COMMUNITY DETECTION IN CITATION NETWORKS USING ATTRIBUTE SIMILARITIES

Arijeet Chakrabarty[1], Jeshuren Chelladurai[2], S.K. Venkateswaran[3]
[1,2,3]Department of Computer Science and Engineering
Thiagarajar College of Engineering, Madurai
Email:arijeet995@gmail.com[1],jeshurench@gmail.com[2],venkateswaranskv@gmail.com[3]

Abstract

The number of research papers published in conferences and journals continues to increase, due to the open availability of information resources and advancement of research facilities. In such a deluge of research papers, it is important to find interconnections among them and better use those insights to help the research work progress. This problem maps to the problem of community detection in a connected network, where each node in the network is a research paper. In this paper, we apply weighting schemes using the node attributes of the network, to detect communities in the network using a Hierarchical clustering algorithm, called the Divide and Link algorithm. We then experiment the method proposed using a manually crawled citation network and evaluate the results using RandIndex.

Keywords: clustering, hierarchical clustering, community detection

I. INTRODUCTION

Community detection is the problem of finding groups in networks. The nodes in these groups are densely connected to each other. This type of groups is found in real networks, such as social networks which contain communities' groups based on interests, location, etc., citation networks which form communities based on research topics, co-authorship, etc., [5] In our paper, we are focusing on the community detection problem in citation networks.

Several methods like Louvian [1], CESNA [3] exist for community detection in a given network. Hierarchical clustering is one such method for community detection.

Hierarchical clustering is of two types:

- Agglomerative, which is a bottom up approach where smaller clusters are merged together into larger clusters.
- Divisive, which is top down approach where large clusters are split into smaller clusters.

In this paper we have used a divisive approach for community detection using the Divide and Link algorithm introduced in [4].
The paper is organized as follows,

In section 2, we do a literature survey of already existing hierarchical clustering methods that are used for the community detection problem.

In section 3, we provide brief definitions to the concepts used in the paper.

In section 4, we describe the model and its application for the problem statement defined.

Section 5 consists of the experimental results and in section 6 we conclude our work and outline the future research work.

II. LITERATURE REVIEW

A. *A Divide-and-Link algorithm for hierarchical clustering in networks*

Daniel Gómez et al, in their paper, [4] have introduced a new hierarchical clustering algorithm and have applied this algorithm in the

problem of community detection in social networks. The algorithm consists of two stages, a divisive stage to divide the original graph and a linking stage to link nodes which are similar to each other. A betweeness measure and similarity measure are considered for the two stages respectively. The analysis of the algorithm was done by applying it to some specific social networks, such as the Karate Club network, the Les Miserables network, the Centrality Authors network, and the Dolphins network.

B. *Graph Clustering based on Structural and Attribute Similarities*

Yang Zhou et al, in their paper, [2] have used structural and attribute similarities in the problem of community detection. They have proposed a general weighting framework along with a hierarchical clustering algorithm to cluster the components into respective clusters, which constitute communities in the network. We have taken the ideas from the same and used different measures for the edge weight calculation.

III. BASIC CONCEPTS

A. *Citation Network*

A citation network is a social network that contains information about papers and their citation relationships. The nodes in a citation network represent documents and the edges represent the relationship between them. If a document d_1 cites another document d_2, then edge exists between them and it is directed from d_1 to d_2. Citation networks are directed and acyclic. [5]

B. *Cosine Similarity*

Cosine similarity is a measure that represents the similarity between two vectors or two documents in the vector space. This measure is obtained by calculating the cosine of the angle between the two vectors. Given two vectors of attributes, *A* and *B*, the cosine similarity$\cos\theta$ is represented by the following formula

$$\cos\theta = \frac{A.B}{|A||B|}$$

The resulting value of Cosine similarity represents the relation between two documents or vectors based on the angle between them. For two very similar documents, the angle between them will be close to 0 degrees, and thus the cosine similarity value will be near to 1. For two dissimilar documents, the angle between them will be close to 90 degrees, and thus the cosine similarity value will be near to 0. [6]

C. *Jaccard Index*

The Jaccard Index or the Jaccard similarity coefficient is a metric that is used for comparing the similarity and dissimilarity of item sets. The formula for Jaccard Index between two item sets is given as the size of the intersection of the item sets divided by the size of the union of the item sets. Given two item sets *A* and *B*, the Jaccard Index is as follows

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

In our scenario, for two papers *A* and *B*, we create two item sets which contain the list of papers referenced by those two papers respectively. Thus, the numerator of Jaccard Index represents the number of papers that are referenced together by both paper *A* and paper *B* and the denominator represents the total number of papers that are referenced individually by paper *A* and paper *B*. [7]

D. *Term Document Matrix*

A term document matrix is a two dimensional matrix that represents the occurrence of terms in a corpus of documents. The rows of the matrix represent the documents of the corpus and the columns represent the individual terms present in each document. Thus, each entry *(i,j)* of the term document matrix represents the frequency of term *j* in document *i*. [8]

E. *RandIndex*

Rand index is an evaluation measure of the quality of two clusterings. For a given set of *n* documents, Rand index provides the percentage of documents that have been correctly assigned to their suitable clusters, i.e. two similar documents are grouped together in the same cluster and two dissimilar documents have assigned to two separate clusters.

Given a set *n* documents *S* and two partitions of *S*, *A* and *B*, such that

- *a* – number of pair of elements of *S* which are similar and are assigned to the same cluster.
- *b* – number of pair of elements of *S* which are dissimilar and are assigned to two different clusters.

- c – number of pair of elements of S which are dissimilar and are assigned to the same cluster.
- d – number of pair of elements of S which are similar and are assigned to two different cluster. [9]

The Rand index R is given as.

$$R = \frac{a+b}{a+b+c+d}$$

IV. ALGORITHM DESCRIPTION

The algorithm proposed for finding communities in the citation networks is described in **Algorithm 1 and Algorithm 2.** The input for the algorithm is a network of research papers, where each node is a research paper data. There is an edge between two nodes, if a paper P_i and P_j if the paper P_i is cited in the paper P_j. Since our algorithm has two types of edges,

i) Divisive Edges
ii) Linking Edges

We require two matrices to be provided to the Algorithm. Hence, for the divisive edge matrix, we find the cosine dissimilarity, by using the cosine similarity formula described in section II. If div (i, j) is an entry in the divisive edge weight matrix, then,

$$div\ (i\ ,\ j) = 1 - Cosine\ Similarity\ (P_i\ ,\ P_j)$$

The linking weight matrix is formed using the jaccard index. If link(i,j) is the linking edge weight between two papers P_i and P_j then it is given by,

$$link\ (i\ ,\ j) = \frac{\#\ of\ common\ papers\ cited\ by\ Pi\ \ and\ Pj}{\#\ of\ total\ \ papers\ cited\ by\ Pi\ \ and\ Pj}$$

Input: D = $\{d_1,d_2,d_3,...d_n\}$ the set of papers to be clustered along with the Edge Set, E = $\{e_1,e_2,e_3,...e_m\}$

Output: Two matrices of size d_n x d_n, representing the divisive and linking weights

Step 1: Form Term – Document Matrix for documents in D.

Step 2: For each pair of document, divisive weight is calculated by,

div (i , j) = 1 – Cosine Similarity (P_i , P_j)

Step 3: For each pair of document, linking weight is calculated by,

$$\textbf{link (i , j)} = \frac{\textbf{\# of common papers cited by Pi and Pj}}{\textbf{\# of total papers cited by Pi and Pj}}$$

Algorithm 1. Preprocessing

Now, as we have the input in the required format, the algorithm proceeds as follows, in each step, the divisive edge is formed by finding edge with maximum divisive weight.

$$d_e = max(div(di,dj))$$

Linking edge set is formed by adding all linking edge except the edge which forms maximum divisive weight.

$$l_e = \{e_1,e_2,e_3,e_4..e_k\}$$

where edge is of form $e_i = (d_i,d_j)$,ei != de
Linking edges are sorted based non-increasing order of edge weights.

$$l_e = \{e_1,e_2,e_3,e_4..e_k\}$$

where e1>=e2>=e3…>=ek
Edge set is formed by the union of divisive and linking edges.

$$E = d_e\ U\ l_e$$

Initialize result edges as null. For each edge (ei = (dj,dk)) in edge set(E) if there is no path between di and dj based on edges in result set then add edge e_i to result set and if there is a path connecting d_i and d_j ignore the edge as adding edge will result in formation of cycle.

Input: Two matrices of size d_n x d_n, representing the divisive and linking weights

Output: Clusters C = $\{c_1,c_2,c_3,...c_n\}$ representing the communities in the network

- Until required number of clusters are formed or each cluster if of size one do the following,
  - Form divisive edge by finding edge with maximum divisive weight

    $$d_e = \max(div(d_i,d_j))$$

  - Form linking edges by adding all link edges except the edge which form max divisive edge,

    $$l_e = \{e_1,e_2,e_3,e_4..e_k\}$$

    where edge is of form $e_i = (d_i,d_j)$
  - Sort linking edges in non-increasing order, edge set of iteration ,

    $$E = d_e \cup l_e$$

  - Initialize result edges as null set and disjoint set **ds** of size n.
  - For each edge ($e_i = (d_j,d_k)$) in edge set E,
    - If d_j and d_k are present in different sets,
      - Merge them and add edge(e_i) to result set
  - remove divisive edge from result set
  - Do Breadth first search or Depth first search on result set to find the clusters,

    $$C = \{c_1 , c_2 . \ ,.c_n\}$$

  - For each edge in graph if an edge connects documents in different clusters remove edge
  - Compute Rand Index for the clusters formed

Algorithm 2. Divide and Link Algorithm

Above strategy guarantees there is at most one path between any two documents in the graph formed using result edges. The graph formed using result edges is spanning tree. Spanning tree obtained is neither minimum nor maximum. Remove divisive edge from result set. Removal of divisive edge from result set results in increase in number of connected components. Perform Depth first search or Breadth first search on result set to find the number of clusters and documents in each cluster.

$$C = \{c1,c2,..cn\}$$

For any i , j $c_i \cap c_j = NULL$ (i!=j)

$$c1 \cup c2 ... \cup cn = D$$

For any edge in graph if an edge connects documents from different clusters remove the edge. Rand Index is computed for the clusters formed.

V. EXPERIMENTS & RESULTS

A.Dataset Collection

The dataset for our experiments were crawled from the IEEExplore website, using a web crawler. We chose to crawl papers from 7 categories under computer science. For each category, we choose an initial seed and started crawling papers cited by it in a breadth firth search manner. For each paper, we considered only the parts which contained more information regarding the paper, such as the Title and the Abstract. The papers cited by that paper were crawled next and the process was continued. The dataset accumulated in such manner consisted of 945 documents from 7 categories.

B.Pre – Processing

The dataset obtained in the above step, was preprocessed by removing the stop words from the document. Then the documents were represented in the form of a Term Document Matrix. The Term Document Matrix was used for constructing the two weighted edge matrices required for the algorithm.

C. Experiments

Our proposed algorithm was run on the obtained dataset till 7 clusters were formed. At each stage, the RandIndex of the clustering was calculated. The values obtained during each of the stage are represented as a graph in **Fig.1**. The experiments were performed on a Core i5 processor with 8GB of RAM.

D. Results

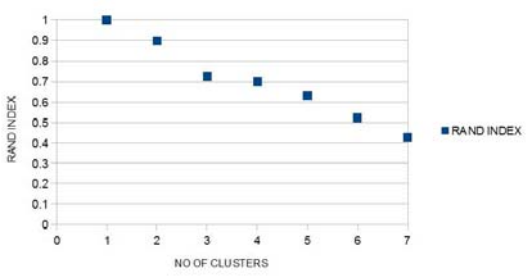

Fig.1. Rand Index values at each stage of the clustering process

Number of class labels increase by one as number of clusters increase by one for each

iteration. For every iteration expected class label is assigned to each document. At the end of each iteration class labels are assigned to each cluster. Class label of a cluster is the class label with maximum number of documents in that cluster. For each iteration expected class label and class label obtained by clustering is computed. Rand index is computed on expected and observed class labels. Random Index of first iteration is one as all documents belong to same cluster. Random Index decreases with increase in number clusters as probability of mislabeling documents is raises with increase in number of clusters.

VI. CONCLUSION

In this paper, we have demonstrated a weighted method to be used in with a hierarchical clustering algorithm known as the Divide and Link Algorithm. Also, we have evaluated our method by measuring the RandIndex at each stage. In the future, we would like to extend the work by proposing a general framework for all kinds of graph networks.

ACKNOWLEDGMENT

We owe our special thanks and gratitude to Mr. S. Karthick, Assistant Professor, Department of Computer Science and Engineering, Thiagarajar College of Engineering, for his guidance and support throughout our project. We would like to thank the members of the “Artificial Intelligence” Special Interest Group of our college for providing information and encouragement throughout our project.

REFERENCES

[1]P. De Meo, E. Ferrara, G. Fiumara and A. Provetti, "Generalized Louvain method for community detection in large networks", *2011 11th International Conference on Intelligent Systems Design and Applications*, 2011.

[2]Y. Zhou, H. Cheng and J. Yu, "Graph clustering based on structural/attribute similarities", *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 718-729, 2009.

[3]H. Elhadi and G. Agam, "Structure and attributes community detection", *Proceedings of the 7th Workshop on Social Network Mining and Analysis - SNAKDD '13*, 2013.

[4]D. Gómez, E. Zarrazola, J. Yáñez and J. Montero, "A Divide-and-Link algorithm for hierarchical clustering in networks", *Information Sciences*, vol. 316, pp. 308-328, 2015.

[5] "Citation Network", *Wikipedia*. [Online]. Available:https://en.wikipedia.org/wiki/Citation_network

[6] "Cosine Similarity", *Wikipedia*. [Online]. Available:https://en.wikipedia.org/wiki/Cosine_similarity

[7] "Jaccard Index", *Wikipedia*. [Online]. Available:https://en.wikipedia.org/wiki/Jaccard_index

[8] " Document-term matrix ", *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Document-term_matrix

[9] " Rand index ", *Wikipedia*. [Online]. Available:https://en.wikipedia.org/wiki/Rand_index

.