The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 7 ||Pages|| 35-41||2013||
ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805

Graph Theoratic Techniques for Web Content Mining

[1]Sandhya (M.Tech CSE), [2,]Mala Chaturvedi (M.Tech CSE), [3,]Anita Shrotriya
Jayoti Vidyapeeth Women's University, Jaipur,
Manav Bharti University, Salon Himanchal Pradesh
Assistance Professor At Jayoti Vidyapeeth Women's University, Jaipur

--ABATRACT--
Web is a large pool of data and information. On web there are various types of contents they may be in the form of text, images, audio, videos, metadata and hyperlinks. Web content mining encompasses resource discovery from web, document categorization and clustering and information extraction from web pages. In this paper we discuss the graph theoretic techniques for web content mining that are commonly used and compare the result.

Date of Submission: 31 June 2013, Date of Publication: 7.July 2013

I. INTRODUCTION

The World Wide Web is a rich source of information and continues to expand in size and complexity. Retrieving of the required web page on the web, efficiently and effectively, is becoming a challenge for our user who is new on the internet pool of data. World wide web is the collection of documents, images, text files and other forms of data in structured, semi structured and unstructured. The data store on web is become main source of information for the users in many domain. This increase the users on web and most of the users are inexperienced. Due to heterogeneity and lack of structure of web data web mining becoming challenging task. The web contains mixture of many kinds of information and this information constantly keep changing and makes web dynamic and noisy. Web mining is dealing with this problem. There are several web mining algorithm that helps user to find information or data for that actually they are looking on web. There are three category of web mining and there are many algorithms. In this paper we focus on the web content mining and the algorithm used in this technique. The aim of this paper is to study the graph theoretical algorithms that are used in web content mining.

II. WEB MINING

Web mining is the retrieving data or information on web. Web mining is data mining technique that automatically discovers the information from web. It is having the following task:

1. **Resource finding**: the task of retrieving intended Web documents.
2. **Information selection and pre-processing:** automatically selecting and pre-processing specific information from retrieved Web resources.
3. **Generalization:** automatically discovers general patterns at individual Web sites as well as across multiple sites.
4. **Analysis:** validation and/or interpretation of the mined patterns.

Web content mining targets the knowledge discovery, in which the main objects are the traditional collections of text documents and, more recently, also the collections of multimedia documents such as images, videos, audios, which are embedded in or linked to the web pages. Data mining because many data mining techniques can be applied in web content mining. It is also related with text mining because much of the web contents are text, but is also quite different from these because web data is mainly semi structured in nature and text mining focuses on unstructured text. *Web structure mining* focuses on the hyperlink structure of the Web. The different objects are linked in some way. Simply applying the traditional processes and assuming that the events are independent can lead to wrong conclusions. However, the appropriate handling of the links could lead to potential correlations, and then improve the predictive accuracy of the learned models. The goal of Web Structure Mining is to generate structured summary about the website and web page. *Web Structure Mining* has a relation with *Web Content Mining*. It is quite often to combine these two mining tasks in an application. *Web usage mining* focuses on techniques that could predict the behavior of users while they are interacting with the WWW. It is the process by which we identify the browsing patterns by analyzing the navigational behavior of user. Web usage mining collects the data from Web log records to discover user access patterns of Web pages. It focuses on techniques that can be used to predict the user behavior while the user interacts with the web. This activity involves the automatic discovery of user access patterns from one or more web servers.

TABLE1. WEB MINING CATEGORIES

Web Mining				
	Web Content Mining		**Web Structure Mining**	**Web Usage Mining**
	IR View	**DB View**		
View Of Data	–Structure –Unstructured	–Semi Structure –Web Site as DB	-Link Structure	-Interactivity
Main Data	- Text documents -Hypertext documents	-Hypertext documents	-Link Structure	-Server Logs -Browser Logs
Representation	-Bag of words, n-gram Terms, -phrases, Concepts or ontology -Relational	-Edge labeled Graph, -Relational	-Graph	-Relational Table -Graph
Method	-Machine Learning -Statistical (including NLP)	-Proprietary algorithms -Association rules	-Proprietary algorithms	-Machine Learning -Statistical -Association rules
Application Categories	-Categorization -Clustering -Finding extract rules -Finding patterns in text	-Finding frequent sub structures -Web site schema discovery	-Categorization -Clustering	-Site Construction -adaptation and management -Marketing, -User Modeling

III. WEB CONTENT MINING ALGORITHM

These are web content mining algorithms. These algorithms are different from the previous web content mining algorithm. Here we are using graph theoretical technique. This is different from other in this we

3.1. THE GRAPH BASED K-MEANS CLUSTERING ALGORITHM

Inputs: The set of n data items (representing by graphs) and a parameter k, defining the number of cluster to create.

Outputs: The centroids of the clusters (represented by median graphs) and for each data item the cluster (an integer in [i, k] it belongs to.

Step1: Assign each data item randomly to a cluster (from 1 to k)
Step2: Using the initial assignment, determine the median of the graph of each cluster.
Step 3: Given the new medians, assign each data item to be in the cluster of its closest median, using a graph theoretic distance measure.
Step4: Re-compute the medians as in step2 repeat step3 and 4 until the median do not changes.

3.2. CLUSTER HIERARCHY CONSTRUCTION ALGORITHM

Summary of Notation Used in CHCA

B the set of row vectors remaining to be processed by the algorithm (the "before set")
C$^{\rightarrow}$ a row vector that is the current candidate for becoming a new cluster
d$^{\rightarrow}$ any cluster (row vector) in the "current set" *K*
d$^{\rightarrow'}$ any cluster (row vector) in the "done set" *D*
D the set of row vectors already processed by the algorithm (the "done set")
K the set of parents of the current cluster candidate
m the number of columns in the membership table *X* (the number of terms)
n the number of rows in the membership table *X* (the number of web pages)
***n*˜** the number of rows in the reduced membership table *X* (number of distinct rows)
r$^{\rightarrow}$ a row vector from *X*, representing the terms appearing on a particular web page
x$^{\rightarrow}$ a cluster in the hierarchy created by the algorithm
X the membership table with *n* rows and *m* columns
˜

X the reduced membership table, created from *X*, with *m* columns and $n^{\sim}$ rows
MCT a user defined parameter, the Maximum Cluster Threshold, which determines the maximum number of clusters to create
MPT a user defined parameter, the Minimum Pages Threshold, which determines the minimum number of pages each cluster can contain
MDT a user defined parameter, the Maximum Distance Threshold, which determines the maximum difference in terms to consider when adding new clusters
$\boldsymbol{\phi_m}^{\rightarrow}$ a row vector with *m* components, all of which are 0 (the empty cluster) – the difference operation between vectors defined above (in Step!3e only, we use this operator as a shorthand for the set theoretic removal of an element from a set)

The algorithm takes a binary matrix (a table) as input. The rows of the table corresponds to the objects we are clustering. Here we describing this algorithm with web pages, but the method is applicable to other domains as well.

Step1: Creating a data structure (the membership table **X**). Table is binary representation of the relationships between each web pages and each term and is the primary input to the algorithm.
Table contains rows and columns.
Row = The number web pages.
Columns =Total no of terms which appears on at least one page.
If page i contains the $\mathbf{j_{th}}$ term , then at row **i** and **j** is the membership table($\mathbf{X_{ij}}$) there will be a 1; otherwise there will be a 0.

The term "membership table" tells that this table is for clustering, not for an information reterival task.
Step2: In this step the duplicate row vector are remove from table **X** and a new reduced membership table **X** is created. In this table only **n□(≤n)** distinct row vector exist.
The new data structure **x□** is created which creates the hierarchy. We can only considering the unique row vector, that's why we only need the smaller **x□** table to create the hierarchy. This is the pre processing step of the algorithm. Since in this step we reduced the table, so that memory requirements and running time of the main parts of the algorithms are reduce.

Step3: The cluster hierarchy from **x□** is created by using user defined values for parameters **MCT**, **MDT** and **MPT**. The description for the produces is as follows:- Let B, the "Before set" , be the set of row vectors not yet assigned to any cluster. It initially contains all the row vector of **x□**, namely $\{\mathbf{r_1}^{\rightarrow}, \ldots\ldots, \mathbf{r_n}^{\rightarrow}\}$.
Set **D,** the ''Done set'' is the set of clusters created by the algorithm so far. **D** is initially set to $\{\boldsymbol{\phi_m}^{\rightarrow}\}$ while $\mathbf{B \neq \emptyset}$(i.e non empty, there are still candidates to process) and $|\mathbf{D}| \leq \mathbf{MCT+1}$ do the following steps:

Step3(a): $\mathbf{C}^{\rightarrow}$, the ''candidate vector" is a row vector from set **B** which has a minimum number of terms (1 bits). IF there is more than one row with the same minimum number of terms, selected one at random for $\mathbf{C}^{\rightarrow}$ or we can also write $\mathbf{C}^{\rightarrow}$ be a row vector from set B such that for all $\mathbf{a_i}^{\rightarrow} \in \mathbf{B}$, $|\mathbf{C}^{\rightarrow}| \leq | \mathbf{a_i}^{\rightarrow}|$.

Step3(b): Search a set of clusters $\mathbf{K} \subseteq$ of **D** such that $\mathbf{C}^{\rightarrow}.\mathbf{d}^{\rightarrow}=\mathbf{d}^{\rightarrow}$ for all $\mathbf{d}^{\rightarrow} \in \mathbf{K}$ and $\mathbf{d}^{\rightarrow} \in \mathbf{D}$.
The parent cluster or existing cluster in the set k must satisfying the following two conditions simultaneously:
1) **Sub set hood:** $\mathbf{C}^{\rightarrow}.\mathbf{d}^{\rightarrow}=\mathbf{d}^{\rightarrow}$ states that K only consists of those clusters. Such that the attributes of the parent clusters (those in **K**) are a subset of those of the child cluster.

2) **Minimum Distance:** when using the vector different operation we defined the standard absolute value operation (abs).
Abs ($|\mathbf{C}^{\rightarrow}| - |\mathbf{d}^{\rightarrow}|$,
The condition $\mathbf{C}^{\rightarrow} - \mathbf{d}^{\rightarrow} \leq \mathbf{C}^{\rightarrow} - \mathbf{d}^{\rightarrow}$ means the direct parents of $\mathbf{C}^{\rightarrow}$ are those clusters that have the least "distance" (difference in terms) from the child (or, put another way , the parents of $\mathbf{C}^{\rightarrow}$ those clusters with the most terms in common). This needed to enforce a proper ordering among the clusters.

Step3(c): If the minimum of $\mathbf{C}^{\rightarrow} - \mathbf{d}^{\rightarrow}$ for all $\mathbf{d}^{\rightarrow} \in \mathbf{K} \leq \mathbf{MDT}$ or **K** contains $\boldsymbol{\phi_m}^{\rightarrow}$ (i.e $\mathbf{C}^{\rightarrow}$ is a base cluster), then candidate vector $\mathbf{C}^{\rightarrow}$ becomes the child of all clusters in **K** by inheritance (or multiple inheritance in **K** has more than one element). Otherwise, skip to (**step3(e)**). Cluster with $\boldsymbol{\phi_m}^{\rightarrow}$ (the empty cluster) as a parent are base clusters. Each child cluster inherits the terms from its parent cluster(s) and adds its new terms.

Step3(d): $\mathbf{D = D \cup C}^{\rightarrow}$ **.** in other words, add the row vector corresponding to $\mathbf{C}^{\rightarrow}$ to set **D**(the " Done Set ").

Step3 (e): B= B – $C^{\rightarrow}$. that is, remove the row vector corresponding to **$C^{\rightarrow}$** from the set **B** (“the before set”).

Step 4: After the loop in step3 ends, the initial cluster hierarchy has been created. However the web pages have not yet been assigned to clusters. For each web pages (row) in the original membership table **X ,**assign the web pages to a cluster using the distance measure:

$$\mathbf{D}\text{is }(r,i) = \mathbf{1/m} \sum_{j=1}^{m} \mathbf{Y_{ij} \cdot (Y_{ij} - X_{rj})} \qquad \text{..........}\mathbf{(4.1)}$$

where **m** is the number of terms(number of columns), $\mathbf{X_{rj}}$ is the jth term in row r of the original membership table , and $\mathbf{Y_{ij}}$ is jth term of cluster i in the hierarchy (vector corresponding to the cluster).

Step 5: starting with the clustering farthest down in the hierarchy , remove those clusters that have a number of web pages assigned to them less than **MPT.** Re assign the pages from the deleted clusters to the remaining clusters.

3.3. THE GRAPH THEORETIC GLOBAL K-MEANS ALGORITHM

The global k-means algorithm proposed by **likas et al** is a way of determining “good” initial cluster centers for the k-means algorithm. The basic procedureis an incremental computation of cluster centers.

starting at the case of one cluster(k=1), the cluster center is defined to be the centroid of the entire data set.

for general case of k-cluster, the centres are determined by taking the centers from the k-1 cluster problem & then determining the optimum location of a new center.

this is done by considering each data item as the new cluster center and then executing the k-means algorithm with that particular set of cluster centers and determining which data point minimize the error as defined by:

$$\mathbf{E(m_1 \ldots\ldots m_M) = \sum_{i=1}^{N} \sum_{k=1}^{M} I(x_i \in c_k) \left\| x_i - m_k \right\|^2} \text{ ....}\mathbf{3.1}$$

where

N= is the number of data items

M= number of clusters

X_i= data item i

m_k = is cluster center k

I(x)= 1 if X is true & 0 otherwise for many applications this will be too time consuming so the author have also proposed a “fast” version of global k-means. According to this version instead of running k-means when considering each data item as a new cluster center candidate, we calculate the following:

$$\mathbf{b_n = \sum_{j=1}^{N} \max\left(d^j_{k-1} - \left\| x_n - x_j \right\|^2, 0\right)} \text{ ....}\mathbf{3.2}$$

where,

$\mathbf{d^j_{k-1}}$= is the distance between data item $\mathbf{x_j}$ & its closet cluster center for the k-1 clustering problem.

We then select the new cluster to be data item $\mathbf{x_i}$ where:

$$\mathbf{i = \arg\max_{n} b_n} \text{ ....}\mathbf{3.3}$$

3.4. A GRAPH BASED EXTENSION OF THE K- NEAREST NEIGHBOUR METHOD

Before understanding extension of the K-NM first we describe the K-NM and then its extension. The basic K-NM algorithm is as follows. First we have a database of training examples (instances). In the basic K-NM approach these will be numerical vector in some real valued feature space.

The basic K- Nearest Neighbour Algorithm

INPUT: A set of pre classified training instance a query instance q, and a parameter K, defining the number of neighbours to use.

OUTPUT: A labels indicating the class of the query instance q.

Step 1: find the K-closest training instances to q according to a distance measure.

Step 2: select the class of q to be the class held by the majority of K- nearest training instances.

The extension that is node on K- nearest neighbours algorithm is, we using graph as data for K-nearest neighbours, by using the method of graph techniques to modeled web documents.
And using graph theoretical techniques distance measure:

$$\mathbf{d_{MCS}\ (G_1, G_2) = 1 - \frac{|MCS\ (G_1,G_2)|}{Max\ (|G_1|,|G_2|)}} \quad \ldots\ldots\ldots\ldots\ldots\ldots 4.1$$

$$\mathbf{d_{WGU}\ (G_1, G_2) = 1 - \frac{|MCS\ (G_1,G_2)|}{|G_1|+|G_2|-|MCS\ (G_1,G_2)|}} \quad \ldots\ldots\ldots\ldots 4.2$$

$$\mathbf{d_{UGU}\ (G_1, G_2) = |G_1|+|G_2|-2|MCS\ (G_1,G_2)|} \quad \ldots\ldots\ldots\ldots 4.3$$

$$\mathbf{d_{MMCS}\ (G_1, G_2) = |G_1|+|G_2|-|MCS\ (G_1,G_2)|} \quad \ldots\ldots\ldots\ldots 4.4$$

$$\mathbf{d_{MMCSN}\ (G_1, G_2) = 1 - \frac{|MCS\ (G_1,G_2)|}{|MCS\ (G_1,G_2)|}} \quad \ldots\ldots\ldots\ldots\ldots\ldots 4.5$$

3.5. GRAPH HIERARCHY CONSTRUCTION ALGORITHIM (GHCA)

1.5.1 PARAMETER

In GHCA we use five user defined parameter to control the properties of resulting cluster hierarchy. Most of these are the similar to the CHCA parameter.

(1) **MTT** (Maximum Terms Threshold). This parameter restricts the maximum number of vertices in the resulting graph representations of documents. We have two options. We can use the *MTT* most frequently occurring terms on each page (where frequency means the number of occurrences on a given page). Or we can create a common set of the *MTT* most frequently occurring terms across all pages (where frequency means the number of pages where a term occurs at least once). The default option is to use the 30 most frequently occurring terms across all pages.

(2) **MPT** (Minimum Pages Threshold). This parameter is used in the pruning section of GHCA by removing clusters that have fewer than *MPT* native pages assigned to them. The default value is 3.

(3) **MDT** (Maximum Distance Threshold). This parameter is used for restricting the growth of the hierarchy. We do not add clusters to the hierarchy whose difference in size from their parent(s) is greater than *MDT*. As we mentioned in Chapter!4, the size of a graph is defined as the sum of the number of edges and vertices in the graph. The default value of *MDT* is 2, which is large enough to allow the addition of one new term to an existing phrase (*i.e.* one node and one edge).

(4) **MCT** (Maximum Cluster Threshold). This parameter is used to limit the overall size of the hierarchy. We stop the hierarchy construction phase of the algorithm once it has created *MCT* clusters or we have no candidate graphs remaining. The default value is 50.

(5) **BCST** (Base Cluster Size Threshold). This parameter is used to limit the size of base (top level) clusters. We do not create a new base cluster if its size exceeds *BCST*. The default value of *BCST* is 3. The default value is large enough to admit a two term phrase (*i.e.* two nodes connected by an edge) as a base cluster.

3.5.2 ALGORITHM

There three basic step for this algorithm they are initial hierarchy construction, document assignment and bottom-up clustering pruning.

Step1: Initial hierarchy construction

1. Find the candidate graph with minimum size, and the size of a graph **G**, **│G│**, is defined as the sum of the number of edges and vertices, **│V│+│E│**, and make it the cluster candidate. If there is a tie than, select one of the graphs at random.
2. Now we have to determine the possible parents of the cluster candidate in the hierarchy, such that any parents of the cluster candidate are sub-graphs of the cluster candidate and the distance, defined as the difference in size between the two graphs, is minimum. .
3. If the cluster candidate is the base cluster (and it has no parents cluster in step 2) and if the size of the graph is less than or equal to BCST, than we have to add cluster candidate as a base cluster in the hierarchy.
4. If the cluster candidate is not a base cluster and the difference in size between the cluster candidate and the parents is less than or equal to MDT, than we have to add the cluster candidate to the hierarchy.
5. Now we have to remove cluster candidate from the set of candidate graph.
6. After doing all this if there is number of cluster in the hierarchy is less than MCT and there is still candidate graph remaining, than move to step1; or proceed to the initial document assignment phase.

Step2: Document assignment phase

For every page that is represented in the set of page graphs, we have to determine that which cluster in the hierarchy have the smallest distance according to the MCS distance measure:

$$\mathbf{d_{MCS}\,(G_1, G_2) = 1 - \frac{|MCS\,(G_1,G_2)|}{Max\,(|G_1|, |G_2|)}}$$

1. IF there is 1 minimum distance, than we have to skip this page and return to 1.
2. Assign the page to the cluster(s) which have minimum distance as a *native page.*
3. Also assign the page to super-clusters above the clusters selected in step 2 in the hierarchy as an inherited page; continue to propagate the inherited page up the hierarchy from child to parent until a base cluster is reached.

Step3: Cluster Pruning Phase

1. Starting with the lowest level in the hierarchy, delete all clusters at that level from the hierarchy that have less than ***MPT*** *native pages* assigned to them.
2. Given the new hierarchy, re-assign all the pages from the deleted clusters as described above in **DOCUMENT ASSIGNMENT PHASE** steps (1) to (3).
3. Fix orphaned clusters by updating the parent information as in **INITIAL HIERARCHY CONSTRUCTION** step (2).
4. Repeat steps (1) to (3) going up one level in the hierarchy each iteration until the top level is reached.

Step4: Results Display Methodology for GHCA

1. For each cluster, first display the longest simple paths (acyclic paths not contained in any other acyclic paths) in the graph as ordered phrases; next show any isolated nodes as single terms.
2. If the cluster is not a base cluster, show only those phrases or terms which are specific to the graph (i.e., those not displayed for a parent cluster).

IV. CONCLUSION

Web mining having three categories and web content mining is one of them. These algorithm used in web content mining, these algorithms using graph based technique. They are different from other web mining algorithm because they do not using vector technique. These algorithms are more optimal than the vector technique. Graph based algorithm makes web content mining more suitable than the vector based technique. This paper is focus only on graph theoretical technique of web content mining. Here we would like to add a hypothesis that if we use fuzzy logic with graph theoretical techniques than web content mining may gives us more better and optimal result.

REFERENCES

[1] Page Ranking Algorithms for Web Mining,Rekha Jain Department of Computer Science, Apaji Institute, Banasthali University C-62 Sarojini Marg, C-Scheme, Jaipur,Rajasthan Dr. G. N. Purohit Department of Computer Science, Apaji Institute, Banasthali University.

[2] Graph-theoretic techniques for web content mining,Adam Schenker, University of South Florida

[3] ON TWO ALGORITHMS USED IN WEB STRUCTURE MINING,Claudia Elena Dinucă Ph. D Student,University of Craiova Faculty of Economics and Business Administration Craiova, Romania Dumitru Ciobanu Ph. D Student University of Craiova Faculty of Economics and Business Administration Craiova, Romania.

[4] Performance Improvement Of Web Usage Mining By Using Learning Based K-Mean Clustering , Ms. Vinita Shrivastava M.Tech (Information Technology) Technocrats Institute of Technology, Mr. Neetesh Gupta Head Of Department (Information technology) Technocrats Institute of Technology,Bhopal india.

[5] Baeza-Yates, R. and Ribeiro-Neto, B., 1999. Modern Information Retrieval. Addison-Wesley-Longman Publishing Co.Harlow, England.

[6] Cooley R. et al, 1997. Web Mining: Information and Pattern Discovery on the World Wide Web. Proceedings of theIEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), pp. 12-23.

[7] Doszkocs, T. E. et al, 1990. Connectionist models and information retrieval. In Annual Review of Information Scienceand Technology (ARIST), 25, pp. 209-260.

[8] Kosala R. and Blockeel H., 2000. Web Mining Research: A Survey. In Newsletter ACM SIGKDD, Vol. 2, Issue 1, pp. 1-15