STRUCTURE OF GRAPH BASED ON WEB COMMUNITIES USING TIME SIMILARITIES

San San Tint and May Yi Aung

Department of Research and Development II, University of Computer Studies, Mandalay, Myanmar

Abstract

In real applications, graphs may be huge in terms of the number of nodes and edges. Many graph drawing algorithms have been developed, but most of them have difficulty dealing with large graphs with thousands of nodes. Clustering graphs is one efficient method to draw large graphs even though other techniques exist, such as fisheye view, hyperbolic geometry and distortion-oriented presentation. A clustered graph can significantly reduce visual complexity by replacing a set of nodes in a cluster with one abstract node. Moreover, a hierarchically clustered graph can find superimposed structures over the original graph through a recursive clustering process. The link structure of the Web is modelled using the Web graph. The result of the studies can appear on Web algorithm such as crawling, searching and tracing of Web communities. This system proposes to track a time similarities communities.

Keywords

Communities, Eigenvalue Decomposition, Hyperlink Structure, Similarity Metric, Web Structure Mining, World Wide Web

1. Introduction

The approach of structural graph is a guide to find the region for a user [1-5]. Online exploratory visualization approach provides major departure from traditional site-mapping methods. The approach has incrementally calculations and storage of the visualization of a small subset of cyberspace without defining the geometrical structure of a specific Website corresponding to the change in the user's opinion. In other words, following the user's orientation, a sequence of Web sub-graphs is automatically displayed with the smooth animation. This attributes can provide to explore the track in which users access links without requiring the whole structure of cyberspace to be known [4][7][8].

Many graph drawing algorithms have been developed, but most of them have difficulty dealing with large graphs with thousands of nodes. Clustering graphs is one efficient method to draw large graphs even though other techniques exist, such as fisheye view, hyperbolic geometry and distortion-oriented presentation. A clustered graph can significantly reduce visual complexity by replacing a set of nodes in a cluster with one abstract node. Moreover, a hierarchically clustered graph can find superimposed structures over the original graph through a recursive clustering process[9-10].

2. RELATED WORKS

The contributions are focused on improving the algorithm performance, others on proposing different graph modeling and others on the application of this technique for real-world applications.

Author presented to solve the web document clustering problem which is graph partitioning, measure the partitioning result using the normalized cut criterion [1]. Combining normalized cut and the scaled Fiedler vector together, this approach forms a global, unbiased algorithm which can effectively extract different topics contained in the web-graph. The solution was to find the normalized cuts problem subject to a set of linear constraints of the form $UTx = 0$ [2]. It can reduce an eigenvalue problem using spectral techniques as well and generalize the set of constraints to $UTx = b$ enforcing not robust when the constraints are noisy. The computational complexity is more significant than using with the basic normalized cuts problem [3].

In this approaches to find interactive image segmentation in computer vision and graphics the steps of the work to find an image segmentation by computing a min-cut/max-flow on a graph which encodes both the user constraints and pair wise pixel similarity [4]. The investigation of further work is to model the foreground and background regions [5].

3. BACKGROUND THEORY

Most useful information is mined by web browser which has proceed much knowledge with tasks such as fascinating and challenging. The case, the large amount of data on the web makes it to access data source in the web. And on the web the coverage of data/ information is wide today [11]. There are several characteristics. Information on the Web is heterogeneous. Because of different authorship of Web pages, users can face multiple pages with the same or similar information using completely different words and/or formats.

The information on the Web is noisy. The noise comes from two main sources. The rest is considered noise. The Web is also about services [2], [5]. Communities of users who share common interests can also be discovered. Web pages can be automatically classified and clustered according to their topics. Furthermore, customer reviews and forum postings can be mined to discover consumer sentiments. These are not traditional data mining tasks.

4. SIMILARITY GRAPHS

Given a set of data points $x_1,...,x_n$ and some notion of similarity $s_{ij} \geq 0$ between all pairs of data points x_i and x_j, the goal of clustering is to divide the data points into several groups.

In this opinion, such that points in the same group are similar. And those points in different groups are dissimilar to each other. By representing the data, it is the form of the similarity graph $G = (V, E)$.

Each vertex v_i in this graph represents a data point x_i.

Two vertices are connected if the similarity s_{ij} between the corresponding data points x_i and x_j is positive or larger than a certain threshold, and the edge is weighted by s_{ij}. To find a partition of the graph, such edges between different groups have very low weights in which the points in different clusters are dissimilar from each other and the edges within a group have high weights in which the points within the same cluster are similar to each other.

4.1. Graph Notation

Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, ..., v_n\}$.

Assume that the graph G is weighted, that is each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} > 0$.

The weighted adjacency matrix of the graph is the matrix $W = (w_{ij})_{i,j=1,....,n}$. If $w_{ij} = 0$ this means that the vertices v_i and v_j are not connected by an edge. As G is undirected we require $w_{ij} = w_{ji}$.

The degree of a vertex $v_i \in V$ is

$$d_i = \sum_{j=1} w_{ij}$$

(1) So over all vertices adjacent to v_i ,

As for all other vertices v_j the weight w_{ij} is 0.

The degree matrix D is the diagonal matrix with the degrees $d_1, ..., d_n$ on the diagonal.

Given a subset of vertices $A \subset V$, its complement is $V \setminus A$ by $\overline{A}$. The indicator vector $1_A = (f_1, ..., f_n) \in \Re^n$ as the vector with entries $f_i = 1$ if $v_i \in A$ and $f_i = 1$ otherwise.

therefore the shorthand notation is $i \in A$ for the set of indices $\{i \mid v_i \in A\}$, in particular,

$\sum_{i \in A} w_{ij}$.

For two not necessarily disjoint sets $A, B \subset V$

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij} \tag{2}$$

two different ways of measuring the "size" of subset $A \subset V$:

$|A| :=$ the number of vertices in A

$$vol(A) := \sum_{i \in A} d_i \tag{3}$$

Therefore, $|A|$ measures the size of A by its number of vertices, while $vol(A)$ measures the size of A by summing over the weights of all edges attached to vertices in A.

$A \subset V$ of a graph (if any two vertices in A).

A subset A is a connected component (if connected).

No connections between vertices in A and $\overline{A}$.

The nonempty sets $A_1,...,A_k$ form a partition of the graph if $A_1 \cup ... \cup A_k = V$.

Data points with pair wise similarities A or pair wise distances d_{ij} into a graph. There are Different Similarity Graphs. These are:

(1)The ε -neighborhood graph,
(2)The fully connected graph,
(3)K-nearest neighbor graph.

Here the goal is to connect vertex v_i with vertex v_j if v_j is among the k-nearest neighbors of v_i. In graph undirected, (1) to simply ignore the directions of the edges, by connecting v_i and v_j with an undirected edge with v_i , the k-nearest neighbours of v_j or v_j , the k-nearest neighbours of v_i. The resulting graph is the k-nearest neighbour graph and (2) to connect vertices v_i and v_j if both v_i, the k-nearest neighbours of v_j and v_j , the k-nearest neighbours of v_i. The resulting graph is called the mutual k-nearest neighbour graph. After connecting the appropriate vertices the edges can be weighted by the similarity of their endpoints. After connecting, all points are positive similarity with each other, and weighted all edges by s_{ij}. In the local neighbourhood relationships, if the similarity models has local neighbourhoods, this construction is only useful. An example for such a similarity function is the Gaussian similarity function $s(x_i, x_j) = \exp(-\| x_i - x_j \|^2 / (2\sigma^2))$, where the parameter σ controls the width of the neighbourhoods. This parameter plays a similar role as the parameter ε in case of the "ε neighbourhood graph [6].

5. HYPERLINK WEB GRAPH STRUCTURE

To obtain the link graph, we first download the web log file which consists of eleven fields. This system use date-time, URL, Referrer fields. Web pages in the Referrer field point to the web pages in the URL field. This is the level one expansion. It is easy to convert the adjacency matrix of and undirected graph.

After the full list of URL and Referrer fields is available, we take the next field to get the time information. Once the time information of all web pages is available, we compute the time similarity.

Given a link graph $G = (V, E)$, which is directed, this system define matrix A to be:

$$A_{ij} = \begin{cases} 1 & if\,(i, j) \in E\, or \quad (i, j) \in E \\ 0 & otherwise \end{cases} \tag{4}$$

A is the adjacency matrix of the link graph [1].

6. SIMILARITY MATRIX

In clustering a graph, similarity between nodes is represented by weight [12]. In the web pages clustering problem, Similarity matrix is calculated using Euclidean distance according to time information containing web log file. Euclidean distance is defined as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{in} - x_{jn})^2} \quad (5)$$

Where $i = (x_{i1}, x_{i2}, ..., x_{in})$ and $j = (x_{j1}, x_{j2}, ..., x_{jn})$ are two n-dimensional data objects. There are examples value of matrixes, Eigenvalues and Eigenvectors.

Weight Matrix

0.0	0.1	0.1	0.6	0.4	0.1
0.1	0.0	0.0	0.5	0.5	0.0
0.1	0.0	0.0	0.5	0.5	0.0
0.6	0.5	0.5	0.0	1.0	0.5
0.4	0.5	0.5	1.0	0.0	0.5
0.1	0.0	0.0	0.5	0.5	0.0

Laplacian Matrix

1.4	-0.1	-0.1	-0.6	-0.4	-0.1
-0.1	1.2	0.0	-0.5	-0.5	0.0
-0.1	0.0	1.2	-0.5	-0.5	0.0
-0.6	-0.5	-0.5	3.2	-1.0	-0.5
-0.4	-0.5	-0.5	-1.0	2.8	-0.5
-0.1	0.0	0.0	-0.5	-0.5	1.2

Eigenvalues

0.00
1.16
1.20
1.49
2.96
4.05

Eigenvector

-0.41	0.00	-0.08	-0.86	-0.26	0.11
-0.41	0.71	-0.38	0.32	-0.29	0.04
-0.41	0.00	0.84	0.21	-0.29	0.05
-0.41	0.00	0.01	-0.04	0.42	-0.81

7. GRAPH MINING ALGORITHM

7.1. Pattern Mining Process in Graphs

The frequent pattern mining has been widely improved in the context of mining transactional data. Recently, the techniques for frequent pattern mining have also been extended to be a structure of graph data. The processes of determining are different cases depending upon the application domains in which graph data. If we have a group of graphs and to determine all patterns supporting a fraction of the corresponding graphs. If we have a single large graph and to determine all patterns supported at least a certain number of times in this large graph [4].

7.2. Clustering Algorithms for Graph Data

This section discusses a variety of algorithms for clustering graph data for web with time [13], [14]. This includes both classical graph clustering algorithms as well as algorithms for clustering XML data. Clustering is important solution of a variety of graph scenarios. Most algorithms have significant applications for users where congestion detection, facility location, and XML data integration. Within the context of graph algorithms, the clustering can be of two types:

- **Node Clustering Algorithms:** There is one large graph, and need to be a cluster the underlying nodes with the use of a distance or time similarity value on the edges in the web. And the edges of the graph are assigned with numerical distance values or time value (in seconds). These numerical distance values are improved to create clusters of nodes as a graph. One has the presence of an edge is a similarity value of 1, and the absence of an edge is a similarity value of 0.
- **Graph Clustering Algorithms:** There is a (possibly large) number of graphs needed to be clustered based on their underlying structural behavior like time(seconds). It is a challenge to need to match the structures of the underlying graphs, and used these structures for clustering purposes. Classical graph data sets and semi-structured data are needed to transform a cluster [3].

8. Design and Implementation

The detailed design of the system is described with the following explanations to understand clearly.

81. Proposed System Architecture

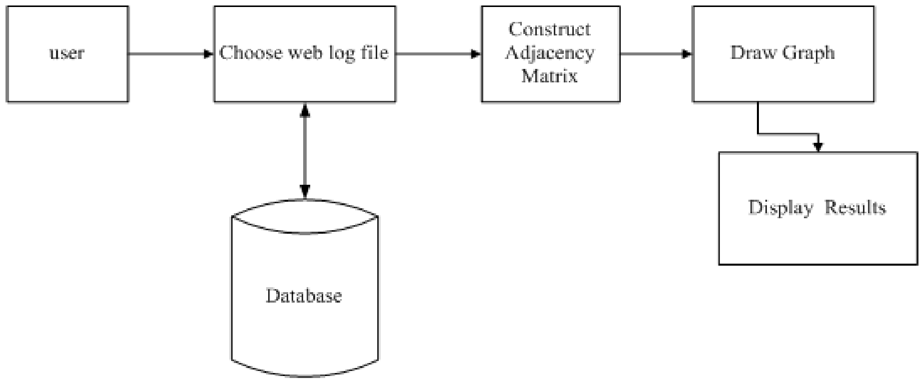

Figure 1. Architecture of Our Proposed System

Figure 1 shows the Architecture of Proposed System . In this system, the user first chooses the web log file. This system is developed for clustering web pages based on the date time information among a set of web pages. The web log file of server is downloaded from a web site. Then these log files are located in the database.. This system focuses on the web graph clustering of web log files according to the different date and time of that file. The sample graphs represents time similarity using various web log file data at certain time.

8.2. Implementation of the System

There are many oriented-programming languages. Our system is implemented by Microsoft Visual Studio 2012 and Microsoft SQL Server Database.

8.3. Import Web Log File

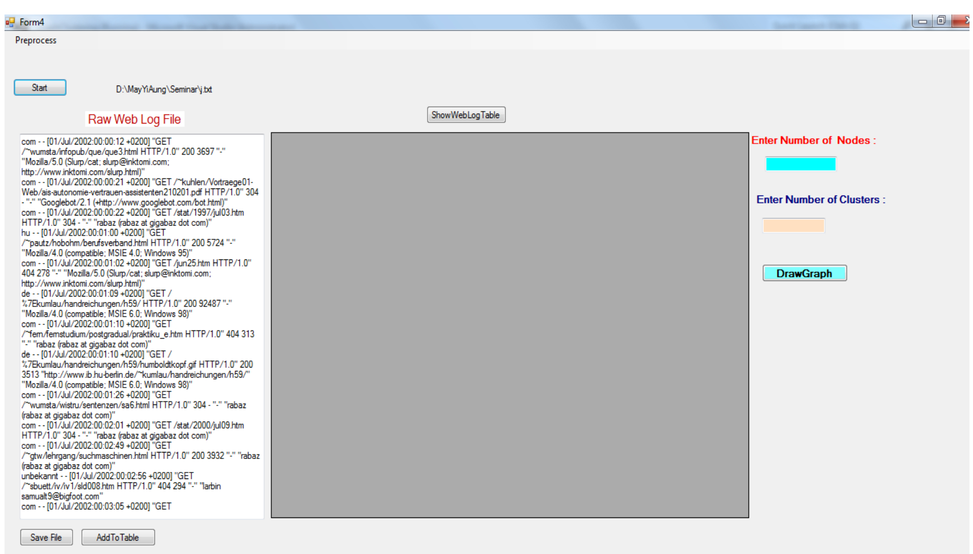

Figure 2. Import Web Log File

Figure 2 shows the user import web log file that is extended web log file type. The user can import the raw data. The start button can choose the raw web log file between sample three sample web log files. The user can view this file with a text.

8.4. Web Log Database

IP	Id	Access	DateTime	Method	URL	Protocol	Status	Bytes	Referrer	Agent
com	-	-	01/Jul/2002:00:00:21...	GET	/~kuhlen/Vortraege01-W...	HTTP/1.0	304	-	-	Googlebot/2.1(+http://www.googlebot.com/...
ar	-	-	01/Jul/2002:00:08:59...	GET	/~pz/zahnpage/sapub81...	HTTP/1.1	200	2787	http://www.google.com....	Mozilla/4.0(compatible;MSIE5.5;Windows98)
net	-	-	01/Jul/2002:00:16:11...	GET	/~wumsta/rehm0.html	HTTP/1.1	200	11249	http://www.google.de/s...	Mozilla/4.0(compatible;MSIE5.2;Mac_Power...
net	-	-	01/Jul/2002:00:29:01...	GET	/~fem/	HTTP/1.1	200	1441	http://www.google.de/s...	Mozilla/4.0(compatible;MSIE5.0;WindowsXP)...
unbekannt	-	-	01/Jul/2002:00:31:10...	GET	/~mh/dynhtml/layers32.htm	HTTP/1.0	200	37803	-	larbinsamualt9@bigfoot.com
com	-	-	01/Jul/2002:00:55:02...	GET	/stat/2001/sep27.htm	HTTP/1.0	200	21869	-	rabaz(rabazatgigabazdotcom)
de	-	-	01/Jul/2002:01:10:30...	GET	/jaw/Html/r_hand.gif	HTTP/1.1	200	163	http://www.ib.hu-berlin.d...	Mozilla/4.0(compatible;MSIE5.5;WindowsNT...
com	-	-	01/Jul/2002:01:31:52...	GET	/stat/2000/feb16.htm	HTTP/1.0	304	-	-	rabaz(rabazatgigabazdotcom)
HUB	-	-	01/Jul/2002:01:44:30...	GET	/~kumlau/handreichunge...	HTTP/1.0	304	-	-	MnoGoSearch-Indexer/3.2(HUB;webtech@...
unbekannt	-	-	01/Jul/2002:01:58:18...	GET	/~mh/selfhtml/tddba.htm	HTTP/1.0	200	311	-	Scooter-3.2.EX
com	-	-	01/Jul/2002:02:16:46...	GET	/stat/1997/may16.htm	HTTP/1.0	304	-	-	rabaz(rabazatgigabazdotcom)
unbekannt	-	-	01/Jul/2002:02:48:02...	GET	/stat/2001/feb23.htm	HTTP/1.0	200	21311	-	Scooter-3.2.EX
unbekannt	-	-	01/Jul/2002:03:08:37...	GET	/stat/2001/jan25.htm	HTTP/1.0	200	21519	-	Scooter-3.2.EX
unbekannt	-	-	01/Jul/2002:03:53:04...	GET	/~kumlau/handreichunge...	HTTP/1.0	200	4044	-	Scooter-3.2.EX
com	-	-	01/Jul/2002:03:57:32...	GET	/~wumsta/Mikau/oben.ht...	HTTP/1.0	304	-	-	rabaz(rabazatgigabazdotcom)
com	-	-	01/Jul/2002:04:34:22...	GET	/werner/lehrgang/selfhtml...	HTTP/1.0	200	675	-	Mozilla/5.0(Slurp/cat;slurp@inktomi.com;http...
com	-	-	01/Jul/2002:05:04:44...	GET	/~wumsta/infopub/pub19...	HTTP/1.0	200	19003	-	Mozilla/5.0(Slurp/cat;slurp@inktomi.com;http...
unbekannt	-	-	01/Jul/2002:05:42:25...	GET	/stat/1998/sep23.htm	HTTP/1.0	200	20033	-	larbinsamualt9@bigfoot.com
com	-	-	01/Jul/2002:06:14:25...	GET	/may05.htm	HTTP/1.0	404	273	-	Mozilla/5.0(Slurp/cat;slurp@inktomi.com;http...
net	-	-	01/Jul/2002:06:24:31...	GET	/~kumlau/handreichunge...	HTTP/1.1	301	345	-	MicrosoftURLControl-6.00.8862
com	-	-	01/Jul/2002:06:35:50...	GET	/~gragert/folien/teil_08.pdf	HTTP/1.0	304	-	-	Googlebot/2.1(+http://www.googlebot.com/...
de	-	student	01/Jul/2002:06:47:01...	GET	/~fem/femstudium/gehei...	HTTP/1.1	200	1239	http://www.ib.hu-berlin.d...	Mozilla/4.0(compatible;MSIE6.0;WindowsNT...
net	-	-	01/Jul/2002:07:13:31...	GET	/~wumsta/uk/HUmwelog...	HTTP/1.1	304	-	http://www.ib.hu-berlin.d...	Mozilla/4.0(compatible;MSIE5.0;Windows98)...

Figure 3. Web Log File Data Set

Figure 3 shows the data entry of web log file that consists of IP, Id, Access, Date Time, Method, URL, Protocol, Bytes, Referrer, and Agent. The system used Date Time, URL, and Referrer. The web log file can be added into database by splitting space. The web pages in URL and Referrer fields are filtered by .html, .htm, .txt, .pdf, .php, .asp, .xml, .css. The same web pages in URL and Referrer are used only one time. The web pages are identified by numbers.

8.5. Retrieving Required Fields Table

Table 1. Web Log Table

DateTime	URL	Referrer
01/Jul/2002:00:01:10+0200	/%7Ekumlau/handreichungen/h59/humboldtkopf.gif	http://www.ib.hu-berlin.de/~kumlau/handreichungen/h59/
01/Jul/2002:00:03:55+0200	/~pbruhn/b-kunst.htm	http://www.das-erste.de/kultur/beutekunst.asp
01/Jul/2002:00:03:55+0200	/~pbruhn/freedom.gif	http://www.ib.hu-berlin.de/~pbruhn/b-kunst.htm
01/Jul/2002:00:03:59+0200	/bkunst/beutesuche.htm	http://www.ib.hu-berlin.de/~pbruhn/b-kunst.htm
01/Jul/2002:00:08:59+0200	/~pz/zahnpage/zapub81_.htm	http://www.google.com.ar/search?hl=es&ie=UTF-8&oe=UTF8&q=Speculum+Humana...
01/Jul/2002:00:10:56+0200	/~pbruhn/b-kunst.htm	http://www.bonzi.com/bonziportal/index.asp
01/Jul/2002:00:10:58+0200	/~pbruhn/freedom.gif	http://www.ib.hu-berlin.de/~pbruhn/b-kunst.htm
01/Jul/2002:00:10:59+0200	/~wumsta/rehm4.html	http://www.google.com/search?q=Personenpost&hl=sv&lr=&ie=UTF-8&oe=UTF8&star...
01/Jul/2002:00:13:14+0200	/~wumsta/sgml/xmldoc.html	http://www.google.de/search?q=beispiel+xml-dokument&ie=UTF-8&oe=UTF8&hl=de&...
01/Jul/2002:00:13:14+0200	/~wumsta/sgml/vpstyles.css	http://www.ib.hu-berlin.de/~wumsta/sgml/xmldoc.html
01/Jul/2002:00:13:34+0200	/~wumsta/rehmvor.html	http://www.google.de/search?hl=de&ie=ISO-8859-1&q=Tontr%E4germedien%2Bgesc...
01/Jul/2002:00:15:16+0200	/werner/lehrgang/html2/bottle1.htm	http://www.google.com/search?q=bmp+format&hl=en&lr=&ie=UTF-8&start=30&sa=N
01/Jul/2002:00:15:21+0200	/werner/lehrgang/html2/bilder/Bottle1.bmp	http://www.ib.hu-berlin.de/werner/lehrgang/html2/bottle1.htm
01/Jul/2002:00:16:11+0200	/~wumsta/rehm0.html	http://www.google.de/search?hl=de&ie=ISO-8859-1&q=Tontr%E4germedien%2Bgesc...
01/Jul/2002:00:16:11+0200	/~wumsta/dhblog.gif	http://www.ib.hu-berlin.de/~wumsta/rehm0.html
01/Jul/2002:00:18:26+0200	/~is/computerkurs/background.jpg	http://www.ib.hu-berlin.de/~is/computerkurs/ms-dos.html
01/Jul/2002:00:18:26+0200	/~is/computerkurs/ms-dos.html	http://www.google.de/search?q=ms-dos+befehle&ie=UTF-8&oe=UTF8&hl=de&meta=
01/Jul/2002:00:18:27+0200	/~is/computerkurs/compu2.gif	http://www.ib.hu-berlin.de/~is/computerkurs/ms-dos.html
01/Jul/2002:00:18:27+0200	/~is/computerkurs/ballblue.gif	http://www.ib.hu-berlin.de/~is/computerkurs/ms-dos.html
01/Jul/2002:00:19:54+0200	/~pz/zahnpage/mazarine/maz7.jpg	http://pub115.ezboard.com/fdocgonzossprechzimmerfrm6.showMessage?topicID=25...
01/Jul/2002:00:20:06+0200	/~pz/zahnpage/mazarine/maz7.jpg	http://pub115.ezboard.com/fdocgonzossprechzimmerfrm6.showMessage?topicID=25...
01/Jul/2002:00:27:24+0200	/~pz/zahnpage/bi1_2g.htm	http://www.google.com/search?hl=en&lr=&ie=ISO-8859-1&q=+Millstadt+benediktiner
01/Jul/2002:00:29:01+0200	/~fern/	http://www.google.de/search?q=fernstudium&ie=ISO-8859-1&hl=de&btnG=Google-Su...
01/Jul/2002:00:29:06+0200	/~fern/fernstudium/images/navigation_pure.gif	http://www.ib.hu-berlin.de/~fern/

Table 1 contains Data Time, URL and Referrer fields. They are used to draw graph and computing edge weight. If the URL fields contain .jpg, .png, .jpeg, .gif, and .bmp extensions, they are not used because of contents of web page.

8.6. Drawing Graph

Figure 4. Draw Web Graph

Figure 4 shows the graph that consists of nodes and edges. The nodes are web pages and the edges are time distance between two web pages.

9. CONCLUSION

This paper is web log analysis that has been widely used to infer near time about web pages. We present an algorithm to solve the web graph clustering problem. This algorithm forms a global, unbiased algorithm which can effectively extract different time web page contained in web graph of web log file. The paper will produce good results applied into the visualization of a graph in which the algorithm is a graph theoretic approach and it thus has wide applications.

This system presents the enhancement of web search using hyperlink structure of the web. In the hyperlink structure, in-links indicate the hyperlinks pointing to a page and the term out-links indicate the hyperlinks found in a page.

ACKNOWLEDGEMENTS

Our heartfelt thanks go to all people, who support us at the University of Computer Studies, Mandalay, Myanmar. This paper is dedicated to our parents. Our special thanks go to all respectable persons who support for valuable suggestion in this paper.

REFERENCES

[1] X. He, H. Zha , C.H.Q. Ding, and H. D. Simon, “Web document clustering using hyperlink structures”, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, US, NERSC Division, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720, USA.

[2] X. Huang, W. Lai, “Web Graph Clustering For Displays And Navigation Of Cyberspace”, The University of Southern Queensland, Australia, Swinburne University of Technology, Australia.

[3] B. Liu, “Web Data Mining”, Department of Computer Science, University of Illinois at Chicago, USA.

[4] Z. Markov And D. T. Larose, “Data Mining The Web”, Central Connecticut State University New Britain, CT.

[5] C. Kaur , R. R. Aggarwal ,“Web Mining Tasks And Types: A Survey”, Volume 2, Issue 2 (February 2012) ,(ISSN 2231-4334), Department of Computer Science & Engineering, Thapar University, Patiala.

[6] D. Gibson, R. Kumar, A. Tomkins, Discovering Large Dense Subgraph in Massive Graphs, VLDB Conference, 2005.

[7] C. C.Aggarwal, Managing and Mining Graph Data, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA,

[8] D. F Kibler and S. Hampson, " Learning Weight Matrices for Identifying Regulatory Elements", Information Computer Science Department, University of California, Irvine, USA.

[9] N.P. Vander, H.G. T. Morsche, and R.R.M. Mattheij, "COMPUTATION OF EIGENVALUE AND EIGENVECTOR DERIVATIVES FOR A GENERAL COMPLEX-VALUED EIGENSYSTEM", Electronic Journal of Linear Algebra, ISSN 1081-3810, A publication of the International Linear Algebra Society Volume 16, pp. 300-314, October 2007.

[10] S. KOUACHI, "'EIGENVALUES AND EIGENVECTORS OF TRIDIAGONAL MATRICES", Electronic Journal of Linear Algebra ISSN 1081-3810, A publication of the International Linear Algebra Society, Volume 15, pp. 115-133, April 2006.

[11] A. Farkas, "The Analysis of the Principal Eigenvector of Pairwise Comparison Matrices", Budapest Tech, Faculty of Economics, 1084 Budapest, TavaszmezÅ út 17, Hungary.

[12] X. Chen, L. Guo, and Z. Fan, " LEARNING POSITION WEIGHT MATRICES FROM SEQUENCE AND EXPRESSION DATA",10:6 WSPC/Trim Size: 11in x 8.5in for Proceedings, May 14, 2007.

Authors

She is Associate Professor, Head of Department of Research and Development II in University of Computer Studies, Mandalay, Myanmar. Her research areas include Information Retrieval, Cryptography and Network Security, Web Mining and Networking. She received her B.Sc. (Physics), M.Sc.(Physics) from Yangon University, Myanmar and M.A.Sc.(Computer Engineering) and Ph.D.(Information Technology) from University of Computer Studies, Yangon, Myanmar.

Author studied computer science at the University of Computer Studies, Monywa, Myanmar where she received her B.C.Sc Degree in 2011. She received B.C.Sc(Hons:) in computer science from the University of Computer Studies at Monywa, Myanmar in 2012. Since 2012, Author has studied computer science at the University of Computer Stu dies, Mandalay, Myanmar where her primary interests include web mining, graph clustering, grouping and web log analysis.