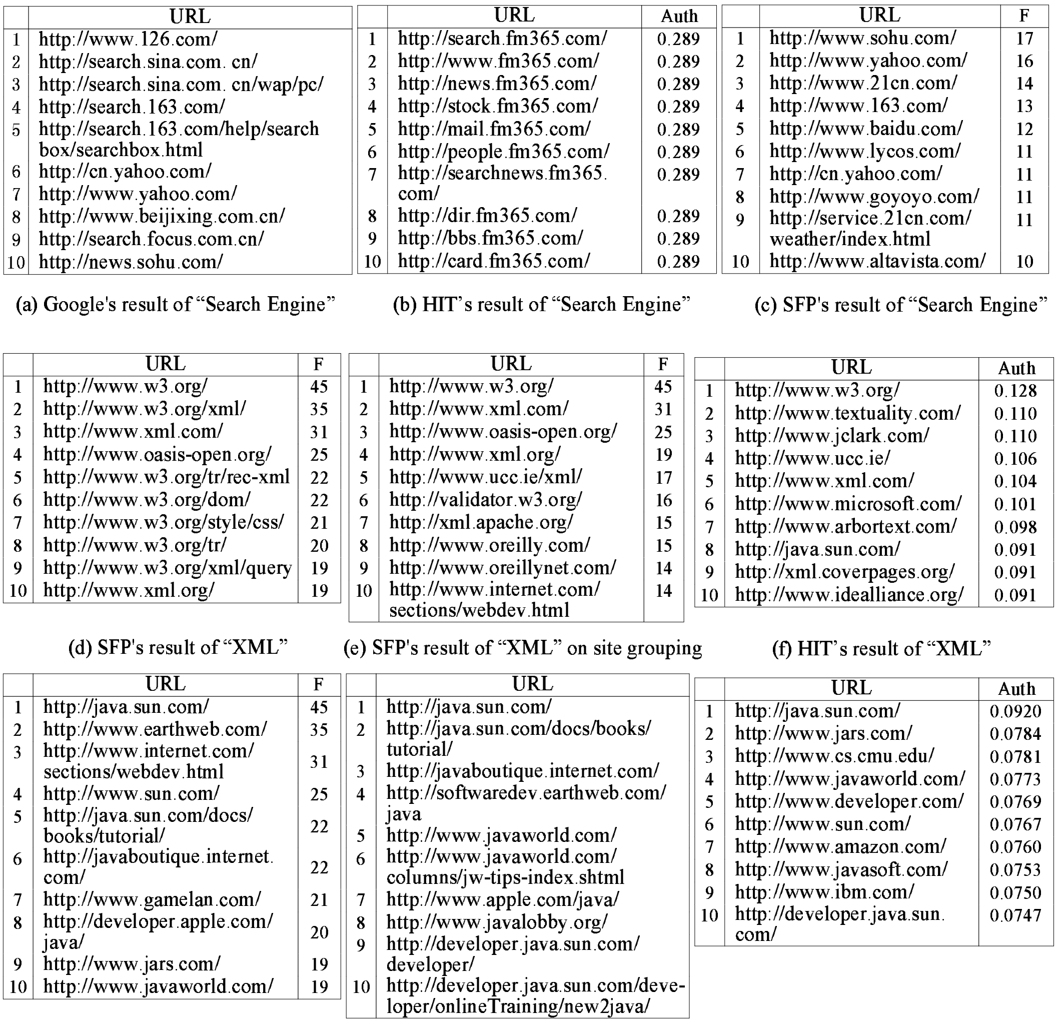
As Google has adopted the methods of PageRank, anchor text, and proximity information to improve search quality, which is a common search engine adopted by users. Therefore, we use Google's search result as a reference to justify our algorithm.

4.1 Searching Authoritative Web Pages

4.1.1 Search key = "search engine" (input in Chinese)

After running the query having the search key "Search Engine"(input in Chinese), we obtain the result in Figure 2(a). But it is far from the list of authoritative sites that we expect. The well-known search engine in China - Sohu only appears as a news page which has little similarity with the search key. Moreover, it ranks even behind some less common search engines, e.g. http://www.beijixing.com.cn/. Other important search engines like Lycos, Google, AltaVista are not listed in Figure 2(a). We find that Lycos ranks 35, and that AltaVista and Google, unfortunately, are not in the top 200. It indicates that current search engines, even the popular one like Google, still have shortcomings in this case. As to the HITS, shown in Figure 2(b), it seems that it has been trapped by the pages from the site fm365.com, which is clearly an unsatisfactory result.

We now discuss the testing result of SFP algorithm. According to the frequency of each page, we show the top 10 URLs in Figure 2(c). This result is much better than that in Figure 2(a). It is because the result in Figure 2(a). are all famous search engines and most of them are the front page of the search engines. One unexpected result is that Google does not appear in the top 10 of the list. The underlying reason is that our authority mining algorithm is based



	URL
1	http://www.126.com/
2	http://search.sina.com. cn/
3	http://search.sina.com. cn/wap/pc/
4	http://search.163.com/
5	http://search.163.com/help/search box/searchbox.html
6	http://cn.yahoo.com/
7	http://www.yahoo.com/
8	http://www.beijixing.com.cn/
9	http://search.focus.com.cn/
10	http://news.sohu.com/

(a) Google's result of "Search Engine"

	URL	Auth
1	http://search.fm365.com/	0.289
2	http://www.fm365.com/	0.289
3	http://news.fm365.com/	0.289
4	http://stock.fm365.com/	0.289
5	http://mail.fm365.com/	0.289
6	http://people.fm365.com/	0.289
7	http://searchnews.fm365. com/	0.289
8	http://dir.fm365.com/	0.289
9	http://bbs.fm365.com/	0.289
10	http://card.fm365.com/	0.289

(b) HIT's result of "Search Engine"

	URL	F
1	http://www.sohu.com/	17
2	http://www.yahoo.com/	16
3	http://www.21cn.com/	14
4	http://www.163.com/	13
5	http://www.baidu.com/	12
6	http://www.lycos.com/	11
7	http://cn.yahoo.com/	11
8	http://www.goyoyo.com/	11
9	http://service.21cn.com/ weather/index.html	11
10	http://www.altavista.com/	10

(c) SFP's result of "Search Engine"

	URL	F
1	http://www.w3.org/	45
2	http://www.w3.org/xml/	35
3	http://www.xml.com/	31
4	http://www.oasis-open.org/	25
5	http://www.w3.org/tr/rec-xml	22
6	http://www.w3.org/dom/	22
7	http://www.w3.org/style/css/	21
8	http://www.w3.org/tr/	20
9	http://www.w3.org/xml/query	19
10	http://www.xml.org/	19

(d) SFP's result of "XML"

	URL	F
1	http://www.w3.org/	45
2	http://www.xml.com/	31
3	http://www.oasis-open.org/	25
4	http://www.xml.org/	19
5	http://www.ucc.ie/xml/	17
6	http://validator.w3.org/	16
7	http://xml.apache.org/	15
8	http://www.oreilly.com/	15
9	http://www.oreillynet.com/	14
10	http://www.internet.com/ sections/webdev.html	14

(e) SFP's result of "XML" on site grouping

	URL	Auth
1	http://www.w3.org/	0.128
2	http://www.textuality.com/	0.110
3	http://www.jclark.com/	0.110
4	http://www.ucc.ie/	0.106
5	http://www.xml.com/	0.104
6	http://www.microsoft.com/	0.101
7	http://www.arbortext.com/	0.098
8	http://java.sun.com/	0.091
9	http://xml.coverpages.org/	0.091
10	http://www.idealliance.org/	0.091

(f) HIT's result of "XML"

	URL	F
1	http://java.sun.com/	45
2	http://www.earthweb.com/	35
3	http://www.internet.com/ sections/webdev.html	31
4	http://www.sun.com/	25
5	http://java.sun.com/docs/ books/tutorial/	22
6	http://javaboutique.internet. com/	22
7	http://www.gamelan.com/	21
8	http://developer.apple.com/ java/	20
9	http://www.jars.com/	19
10	http://www.javaworld.com/	19

(g) SFP's result of "Java"

	URL
1	http://java.sun.com/
2	http://java.sun.com/docs/books/ tutorial/
3	http://javaboutique.internet.com/
4	http://softwaredev.earthweb.com/ java
5	http://www.javaworld.com/
6	http://www.javaworld.com/ columns/jw-tips-index.shtml
7	http://www.apple.com/java/
8	http://www.javalobby.org/
9	http://developer.java.sun.com/ developer/
10	http://developer.java.sun.com/developer/onlineTraining/new2java/

(h) Google's result of "Java"

	URL	Auth
1	http://java.sun.com/	0.0920
2	http://www.jars.com/	0.0784
3	http://www.cs.cmu.edu/	0.0781
4	http://www.javaworld.com/	0.0773
5	http://www.developer.com/	0.0769
6	http://www.sun.com/	0.0767
7	http://www.amazon.com/	0.0760
8	http://www.javasoft.com/	0.0753
9	http://www.ibm.com/	0.0750
10	http://developer.java.sun. com/	0.0747

(i) HIT's result of "Java"

Figure 2. Figures of the Experiment result

on Generated Graphs of the search results, a search engine is authoritative only if it is contained by many Generated Graphs, i.e. it is linked by many search results. As Google is a brand new search engine for ordinary people (especially to Chinese due to the impact of the language and the range), it is inevitably lower in the list using our approach. Moreover, using a search key in Chinese in this test also implies that Chinese web sites are considered to be more important in the returned result. In fact, Google has many specific search pages, with respect to different languages and different search methods. For example,

http://www.google.com/dirhp?hl=en,
http://www.google.com/,
http://www.google.com/en,
http://www.google.com/imghp?hl=zh-cn,
http://www.google.com/advanced_search?hl=en,
http://www.google.com/grp-hp?hl=zh-cn,
http://www.google.com/grphp?hl=en,
http://www.google.com/intl/zh-cn/.

The total frequency of all these pages will be 36, which is much greater than 17 or Sohu. In fact, this is a common happening for multilingual search engines, e.g. Yahoo!.

We note that when the search key is in English input, Google performs well. Therefore, SFP is useful to improve the quality of the search results when a search engine suffers from the poor interpretation of Chinese search keywords.

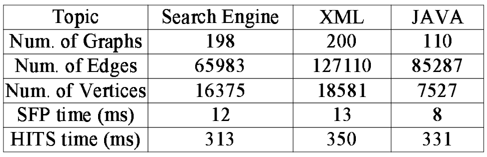
4.1.2 Search key = "XML"

XML has been greatly promoted by the research community and industry. Many XML related standards have been established. W3C (World Wide Web Consortium, http://www.w3c.org) is commonly regarded as the authority of the language standard. In this experiment, we set the parameters to the same values as the previous one and we obtain the top 200 Generated Graphs. The result we obtain for searching “XML” is as follows: 4932 pages, 160.4MB page content, 18,581 links, and 127,110 edges. Figure 2(d) shows the top 10 results obtained by running SFP algorithm. Figure 2(e) lists the result by grouping the pages in the same site. As expected, XML standard pages from W3C occupy most of the positions. The result is much better than that obtained from HITS, which is shown in Figure 2(f). Herein, http://www.xml.com/ and http://www.xml.org are listed the third, fourth and fifth, sixth respectively in Google result. http://www.oasis-open.org is an international nonprofit consortium that designs and develops industry specifications for interoperability based on XML. The ninth and tenth result of Google are linked to this site.

4.1.3 Other Search Results

Java is a new object-oriented language in recent years, since the feature of platform-independence makes it a favorite tool for programmers. In this experiment we set $MaxPages = 100$, $MaxLinks = 100$, $Level = 2$ and key=“java”, by applying SFP to analyze the first 110 entries of the Google search result. After running SFP algorithm ($minfreq = 7$), the site being ranked first is http://java.sun.com/. Again, this result demonstrates the accuracy of our algorithm. We achieve the result in Figure 2(g), which is not in the shade of those in Google and HITS shown in Figure 2(h) and Figure 2(i), respectively.

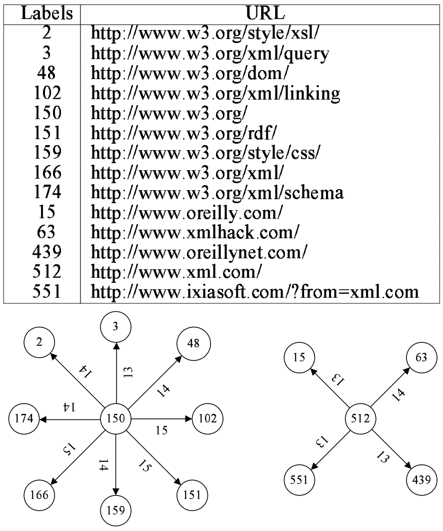
In the algorithms based on matrix iteration, many navigation and advertisement links will have illegal weights due to the spread of weights. However, in our algorithm, these kinds of links do not possess high frequency and eventually they are filtered out.



Topic	Search Engine	XML	JAVA
Num. of Graphs	198	200	110
Num. of Edges	65983	127110	85287
Num. of Vertices	16375	18581	7527
SFP time (ms)	12	13	8
HITS time (ms)	313	350	331

Figure 3. Performance Comparison of SFP and HITS on some topics

As shown in Fig. 3, when we analyze the first 200 entries in the google search results, HITS is relatively stable but much higher in processing time, no matter which query topic is chosen. SFP avoids the iterating calculation, which is used by HITS, by counting the frequency of hyperlinks, which shortens the processing time considerably. Clearly, SFP performs much better than HITS, as shown in this experiment.



Labels	URL
2	http://www.w3.org/style/xsl/
3	http://www.w3.org/xml/query
48	http://www.w3.org/dom/
102	http://www.w3.org/xml/linking
150	http://www.w3.org/
151	http://www.w3.org/rdf/
159	http://www.w3.org/style/css/
166	http://www.w3.org/xml/
174	http://www.w3.org/xml/schema
15	http://www.oreilly.com/
63	http://www.xmlhack.com/
439	http://www.oreillynet.com/
512	http://www.xml.com/
551	http://www.ixiasoft.com/?from=xml.com

Figure 4. XML Authoritative Community

4.2 Authoritative Communities

We observe that a common feature of many web sites share resources through collaboration in order to strengthen the content for each others, which lead to the concept of *community* in the web. We view a web page as a member unit of a community and aim at finding out authoritative community and important information within a web site. In this respect, SFP is able to find both the single pages (0-rank sub-graphs) and higher rank frequent sub-graphs. Thus, our approach is effective in detecting authoritative communities in the web, since SFP provides a more explicit understanding of the structure of the web through related to authoritative communities.

For example, in the “XML”experiment in Section 4.1.2, there are seven out of top ten which are from the same web site http://www.w3c.org. They are linked to the XML related standards. It shows that multiple authoritative pages can occur within the same web site. If we set 13 to be the minimum frequency, we are able to obtain the two most important authoritative communities in Figure 4.

In the “java” experiment, it can easily find that the authoritative community led by http://java.sun.com/. When illustrating these search results, we can simply show all of them with an extensible tree rooted by http://java.sun.com/, which will be helpful for users to find the important pages and their relationships. Furthermore, this is also helpful for advertisers to decide which page is the best place for advertisement. We believe that authoritative community deserves a further study to discover the potential in real applications.

5 Conclusions

In this paper, we take into account the link structure in the Web and adapt our early proposed SFP algorithm in the context of web mining, which is able to refine authoritative pages and communities based on their frequency of users' accessing.

From the experiments we discussed in Section 4, we observe the following desirable features in our result: (1) SFP is subject-sensitive, (2) SFP reflects user expected authoritative pages, and (3) SFP is able to discover authoritative communities. Overall, SFP alleviates the problem that a user needs to consume too much time to filter out those poor results returned from a search engine. Our method performs relatively better in terms of the quality of search result and the processing time for organizing and filtering the search results. We notice that there are some shortcomings in mining authoritative communities, e.g., some community members deviate from the subject. We are still researching on how to calculate the authority of multi-version pages more effectively.

Acknowledgement

Here we want to give our many thanks to Jian Ma for the initial literal work of this paper.

References

[1] R. Agrawal, et al. Mining association rules between sets of items in large databases. In *Proc. of SIGMOD'93*, pp. 207–216.

[2] R. Agrawal, et al. Fast algorithms for mining association rules in large databases. In *Proc. of VLDB'94*, pp. 487–499.

[3] T. Asai, et al. Efficient substructure discovery from large semi-structured data. In *Proc. of SDM'02*, pp. 158–174.

[4] S. Brin, et al. Dynamic itemset counting and implication rules for market basket data. In *Proc. of SIGMOD'97*, pp. 255–264.

[5] S. Brin, et al. The anatomy of a large-scale hypertextual web search engine. In *Proc. of WWW'98, Computer Networks*, vol. 30, no. 1-7.

[6] S. Chakrabarti, et al. Mining the web's link structure. *IEEE Computer*, 32(8):60–67, 1999.

[7] L. Dehaspe, et al. Finding frequent substructures in chemical compounds. In *Proc. of KDD'98*, pp. 30–36.

[8] D. Florescu, et al. Database techniques for the world-wide web: a survey. *ACM SIGMOD Record*, 27(3):59–74, 1998.

[9] D. Gibson, et al. Inferring web communities from link topology. In *Proc. of the 9th ACM conference on Hypertext and hypermedia*, pp. 225–234.

[10] J. Han, et al. Mining frequent patterns without candidate generation. In *Proc. of SIGMOD'00*, pp. 1–12.

[11] J.D. Herbach. Improving authoritative sources in a hyperlinked environment via similarity weighting. BSE Thesis, Dept. of Computer Science, Princeton University, 2001.

[12] A. Inokuchi, et al. An apriori-based algorithm for mining frequent substructures from graph data. In *Proc. of PKDD'2000*, *LNCS*, vol. 1910, pp. 13–23.

[13] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of SODA'98*, pp. 668–677.

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

[15] J. Kleinberg et al. Applications of linear algebra in information retrieval and hypertext analysis. In *Proc. of PODS'99*, pp. 185–193.

[16] R. Kumar, et al. Trawling the web for emerging cyber-communities. In *Proc. of WWW'99, Computer Networks*, vol. 31, no. 11-16.

[17] M. Kuramochi,et al. Frequent subgraph discovery. In *Proc. of ICDM'01*, pp. 313–320.

[18] T. Matsuda, et al. Extension of graph-based induction for general graph structured data. In *Proc. of PAKDD'2000*, *LNCS*, vol. 1805, pp. 420–431.

[19] J.Park, et al. An effective hash based algorithm for mining association rules. In *Proc. of SIGMOD'95*, pp. 175–186.

[20] P. Reddy, et al. Inferring web communities through relaxed-cocitation and power-laws. In *Proc. of the 12th Data Engineering Workshop*.

[21] K.Wang, et al. Discovering typical structures of documents: A road map approach. In *Proc. of SIGIR'98*, pp. 146–154.

[22] Q. Yuan, et al. Extract frequent pattern from simple graph data. In *Proc. of WAIM'2002*, *LNCS*, vol. 2419, pp. 158–169.

[23] M. Zaki. Efficiently mining frequent trees in a forest. In *Proc. of SIGKDD'02*.