



# International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

[www.iasir.net](http://www.iasir.net)

## Community Detection in Temporal Networks

Anupama Angadi<sup>1</sup>, Dr. P. Suresh Varma<sup>2</sup>

<sup>1</sup>IT Department, GMR Institute of Technology, Rajam-532127, Andhra Pradesh, INDIA

<sup>2</sup>Department of CSE, University College of Engineering, Adikavi Nannaya University  
Rajahmundry-53329, Andhra Pradesh, INDIA

**Abstract:** Many complex systems in nature, society and technology- from the online social networks to the internet from the nervous system to power grids- can be represented as a graph of vertices interconnected by edges. Small world network, Scale free network, Community detection are fundamental properties of complex networks. Community is a sub graph with densely connected nodes typically reveals topological relations, network structure, organizational and functional characteristics of the underlying network, e.g., friendships on Facebook, Followers of a VIP account on Twitter, and interaction with business professionals on LinkedIn. Online social networks are dynamic in nature, they evolve in size and space rapidly. In this paper we are detecting incremental disjoint communities using dynamic multi label propagation algorithm. It tackles the temporal event changes, i.e., addition, deletion of an edge or vertex in a sub graph for every timestamp. The experimental results on Enron real network dataset shows that the proposed method is quite well in identifying communities.

**Keywords:** Social Network, Community Detection, Network Evolution, Temporal Events, Dynamic Network

### I. Introduction

In recent years online social networks, such as Facebook and Twitter, have been growing rapidly. Such a network can be modelled as graphs, where vertices represent the objects and edges represent interaction among these objects. These interactions represent friendships or the likes, as in Facebook, or followers as on Twitter. Community is a subgraph where a group of nodes densely connected inside and sparsely connected with the rest of the network. Community detection is one of the most significant tool for analyzing complex networks by enabling the study of interconnected groups in large-scale online networks including genetic networks, communication networks, social networks and so on. The cognition of this feature in social networks is important where members of community share common characteristics such as beliefs, demands or preferences, location, occupation, citation networks form communities by research topic, etc. In this regard there is a huge demand for efficient community detection algorithms.

There are many community detection algorithms in use, but are designed to discover communities in static networks and do not scale well on social nets, as they build up rapidly in terms of space and size. Static community algorithms can pose problems in dynamic environment. This paper leverages community detection in dynamic networks. The proposed approach for solving a problem is to track the changes of network in a sub graph, obtain the differences in edges between the graphs for every time stamp. The updated network with changed nodes will be sent to speaker and assign it to the corresponding communities.

### II. Related Work

The LPA [1] is a simple, effective algorithm uses network structure alone to detect disjoint communities, with no overlaps. The main idea behind LPA is to propagate the labels of node throughout the network. At every step each node updates based on the labels of its neighbors. At the end of the propagation, most labels disappear and only a few labels exist in the network and nodes with the same label are grouped together after convergence. It reaches convergence when each has the majority of its neighbors. At the final stage of convergence, nodes with the same label form a community.

To improve the execution time of LPA, Xie and Szymanski (2011) [2] proposed an update rule based on neighborhood strength driven label propagation. The idea behind this work is to avoid unnecessary updates at each iteration of the algorithm, while keeping the overall demeanor of the algorithm and to better the tone of the community detection. By preserving the information about the boundaries of the community to avoid recalculation, reduced the time complexity of LPA.

Leung [3] extended LPA using the hop attenuation technique. The idea behind this work is to prevent community identifier propagate too far, the node is checked after every propagating step, whether it is still among the community identifier and all others are deleted.

COPRA [5] and SLPA [4] incorporate LPA with the modification of each node to use more than one label to discover overlapping communities. The main idea is, every vertex is given a label with belonging coefficient set to 1, every vertex updates its label by summing and normalizing the belonging coefficients of its neighbors,

require a parameter  $v$  to limit the communities. If the value of  $v$  is greater, the algorithm deletes overlaps between communities. At the end of propagation it remove communities that are totally contained by others. Speaker-Listener label propagation (SLPA) distributes labels, according to interaction rules and maintains label distributions of each node in memory. Each node can be a listener or a speaker. The characters are swapped depending on whether a node serves as information provider or consumer. A node can hold as many labels as it likes, depending on the underlying network structure. The more a node observes a label the more likely it will spread this label to other nodes. At the end of iteration it identifies overlapping communities.

Markov Cluster Algorithm (MCL) [6] is based on simulations of flow (random walk), the idea behind the algorithm is to add self-loop to each node, normalize the adjacent matrix using stochastic process followed by inflation operator until it reaches steady state.

In Label Rank [7] a set of operators was introduced to control and stabilize the propagation dynamics. Label Rank relies on four operators namely propagation, inflation, cut off and conditional update. Label Rank differs from MCL in two aspects. First, Label Rank applies the inflation to the label distributions and not to adjacent matrix. Second, the update of label distribution on each node in LabelRank requires only local information.

MLPA [8] comprises of six phases, Pre-process, Initialization, Inflation, Post process, Listener, Speaker. The first four phases are like LabelRank and the next two are like SLPA. Each node can broadcast multiple labels to the neighbor nodes and at the same time receives multiple labels from its neighbors to detect overlapping communities.

LabelRankT [9] can be seen as a LabelRank with conditional update rule where changed nodes accept the new distribution and detects evolving communities.

### III. Dynamic Multi-Label Propagation Algorithm(DMLPA)

Our approach is based on the algorithm MLPA with a modification in Speaker-listener propagation to detect disjoint communities, where at each iteration listener listens to all labels sent by the neighbor nodes but accepts a label with high probability. The algorithm is described in Algorithm. 1. The main idea is to allow a change in the structure of a network, such as an edge being added or removed, would affect the neighborhood between the nodes that are incident to this edge. The changes in nodes and edges are reflected in the edges alone. The algorithm track these edges in a sub graph and find the differences in the edges between initial graph and sub graph for every time stamp to recognize the structural changes in a graph, and update the graph with remainders to find incremental communities. Moreover, we update the neighbors list, label distribution matrix, the propagation process of the changed nodes and assign them to the corresponding communities.

An example that shows different temporal events in three consecutive snapshots.  $G_0$ ,  $G_1$ ,  $G_2$  is shown in Figures. 1,2 and 3. During the evolution edges are added or removed and communities form, dissolve, survive and split all these are captured by DMLPA.

**Figure 1: (i) The example network  $G_0$  with 7 nodes. (ii) Colors represent communities discovered by DMLPA**

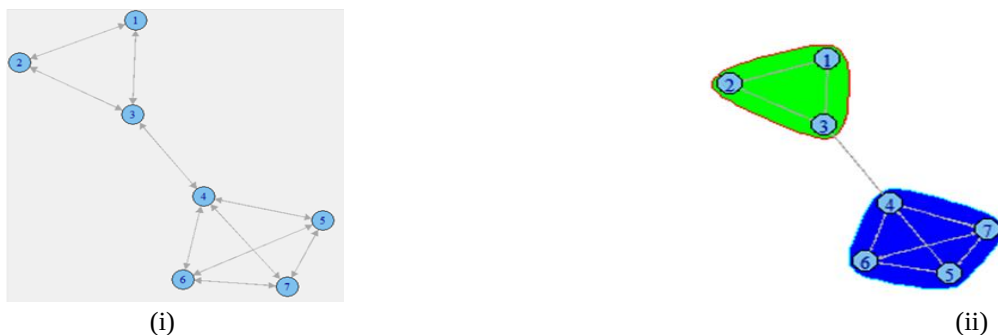


**Figure 2: (i) The event changes at first timestamp. Four edges were added. (ii) The remainder edges between initial graph and sub graph (iii) the example network  $G_1$  with evolved edges.**

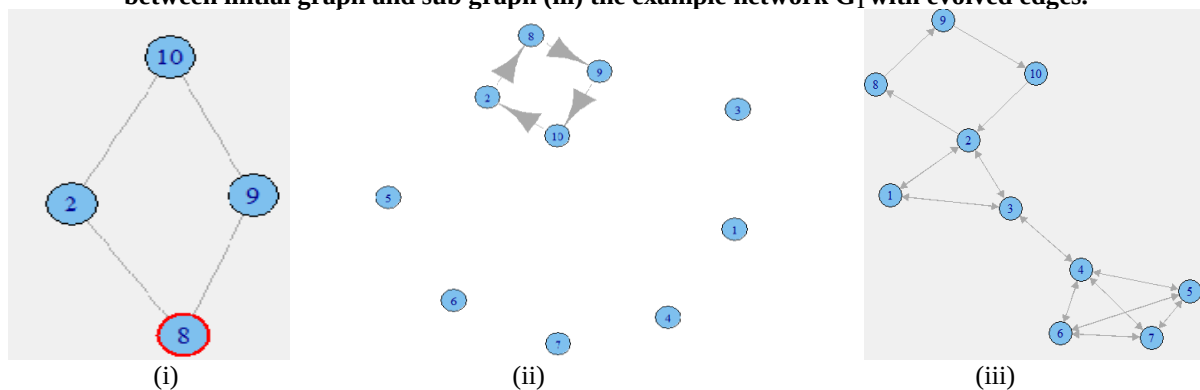


**Figure 3: (i) The event change at next timestamp (ii) The remainder edges between graphs with changed events. (iii) Node 10 was removed, one edge was added. Colors represent communities discovered by DMLPA**

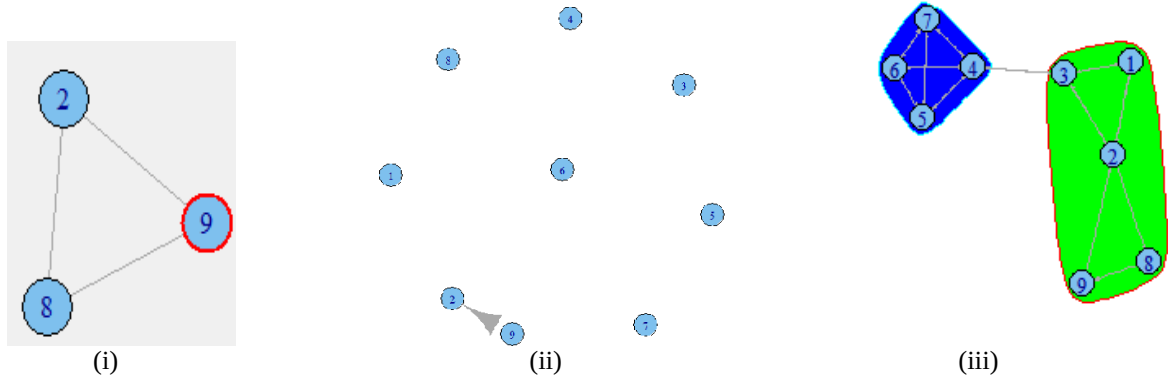


**Table 1: Notations**

| Symbols     | Definition                          |
|-------------|-------------------------------------|
| T           | Snapshots                           |
| E           | edge list                           |
| A           | Adjacency Matrix                    |
| $A_e$       | Adjacency of edge e                 |
| P           | Label Distribution Matrix           |
| $P_e$       | Label Distribution of edge e        |
| $A_{(x,x)}$ | Self loop for x in adjacency matrix |

**Algorithm 1: DMLPA**

Input: Graph Timestamps(0.....T)  
run DMLPA for  $G_0$   
for t in 1 thru T:  
     $E = \text{edgelist}$   
    for each e in E  
        if e is an addition  
            add e to  $G_t$   
        else  
            remove e from  $G_t$   
        end if  
    end for  
    get the subgraph  $G_t$   
     $\Delta E = \text{graph.difference}(G_t, G_{t-1})$   
    updateNetwork( $G_{t-1}, E$ )  
end for

**Algorithm 2: updateNetwork(G,E)**

Input: Graph G, edge changes E  
for each e in E  
     $u, v = \text{nodes of } e$   
     $x = u$   
    if x is in list of edges of G and label="add"  
        add e to G  
         $A_e \leftarrow -1$   
        Update neighbor list of x  
         $K \leftarrow \text{sum of all adjacent neighbors of } x$   
         $P_e \leftarrow -1/K$   
        Speaker(x)  
    else:  
        if x is not in list of edges of G and label="add"

```

    addVertex x to G
     $A_{(x,x)} < -1$ 
    add e to G
    add x to A
     $A_e < -1$ 
    Update neighbour list of x
     $K \leftarrow$  sum of all adjacent neighbors of x
     $P_e < -1/K$ 
    Speaker(x)
else:
    if label="delete"
        delete e from G
         $A_e < -0$ 
         $P_e < -0$ 
        Speaker(x)
end if
end for

```

#### IV. Evolution on Real-World Networks

The proposed framework is applied on Enron email dataset in order to show the feasibility of evolving events. The Enron Corpus is a dataset contains of over 600,000 emails generated by 158 employees of the Enron Corporation [11], [12]. The information from the dataset made it possible to create timestamps for evolving networks. Our network dataset is limited to a period from 02-01-2002 to 22-04-2002 in order to reduce the graph size. Our timestamps for this network are by days that show email exchange between two individuals. During evolution communities form, dissolve, survive and split were shown using DMLPA. Modularity [10] is defined as the ratio of difference between the actual and expected number of edges within the community. It is the most used metric in evaluating the quality of communities found by community detection algorithms. It is defined as

$$Q = \frac{1}{2|E|} \sum_{ij} \left[ A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta_{c_i, c_j},$$

$$\delta_{c_i, c_j} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ in the same community,} \\ 0 & \text{otherwise.} \end{cases}$$

Where E is the sum of all edges, A is the adjacency matrix of the network,  $k_i$  is the degree of node i,  $k_j$  is the degree of node j.

**Figure 4: Shows the structural changes in Enron dataset, including the number of edges added and deleted as well as the number of nodes evolved from 02-01-2002 to 22-04-2002.**

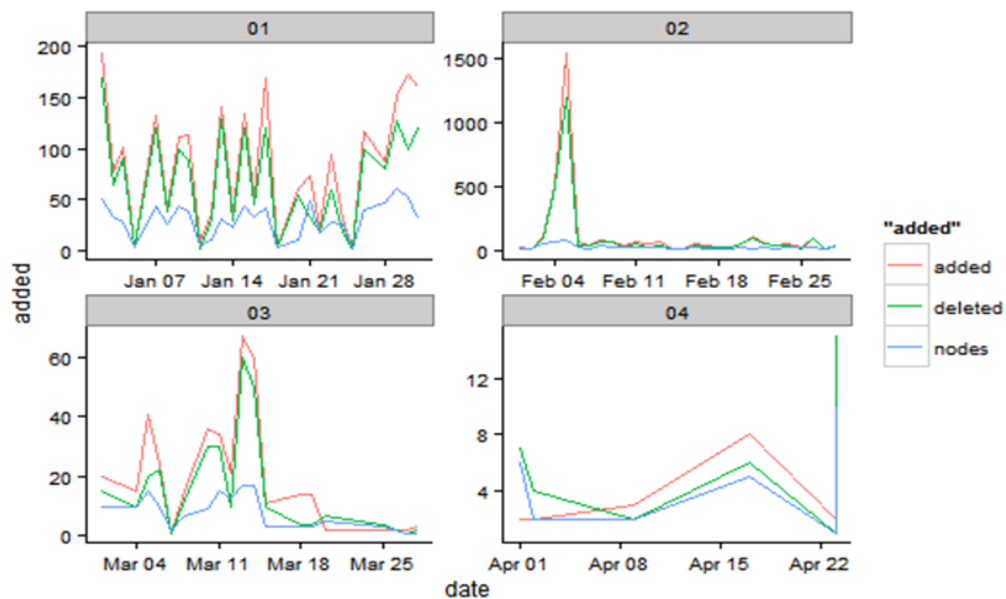


**Figure 5: Shows the evolution of communities in Enron dynamic dataset.**

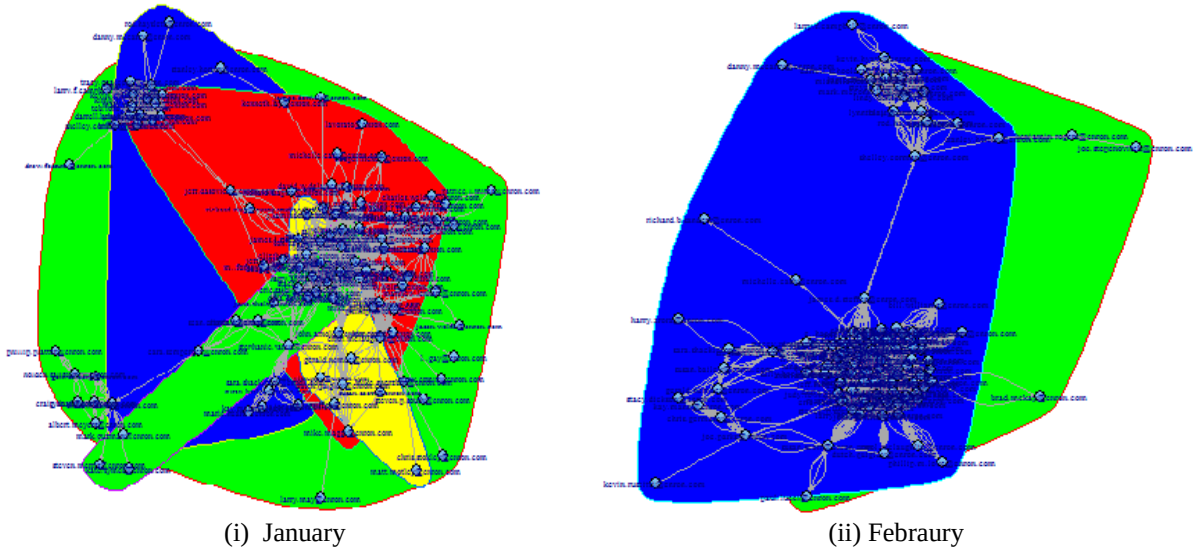
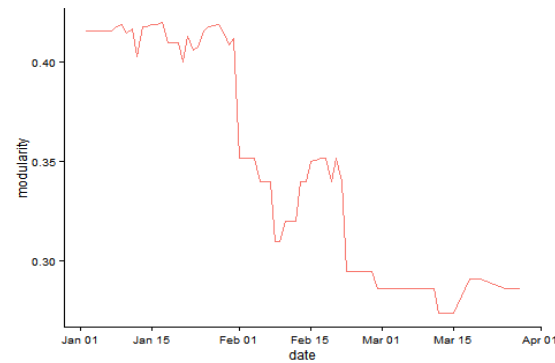


**Figure 6: Shows the modularity of the communities for each timestamp of the Enron dataset with MLAP and DMLAP**



## V. Conclusion

With the rapid growth in the size and space of online social networks, it is not feasible to analyse community detection techniques using static approach. In this paper we propose DMLPA framework for tracking the structural changes and community evolutions over a period of time. Experimented our framework on the Enron Email dataset, and able to track the changes for every timestamp and detected incremental communities. Modularity was formulated to find the goodness of the community structure.

## References

- [1] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, 2007.
- [2] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Surveys*, 45(4), 2013.
- [3] I. X. Y. Leung, P. Hui, P. Li, et al., Towards realtime community detection in large networks, *Physical Review E*, 79, 6, 2009, pp. 066107.
- [4] J. Xie and B. K. Szymanski. Towards linear time overlapping community detection in social networks. In *PAKDD*, pages 25–36, 2012.
- [5] S. Gregory. Finding overlapping communities in networks by label propagation. *New J. Phys.*, 12:103018, 2010.
- [6] S. van Dongen. A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science, 2000.
- [7] J. Xie and B. K. Szymanski. Labelrank: A stabilized label propagation algorithm for community detection in networks. In *Proc. IEEE Network Science Workshop*, West Point, NY, pages 138–143, April 2013.
- [8] Prabavathi G.T., Thiagarasu V., Design and Development of Overlapping Community Detection Algorithm using Multi-Label Propagation, *International Journal of Advance Research in Computer Science and Management Studies*, Vol.2(2), 2014, pp. 195-199.
- [9] J. Xie, M. Chen, and B. K. Szymanski, "LabelrankT: Incremental community detection in dynamic networks via label propagation," *arXiv preprint arXiv:1305.2006*, 2013.
- [10] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004.
- [11] [http://en.wikipedia.org/wiki/Enron\\_scandal](http://en.wikipedia.org/wiki/Enron_scandal).
- [12] <http://www.isi.edu/~adibi/Enron/Enron.htm>.