An approach to relate the web communities through bipartite graphs

P.Krishna Reddy and Masaru Kitsuregawa
Institute of Industrial Science, The University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo- 1538505, Japan
{reddy, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract

The Web harbors a large number of community structures. Early detection of community structures has many purposes such as reliable searching and selective advertising. In this paper we investigate the problem of extracting and relating the web community structures from a large collection of Web-pages by performing hyper-link analysis. The proposed algorithm extracts the potential community signatures by extracting the corresponding dense bipartite graph (DBG) structures from the given data set of web pages. Further, the proposed algorithm can also be used to relate the extracted community signatures. We report the experimental results conducted on 10 GB TREC (Text REtrieval Conference) data collection that contains 1.7 million pages and 21.5 million links. The results demonstrate that the proposed approach extracts meaningful community signatures and relates them.

Index terms Community detection, Trawling, Link analysis, Web mining, Data mining, Relation, Bipartite graph.

1 Introduction

The Internet (or Web) has rapidly grown into being an integral element of the infrastructure of the society. One of the most powerful socializing aspects of the Web is its ability to connect a group of like-minded people independent of geography or time zones. The Web lets people join communities across the globe by providing an opportunity to form the associations among the people. In the Web environment, one is limited only by his/her interests. As a result, the Web dramatically increases the number of communities one can bond to. For instance, in the past one might have had time to be a part of his/her neighborhood community and one or two social organizations. However, in the Web environment, one gets vast opportunity to form connections as entire world is at his/her disposal. Thus, community forming is one of the important activity in the Web. The Web has several thousand well-known, explicitly defined communities -- groups of individual users who share a common interest. Most of these communities manifest themselves as news groups, Web-rings, or as resources collections in directories such as Yahoo and Infoseek, or home pages of Geocities.

In this paper we focus on the problem of finding and relating the communities in a given data set. Such communities include those emerging communities which are not manifested or not well-known as those listed in the Yahoo or other search engines. Some of such emerging communities have a potential to become the full-fledged communities in future. If we find these communities early it may serve many purposes. These communities provide valuable and possibly the most reliable resources for the user who is interested in them. They also represent the sociology of the Web. By enabling the people to know the existence of such communities, they can target their advertising selectively. Also since interest-based communities are forming with members from all over the world, the governments can engage (or disengage) these communities to meet their objectives. For instance, communities can enable people to shop, get news, meet each other, be entertained, and gossip or in other ways.

In the context of the Web, we consider community as a group of content creators that manifests itself as a set of interlinked pages. We abstract a community as a set of pages that form a dense bipartite graph. The proposed algorithm extracts communities by extracting the potential DBG structures in the given data set (web pages). Further, the proposed approach can be used to relate the extracted communities. We consider a group of communities related if they have common interests on some topic. The related community structures are extracted by extracting the DBG structures among the extracted communities. By extending this approach to higher levels, one can build an hierarchy of communities for a given data set. We report experimental results on 10 GB TREC (Text REtrieval Conference) data collection that contains 1.7 million pages and 21.5 million links. The results demonstrate that the proposed

approach extracts meaningful community as well as related community patterns.

The rest of the paper is organized as follows. In the next section, we review related research. In section 3, discuss abstraction of a community through dense bipartite graphs. In section 4 we explain $cocite$ and $relax\_cocite$ relationships, and present the community extraction algorithm. In section 5 we report the experimental results conducted on the 10GB TREC data. The last section consists of the summary and the future research.

2 Related work

We review the approaches proposed in the literature related to data mining and link analysis and, community detection.

Data mining and link analysis

The data mining approach [1] focuses largely on finding the association rules and other statistical correlation measures in a given data set. The notion of finding communities in the proposed approach differs from data mining since we exploit co-citation whereas data mining is performed based on the support and confidence.

One of the earlier uses of link structure is found in the analysis of social networks [19], where network properties such as cliques, centroids, and diameters are used to analyze the collective properties of interacting agents. The fields of both citation analysis [13] and bibliometrics [25] also use citation links between works of literature to identify patterns in collections.

Most of the search engines perform both link as well as text analysis to increase the quality of search results. Based on link analysis many researchers proposed schemes [8, 9, 11, 7, 17, 16, 4] to find related information from the Web. In this paper we extend the concept of cocitation to the web environment to extract communities from a large collection of Web pages.

Community related research

In [14], communities have been analyzed which are found based on the topic supplied by the user by analyzing link topology using HITS (Hyper-link-Induced Topic Search) algorithm [16]. The HITS is one of the widely used algorithm in search engines to find authoritative resources in the Web that exploits connectivity information among the Web pages. The intuition behind the HITS algorithm is that a document that many documents point to is a good authority and the document that points to many others is a good hub. Transitively, a document pointed to by many good hubs is an even better authority, and similarly a document that pints to many good authorities is an even better hub. The HITS algorithm repeatedly updates authority and hub scores so that documents with high authority scores are expected to have relevant contents, whereas documents with high hub scores are expected to contain links to relevant contents. In that paper the community is defined as a core of central $authoritative$ pages linked together by hub pages. The motivation behind the HITS algorithm is to find good authority pages given a collection of pages on same topic. Our motivation is to detect the potential communities in a larger collection of pages that covers a wide variety of topics.

Ravi Kumar et al. [18] proposed a trawling method to find potential communities by abstracting a core of the community as a group of pages that form a complete bipartite graph (CBG)(by considering web-page as a node and link as an edge between two nodes). A CBG is a bipartite graph with two groups of nodes that contains every possible edge between two groups. Given a large collection of pages, the trawling algorithm extracts all the potential CBGs to find the cores of all the potential communities. Thus, a community core extracted by trawling approach is a small group of pages that form a CBG. The community detection in the trawling algorithm [18] is based on the assumption that web communities contain at least one CBG which is called the core of the community. Given a large collection of pages, the trawling algorithm extracts community cores by extracting all the potential CBGs. In this paper we relax the criteria of existence of a community by defining a DBG structure. Also, the DBG abstraction is extended to relate the extracted communities.

In [12], given a set of crawled pages on some topic, the problem of detecting a community is abstracted to maximum flow /minimum cut framework, where as the source is composed of known members and the sink consist of well-known non-members. Given the set of pages on some topic, a community is defined as a set of web pages that link (in either direction) to more pages in the community than to the pages of outside community. The flow based approach can be used to guide the crawling of related pages.

In [5], an approach to find the related pages of a seed pages presented by specializing the HITS algorithm exploiting link weighting and order of links in a page. Companion first builds a subgraph of the Web near the seed, and extracts authorities and hubs in the graph using HITS. The authorities are returned as related pages. In [21] companion algorithm is extended to find related communities by exploiting the derivation relationships between pages.

The proposed approach differs from preceding approaches as we used a DBG abstraction to extract and relate the web communities.

3 Bipartite graphs and communities

We first explain some terminology used in this paper. Web pages are denoted by P_i, P_j ...; where i, j are integers. A page is referred by its URL, which also denotes a node

in a bipartite graph (BG). We refer a page and its URL interchangeably. If there is an hyper-link from page P_i to page P_j, we say P_i is a parent of P_j and P_j is a child of P_i. An hyper-link from one page to other page is considered as an edge between the corresponding nodes in the BG. For a page P_i, parent(P_i) is a set of all parent pages (nodes) of P_i and child(P_i) is a set of children pages of P_i.

3.1 Bipartite graphs

Here, we give the definition for a bipartite graph.

Definition 1 Bipartite graph (BG) *A bipartite graph BG(T,I) is a graph whose node-set can be partitioned into two non-empty sets T and I. Every directed edge of BG joins a node in T to a node in I.*

In this paper we extract communities by performing only hyper-link analysis. For a page, we only consider only the link information and ignore the text information. In this paper we investigate how only link information is helpful to extract community information. (As a part of future work, we will investigate how the proposed approach can be used to extract the communities by using both text and link information.)

A web page can be represented as BG (Here, we ignore the links from a page to itself). A BG for P_i is denoted by BG(T,I), where T contains the P_i and I contains its children.

A community consists of members. Similar to a web page, the community can be represented as a BG(T,I), where T consists of community identifier and I contains the identifiers of its members.

Note that a BG is dense if many possible edges between T and I exist. In BG, the linkage denseness between the sets T and I is not specified. Here, we define a dense bipartite graph that captures the linkage denseness between the sets T and I as follows.

Definition 2 Dense bipartite graph (DBG) *Let p and q be nonzero integer variables and tc and ic be the number of nodes in T and I, respectively. A DBG(T, I, p, q) is a BG(T,I), where (i) each node of T establishes an edge with at least p ($1 \leq p \leq ic$) nodes of I, and (ii) at least q ($1 \leq q \leq tc$) nodes of T establish an edge with each node of I.*

Now we define a complete bipartite graph that contains all possible edges between the nodes of T and the nodes of I.

Definition 3 Complete bipartite graph (CBG) *A CBG(T,I, p, q) is a DBG(T,I, p, q), where p = ic and q = tc.*

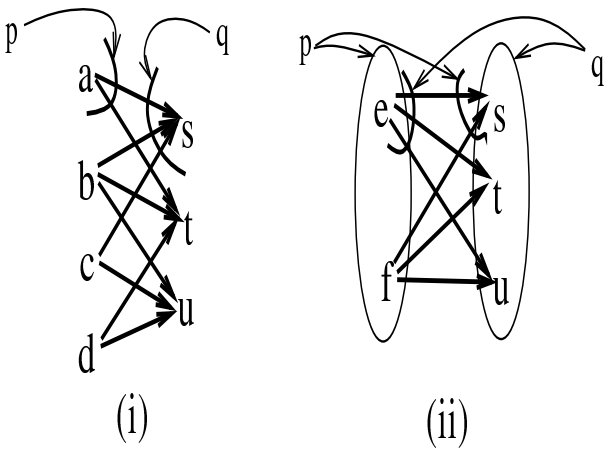

Figure 1. Graphs: (i) DBG(T,I, p, q) (ii) CBG(T, I, p, q)

It can be observed that in DBG(T,I, p, q), both p and q specify the linkage denseness whereas in CBG(T, I, p, q) same denote both the number of nodes in I and T and the linkage denseness. Figure 1 shows the difference between a DBG(T,I, p, q) and a CBG(p,q).

Theorem 1 *For a given data set, r and s, let dense bipartite graph set, DBGS(r,s) = $\{$DBG(T,I, p, q) $\mid p \geq r$ and $s \geq q\}$ and complete bipartite graph set, CBGS(r,s) = $\{$CBG(T, I, p, q) $\mid p \geq r$ and $s \geq q\}$. Then, CBGS(r,s) $\subseteq$ DBGS(r,s).*

Proof: *We say that all the CBGs are the instances of DBGs. That is, at fixed r and s values, if we extract all DBGS(r,s), all the CBGs in CBGS(r,s) are automatically extracted. Note that, DBGS(r,s) includes all the DBG(T,I, p, q) patterns such that $p \geq r$ and $q \geq s$. This implies that DBGS(r,s) includes a DBG(T,I, p, q) with p = ic and q = tc. So, CBGS(r,s) $\subseteq$ DBGS(r,s).*

From the preceding theorem, one can note that if we extract DBGs from a data set corresponding CBGs are also extracted atomatically. However, since CBGS(r,s) $\subseteq$ DBGS(r,s) (for any r $\geq$ 1 and s $\geq$ 1), if there is a DBG(T,I,r,s) pattern, there is no guarantee that corresponding CBG(T,I,r,s) pattern exists.

In this paper we consider community as a set of closely associated pages that form a DBG. Similarly, we consider a DBG over a set of communities as an abstraction of a higher level community. In this way we define higher level communities in terms of lower level communities. By extending this notion, a community hierarchy can be formed for a given data set. Here, we describe the notion of community hierarchy for the given data set.

Definition 4 Community hierarchy *Let the variable num_levels denote the number of levels in a hierarchy for a given data set. A community is denoted with $C(i, j)$, where i ($1 \leq i \leq num\_levels$) is a nonzero integer value that denotes the level and j is an integer value which denotes unique community identifier at level i. Then,*

- *If i=1, members of C(i,j) are the Web pages.*
- *If $i > 1$, members of C(i,j) are the communities at level ''i-1''.*

Note that when i=1, the input is a set of BGs of the web pages; and when $i > 1$, the input is a set of BGs of the communities of ''i-1'' level. For the sake of simplicity we use the term node for both web page and community. A node at i'th level can be a member of multiple communities at (i+1)'th level. The input at any level is a set of nodes. Note that web pages are treated as nodes at level zero. We now define C(i,j) based as follows.

Definition 5 Community (C(i,j)) *Let p_t and q_t be integer variables that represent threshold values. The community C(i,j) = T, if there exist a DBG(T, I, p, q) over a set of nodes at level ''i-1'' with $p \geq p_t$ and $q \geq q_t$.*

Not all the DBGs form meaningful community patterns. So we select potential DBG patterns by fixing the threshold values for both p and q as p_t and q_t, respectively. The values of both p_t and q_t are fixed after examining the potential correspondence with the real community patterns.

3.2 Discussion

We consider a community as a collection of pages that form a linkage pattern equal to a DBG. Our definition is based on the following intuition: *Web communities are characterized by DBGs.* In the Web environment, a page-creator (a person who creates the page) creates the page by putting the links to other pages of interest in isolation. Since a page-creator mostly puts the links to display his interests, we believe that if multiple pages are created with similar interests, at least few of them have common interests. Our intuition is that such a phenomena can be captured through a DBG abstraction.

A community phenomena can also be captured through a CBG abstraction[18]. A CBG abstraction extracts a small set of potential members to agree on some common interests. However, it is not possible to find the large communities through CBG abstraction because page-creators put links in a page in an arbitrary manner. So it rarely happens that a page-creator puts links to all the pages of interest in particular domain.

Given a very large collection of pages, for each community there might exist few pages that could form CBG. However, given the size of the Web it is not easy (impossible) to crawl a very large collection of Web pages. Collecting a very large collection of pages is a time consuming process. Also, for effective search, focused crawling is recommended that covers all the Web pages on few topics. In this situation, given a reasonably large collection of pages, there is no guarantee that each community formation is reflected as a CBG core. Because, a data set may not contain the potential pages to form a CBG.

Normally, each member in a community shares interests with few other members. Therefore, as compared to CBG abstraction, the abstraction of a community pattern through a DBG matches well with real community patterns. In general community can be viewed as a macro-phenomena created by complex relationships exhibited by corresponding members. At micro-level, each member establishes relationships with few other members of the same community. Integration of all members and their relationships exhibit a community phenomena. In the context of Web, a DBG abstraction enables extraction of a community by integrating such micro-level relationships.

Also, it can be noted that the proposed approach based on DBGs can be extended to find higher level communities among lower level communities. This is interesting in the sense that if we extend from bottom to top, we can build an hierarchy of communities for a given data set. In general, given a set of nodes (of any type) and association information among them, the DBG abstraction helps to extract the communities from the given set of nodes.

4 Proposed approach

Web-page creators keep links in a page for different reasons. For example, one may put a link to other page to direct the relevant information, to promote the target page or as an index pointer. In this paper we consider the existence of a link from one page to another page as a display of interest by the former on the later page.

In the web environment, web pages can be grouped based on the type of relationship (association, pattern, or criteria) defined among pages. For example, in an information retrieval environment, the documents are searched based the notion of syntactic relationship that is measured based on the existence of number of common keywords. Similarly, one could define any type of relationship among the web pages and investigate the efficiency through experiments. In the Web environment researchers have defined different types of relationships to group the web pages. Existence of a link, cocitation, coupling, number of paths between web pages are some examples of relationships.

In this paper we have investigated finding communities based on the $relax\_cocite$ relationship which is a relaxed version of the $cocitation$ relationship. We first discuss about the $cocite$ relationship to search related information in the Web. Next. after explaining $relax\_cocite$, we present the proposed algorithm. Also, note that we explain cocite and $relax\_cocite$ relationships for web pages. However, these relationships can be extended to nodes (communities, for instance) of any type.

4.1 Cocite

The fields of citation analysis [13] and bibliometrics [25] also use citation links between works of literature to identify patterns in collections. Co-citation [20] and bibliographic coupling [15] are two of the more fundamental measures used to characterize the similarity between documents. The first measures the number of citations in common between two documents, while the second measures the number of documents that cite both of two documents under consideration.

Also, in the information retrieval literature, relationship between the documents can be established with the keywords that exist in the both documents. Similarly, in a web environment as we have considered link as a display of interest on the target page, by dealing with only links we can establish an association among pages based on the existence of common children (or URLs). That is, we can establish the association among the pages through the number of common children. We call this relationship $cocite$ as in bibliographical terms if two documents [20] refer a collection of common references, we say, they cocite[1] them. We formally define the $cocite$ relationship in the context of Web environment as below. Figure 2(i) depicts the cocite relationship between the pages P_1 and P_2 with $cocite\_factor$ = 3.

Definition 6 Cocite *Let P_i and P_j be pages. cocite(P_i, P_j)=true, if $\mid child(P_i) \cap child(P_j) \mid \geq cocite\_factor$, where $cocite\_factor$ represents a nonzero integer value.*

4.2 Relax_cocite

According to $cocite$, a set of pages is related, if there exist a set of common children. Even though $cocite$ is defined to establish a relationship between two documents, it could form the association among the multiple documents in the following way. We consider two pages P_i and P_j in the data set are related if both have common links at least equal to $cocite\_factor$. Similarly, n ($n \geq 2$) pages are related under $cocite$ if these pages have common children at least equal to $cocite\_factor$. If a group of pages are related according to $cocite$ relationship, these pages form an appropriate CBG.

However, to extract a DBG, we have to retrieve a collection of pages loosely related. So we relax the $cocite$ relationship to find loosely related pages in the following manner. We allow pages P_i, P_j and P_k to group if cocite(P_i,P_j) and cocite(P_j, P_k) are true. This modification enables relationship between a page and multiple pages taken together. That is, if a page could not form association with another page according to $cocite$, it does not imply that they are different. Even though a page fails to satisfy a certain minimum criteria page-wise, however, it could satisfy minimum criteria with multiple pages taken together. We define the corresponding new definition, $relax\_cocite$ as follows.

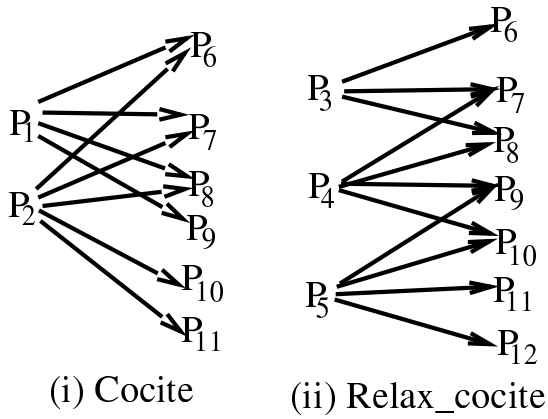

Figure 2. Depiction of $cocite$ and $relax\_cocite$.

Definition 7 Relax_cocite. *Let T be the set of pages and P_j be the another page ($P_j \notin T$). For any page $P_i \in T$, $relax\_cocite(T_i, T_j)$=true if $\mid child(P_j) \cap child(T) \mid \geq relax\_cocite\_factor$. Here, $relax\_cocite\_factor$ is nonzero integer variable and child(T) contains the children of the pages of T.*

It can be observed that for a new page P_k, as compared to cocite, the $relax\_cocite$ relationship increases the probability of association with P_i ($P_i \in T$) as child(T) is larger than child(P_i). Figure 2(ii) depicts the $relax\_cocite$ relationship among web pages P_3, P_4 and P_5, with $relax\_cocite\_factor$ equal to 2.

However, note that for a given page, $relax\_cocite$ may gather pages that are semantically different from the starting page. However, after collecting a reasonable number of pages we employ effective pruning methods to extract a DBG pattern by pruning non-potential pages.

4.3 Algorithm

We present a community extraction algorithm which extracts community structures from a large collection of nodes (pages or communities). Note that the proposed algorithm can applied to extract the communities at all the levels of a community hierarchy for a given data set. We use notation n_{ij} to denote the j'th node at i'th level. We consider web pages as nodes at level zero. For the first level communities, the input consists of a a set of BGs of a given data set (web pages). At higher levels, the input consists of

[1]Note that we consider two documents are related as per $cocite$ if they cite a group of documents and as per $couple$ if a group of documents cite them. In this paper, we propose community extraction algorithm based on the relaxed form of cocitation.

a large set of BGs of preceding level communities. So the input is a large number of nodes at level i and the output is DBGs (communities) at level $(i+1)$.

4.3.1 Community extraction

Given a large collection of nodes, an algorithm to extract DBG structures consists of two steps: gathering related nodes and the extraction of DBGs. For each node, we gather related nodes during gathering phase through the $relax\_cocite$ relationship. We then apply the iterative pruning technique to extract a DBG(T, I, p, q). The corresponding routines are as follows.

1. **Gathering related pages**
   In this step for a given node, n_{ij}, we find T (set of the nodes). We set $relax\_cocite\_factor$ to 1. The integer variable $num\_iterations$ (> 0) is which is set to 0. The iteger variable $max\_iterations$ is set to maximum number of iterations.
   (a) Set T = $\{ n_{ij} \}$.
   (b) While $num\_iterations \leq max\_iterations$
      i. At the given $relax\_cocite\_factor$ value, find all n_{ik} such that $\mid child(n_{ik}) \cap child(T) \mid \geq relax\_cocite\_factor$.
      ii. T = $\{ n_{ik} \} \cup$ T.
   (c) Output T.
2. **DBG extraction**
   In this step the input is the set T produced from the preceding step and the output contains a dense bipartite graph, DBG(T, I, p, q). Let $edge\_file$ be the set of elements $< n_{ij}, n_{ik} >$ where n_{ij} is a parent (source) of child n_{ik} (destination). The $edge\_file$ is set to ϕ.
   (a) Select the values of both p and q.
   (b) For each $n_{ij} \in T$, if $n_{ik} \in child(n_{ij})$, insert the edge $< n_{ij}, n_{ik} >$ in $edge\_file$.
   (c) While the $edge\_file$ is not converged the following steps are repeated.
      i. Sort the $edge\_file$ based on the source. If $\mid child(n_{ij}) \mid < p$, remove all the elements in which n_{ij} is the source node (of type $< n_{ij}, n_{ik} >$) from the $edge\_file$.
      ii. Sort the $edge\_file$ based on the destination. If $\mid parent(n_{ik}) \mid < q$, remove the elements in which n_{ik} is the destination node (of type $< n_{ij}, n_{ik} >$) from the $edge\_file$.
   (d) The resulting $edge\_file$ represents a DBG(T, I, p, q) where, T=$\{ n_{ij} \mid < n_{ij}, n_{ik} > \in edge\_file \}$ and I= $\{n_{ik} \mid < n_{ij}, n_{ik} > \in edge\_file \}$. The set T contains the members of the community.

5 Experiment results

In this section we explain about the TREC data collection, preprocessing and report experiment results conducted on 10GB TREC data.

5.1 Description of data-collection

We report experimental results conducted on 10 GB TREC [23] (Text Retrieval Conference [22]) data collection. It contains 1.7 million web pages. We reproduce the following text on the web page that explains properties of the data collection.

The purpose of the Web Track is to have a framework, based on a snapshot of the World Wide Web, within which new search techniques can be reliably evaluated and within which repeatable experiments may be conducted.

Web Collections: *ACSys (Advanced Computational Systems) has developed three Web document corpuses based on a 320 gigabyte crawl of the World Wide Web by the Internet Archive in early 1997.*

The VLC2 (Very Large Collection No.2) consists of the first 100gB of Web data from the crawl which was then minimally reformatted. This dataset is also known as WT100g, and is used in the Large Web Task.

The newest collection is WT10g, a 10.3gB subset of the VLC2 collection. It has been developed for use in TREC-9's Main Web Task. WT10g has various properties that we hope will make it more suitable for conducting particular kinds of Web retrieval experiments, including those involving link-based methods and distributed information retrieval methods.

5.2 Preprocessing and link-file preparation

For a given page collection, link-file contains all the links of the form $< p, q >$ where $p \in parent(q)$. We prepare a link-file through the following steps (for details see [18]): extracting all the links, eliminating the duplicates and removing both popular and unpopular pages.

The pages are in the text format with html marking information. We have extracted links by ignoring all the text information. We then created a link-file for entire page collection in the following manner. We employed 32 bit fingerprint function to generate a fingerprint for each URL. Each page is converted into a set of edges of the form <source , destination>, where source represents the title

URL and destination represents the other URL in the page. The total number of pages and edges comes to 1.7 million and 21.5 million respectively.

Next, we removed the possible duplicates by considering two pages as duplicates if they have a common sequence of links. We employed the algorithm proposed in [3] to remove the duplicates. We have selected shingle window size as four links. We kept at most three shingles per page. We have considered two pages as duplicates even one shingle is common between them. We found that considerable number of pages are duplicates. After the duplicate elimination, the total number of edges comes to 18 million.

Next we have removed edges derived from both extreme popular and unpopular pages. The popular pages are those which are highly referred in the Web such as WWW.yahoo.com. Also the unpopular pages are those which are least referred. We considered a page as popular if it has more than 50 parents (we have adopted this threshold from [18]). We considered a page as unpopular if it has less than two parents. After sorting the link-file based on the destination, those pages having number of parents greater than fifty and less than two are removed. Also, we removed pages with one child by considering that these do not contribute to community finding. So, after sorting based on the source, the links which have number of children less than two are removed. The above two steps are performed repetitively until the number of edges converge to a fixed value. After this step the number of pages and corresponding edges comes to 0.7 million and 6.5 million respectively.

This link-file is used to retrieve both parents and children of a given page during community extraction.

5.3 Community extraction results

We first report the results during gathering phase. We then discuss community extraction using proposed approach. Next, we show some examples of real community patterns extracted using proposed approach from the TREC data collection.

In the gathering phase, it has been observed that with number of iterations beyond 1, the pages in T are found to be too loosely related. Since our aim is to find all communities, we extracted communities by restricting number of iterations to one. Among these pages, we extract DBG(T,I, p, q).

Figure 3 shows the number of DBG(T,I, p, q) patterns for all the pages that constitute link-file. The total number of pages that constitute link-file is around 0.7 million. For a DBG(T,I, p, q), the column ''(avg(T), avg(T))'' indicates average number of pages in T and I. (Note that these include duplicate communities.) In this, the node set T contains members of the community.

(p, q)	# of DBG(T, I, p, q)	(avg(T), avg(I))
(2,3)	110422	(36.21, 162.6)
(2,4)	81135	(36.98, 109.65)
(2,5)	61566	(36.15, 83.465)
(3,3)	90129	(32.86, 192)
(3,4)	59488	(32.26, 140.56)
(3,5)	40708	(30.17, 114.93)
(4,3)	66670	(34.29, 244.81)
(4,4)	49051	(27.75, 159.62)
(4,5)	32309	(24.97, 134.33)
(5,5)	28296	(21.07, 145.09)
(6,6)	17335	(19.03, 161.67)
(7,7)	10960	(18.97, 198.17)

Figure 3. Graph details: # of DBG(T,I, p, q) patterns, average # of pages in T and I.

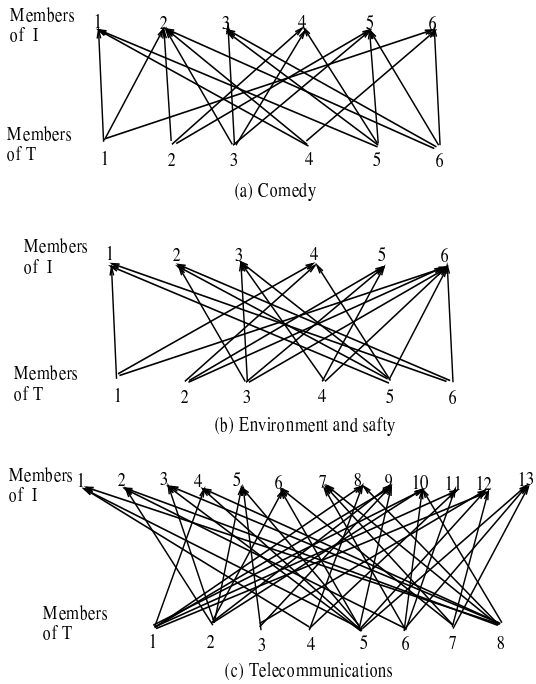

Figure 4. Community examples: Kids, environment and safty, and telecommunitycations.

Community Examples of 1-level

Here we provide three potential community examples extracted from 10GB TREC data collection. The set T represents the potential members of the community (corresponding topics are indicated in the brackets) and the set I represents the potential children of the community. All the graphs represent DBG(T, I, 3, 3); i.e., each member of T has at least 3 children in I and at least 3 members in T have one common child in I. Figure 4, shows corresponding graphs.

Example 1. Topic: Comedy We extracted DBG(6,6,3,3).

Members of T
1. http://www.tnef.com/jim_carrey.html (Jim Carrey - (15 links) Actors)
2. http://www.comedyweb.co.uk/cwlinks.htm (Comedy Web Links Page)
3. http://www.starcreations.com/abstract/laughriot /lr-fam01.htm (LAUGH RIOT - FAMOUSLY FUNNY)
4. http://www9.yahoo.com/Business_and_Economy/Companies /Entertainment /Comedy/Comedians/Carrey_Jim/ (Yahoo! - Business and Economy: Companies: Entertainment: Comedy: Comedians: Carrey, Jim)
5. http://www.scar.utoronto.ca/ 93kolmeg/starp.html (Personalities on Chog)
6. http://www.allny.com/comedy.html (New York Comedy Clubs)

Members of I
1. http://q.continuum.net/ scout/jimpage.htm
2. http://www.halcyon.com/browner/
3. http://www.nd.edu/ jlaurie1/dmhome.html
4. http://www.cheech.com/
5. http://meer.net/ mtoy/steven_wright.html
6. http://www.en.com/users/bbulson/jim.html

Example 2. Topic: Environment and safty We extracted DBG(6,6,3,3).
Members of T
1. http://www.saul.com/env/index.html (Saul, Ewing, Remick & Saul - 10: Environmental Law (PA, NJ, DE))
2. http://www.crystalcity.org/cfd/sitelinks.html (CFD links to other sites)
3. http://www.safetylink.com/ (Safety Link)
4. http://wwell.com/safety-resources/related-links.html (Safety Resources on the Web)
5. http://www.pixelmotion.ns.ca/WCB/links.html
6. http://www.mcaa.org/safety.htm (Safety & amp; Health)

Members of I
1. http://atsdr1.atsdr.cdc.gov/toxfaq.html
2. http://www.ccohs.ca/
3. http://turva.me.tut.fi/ oshweb/
4. http://atsdr1.atsdr.cdc.gov/hazdat.html
5. http://www.wpi.edu/ fpe/nfpa.html
6. http://www.osha-slc.gov/

Example 3. Topic: Telecommmunications We extracted DBG(8, 13, 3, 3).

Members of T
1. http://gatekeeper.angustel.com/links/l-mfrs.html (Telecom Resources: Manufacturers)
2. http://gemini.exmachina.com/links.shtml (Wireless Links)
3. http://millenniumtel.com/ref-voic.htm (Millennium Telecom:References)
4. http://www.buysmart.com/phonesys/phonesyslinks.html (BuyersZone: Phone systems)
5. http://www.commnow.com/links.htm (WirelessNOW Links Page)
6. http://eserver.sms.siemens.com/scotts/010.htm
7. http://www.searchemploy.com/research.html (Search & Employ)
8. http://www.electsource.com/elecoem.html (Electronics OEM's)

Members of I
1. http://www.harris.com/
2. http://www.nb.rockwell.com/
3. http://www.cnmw.com/
4. http://www.mpr.ca/
5. http://www.brite.com/
6. http://www.pcsi.com/
7. http://www.ssi1.com/
8. http://www.mitel.com/
9. http://www.centigram.com/
10. http://www.adc.com/
11. http://www.dashops.com/
12. http://www.octel.com/
13. http://www.isi.com/

5.4 Related community exaples (2-level)

With p = 3, q =4, we have extracted 59488 communities of 1-level communities. After removing the duplicates among these communities, we extracted related community sets (2-level), using the proposed approach. Here, we show two examples of related community structures.
Example 4. The following community structures are about Medicine and Health information.

- 1. http://seamless.seamless.com/talf/txt/ resource/ medical.shtml (The Consumer Law Page: Resources: Medical Resources)
  2. http://www.keenesentinel.com/ clinic/medlinks.shtml (Medical WWWW Links)
  3. http://www.alexanderlaw.com/txt/ resource/ medical.shtml (The Consumer Law Page: Resources: Medical Resources)
  4. http://eserver.sms.siemens.com/siemrad.htm (Radiology Related Sites)
  5. http://www.masalink.org/yps/YPSMEDSI.HTM (Medicine Links - Medicine)
- 1. http://yarra.vicnet.net.au/ stjohn/www/sja_fs.htm (St John WWW Links)
  2. http://vision911.com/pg10_alr.htm (Vision Software, Inc. - Other Helpful Links)
  3. http://www9.yahoo.com/Health/Public_Health_and_Safety/ Fire_Protection/Fire_Departments/ (Yahoo! - Health: Public Health and Safety: Fire Protection:Fire Departments)
  4. http://innonyc.com/eslinks.htm (Innovations BBS: Emergency Services Links)
  5. http://www.olympus.net/personal/cline/fire.html (Fire)
- 1. http://fs01.hwp0.ocps.k12.fl.us/health.html (WPHS health)
  2. http://www.dsmo.com/archive.htm (Not So New on the Web)

3. http://scratchy.hcrhs.hunterdon.k12.nj.us/ othersites/ health.html (health.html)
4. http://smiley.logos.cy.net/CHARLIE/nutr.html

- 1. http://demonmac.mgh.harvard.edu/nationalhealth council.html (National Health Council - Member Web Sites)
  2. http://www.msma.org/public/links.html (MSMA Links)
  3. http://haas.berkeley.edu/ ehsu/top_500l.html TOP 500 (lynx)
  4. http://www.chugai.co.uk/links.html (Pharmaceutical Links)
- 1. http://www.sandriniclinic.com/Links/soclinks.htm (National Specialty Societies and Health Related)
  2. http://www.bnet.att.com/industries/group80.htm (Health services)
  3. http://medsource.com/linkpr2.html (Provider MedLinks--Clinical)
  4. http://www.medsocdel.org/resource.html (Medical Resources and Research)
- 1. http://dem0nmac.mgh.harvard.edu/hospmed.html (Hospital/Medical Resources)
  2. http://www.globalmednet.com/medweb/ma.htm (HOSPITALS IN MASSACHUSETS)
  3. http://medicineonline.com/hospit.htm (Medicine Online HOSPITALS)
  4. http://www.community-care.org.uk/health/ usahosp.html (US Hospitals 'On-Line')

Example 5. The following community structures are about computer companies and computer manufactures.

- 1. http://ioc1.concordnc.com/Gamelink.htm (Internet Of Concord Games Link)
  2. http://www.cybersurvey.com/links.htm (links)
  3. http://ameristar.net/manuf.htm (AmeriStar - Manufacturers)
  4. http://www.recorder.ca/panther/games.htm (Hot Links)
  5. http://www.master.net/chad/gamlinks.html (Chad's Computer Game Links)
- 1. http://www1.windows95.com/drivers/video.html (Video Adapters and Monitors)
  2. http://prodata.kneehill.com/sound.htm (Sound and Multimedia Devices)
  3. http://www3.windows95.com/drivers/sound.html (Sound and Multimedia Devices)
  4. http://shade-tree.com/webdoc4.htm (webdoc4.htm)
  5. http://msg2.ucr.edu/techsupp.html (Windows95 Annoyances (Obtaining Technical Support and Drivers)
- 1. http://www1.windows95.com/drivers/video.html (Video Adapters and Monitors)
  2. http://www.cts-bfs.com/cts-manufacturers.shtml (CTS - Manufacturer Index)
  3. http://www.users.dircon.co.uk/ andrewh/hardware.htm (Andrew's Web Resources - Computer Hardware Page)
  4. http://msg2.ucr.edu/techsupp.html (Windows95 Annoyances (Obtaining Technical Support and Drivers)
  5. http://home1.inet.tele.dk/bel/bel6.htm (bel6)
- 1. http://www.c2000.com/hotlinks/it_sites.htm (Centreline 2000 - IT Web Sites)
  2. http://cnworks.com/html/industry_links.html (Industry Links)
  3. http://ameristar.net/manuf.htm (AmeriStar - Manufacturers)
  4. http://www.nd.edu/ jtracey/starting_points.html
  5. http://most.robohack.planix.com/ woods/netscape-bookmarks.html (Greg A. Woods's Bookmarks)
- 1. http://www.macsource.com/links_vendors_jq.html
  2. http://www.lightwave.com/company.htm (Digital Lightwave Inc. Company Index)
  3. http://www.ecin.com/jumping/ (ECI's - Jumping off Points)
  4. http://home1.inet.tele.dk/fenger/firma2.html (Erling Fenger HARDWARE/SOFTWARE)
- 1. http://eserver.sms.siemens.com/scotts/070.html
  2. http://delec.com/vendorIndex/e.htm (Vendor Index)
  3. http://ameristar.net/manuf.htm (AmeriStar - Manufacturers)
  4. http://www.123go.com/drw/webs/vendors.htm
- 1. http://www.avinfo.com/coolweb.htm (avinfo - WebMedia: Video, Audio, Multimedia, VRML)
  2. http://reality.cowhouse.com/Home/Links/links.html (Cow House Production's bookmarks to other sites)
  3. http://www.wvinter.net/plugins.html (WVInter.Net - Plug Ins)
  4. http://www.ccon.org/hotlinks/hotlinks.html (Contact Consortium HOT Links to Virtual Worlds Sites)

6 Summary and conclusions

In this paper we proposed a simple and efficient approach to extract and relate community signatures from a large collection of web pages by performing hyper-link analysis. A community signature is mathematically abstracted as a DBG over a set of pages. For each page, the algorithm gathers related pages based on the proposed $relax\_cocite$ relationship and then follows an iterative pruning technique to extract a potential DBG structure. The algorithm scales-up well as the time to find all the communities and related communities increases linearly with number of pages in the data set. Also, by copying the $edge - file$ at different nodes, the algorithm can be operated in parallel.

As a part of future work we will investigate the the following issues. from the experimental results, it was observed that not all the community signatures (especillay bigger) are meaningful. We will perform experiments by putiing constains on the number of nodes in DBG so that all the extracted communities are meaningful. We will also conduct experiments at higher levels to build a community hierarchy for the given data set. Also, in addition to link information, we will perform experiments by including key words. With this method we hope to extract all the potential communities in a data set.

In general, a community is a macro phenomena created

by complex relationships exhibited by corresponding members. At micro level, each member establishes relationship with few other members of the same community. Integration of all members and their interests exhibit a community phenomena. The DBG abstraction enables detection of potential community signatures from a given data set by integrating such micro-level relationships.

Acknowledgments
This work is supported by ''Research for the future'' (in Japanese Mirai Kaitaku) under the program of Japan Society for the Promotion of Science, Japan.

References

[1] R.Agrawal and R.Srikant. Fast algorithms for mining association rules, in Proc. VLDB, 1994.

[2] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, Janet Wiener, Graph structure in the Web: experiments and models, in Proc. 9th WWW, May 2000.

[3] Andrei Z.Broder, Steven C.Glassman, Mark S.Manasse, and Geoffery Zweig, Syntactic clustering of the Web, in Proc. 6th WWW, 1997.

[4] K.Bharat and M.Henzinger, Improved algorithms for topic distillation in hyper-linked environments, in Proc. 21st SIGIR, 1998.

[5] Jeffrey Dean, and Monica R.Henzinger, Finding related pages in the world wide web. in Proc. 8th WWW, 1999.

[6] Bill Gates, Business@the speed of thought, Warner Books, 1999.

[7] S.Brin and L.Page, The anatomy of a large scale hypertextual web search engine, in Proc. 7th WWW, April 1998, pp. 107-117.

[8] J.Carriere and R.Kazman. Web query: Searching and visualizing the web through connectivity. In proceedings of 6th WWW Conference, pp. 107-117, April 1997.

[9] S.Chakrabarti, B.Dom, D.Gibson, J.Kleinberg, P.Raghavan and S.Gopalan, Automatic resource compilation by analyzing hyper-link structure and associated text, in Proc. 7th WWW, 1998, pp. 65-74.

[10] Mark E.Crovella and Azer Bestavros, Self-Similarity in World Wide Web traffic evidence and possible causes, in Proc. ACM SIGMETRICS, pp. 160-169, 1996.

[11] Ellen Spertus. Parasite: Mining structural information on the Web. In Proc. 6th WWW, pp. 587-595, April 1997.

[12] G.W.Flake, Steve Lawrence, C.Lee Giles, Efficient identification of web communities, in Proc. 6th ACM SIGKDD, August 2000, pp.150-160.

[13] E.Garfield. Cocitation analysis as a tool in journal evaluation, Science, 178, 1772.

[14] D.Gibson, J.Kleinberg, P.Raghavan. Inferring web communities from link topology, in Proc. ACM Conference on hypertext and hyper-media, 1998, pp. 225-234.

[15] M.M.Kessler. Bibliographic coupling between scientific papers. American Documentation, 14, 1963.

[16] J.Kleinberg, Authoritative sources in a hyper linked environment, proc. of ACN-SIAM Symposium on Discrete Algorithms, 1998.

[17] Loren Terveen and Will Hill. Evaluating emergent collaboration on the Web. In Proc. ACM CSCW'98, Conference on Computer Supported Cooperative Work, Social Filtering and Social influences, pp. 355-362, 1998.

[18] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, Trawling the Web for emerging Cyber-communities, in pRoc. 8th WWW, May 1999.

[19] John Scott. Social Network analysis : a handbook. SAGE Publications, 1991.

[20] Small, H.G. Co-citation in the scientific literature: A new measure of the relationship between two documents. Journal of American Society for Information Science, 24, no. 4, pp.265-269, 1973.

[21] Masashi Toyoda and Masaru Kitsuregawa, Creating a Web Community Chart for navigating Related Communities, ACM Hypertext 2001.

[22] TREC: Text REtrieval evaluation (http://trec.nist.gov).

[23] http://pastime.anu.edu.au/TAR/vic2.html

[24] White, Howard D., and Belver C. Griffith. 1980. Author cocitation: A literature measure of intellectual structure. Journal of American Society for Information Science, 28, no. 5, pp.345-354, 1980.

[25] H.D.White and K.W. McCain, Bibliometrics, in Annual Review of Information Science and Technology, Elsvier, 1989, pp. 119-186.