INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Incorporation of contextual information through Graph Modeling in Web content mining

Kaushik Kishore Phukon[1]*

1 Pandit Deendayal Upadhyaya Adarsha Mahavidyalaya, Gauhati University, Bongaigaon, 783383, Assam, India. Tel.: 917002551636

OPEN ACCESS

Received: 13.09.2020

Accepted: 07.12.2020

Published: 19.12.2020

Citation: Phukon KK (2020) Incorporation of contextual information through Graph Modeling in Web content mining. Indian Journal of Science and Technology 13(46): 4573-4578. https://doi.org/10.17485/IJST/v13i46.1660

* **Corresponding author**.

Tel: 917002551636
kaushikphukon@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The objectives of this research article is to deal with the problem of web document clustering by modeling the web documents as directed completely labeled graphs that incorporate contextual information in the computation process to the extent required. The computational complexity of the MCS algorithm based on this graph model is $O(n^2)$, n being the number of nodes. As graph similarity using MCS is an NP-complete problem, so this is an important result that allows us to forgo sub-optimal approximation approaches and find the exact solution in polynomial time. **Method:** The first step towards this new approach of web document clustering is the representation of the web documents with the help of a directed completely labeled graph that can retain contextual information of the document under consideration. After graphical modeling of the document, the next step is the calculation of similarity between the graphical objects. For this purpose, a customized algorithm proposed as Algorithm for Maximum Common Subgraph Isomorphism (AMCSI)[(1)] based on a backtracking search scheme is being used. The proposed AMCSI algorithm is solving the problem of maximum common subgraph isomorphism in polynomial time. After obtaining the value for the similarity between the graphical objects we are again using a customized fuzzy-c means algorithm to produce clusters from the target set of web documents. We are using multidimensional scaling to express the distance values between the web pages (graphs) in two coordinates (x,y) and deterministic sampling to calculate the graph median in the process of fuzzy c-means clustering. **Findings:** We present an alternative method for web document clustering by representing the web documents as directed completely labeled graphs where the computational complexity of the MCS algorithm is $O(n^2)$[(1)]. A new distance measure is also developed based on the directed completely labeled graph representation which is giving 16.9% better result than the prevailing methods[(2)]. For the clustering purpose, we have chosen the fuzzy c-means clustering algorithm and customizing the original algorithm to fit with graphical objects. This approach enables us to model the web documents as graphs without discarding contextual information and then cluster these graphical objects with the help of a well-established clustering algorithm.

Keywords: Vector; graph; web document; subgraph; isomorphism; fuzzy; clustering

1 Introduction

A good number of data mining methods are available that deal with plain text documents. But the document mining methods that are suitable for text documents are not fit for web documents. Web documents are different from plain text documents as they consist of certain markup elements. The incorporation of these markup elements or the tags into the mining process can lead to greater accuracy, efficiency and effectiveness of the results obtained henceforth. This is only possible when we have ways to incorporate the information regarding the markup elements into the mathematical model representing the object under consideration (web document). Discarding markup information or amalgamation of the markup information with the help of an additional process is one of the major limitations of existing web content mining methods.

Another issue is the non-homogeneity of the web documents. As the web is highly non-homogeneous in nature it is not always feasible to apply unambiguous knowledge. For instance, we may consider the term "Himalaya". This term can imply many things on the internet such as a commercial website, a mountain, a person, a locality, a forest, a company, etc. Thus, for web resources, it may not be feasible to apply unambiguous knowledge or information from a domain, that can otherwise be used effectively for a structured knowledge discovery process.

Also, the web content is dynamic in nature. The web pages are very much porn to have changes in terms of contents as well as location. This dynamic nature of the internet restricts the use of traditional data mining or knowledge discovery methods which may possibly be unable to trace the nature of web resources. Such techniques may lead to out of date results. A significant fraction of the huge volume of web resources changes frequently and must be processed periodically to have meaningful up to date results. Some other issues that we must deal with when processing web documents for the purpose of content mining are the variation in size, style, layout, languages etc.

The traditional methods generally require some training set to learn about the document corpus. The creation of a training set that is capable of managing ambiguity and which can incorporate changes as it happens is hardly achievable. In the light of the above-mentioned drawbacks of the traditional methods, in this paper, we are presenting a graph model that is capable of representing web documents without discarding any valuable information that can be used for clustering the documents.

This graph-based modeling of web documents happens to be more beneficial with respect to both representational and clustering perspective. In general graph similarity using maximum common subgraph is an NP- complete problem. But our completely directed graph-based representation is able to keep the contextual information of the documents reducing the time complexity of determining the MCS to $O(n^2)$. The new graph distance measure based on the new graph representation is also giving 16.9% better result. Further, the need for clustering the objects under consideration can be met with straightforward extensions of classical clustering algorithms such as fuzzy c- means algorithm to work with graphical objects.

2 Web document content mining process

Web Document Content mining processes are activities done to discover and extract useful information from the web documents usually with the assistance of a machine. Certain instances of these processes are clustering, classification, retrieval, filtering, extraction and summarization.

All these processing activities require a pre-processing step to accomplish their task. This preprocessing step generally consists of converting the web documents or the set(s) of web documents into some mathematical model of values that facilitate the application of further treatment available to calculate different essential parameters. The general procedure that may be followed by each of the above web document content mining process is as below:

Step 1: Conversion of the respective web documents into a computer processable format.

Step2: Similarity measurement among the documents in the converted format.

Step3: A clustering process that can utilize the outputs provided by the above two steps.

As stated, most of the document clustering methods consider a document as a collection of some dictionary words and do not bother about the contextual organization and the ideas conveyed by the document. As a result, the similarity measures based on such a representation model cannot perceive contextual similarity due to the lack of contextual information [3]. Numerous attempts have been made to discover some new way to represent text apart from the term frequency approach. Some of the significant efforts are N/grams [4], bigrams [5], extraction of words semantic relations through corpus statistics [6], the use of background knowledge by replacing words with their higher concepts in the ontology [7], and the consideration of word sequences with the help of a graphical model [8].

We found six different graph models for representing web documents in the literature [9]. They are: standard, simple, n-distance, n-simple distance, absolute frequency, and relative frequency. Numerous research works have been perused in the area of graph similarity in order to incorporate the additional information allowed by graph representations. In the literature, the work comes under several different topic names including graph distance, graph matching, inexact graph matching, error-tolerant graph matching or error-correcting graph matching. There are many clustering techniques in the literature, each adopting a certain strategy such as K-means algorithm [10], hierarchical clustering [11,12], Graph-based clustering [9,13–15] and many others [11,16–18]. A variant of K-means that allows instances to be a member of more than one cluster at a time is known as Fuzzy C-means (FCM). Instead of having a single membership of objects to their respective clusters, FCM allows the objects under consideration to belong to several (not necessarily) clusters simultaneously [19,20]. The FCM algorithm, proposed by Dunn in the year 1974 and extended by Bezdek in the year 1981, can be applied if the objects of interest are represented as points in a multi-dimensional space.

3 The composite graph model for web document representation

The closer two words are to each other; the stronger their connection tends to be and graphs are the right scientific structures to represent such a relationship [21]. There is a very close relationship between graph theory and calculus of binary relations. A binary relationship

between two objects x_i and x_j may be stated as a binary relation R between the pair (x_i, x_j), Where X can be defined as set of objects, X={x_1, x_2, …}. Symbolically this representation can be stated as $x_i R x_j$ and say that x_i has relation R to x_j. The obvious or the most natural way of representing a relation is through directed graphs. In a directed graph each x_i belongs to X is represented by a vertex x_i. If x_i has some specified relation R to x_j, a directed edge is drawn from vertex x_i to x_j, for every pair (x_i, x_j). For example, Figure 1 represents a relation "is the successor of" on a set of 5 consecutive numbers {1, 2, 3, 4, 5}, with each edge labeled with the members' respective distances from all the other predecessors. It seems to be obvious from the above definition that every binary relation on a finite set can be represented by a digraph without parallel edges.

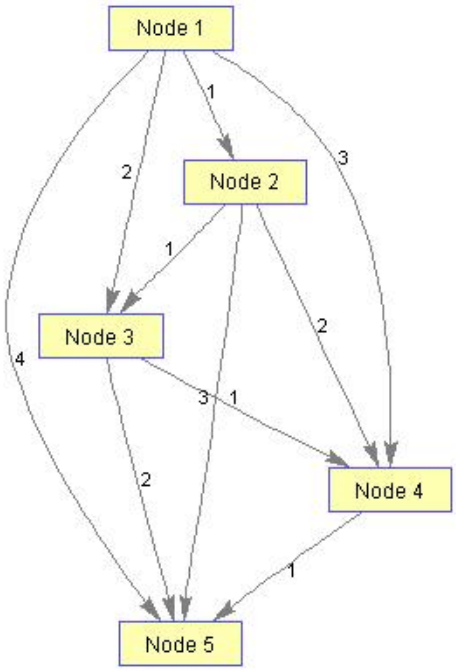

Fig 1. Representing a relation "is successor of" on a set of 5 consecutive numbers {1, 2, 3, 4, 5}

The composite graph model (CGM) [22] which was modeled by this author in the year 2012 which represents a web document as a directed and completely labeled graph. The CGM was developed with help of the Tag Sensitive Graph Model (TSGM) [22] and Context-Sensitive Graph model (CSGM) [22]. In the composite graph representation, we are using the TSGM to represent three sections of a general web page namely head, link and address. The use of TSGM enables us to utilize the markup information available in the web document. We are using CSGM to represent the text section because of its efficiency to represent a large text section.

The composite model successfully retains all the advantages of both TSGM and CSGM and eliminates the drawbacks of both the parent models.

3.1 Algorithm for adjacency matrix representation of a web document in accordance with the CGM

A binary relation R on a set can also be represented by a matrix, called a relation matrix. It is commonly a (0, 1) n by n matrix where n is the number of elements in the set. The i, j^{th} entry in the matrix is 1 if $x_i R x_j$ is true and is 0, otherwise. For example, the relation matrix for the sentence: "Not everything that can be counted counts, and not everything that counts can be counted" is (Table 1):

Table 1. The relation matrix representing the sentence

Nodes	not	everything	that	can	be	counted	counts	,	and
not (Node1)	0	**2**	0	0	0	0	0	0	0
everything (Node2)	0	0	**2**	0	0	0	0	0	0
that (Node3)	0	0	0	**1**	0	0	**1**	0	0
can (Node4)	0	0	0	0	**2**	0	0	0	0
be (Node5)	0	0	0	0	0	**2**	0	0	0
counted (Node6)	0	0	0	0	0	0	**1**	0	0
counts (Node7)	0	0	0	**1**	0	0	0	**1**	0
, (Node8)	0	0	0	0	0	0	0	0	**1**
and (Node9)	0	0	0	0	0	0	0	**1**	0

The CGM transforms the original text into a wordnet (graph) in which all relationships between adjoining words are retained. We are going to present how the CGM performs on a text T of length n words, drawn from a vocabulary $\sum$ containing σ different words.

To implement the word net a digraph G = (V, E) is considered in accordance with the following features:

- Each unique term appearing in the document becomes a node in the graph representing the document. Each node is labeled with the term it represents and a non-negative integer value $x \in [1, \sigma]$ where σ is the total word count in T. The node labels in a document graph are unique, since a single node is created for each keyword even if a term appears more than once in the text.
- Second, if word a immediately precedes word b somewhere in the document, then there is a directed edge from the node representing term a to the node representing term b with an edge labeled as (p:q), where p is a user-specified parameter representing the distance between the two words and q is the number of times that the edge has been traversed.
- There are no multiple edges in G. If there is more than one transition between two consecutive words, only a single edge is modeled.

The above-mentioned steps for modeling a document into a graph can also be used in conjunction with other modeling ways to convert natural language texts into a suitable mathematical format for further processing. The separator modeling also plays a vital role in modeling with the CGM technique by directly affecting the space/time tradeoff. We choose spaceless words to model separators for T on CGM. The following algorithm formalizes the CGM transformation process.

```
1: V ← ϕ
2: E ← ϕ
3: current ← StartVertex;
4: previous ← null
5: while there are more words in T do
6:          word ← T.ParseWord();
7:          if word ∉ V then
8:          V ← V ∪ word;
9:          destination ← V.retrieveID(word);
10:         If current ≠ destination
11:                edge ← (current, destination);
12:                If prev ≠ current
13:                       edge ← (prev,destination)
14:                else goto step 17
15:                end if
16:                E ← E ∪ edge;
17:                G_text.encode(V);
18:                G_vertex.encode(word);
19:                 G_edge.encode(edge)
20:         else previous ← current
21:                current ← destination;
22:         end if
23:         else
24:                destination ← V.retrieveID(word);
25:                edge ←(current, destination);
26:                if edge ∉ E then
27:                       E ← E ∪ edge;
28:                       G_edge.encode(edge);
29:                else id ← E.retrieveID(edge);
30:                       E.update(edge);
31:                end if
32:         end if
33:                       previous ← current
34:         current ←destination;
35: end while
```

Algorithm 1: CGM transformation process

After constructing graph representations for a set of web documents, the next job is to find out the similarities among them for the purpose of clustering. Several techniques are there to measure the similarity among different graphical objects. All these techniques accomplish their job through graph matching. This particular problem of graph matching refers to the topics of inexact graph matching or graph similarity. We have several instances where we can see the application of graph matching to solve different complex problems in the field of image processing, pattern matching, job scheduling, structure analysis, bioinformatics, network management, route management, etc.

The composite graph model as proposed has been developed for representing web documents which can retain more structural information about the contents of a web page than the prevailing models. This necessitates the development of an enhanced similarity

measure which is essentially based on some established method that enables us to pass the additional information into the calculation process.

4 New distance measure based on maximum common subgraph approach

Although the maximum common subgraph approach is a widely accepted graph similarity measure, there are no reported findings to indicate a de facto standard. We decided to consider the maximum common subgraph approach to pursue our study because of its simplicity, efficiency and popularity.

The maximum common subgraph approach is an effective and efficient way to calculate graph distances. But with our newly developed composite graph model, the prevalent method will not work well as the method is based only on number of nodes of the graphs under consideration. As our composite model is capable to hold more information than that of node information only, so to have full benefit from the proposed composite model we are enhancing the MCS distance measure as blow:

$$\text{dist}_{MCS}\left(G_{1'}, G_2\right) = 1 - \left(\frac{\sum_{som} d^{\pm}\left(MCS\left(G_1, G_2\right)\right)}{\max\left(\sum d^{\pm}\left(G_1\right), \left(\sum d^{\pm}\left(G_2\right)\right)\right)}\right)$$

where $\Sigma d^{\pm}$ represents the sum of in-degree and out-degree of the directed graph and $\sum_{som} d^{\pm}$ represents the sum of the minimum of the degrees generated by each common node of the graphs which are included in the MCS. In case when there are two or more MCS then we should consider max ($\sum_{som} d^{\pm}$). max(x,y) is the usual maximum of two numbers x and y.

In the above newly constructed measure of calculating graph distance, we are using the in and out degrees of all the nodes concerned, instead of simply considering the number of nodes of the respective graphs. Since we are representing a document (web) with the help of a graph therefore most of the nodes (representing words of the concerned document) of the graph will have a relationship in the form of edges with some other nodes (words) of the graph. Since we are following the composite model, therefore, the nodes representing a word in a sentence will have a relationship with the successor or predecessor words and also with the words having a distance n from it, where n is the user-provided value. As we are creating a unique node for each word occurring in the document, therefore the word having a higher frequency then others will have higher in and out degrees compared to that of others. By considering in and out degrees to calculate the graph distance, we are trying to include all such information in the calculation process in order to get more accurate and reliable results.

5 The modified fuzzy C -means algorithm to fit with graphs

The main problem on the way of applying fuzzy c-means for graphs is the computation of cluster representative. In the case of the typical fuzzy c-means algorithm, the initial partition matrix U may be generated randomly or can be calculated from the initial cluster centers. For example, let us have a set S of n graphical objects representing web pages, S= (G_1, G_2, … G_n).

The following are the different steps for clustering these n web pages according to our new methodology.

Step 1: Represent these n web pages according to our proposed composite model.

Step 2: Compute the pairwise distances between all possible pairs of graphs using the MCS distance measure as proposed. These distances can be suitably represented with the help of a symmetric hollow matrix.

Step 3: Perform multidimensional scaling on the n

Step 4: Distinguish some points as cluster representatives randomly in accordance with the specified number of clusters.

Step 5: Calculate the initial partition matrix on the basis of the initial cluster representative.

Step 6: Based on the initial partition matrix recompute the cluster representative say g_i (the graph median).

The cluster centers (i.e. g_i's) can be computed with a weighted averaging that takes into account the membership values of each data item. We cannot use the graph median directly without considering the respective weights assigned to each graph in each iteration. It is also not feasible to multiply a graph with a scalar. Therefore the following approach for calculating the graph medians by considering the respective weights may be used.

For each cluster j, use deterministic sampling to compute the number of copies of each graph g_i to use, say $x_j(i)$ which is defined as:

$$x_j(i) = \frac{a_{ij}}{\sum_{\nabla} a_{ij}} \times n$$

where n is the total number of items in the data set and a_{ij} is the membership value of the respective object 'i' (i.e. graph 'i' copies of graph 'i' and compute the median graph of this set to be the representative of cluster j.

Now, the median of a set of graphs S is a graph $g \in S$ ($S = \{G_1, G_2, \ldots, G_n\}$) such that g has the lowest average distance to all elements in S [(1)]

$$g_i^{(l)} = \arg\min_{\nabla s \in S}\left(\frac{1}{|s|}\sum_{y=1}^{|s|}\text{dist}\left(s, G_y\right)\right)$$

where l is the no of iteration.

Step 7: Compute distances based on the new cluster representative

$$D_{ikA}^2 = \left|\text{dist}_{MCS}\left(G_k, g_i^l\right)\right|^T A \left|dist_{MCS}\left(G_k, g_i^l\right)\right|, 1 \le i \le c, 1 \le k \le n$$

where c is the no of clusters and A is the norm inducing matrix.

Step 8: Update the partition matrix by updating the membership values μ
for $1 \leq k \leq n$

$$\text{if } D_{ikA} > 0 \text{ for all } i = 1, 2, \ldots, c$$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA}/D_{jkA}\right)^{2/(m-1)}}$$

where m is the fuzziness parameter
otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ikA} = 0 \text{ and } \mu_{ik}^{(l)} \in [0, 1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$$

Step 9: Repeat the process until termination tolerance is reached.

6 Conclusion

The composite graph model for web document representation takes into account the additional web-related content information which is not done in traditional information retrieval models. It can hold almost all the necessary information such as the order, proximity of word occurrence, markup information and location of a word within a document that is necessary for the purpose of clustering. This model along with the enhanced distance measure is giving increased effectiveness (16.9%) in the graph distance measure compared to that of the prevailing one, even though the maximum common subgraph (MCS) is the same in both the graphs [1,2].We have introduced a new approach for graph distance measure which is essential for clustering data sets. Using the composite model of graph representation we can perform the graph similarity task in $O(n^2)$ time.

The basic idea behind the extension for FCM algorithm is the calculation of cluster center in case of graphical objects. We have modified the step 1 and 2 of the original fuzzy c-means algorithm which will arm it to handle graphical objects. These changes are made without changing the fundamental concepts of the FCM algorithm. This method will enhance the efficiency and effectiveness of the FCM algorithm, as the graphical objects will boost the clustering method with information as needed. This enhancement will allow web documents to be represented by graphs that have the potential to retain information that is usually discarded when using a vector representation. This extended graph-theoretical version of FCM algorithm does not limit the form of graphs which allow the change of graph model or even application domains without reformulating the algorithm.

References

1) Phukon KK. Maximum Common Subgraph and Median Graph Computation from Graph Representations of Web Documents Using Backtracking Search. *International Journal of Advanced Science and Technology*. 2013;51:67–80.
2) Schenker A, Last M, Bunke H, Kandel A. Clustering of web documents using a graph model. In: and others, editor. Series in Machine Perception and Artificial Intelligence. 2003;p. 3–18.
3) Shaban K. Semantic Graph Model for Text Representation and Matching in Document Mining. Canada. 2006. Available from: http://hdl.handle.net/10012/2860.
4) Suen CY. n-Gram Statistics for Natural Language Understanding and Text Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1979;PAMI-1(2):164–172. Available from: https://dx.doi.org/10.1109/tpami.1979.4766902.
5) Martinez AR, Wegman EJ. Text Stream Transformation for Semantic-Based Clustering. vol. 34. 2002.
6) Hasan M, Matsumoto Y. Document Clustering: Before and After the Singular Value Decomposition. *Information Processing Society of Japan* . 1999;p. 47–55.
7) Hotho A, Staab S, Stumme G. Wordnet improves Text Document Clustering. *Proceeding of the Semantic Web Workshop at SIGIR, 26th Annual International ACM SIGIR Conference*. 2003. Available from: https://doi.org/10.1109/ICDM.2003.1250972.
8) Hammouda K, Kamel M. Phrase-based document similarity based on an index graph model. *Proceedings of the 2002 IEEE Int'l Conf on Data Mining (ICDM'02)*. 2002. Available from: https://doi.org/10.1109/ICDM.2002.1183904.
9) Schenker A, Bunke H, Last M, Kandel A. Graph Theoretic Techniques for Web Content Mining. vol. 62 of Series in Machine Perception and Artificial Intelligence. World Scientific Publishing Co. Ltd. 2005.
10) Hartigan JA, Wong MA. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*. 1979;28(1):100–100. Available from: https://dx.doi.org/10.2307/2346830.
11) Jain AK, Dubes RC, '. Algorithms for Clustering Data. NJ. Prentice Hall. 1988.
12) Dutta A, Riba P, Lladós J, Fornés A. Hierarchical stochastic graphlet embedding for graph-based pattern recognition. *Neural Computing and Applications*. 2020;32(15):11579–11596. Available from: https://dx.doi.org/10.1007/s00521-019-04642-7.
13) Duda RO, Hart PE. Pattern Classification and Scene Analysis. New York. Wiley. 1973.
14) Gerritse EJ, Hasibi F, Vries AP. Graph-Embedding Empowered Entity Retrieval. In: Advances in Information Retrieval", 42nd European Conference on IR Research. 2020;p. 97–110.
15) Buluz B, Yilmaz B. Graph mining approach for modeling academic success. In: 25th Signal Processing and Communications Applications Conference (SIU). 2017;p. 1–4. Available from: https://doi.org/10.1109/SIU.2017.7960621.
16) Chintalapudi RS, Prasad MHK. Mining Overlapping Communities in Real-world Networks Based on Extended Modularity Gain. *International Journal of Engineering (IJE), TRANSACTIONS A: Basics*. 2017;30(4):486–492.
17) Berkhin P. Survey of Clustering Data Mining Techniques. 2002.
18) Hubert LJ. Some applications of graph theory to Clustering. *Psychometrika*. 1974;38:435–475.
19) Mehrotra D, Nagpal D, Srivastava R, Nagpal R. Analyse Power Consumption by Mobile Applications Using Fuzzy Clustering Approach". *International Journal of Engineering (IJE), IJE TRANSACTIONS C: Aspects*. 2018;31(12):2037–2043.
20) Mahdizadehand M, Eftekhari M. A Novel Cost Sensitive Imbalanced Classification Method based on New Hybrid Fuzzy Cost Assigning Approaches, Fuzzy Clustering and Evolutionary Algorithms". *International Journal of Engineering (IJE)*. 2015;28(8):1160–1168.
21) Martinez MA, Adiego J, Fuente P. Natural Language Compression on Edge-Guided Text Preprocessing". In: and others, editor. Proc. of 14th International Symposium on String Processing and Information Retrieval (SPIRE'07). 2007;p. 14–25. Available from: https://doi.org/10.1016/j.ins.2011.07.039.
22) Phukon KK. A Composite Graph Model for Web Document and the MCS Technique". *International Journal of Multimedia and Ubiquitous Engineering*. 2012;7(1):45–51.