Chapter 4

Analysing Social Media Networks by Algorithmic Approach

4.1 Network Communities

This chapter focuses on communities or clusters, the sets of nodes with lots of links within and few outside the network. The chapter explains the concept of online generation communities and their framework in Online Social Media Networks (OSMNs), the largest networking site, i.e., Facebook. Many popular methods are available for community identification like Walk trap, Nibble, Label Propagation algorithm (LPA), Fast Community Network Algorithm (FCNA) which has been explored earlier. The Community Framework (CF) is the important and integral part of the OSMNs, but still nobody has to find the correct definition of the community in the real-world networks. In this chapter, the study tries to give a correct definition of the community with its few important traits and from that, we recommend or propose a different, simple and innovative algorithm which will resemble the real-world network. In our approach, we try to incorporate or consider those nodes overlapped in the community framework with the concept of the shortest paths. The study believes that our approach will be more favourable than other network methods which generate the partitions.

Before we define community, we should know network science and types of networks, which includes Sociology (social network analysis-SNA), Mathematics,
Computer Science (Graphs), Economics, Bio-informatics (Networks), Statistical Science (Complex Networks). The two networks are (i) Dynamic network-evolves with time for e.g. Communication or social network (efficient and less time consuming) and (ii) Static Network tests the community framework from scratch for each change in network structure (long running time, the same reaction to a small change).

What makes a community? A cohesive subgroup which has (i) Mutuality of Ties—which define everyone in the group has ties (edges) to one another, (ii) Compactness—closeness reachability of the group with a few steps, (iii) Density of edges—high frequency of ties within the group and (iv) Separation—higher frequency of ties among group members compared to non-members.

An important challenge for the researchers are to discover the community architecture from the different large real and complex online networks. Researchers in [11, 13, 14 and 143] have exploited and mined Online Social Media Networks (OSMNs) by using predictive analysis through text, link, and spatial-temporal information Facebook networks. Many other researchers during the last decade have developed several algorithms to find a community framework or structure in a complex network but with limitations [144]. For example, few algorithms performed worse on huge complex networks, some required few prerequisites like community members, few not able to detect the coinciding communities, some required few parameters to start the algorithm, few not able to explore the mall or sparse classes (communities), some specific to the domain, few able to work with specific structures and handful able to generate good and stable partitions.
The study worked on an innovative approach, identified the total number of communities and found non-connected or secluded nodes in communities and divided the network into stable segments or partitions. The study believes that this approach or method can produce different communities like coinciding, cohesive (less connected), small and even sparse (scattered) communities. It is important to find or locate the smallest and sparse communities of their number is huge in real work networks like biological networks.

Today, most of the researchers related to network analysis focused on community identification, its structure and to list its properties to give an exact picture of the community in real-world networks. One of the most popular methods that are based on the modularity optimization or the \([Q]\) factor [93, 145-147] of a sub-graph or sub-network. Few approaches [66, 93,148-152] tried to compel each vertex to be allocated to one community. This is not possible in real-world networks. Many algorithms had been developed, explored and tested to detect the overlapping or coinciding communities [153-160]. To find link communities, many algorithms had been developed [157,161]. Link communities help us in finding coinciding communities since edges have their own unique identity as compared to nodes or vertices. Other approaches like statistical [162], static-informational approach [144,163,164] and dynamical or synchronization approaches [165-168].

Community identification (CI) is clustering. But why researchers use CI in physical aspects rather than clustering because of the “network big data” pose challenges to the old classical clustering method. Community identification can be defined as the gathering of network vertices into groups in such a way that the nodes in each group are connected densely from inside and sparser from outside. The various applications are social network problems, Routing in Mobile Ad-Hoc
networks, worm containments in cellular networks etc. The process of discovering the community framework (CF) of a social media network is known as community identification (CI) or can define as gathering of network vertices into groups in such a way that nodes in each group are densely connected inside and sparser outside (for example applications like social network problems, routing in mobile Ad-Hoc Networks or Worm Containments in cellular Networks).

As per literature review, the real-time networks, i.e., online social media networks (OSMNs) are neither regular nor random, but also non-trivial in their topology, scale-free and having universal properties like complex systems and present everywhere. The empirical network features which define real-world networks is discussed in [10] are; (a) Power-law (heavy-tailed) degree distribution, (b) Small average distance (graph diameter), (c) Large clustering coefficient (transitivity) (d) Giant connected component, hierarchical structure, etc. Our research article [10] discusses the various network analysis models which were: (a) Random graph model (Erdos & Renyi), (b) preferential attachment model (Barabasi & Albert) and (c) Small world model (Watts & Strogatz).

Network communities are groups of vertices similar to each other and Community identification (CI) is the process of assigning vertices to communities. The non-coinciding communities are those where every vertex belongs to a single group. The aim of CI and Graph Clustering (GC) is same i.e., partitioning network nodes into groups. In graph clustering (GC) the number of clusters is predefined or given as part of the input since in CI the number of communities is unknown. Further community identification can be done in two ways (a) Vertex clustering (vertex similarity) and (b) Graph partitioning (sparse cuts).
In [169] explored, that for any random graph, the degree of all the vertices (i.e., Distributions of edges) are either equal or homogeneous because of the Poissonian distribution. But the real-world networks are not like the random graphs, as they display big inhomogeneities. Also, the degree of distribution is broad and scale-free it means many nodes with lower degree and less number of nodes with higher degree exists in the network.

4.2 Earlier Communities Detection Algorithmic Approaches

Before our research discusses the algorithmic part of community identification, the study wants to throw light on three basic and important properties [66, 150] which helped us in developing an improvised algorithm to find communities. Those properties are (i) community architecture, (ii) community (association) membership and (iii) coinciding (overlapping) member properties. The first property is used to describe the community; is a sub-network that connects the internal members more densely rather than externally. The strong ties community is one who has more connections internally than externally since weak ties communities are those which has more densely connected externally rather than internally. The second property is the community association or the membership property of a member which defines that the respective non-coinciding member has more neighbours within its own community rather than outside in any other community (i.e., Each node or vertex not coinciding with other communities should join the community which has the largest number of member neighbours). The third property is coinciding (overlapping) member properties used to represent the members of a community.
Further, two categories of coinciding vertices exist, *first category*, depends on the total number of connections between that vertex and corresponding communities and the shortest path exists between that vertex and the hub members (comprises the largest number of neighbours within a community) of the corresponding communities, while the *second category* depends on the community architecture or layout or topology. For the *first category* of coinciding nodes not only the number of connected links between coinciding node and the matched communities should be same, but also the path (should be equal and shortest) between that vertex and the core members of the comparable or matched communities. In the *second category*, the coinciding node should be strongly affiliated with the matched communities and least number of connections between the two matched (corresponding) communities if such types of nodes have to remove.

![Diagram of Community Algorithms](image)

**Figure 4.1:** Diagram to depict the taxonomy of community algorithms
The study has explained the following methods or algorithms of community identification with their basic notion, constraints and their time complexity as mentioned in (Figure 4.1).

### 4.2.1 Edge–Betweenness

**Basic Notion:** The improvised version of LPA [94]. This algorithm divides the network into smaller pieces by finding edges that “bridge” communities [66].

**Constraints:** Can be adapted to direct networks (digraph), Can be adapted to weights (no public software), and do not able to draw conclusions about small communities (i.e., gives a big picture).

**Time Complexity:** $O(|V|^3)$ in general, $O(|V|^2 \log |V|)$ for special cases.

### 4.2.2 Fast Greedy

**Basic Notion:** [94, 98, 99, 110] try to assemble randomly larger and larger communities from the ground up. Start by placing each vertex in its own community and then combine communities that produce the best modularity at that step.

**Constraints:** Can be adapted to directed edges (digraph), Can be adapted to weights (no public software), tends to create aggressively larger communities to the detriment of smaller communities.

**Time Complexity:** $O(|E| \cdot |V| \log |V|)$ the worst case.

### 4.2.3 Leading Eigenvector

**Basic Notion:** [170,171] Use the sign on the components of the leading Eigenvector of the Laplacian to divide sequentially the network.
**Constraints:** Can be adapted to directed edges (no public), Can be adapted to weights (digraph).

**Time Complexity:** \( O(|V|^2) \).

### 4.2.4 WalkTrap

**Basic Notion:** [172] simulates many short random walks on the network and compute pairwise similarity measures based on these walks. Use these similarity values to aggregate vertices into communities.

**Constraints:** Can be adapted to direct edges (digraph), Can be adapted to weights, and can alter the resolution by walk length.

**Time Complexity:** Depends on walk length \( O(|V|^2 \log |V|) \) typically.

So far, none of the method discussed above shows the standard quantitative definition to find the community structure that can be acceptable in the real-world. The research should be able to develop the algorithm simple, stable enough to understand the partitions. It should have linear time complexity. It should be an unsupervised approach and heuristic.

### 4.3 Our Proposed and Improvised Version of LPA Algorithm

Our study proposes that every algorithm must have following properties to detect community for a graph \((V, E)\) in online media networks.

a) **Scalable**, where time and space complexity are strict sub-quadratic w.r.t. The number of nodes.
b) **Non-parametric**, where the number of communities need not be specified a priori.

c) **Consistent**, where effectiveness is consistently high across a page wide range of domains.

d) **Effective**, where global connectivity patterns are successfully factored into communities that are predictive of individual links and robust to small perturbations in the network structure.

The community identification is a challenging problem and for huge networks coming up day by day, it requires fast and correct algorithms. The existing algorithm has a high order of time complexity. LPA [94] is using a *heuristic approach, linear time complexity, fast convergence and can work on large OSMNs*. The algorithm works as follows:

Step 1  Let G = (V, E) be the given graph of N nodes.
Step 2  Initialize all nodes with a unique label Li.
Step 3  In each iteration, a node searches for the label maximally occurring in it Neighbourhood.
Step 4  It now changes to that label.
Step 5  Ties are broken uniformly randomly.
Step 6  Converge when all nodes have the same label as the one carried by a maximum of their neighbours.

The algorithm depicted as a flowchart in Figure 4.2 is discussed below has the same spirit as *LPA (Label Propagation Algorithm)*. Our notion is based on the concept of community architecture which can be identified from sub-graphs by testing the total count of in and out degree of each community and comprises following steps:
(i) To load the adjacency matrix or lists with connected nodes or vertices, (ii) To locate-find-remove strong and weak communities from adjacency lists, (iii) Repeatedly assign and un-assign nodes to detect communities depend on the community membership property of the respective nodes and (iv) Continue the process until the study gets the final identified communities and isolated nodes.
Figure 4.2: The proposed algorithm flowchart.

Move 1: Load the Adjacency List with connected nodes or vertices.
If the list is empty then go to RESULT/OUTPUT option (we get the isolated node), else go to MOVE 2.

Move 2: Locate, Identify and Remove from Adjacent Lists
Weak Ties Communities

Move 3: Locate, Identify and Remove from Adjacent Lists
Strong Ties Communities

Move 4: Select and Designate non-assigned vertices
To that Community in which its neighbours are present

Move 5: Discard the repetitious
Delete all those communities or groups which are equal in number or proper subset of other communities, in case of overlapping communities exist remove the intersection part from the smaller community.

Move 6: Accomdate all the missed out vertices or nodes
Accustom all missed out nodes to their specific community in which their membership property belongs to.

Move 7: Authenticate various overlapping vertices
Cross check each and every overlapping vertices that exist between two communities. If vertex shows the community overlapping membership definition, then put the same, else allocate the vertex to the community which has greater number of neighbors.

Move 8: Check Identified communities
Atlast if community doesn’t represent the definition of weak tie community, then integrate it with the community which has the mostly populated connections.

Move 9: Look for any change required for each community?
If it is true, then go to MOVE 6: Else next step.

Move 10: Look for any unassign vertex or node?
If it is False, then go to Final Result or Output, else next step.

Move 11: Discard all the non-connected components?
Remove all the communities which are alone or isolated depends on the adjacency list of each unassign node or vertex.

RESULT/OUTPUT
First: We get all the Identified Communities. Second: We get Isolated Communities, if they are present.
The following are the steps of the improvised algorithm:

MOVE 1: *Load the Adjacency List* with connected nodes or vertices.
        if the list is empty, then got to RESULT/OUTPUT Choice (study get
        the isolated node), else Move 2:
MOVE 2: Locate, identify and remove from adjacent lists *Weak Ties communities*.
MOVE 3: Locate, Identify, and Remove from adjacent lists, *Strong Ties Communities*.
MOVE 4: *Select and Choose non-assigned* vertices to that Community in which
        its neighbors are present.
MOVE 5: *Discard the repetitious communities* or groups having an equal number
        or the proper subset of another community, in a case of overlapping
        communities, exist to remove the intersecting part of the small
        community.
MOVE 6: Accommodate all the missed out vertices or nodes to their specific
        community in which they’re membership property belongs to.
MOVE 7: Authenticate various overlapping vertices i.e., cross check each and
        every overlapping vertices that exist between two communities. If
        vertex shows the community overlapping membership definitions, then
        put in the same, else allocate the vertex to the community which has
        the greatest number of neighbors.
MOVE 8: Check Identified communities (at least, if the community doesn’t
        represent the definition of weak tie community, then integrate with
        the community which has the most populated connections).
MOVE 9: Look for if any change is required in each community? If is true, then
        go to MOVE 6: Else next step.
MOVE 10: Look for any unassigned vertex or node? If is False, then go to the
final result or output, else next step.

MOVE 11: Discard all the non-connected components, i.e., Remove all the communities which are alone or isolated depends on the adjacency list of each unassigned node or vertex.

MOVE 12: Study gets all the identified communities and the isolated Communities if they are present.

### 4.4 Similarity Between LPA and Proposed Algorithm

The study believes that the greatest similarity between LPA and our approach is that both are following the same procedure, i.e., they both allocate each vertex in the network graph to a preferable community (which has the most numbers of its neighbours). The study also believes that the possible outcome partitions generated by LPA as well as by our algorithm will favour the same configuration.

### 4.5 Technical Differences Between LPA and Proposed Algorithm

The technical difference between our algorithm and LPA is that LPA loads the node communities, whereas our algorithm load connected edge or arc communities in the adjacency matrix or list (where each vertex corresponds to all its related edges or arcs). The LPA has the constraints like it requires prerequisite conditions for its execution, the tie-break rules and randomly defined roots or seeds. Our approach tries to achieve near linear times complexity (i.e.,
Deterministic approach) for community identification which LPA cannot achieve. The LPA looks for communities based on the process of propagation through the network by dynamically allocating the labels while our algorithm works on static network layout or topology.

Furthermore, in our approach, the study focuses overlapping/coinciding nodes that combine community structure with their shortest paths which LPA ignores. Our approach is simple, stable, free of prerequisite parameters, able to detect overlapping communities, as the edges are more likely to have unique identities than nodes which instead tend to have multiple identities. Moreover, with our improvised approach tried to find not only large and cohesive communities, but also small and sparse communities.

4.6 Possible Experiments Datasets - Facebook Higher Education (HE) Groups

Our study still has to work on our algorithm execution phase, it’s a just a proposed or theoretical model to detect not communities but also the overlapping nodes. In the future, research will work, to test the performance of our algorithm, with the LPA algorithm partitions. By literature review, a study came to know that empirical networks can be tested by NMI (normalized mutual information) values, as discussed below since many others can be tested by calculating the modularity Q-metric mentioned below. As LPA works well on synthetic networks, so study expects our improvised algorithm to behave in the same manner, as both are based on the same notion. But, for empirical real-world networks, the study
believes our algorithm will do better than the other methods and also expects higher stability and modularity. The study concludes that our algorithm will perform well and competes with other methods.

To detect communities in the possible datasets which were collected in our earlier work by using Network Analysis Software Applications (NASA) [15], from the largest social networking site i.e., Facebook is having Higher Education (HE) Online Groups. A social network directed user-user friendships, the network having 367 nodes and 6468 vertices (figure 4.3) as compiled by us by using NASA in January 2015. The figure 4.3 below describes the graph network of the fetched online group, its dataset and communities detected by NASA. Our study found 367 strongly connected components (Fig 4.4) for our fetched datasets which may be compared or tested in the future with the ground truth communities.

Figure 4.3: The graph network for Higher Education group on Facebook with nodes 367 and edges 6468 (NASA tool- Gephi).
Figure 4.4: Total number of strongly connected components ID which has been found to be 367 for Higher education group on Facebook (NASA tool- Gephi).

Figure 4.5: Total number of communities as per modularity report, which has been found to be 6 for Higher education group on Facebook (NASA tool- Gephi).
Figure 4.6: Total number of strongly connected components which have been found to be 367 and weakly connected components is 2 for Higher education group (NASA tool- Gephi).

and weakly connected component (2) in figure 4.6 and total communities found (6) (figure 4.5) are detected by NASA with our proposed algorithm.

4.7 Algorithm Evaluation

At present, there is no universal technique to check the correctness of the segments created by the algorithms because it is a cumbersome process to predict the architecture of real-world networks beforehand. Now to evaluate the quality of community segments, the study will be using the (NMI) normalized mutual information measure [153] which is defined below.
In above formula X and Y represent reality and predicted communities, H (X) is the random entropy of X, and H (X, Y) H (X, Y) represents the joint entropy of X and Y.

\[
NMI(X|Y) = \frac{(H(X)+H(Y)-H(X,Y))}{(H(X)+H(Y))/2}
\]

(4.7)

As per literature survey, other networks are using the above mathematical expression which calculates Modularity [173] property to calculate the quality of a community segment (partition). It is generally believed that networks which are random in nature do not exhibit community architecture. To calculate Modularity study has to define a matrix e where the each element \(e_{ij}\) depicts the fraction of the total number of connections between two contrasting communities, and the absolute fraction of ties particularly within the reach of a community it is \(e_{ii}\). Finally, the sum of any row of \(e_{ij}\) = \(\sum_{j}^{} e_{ij}\) corresponds to the fraction of links connected to the community \(i\), and the expected count of intra-community ties are just \(a_{i}^{2}\). The study can compare \(e_{ij}\) and \(a_{i}^{2}\) directly, and the total of all the communities present in the real-world network.

Our motive is to develop an algorithm which has less time complexity, so that it can be used on large-scale complex real networks. It has two phases: (i) to identify communities, and study expects its time-cost complexity which comes out to be \(O (m^{2} + \text{lamb})\), where \(v\) is the total count of vertices, \(m\) is the total count of
identifying communities and $l$ is the maximal size of the initial adjacency lists, (ii) after adjusting membership among communities, and the study gets $O(nm^2+lm^2+mv)$ time-cost complexity, where $n$ is the maximum number of coinciding nodes. Therefore, the final calculated and expected time complexity of both phases that may come out to be $O((n+l)m^2+lmv)$. In the future, we are planning to use this algorithm on a real large scale network due to its lesser time complexity.

4.8 Conclusion and Discussion

Communities are not only groups of associated members which connect densely but also sparsely with the remaining part of the network. Many hidden patterns can be disclosed out by detecting the community and by its architecture. But still our study has done a great deal of research to uncover the real-world systems whose structure is not being fully understood. With the help of variously discussed approaches in section 3 of this chapter, it is possible to throw some light on the layout or structure of these networks to improve, maintain, manage, control and renovate them. In this chapter, the study proposes a simple, different and innovative algorithm to identify a community framework of complex networks, especially social media networks based on network topology configuration. Comparing other popular approaches or algorithms discussed in the literature, the study believes that our approach will work competitively on both artificial and real-world social media networks, but we know it is a long journey to get a breakthrough in community detection.
In the future, we still have to work on its practical execution, predictability and furthermore its utility by identifying another category of coinciding vertices and explore the hierarchical architecture of the network. Presently our approach is to detect the most coinciding vertices where we are ignoring least coinciding (overlapping) vertices thereby making it difficult to divide larger communities into smaller ones. In the future, we may try to work on the optimization of modularity property of the network, which is an NP-hard problem.