Node-Centric Community Detection in Evolving Networks

Oluwafolake Ayano
Computer and Information Technology Dept
Purdue University
West Lafayette, IN
+17654094692
oayano@purdue.edu

John Springer, PhD
Computer and Information Technology Dept
Purdue University
West Lafayette, IN
jaspring@purdue.edu

ABSTRACT

Advance in technology has led to the availability of data from different platforms such as the web and social media platforms. Much of this data can be represented in the form of a network consisting of a set of nodes connected by edges. The nodes represent the items in the networks while the edges represent the interactions between the nodes. Community detection methods have been used extensively in analyzing these networks. However, community detection in evolving networks has been a significant challenge because of the frequent changes to the networks and the need for real time analysis. Using Static community detection methods for analyzing dynamic network will not be appropriate because static methods do not retain a network's history and cannot provide real-time information about communities in a network.

Existing incremental methods treat changes to the network as sequence of edge additions and/or removals; however, in many real-world networks, changes occur when a node is added with all its edges connecting simultaneously.

For an efficient processing of such large networks in a timely manner, there is need for an adaptive analytical method that can process large networks without recomputing the entire network after its evolution and treat all the edges involved with a node equally.

We proposed a node centric community detection method that incrementally updates the community structure in the network using the already known structure of the network to avoid recomputing the entire network from the scratch and consequently achieve a high-quality community structure. The preliminary results from our experiments suggest that our approach is efficient for incremental community detection of node centric evolving networks.

1. INTRODUCTION

A dynamic network is a network that changes frequently with time. There are several real-life networks that are dynamic in nature; some examples include social networks, molecular interactions, communication networks, chemical reactions, and transportation networks. Most networks are not static and instead continually evolve through the addition of new nodes and edges; in many cases, the existing nodes and edges are still retained in the networks resulting in very large networks. A dynamic network captures time dependent information that needs to be unraveled and evaluated. Identifying communities in a network that changes rapidly and frequently in a timely manner is necessary to meet the demand of the various real-world applications. Moreover, the evolutionary succession of the network over time is also useful to predict future communities.

2. PROBLEM AND MOTIVATION

Most Community detection methods treat network as static; however, real-world networks are not static and instead are very large and dynamic in nature. Static methods do not retain a network's history and cannot provide real-time information about communities in a network since they need to first decompose the entire network even in cases involving very few changes to the network. Accordingly, these methods can be very inefficient.

We are in the era of dynamicity and streaming networks will continue to emerge and grow. The existing static community detection methods neglect many of the details of real-world operation, such details are essential to reflect reality and for proper analysis of the network. Dynamic network captures time dependent information that needs to be unravel and evaluated. Identifying communities in network that changes rapidly and frequently in a timely manner would therefore be a necessity in other to meet the demand of the various real-world applications

3. BACKGROUND AND RELATED LITERATURE

There are two approaches in literature for analyzing dynamic network; one approach analyzes and identifies communities for each timestep and thereby processes the entire network every time. Existing static methods [1][3][4] can be used to achieve this. As mentioned earlier, these methods are very inefficient as the entire network is processed for even a very small change to the network; additionally, any one change to the network may not be especially important depending on the type of network involved or the nature of the analysis performed plus the change itself may actually be "rolled back" in networks that are rapidly evolving.

The second approach known as incremental community detection propagates the current community structure from the previous timestep. The method has the advantage of converging quickly because it does not decompose the entire network when there is a change and instead only updates the network impacted by said change. [5] presented a real time detection algorithm for tracking community structure of dynamic networks by first applying a static method to obtain the first community structure and then using an updating strategy to track dynamic communities. Similarly, Nguyen and others [4] presented an adaptive modularity-based method for detecting communities in social network while Shang and others [5] analyzed dynamic communities without prior information or any user defined parameters.

4. APPROACH AND UNIQUENESS

4.1 Approach

This study proposed an online analysis of evolving networks with a focus on sequential addition of nodes as many real-world networks evolve through the addition or deletion of nodes; examples of such networks include citation networks and collaborative networks. In a citation network, where the articles represent the nodes and citation between articles represent the edges, when an article is added to the network, the article is added with all its citations at the same time and there is no reason to assume that a citation is added before another. Existing incremental methods model evolving network as sequence of edge addition or deletion. As mentioned earlier, Shang and others [5] presented an incremental method that handles four types of changes including node addition or removal where the node additions or removals are transformed into a sequence of edge additions/deletions. As this method enforces an assumed order for the edges whereas in actuality edges are often added simultaneously, processing edges that actually appear simultaneously in the network in a sequential manner could lead to unstable result because community detection methods are sensitive to the processing order of the edges. Edges should therefore be processed in the order of creation [7].

Our proposed method requires an initial community structure using a static community detection method. In this case, the Louvain algorithm will be used to identify community structure in the first snapshot known as the base structure, and then address updates as the network evolves without recomputing the entire network. Doing this, makes it possible for the new method to process large networks in a timely manner and the base structure also enhances the quality of the community structure identified.

Louvain algorithm was selected because experimental results shows that it is efficient in identifying high quality community structures [6].

The network change which could be addition or deletion of node can lead to one of the three scenarios:

1. New nodes added without connecting edges
2. New nodes with connecting edges added simultaneously into the network.
3. Nodes are removed from the network.

In the first scenario, when a new node joins a network without any connecting edge, such nodes will be treated in isolation and as such it forms a community on its own.

For the second scenario, nodes joining a network with connecting edges can result in the nodes joining an existing community or splitting the community depending on which operation leads to modularity maximization. As for the third scenario, when a node is deleted from the network, the adjacent edges are removed, as a result, the community where the node was deleted can remain unchanged, split or merge with another one.

In order to evaluate the performance of the new method, it will be tested on large scaled real-world networks and compared with other incremental community detection methods.

Modularity measure was used to evaluate the quality of the community identified.

4.1.1 Modularity

Girvan and Newman [6] introduced the concept of modularity to measure the quality of community structure. Modularity denoted as Q is the fraction of the edges that fall within the given community minus the expected fraction if the edges were distributed randomly.

Modularity can be calculated by:

$$Q = \sum_{s=1}^{k} \left[\frac{l_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right]$$

Where k is the number of communities, m is the total number of the edges in the network, l_s is the edges within the community and d_s is the sum of the degree of nodes in the community. Higher value of modularity signifies a good quality of community structure.

4.2 Uniqueness of Approach

Existing incremental community detection methods are edged-based and treat network evolution as sequence of edge addition. The proposed approach is different from existing solution because it treats multiple edges that are added simultaneous to the network equally.

4.3 Dataset

ArXiv HEP-PH (high energy physics phenomenology) citation graph from the e-print ArXiv was collected from Stanford Large Network Dataset Collection. It contains 34,546 papers (known as nodes) with 421,578 edges. The dataset covers papers in the period from January 1993 to April 2003 (124 months).

4.3.1 Pre-processing

Dynamic network is a sequence of network snapshots $G = \{G_0, G_1, G_2, \ldots, G_T\}$ where each snapshot is the network at a given time, in order to model the evolutionary nature of dynamic network the selected dataset was divided into 6 snapshots using a defined time interval. Figure 1 depict the splitting of the dataset into snapshots.

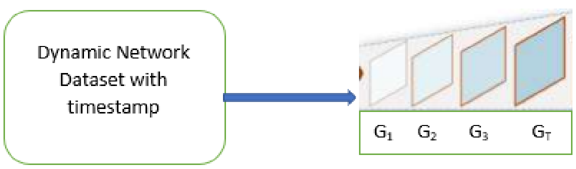

Figure 1: Network Snapshots

4.4 Evaluation

To evaluate the effectiveness of the proposed approach to the problem, the authors selected the papers from 2001 for use in the experiment. To simulate the evolving nature of dynamic networks, the selected dataset was divided into 6 timesteps using two-month durations from January through December, and each timestep was tested using the proposed method and the Louvain method by [1]. The modularity and runtime of each method were then compared.

5. RESULTS AND CONTRIBUTIONS

5.1 Results

Figure 2 shows the runtime of the proposed method and the Louvain algorithm. For the first timestep, the two methods have the same runtime because the Louvain algorithm was used for the initial community structure; however, the proposed method is faster than the Louvain algorithm for the subsequent timesteps because it does not have to recompute the whole network for each iteration. The modularity values computed by the proposed method and Louvain were evaluated. Table 1 contains the results indicating that the proposed method achieves higher modularities than the Louvain method.

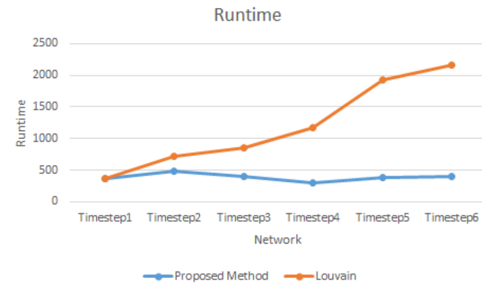

Figure 2: Runtime

Table 1. Comparison of the Modularity

	Louvain	Proposed Method
Timestep1	0.844	0.844
Timestep2	0.8	0.829
Timestep3	0.788	0.831
Timestep4	0.776	0.83
Timestep5	0.771	0.843
Timestep6	0.768	0.84

5.2 Contributions

The key contributions of this study are:

- Developed a node centric incremental community detection that treat edges that appear simultaneously at the same time instead of the conventional sequential approach.

  Build an algorithm for updating communities after its evolution.

  Identified stable communities across the snapshots in the dynamic network for tracking of evolution of community in IP networks
- Develop a model that is efficient for dynamic networks.
-

5.3 Future Work

The proposed method with be evaluated against other incremental algorithms and more datasets to further test its effectiveness.

For graphs with billions of edges only near linear -time community detection methods are practical. There are several fast algorithms that have been developed in recent times but there is still a lot to be done in adapting these methods to take advantage of parallelism.

6. REFERENCES

1] Blondel, V., Guillaume, J., Lambiotte, R. and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*. 2008, 10 (2008), P10008.

[2] Han, J., Li, W., Zhao, L., Su, Z., Zou, Y. and Deng, W. 2017. Community detection in dynamic networks via adaptive label propagation. *PLOS ONE*. 12, 11 (2017), e0188655.

[3] Newman, M. 2004. Fast algorithm for detecting community structure in networks. *Physical Review E*. 69, 6 (2004).

[4] Nguyen N. P., Dinh T. N., Xuan Y. and Thai M. T. 2011. Adaptive algorithms for detecting community structure in dynamic social networks. *In Proceedings of the 30th International Conference on Computer Communications (INFOCOM'11). IEEE, Los Alamitos, CA, 2282--2290*. DOI: http://dx.doi.org/10.1109/infcom.2011.5935045

[5] Shang, J., Liu, L., Xie, F., Chen, Z., Miao, J., Fang, X., Wu, C. 2012. A Real-Time Detecting Algorithm for Tracking Community Structure of Dynamic Networks. *In proceeding of the 6th SNA-KDD workshop (SNA-KDD'12)*. Beijing China

[6] M. E. J. Newman, "Modularity and community structure in networks", *PNAS*, vol. 103, no. 23, pp. 8577-8582, 2006.

[7] Wangsbeng, Z., Gang, P., Zhaohui, W., and Shijian L. 2013. Online Community Detection for Large Complex Networks. In proceeedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13). AAAI Press, 1903-1909.