A Graph Clustering Algorithm Based on Adaptive Neighbours Connectivity

Israa Hadi, Firas Sabar Miften
Faculty of Information Technology, University of Babylon, Iraq
israa_hadi1968@yahoo.com

***Abstract*—This paper deals with graph clustering algorithm which partitions a set of vertices in graphs into smaller sets (clusters). Such vertices of the same set are related to each other rather than to those in the other sets. This means that most graph clustering algorithms are based on the topological shape or feature similarity. Nevertheless, these algorithms suffered from scalability because of the height computation requirements for similarity estimation. This paper represents a stimulus for the current study to introduce an algorithm that automatically finds the number of clusters based on shared neighbours among vertices. The study is based on the hypothesis that the proposed algorithm is able to efficiently find the graph clustering partitions for the whole graphs.**

***Index Terms*—Automatic Clustering; Connectivity; Graph Clustering; Jaccard Similarity.**

I. INTRODUCTION

Clustering refers to the division of data into various sets of mini-objects. In this regard, each set, known cluster, encompasses objects that are similar in comparison to each other but different those of other sets [1], [2].

Broadly, the issue in question has gotten critical consideration amid the most recent years in view of its significance in different fields of science, for example, the discovery of community in social networks, sensor networks, telecommunication and the Web. Its importance is reflected by its vital role in pattern recognition. Phrased differently, it allows distinguishing groups of profoundly related vertices in a graph, also called clusters [3].

As far as the nature of clustering algorithms is concerned, these algorithms represent a case of multiplicity. Indeed, this fact does not prevent a rarity of such algorithms can consequently find groups without the details of the sum of groups. For instance, automatic graph clustering algorithms, which are ready to characterize independently, in isolation, the totality of groups, are equipped efficiently to analyze data of the group. Regarding the analyzed data, these groups have the property of permitting a more productive use of clustering algorithms to be applied to a dataset regardless an earlier learning of the information adaptation. Accordingly, the examination of novel clustering algorithms enables to manage graph clustering issues and identify automatically the collection of groups as a critical research matter.

In conventional clustering of sets of data, the way of distance measure can basically be based on the identification of attribute, e.g., Euclidian distance comparing the two attributes. As opposed to the current approach, graph clustering categorizes the vertex closeness depending upon connectivity, neighbourhood similarity, attribute or contextual similarity. Many current algorithms of graph clustering regard the topological construction of a graph to fulfil the durable interior construction. This approach incorporates clustering based upon max flow min-cut problem [4, 5], normalized cut [6], structural density and modularity [7, 8]. Such methods divide the classes of nodes into various groups as well as gauge the cost of edge cut, i.e. sum of edges relating vertices in various groups or edge cost relying on the connected weights. Such methodologies segment the order of vertices in different collections and gauge the cut cost edge, i.e. edges number interfacing vertices in various gatherings or edge cost in view of the related weights.

About the approach developed to treat clustering of graph node [9], it introduces the measure of collaborative similarity (CSM) aiming at clustering of intra-graph. Instead of the different paths, CSM depends on the strategy of the shortest path to clarify the relevance of structure as well as semantics between vertices. Thus, the method surveyed in [10] suggests the name of congruent approximate graph clustering (CAC). It may keep on the notion of non-negativity severely and may arrive the orthogonality definitely through congruency approximation. On the other hand, the technique given by [11] concerns the arbitrary-pair attributes of vertices. Consequently, the values of the similar attribute are gathered under either specific partition or cluster. As such, it stands as a sufficient way of graph summarization depending upon OLAP processes. As for the first process, known as SNAP, it yields a summary graph via collecting nodes by means of the node attributes and connections of the user-selected node. Concerning the second process, in k-SNAP one, it further permits users to override summary resolution. In order to arrive better analogous of graph summarization to OLAP processes, vertices partition has taken place relying on their feature and then initiating summaries whereas ignoring the connectivity.

II. EXPERIMENTAL

A. Terminology and Definitions

To simplify the discussion, it seems necessary to put forth this symbol: a weighted, an undirected, a graph G consists of an ordered pair G = (V, E), where V stands for a class of vertices and E represents a class of edges. In addition, the matrix of similarity (matrix of affinity) of G graph on n vertices can be expressed by $W = (w_{ij})$ i, j=1, …, $n \in R_{n \times n}$. The positive entry w_{ij} in W refers to vertex i while vertex j seems related together a weighted edge. If $w_{ij} = 0$, it indicates the i as well as j vertices that cannot be related by the edge. Moreover, the **Matrix of Similarity** W stands for symmetric for undirected graphs.

$$Similarity(X.Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (1)$$

Jaccard Similarity refers to coefficient measure [12], schematized in Equation 1. It is, generally speaking, numerously used and acceptable in the area of data mining [13]. Because of its simplicity, it is applied in several areas to detect the relevance between the objects. In this work, it will be used to redefine edge weights between vertices via similarity of Jaccard.

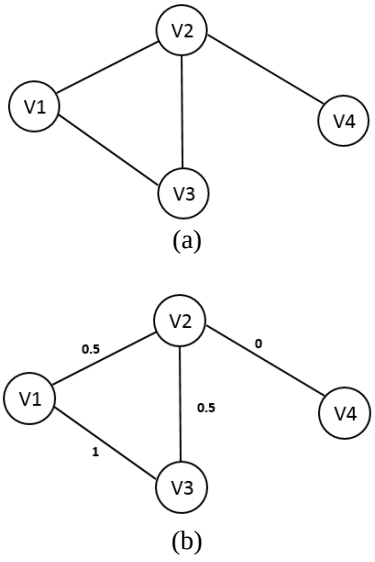

Figure 1: Similarity among vertices. (a) Unweighted graph. (b) Construct weighted graph by Jaccard similarity

The similarity between v1 and v1, by utilizing the neighbourhood of these two vertices. Worded differently, it stands for shared neighbours ratio in relation to all types of the neighbours of the two vertices.

$$SIM(v1.v2) = \frac{|v1 \cap v2|}{|v1 \cup v2|} = \frac{1}{2} = 0.5$$

B. Density

Of vertex refers to a number of neighbourhoods of the vertex. Here, Density is a vector of the number of adaptive neighbours for each element while Density (a) is the number of the adaptive neighbours of the individual a.

C. Methodology

The suggested method can work on unweighted, undirected or weighted. Besides, there is no need for cluster numbers to be discovered. Algorithm 1 schematizes the outline of the suggested method.

As far as the input of algorithm is concerned, it refers to the adjacency matrix of the undirected graph. Step 1 and 2 determine the adaptive neighbours for each Node which, in turn, depend on the Jaccard similarity. It is this type of similarity that measures connectivity or the power of relationship among the pair of nodes. In Step 4 Compute the density of each Node as follows:

Where Density is a vector (V) of the number of neighbors for each Node; Density(x) is the number of the neighbors of the individual x; Step 5 descends sorting the items of the Density (V) vector; and Steps 6 to 12: The first node in vector V must construct (create) the first cluster since it has the largest number of adaptive neighbors in Density.

Algorithm 1
Graph Clustering

Input: Adjacency matrix $W(N \times N)$ $for\ Graph\ G(V.E)$.
Output: Clustering C.
Begin
1: for each vertex pair $v_i\ .v_j\ \in V : i.j = 1\ to\ N\ and\ i \neq j$ **do**
2: $S[i.j] = SIM(v_i.v_j)$ Compute the similarity by Equation (1).
3: for each vertex v_i $in\ V\ where\ i = 1\ to\ N$ **do**
4: $Density(i) = the\ no\ of\ neighbors$ of vertex v_i
5: Sort the items of the vector Density (V) in descending order**.**
6: $C(i) = 0\ where\ i = 1 \ldots N : c = 0$
7: for each vertex v_i in Density (V) $where\ i = 1 \ldots N$ **do**
8: **if** $C(i)\ is\ zero$ ***then***
9: $c = c + 1\ and\ C(i) = c$
10: **for** each vertex v_j $where\ v_j \in (neighbors$ of vertex $v_i)$ **do**
11: ***if*** $S[i.j] = \arg\max(S[j.k])\ where\ k \in (neighbors$ of vertex $v_j)$ **then**
12: $C(j) = C(i)$
13: Return C
14: End

All the adaptive neighbours of the first node in V must be located in this cluster with the condition that it has the highest similarity with the first node. Therefore, the second node in V whose position correspond the second element in Density must be taken as candidate node. If (this candidate has been assigned to any existed cluster), then, all its adaptive neighbours must be located in that cluster with the condition it has the highest similarity with candidate node else, this candidate will construct another new cluster. Besides, all its adaptive neighbours must be located in this new cluster. Moreover, it is conditioned by having the highest similarity with candidate node. As such, the process will continue until the last element in V has been clustered in its corresponding clusters.

D. Metrics of Cluster Quality

As far as cluster quantity is concerned, it is normally categorized as a class of heavily related vertex which appears in connection with various sets in a certain graph. As such, lack of general, as well as exact scientific cluster meaning, is handed in the process of writing [14]. On the other hand, assortments of different, measurements which attempt to test the clustering quality, take place via catching the density of intra-cluster as well as sparsity of inter-cluster. Regarding $G = (V, E)$ is an undirected graph in association with an adjacency matrix, three standards of measuring cluster quality are adopted in the current paper: modularity, conductance and coverage. All of them are standardized in relation to the ultimate goal which scores range starting by 0 up to 1 where 1 represents the score that can be described as the ideal.

E. Modularity

Concerning modularity, it compares the existence of every edge of intra-cluster of a certain graph with the edge probability that might be found in a haphazard graph [15, 16]. As a limit of resolution [17], its algorithms of popular clustering functions objectively [18, 19]. Modularity is presented by Equation (2).

$$\sum_k (e_{kk} - a_k^2) \quad (2)$$

where e_{kk}, stands for the intra-cluster probability of edges through cluster S_k, whereas a_k, refers to the probability of one of two edges: an intra-cluster within cluster S_k an inter-cluster

incident in cluster S_k, as in Equation (3).

$$e_{kk} = \frac{|\{(i.j): i \in s_k, j \in s_k, (i,j) \in E\}|}{|E|}$$
$$a_k = |\{(i,j): i \in S_k, (i,j) \in E\}|/|E| \quad (3)$$

where $S_k \in V$.

F. Conductance

It refers to the cluster conductance that can be identified via inter-cluster edges numbers. It, in turn, is divided by the number of edges and an end point within the cluster. Moreover, another way of division is by the edges number which has not an end point within the cluster that appears lesser. The conductance a cluster is introduced in the form Equation (4).

$$\emptyset(S_k) = \frac{\sum_{i \in s_k, j \notin s_k} A_{i,j}}{\min\{A(S_k), A(\overline{S_k})\}} \quad (4)$$

Where $S_k \in V$ *and* $A(S_k) = \sum_{i \in S_k} \sum_{j \in V} A_{ij} - \sum_{i \in S_k} \sum_{j \in S_k} A_{ij}$ reflects edge numbers in the endpoint within S_k. The graph conductance G is defined as the conductance average for every cluster in relation to the graph, schematized from one. It involves the range extends from (0 to 1) whereas the subtract has one the best score. Therefore, the graph conductance is presented in Equation (5),

$$\emptyset(G) = 1 - \frac{1}{k} \sum_k \emptyset(S_k) \quad (5)$$

G. Coverage

It [20] refers to the comparison of the division of intra-cluster edges of the graph to whole edges of the graph. It is introduced as Equation (6).

$$\frac{\sum_{i,j} A_{ij} \delta(S_i, S_j)}{\sum_{i,j} A_{i,j}} \quad (6)$$

where S_i refers to the cluster of the node i which is allocated whereas δ(a, b) represents 1 if a = b and 0 otherwise. It, coverage, consists of the range of 0 to 1, since 1 stands for that optimal score. On the other hand, it manages the concept of intra-cluster density as well as improves greatly the measure ends in a small clustering wherein all nodes are allotted to the identical cluster.

III. Results and Discussions

The suggested node of the graph clustering of the algorithm has been examined in the data of real-world sets. Frankly, it enriches promising clustering outcomes. Throughout this work, three real-world graphs are used through analyzing a dataset ego-Facebook [21], Arxiv ASTRO-PH (Astro-Physics) collaboration network [22] and Enron email network [23, 24]. The ego-Facebook network has 4,039 nodes whereas 88,234 undirected edges. The ASTRO-PH network has 18,772 nodes and 198,110 undirected edges. The email-Enron network has 36,692 nodes and 18,3831 undirected edges. In this concern, the data sets under scrutiny can be easily obtained by the Stanford Network Analysis Project (snap.stanford.edu/data/) providing reproducibility of the tests.

The clustering result is evaluated by Modularity, Conductance and Coverage Quality Metrics. These results are best shown in Table 1 below. Our proposed method has Quality Metrics between 0.7 and 0.9 for all metrics which is acceptable for graphs clustering.

Table 1
Clustering Quality

Dataset	Nodes	Edges	Modularity	Conductance	Coverage
ego-Facebook	4,039	88,234	0.80328	0.78452	0.81283
ASTRO-PH	18,772	198,110	0.86783	0.99215	0.78047
email-Enron	36,692	18,3831	0.79871	0.98862	0.83918

IV. Conclusion

As far as this work is concerned, it surveys a sufficient strategy of graph clustering to partition the vertices depending upon connectivity among vertices. Nevertheless, the more frequent strategy of connectivity is adopted to evaluate the relevance among vertices. In this regard, every cluster quality is concurrently estimated by coverage quality measures, modularity and conductance. However, the experiments on datasets of the real graph show competitive findings in relation to the quality of the cluster. As such, the current notion appears suitable to the distributed graph processing in relation to the partition of the whole graph of K sub-graphs. Hence, the cluster numbers can be specified automatically.

References

[1] O. Maimon and L. Rokach, Data mining and knowledge discovery handbook vol. 2: Springer, 2005.

[2] J. Kleinberg and E. Tardos, "Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields," Journal of the ACM (JACM), vol. 49, pp. 616-639, 2002.

[3] S. E. Schaeffer, "Graph clustering," Computer science review, vol. 1, pp. 27-64, 2007.

[4] R. Andersen and K. J. Lang, "Communities from seed sets," in Proceedings of the 15th international conference on World Wide Web, 2006, pp. 223-232.

[5] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: a structural clustering algorithm for networks," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 824-833.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, pp. 888-905, 2000.

[7] H. Ino, M. Kudo, and A. Nakamura, "Partitioning of web graphs by community topology," in Proceedings of the 14th international conference on World Wide Web, 2005, pp. 661-669.

[8] M. E. Newman, "Detecting community structure in networks," The European Physical Journal B-Condensed Matter and Complex Systems, vol. 38, pp. 321-330, 2004.

[9] W. Nawaz, K.-U. Khan, Y.-K. Lee, and S. Lee, "Intra graph clustering using collaborative similarity measure," Distributed and Parallel Databases, vol. 33, pp. 583-603, 2015.

[10] [10] W. Ren, G. Li, and D. Tu, "Graph clustering by congruency approximation," IET Computer Vision, vol. 9, pp. 841-849, 2015.

[11] S. M. Van Dongen, "Graph clustering by flow simulation," 2001.

[12] P. Jaccard, Etude comparative de la distribution florale dans une portion des Alpes et du Jura: Impr. Corbaz, 1901.

[13] B. Everitt, S. Landau, and M. Leese, "Cluster analysis. 4th," Arnold, London, 2001.

[14] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," Physical review E, vol. 80, p. 056117, 2009.

[15] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical review E, vol. 69, p. 026113, 2004.

[16] M. E. Newman, "Fast algorithm for detecting community structure in networks," Physical review E, vol. 69, p. 066133, 2004.

[17] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," Proceedings of the National Academy of Sciences, vol. 104, pp. 36-41, 2007.
[18] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," Journal of statistical mechanics: theory and experiment, vol. 2008, p. P10008, 2008.
[19] L. Waltman and N. J. van Eck, "A smart local moving algorithm for large-scale modularity-based community detection," The European Physical Journal B, vol. 86, pp. 1-14, 2013.
[20] S. G. Kobourov, S. Pupyrev, and P. Simonetto, "Visualizing graphs as maps with contiguous regions," EuroVis14, Accepted to appear, vol. 4, 2014.
[21] J. J. McAuley and J. Leskovec, "Learning to Discover Social Circles in Ego Networks," in NIPS, 2012, pp. 548-56.
[22] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 1, p. 2, 2007.
[23] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," Internet Mathematics, vol. 6, pp. 29-123, 2009.
[24] B. Klimt and Y. Yang, "Introducing the Enron Corpus," in CEAS, 2004.