Chapter 5

Membership based Disjoint Community Detection

5.1 Introduction

Size of global datasphere had grown from 0.5 exabyte in 2009 to 18 exabyte in 2018. Data is being generated continuously which brings a new parameter, time, in study. Static community detection algorithms consider network as a fixed graph. Using a static algorithm to identify communities can be very resource consuming for present scenario. These algorithms are also not being able to incorporate the evolution of network over time. Time parameter in data plays a key role in emergence of a new category in community detection which is dynamic networks. Owing to always changing nature, dynamic networks often considered as a stream. It can be a stream of events or a set of snapshots as already discussed in section 2.2.

Community of a node is influenced by its neighbours as it is more likely to join the community of one of its neighbors. A node can be a member of a community with two types of belongingness, fully or partially. State-of-the-art algorithms for disjoint community detection often use crisp or full membership for classification. This comes with a major drawback of losing flexibility is community label assignments. With partial or fuzzy memberships, algorithms can control the classification criteria by adjusting the threshold value and also it allows authors to compare and adjust the membership requirements. Many

researchers lack the membership value of a node for a community in disjoint community detection.Motivated by this, the proposed work has considered the notion of membership value to classify nodes more precisely.

Assignment of a node's community depends on their local as well as global surrounding. Considering global knowledge may increase the complexities of algorithm whereas local knowledge may cause loss of valuable network information affecting the community structure. A trade-off between global v/s local is proposed by authors in [76]. Neighborhood of neighbors plays an important role in community decision process [90]. The work proposes a fuzzy based community detection algorithm by exploiting these two important observations of literature. Community of a node is not only influenced by their neighbors but also by characteristics of their neighbors surrounding. For example if a node is a neighbor of densely connected nodes, it is very likely that the node will be influenced by their neighbors and join their community. These literature observations are inspiration of the novel membership function proposed in the work. Membership function is used to calculate belongingness of a node to existing communities. Mathematical formulation and description of function is discussed in detail in section 5.2.1.

Contribution The work presented in two previous chapters focuses on identifying a community for a given node, while in this chapter we consider the notion of membership of a node into a community. The work presents a community detection algorithm for dynamic networks. It uses fuzzy membership function for determining belongingness of a node to any community. Main contributions of the work is listed below:

- A novel fuzzy membership function is proposed in the work. Affect of neighbourhood on node's community label is an important factor, also the neighbourhood of neighbour presents sufficient information to effectively analyze community transition over time. These two important observations from literature are considered in its formulation.
- We have considered dynamic social network for community detection problem. It presents global community structure of network at time t. At time of arrival, a new node is assigned to a community based on the value of membership function.

- Proposed algorithm is evaluated on seven well-known quality and accuracy metrics. Three real and five synthetic datasets are tested for these metrics. The work also presents a comparative analysis of proposed algorithm against ten state-of-the-art algorithms. Proposed algorithm shows better performance against other algorithms.

5.2 Proposed Work

This section elaborates the proposed algorithm as illustrated in figure 5.1. For each upcoming edge, the algorithm take action based on community event. Birth, expansion, diffusion and death are four possible events that can happen in communities (refer to section 4.2). Initially graph is empty with zero communities. When first edge arrive, one node will form a new community and second one will join it. After it, two parameters, node's newness and its community, are checked to decide whether it will join an existing community or form a new one. The whole process is described in detail in section 5.2.2. Community selection of a node is guided by the membership function used in the work. Inspiration and formulation of membership function is presented in 5.2.1.

5.2.1 Membership Function

Nodes' belongingness to a community is determined by membership function. It is an indicator of a node's affiliation with a community. It can be calculated by considering various factors associated with node for example degree, centrality, neighbourhood, etc. The paper proposes a novel membership function based on two prominent observations derived from literature. Local information is easily available with less complexities but it also raises possibilities of might missing an information that can not be viewed at local level. Li et al. [90] has observed that 2-hop neighbours have high impact on a node's community. Based on these two observations, proposed membership function is formulated.

Membership of node u in community C_i is calculated using equation 5.1,

$$\mu_{C_i}(u) = \frac{1}{|N(u)|}\left[\sum_{v \in N(u)} \left\{ \frac{1}{Degree(v)} \sum_{x \in N(v)} (w_{vx} \cap x.isCommunity == C_i) \right\}\right] \quad (5.1)$$

where,

- $|N(u)|$: number of neighbours of node u
- $Degree(v)$: degree of node v
- w_{vx} : weight of edge (v,x)

Membership of node u for community C_i is calculated at two levels. Initially the 1-hop neighbours are identified. For each neighbour v of u, their neighbours x (2-hop neighbors of u) are examined to check if they belong to community C_i or not. If x is in community C_i, $x.isCommunity == C_i$ will return 1 and weight of edge (v,x) is added to the equation. This summation is normalized by overall degree of node v. Here degree refers to weighted degree of node. Similarly the overall summation is normalized by total number of neighbors of u.

5.2.2 Algorithm

The work proposes an algorithm which is capable of handling dynamic network graphs and identifying underlying communities. An overview of algorithm is presented in figure 5.1. The flow chart incorporates actions taken by algorithm based on newness of nodes forming edges. For each incoming edge, associated nodes are examined to find whether they are new to graph or not. If node u is new to graph, they gives value 0, 1 otherwise.

Algorithm 5 presents pseudo code of the underlying algorithm. It takes two inputs, a new edge (u,v) and the dynamic graph $G(V,E)$. Initially $G(V,E)$ is an empty graph with 0 vertices and edges. Algorithm keeps updating graph in accordance with upcoming edges. It outputs a set of communities at given time t. For each upcoming edge (u,v), it checks whether both nodes are new to network or not. If both are new, a new community ID is generated by first node u and its membership value is set to 1. Since node v is also new and have no neighbour other than u, it is then added to u's community with full membership value. Line 1-5 in algorithm 5 shows the aforementioned steps.

If both nodes are old, in lines 6-9, then algorithm checks if they belong to same community or different. Any new connections between two nodes of same community will strengthen

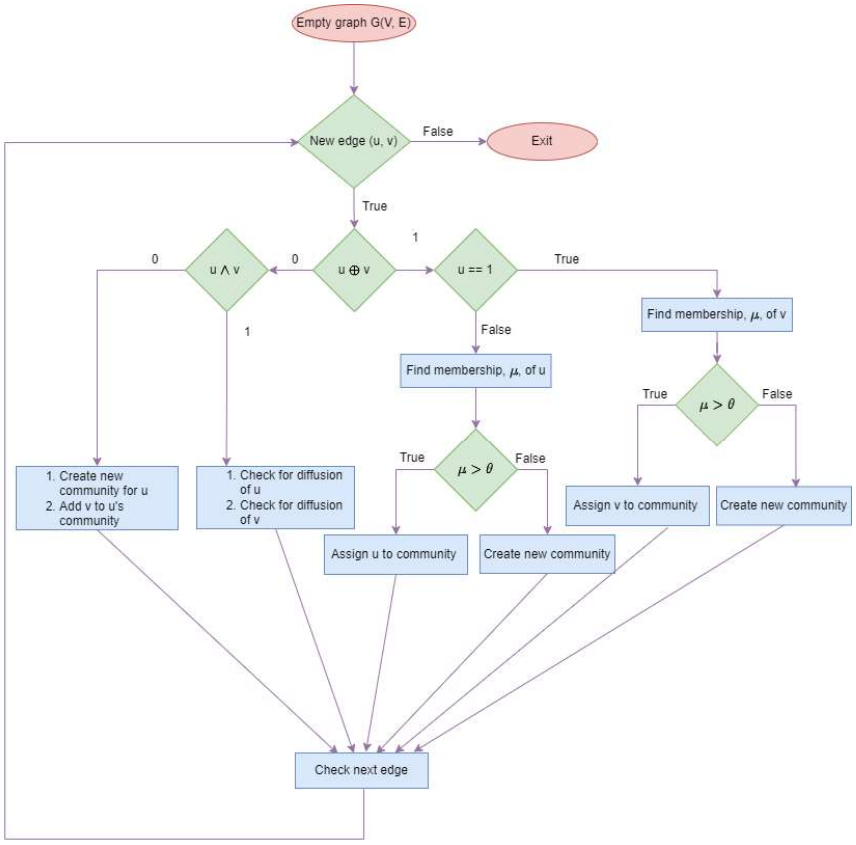

FIGURE 5.1: Flow chart of algorithm

the community. So, it will not change the existing community structure of network and no action is performed by algorithm. On contrary, inter community connections may cause shifting of a node from existing community to another. Hence, if both nodes are from different communities then algorithm will check for node's diffusion. Algorithm 6 describes steps involved in diffusion testing.

Line 10-26 demonstrate the steps taken by algorithm when either u or v is a new node and other one is old. A variable θ is used in this segment. It is used to define minimum membership value of any node in a given community. The value of θ is predefined by expert and empirical analysis. Here θ is set to 0.3. Node which is new will calculate its community label and membership value using algorithm 7 and then it will be added to the community.

Membership value of a node is recalculated in diffusion testing step. If algorithm 7 returns community label which is same as old community label of the node, it will update the new

membership value of node for the community. Whereas in case of different community label, both the community and membership value is updated for the node.

Algorithm 7 presents pseudo code for a node's membership value calculation. It takes a node u and graph $G(V,E)$ as an input and return community label and membership value of node u. Line 3-7 calculates candidate communities for node u. Candidate communities is a set of probable communities for u. A node can join any neighbour's community so, the set will contain all distinct neighbour's communities. For all candidate communities, membership value of u is calculated (line 8-9). Node will be assigned to the community for which its membership value is maximum.

5.2.3 Complexity Analysis

Time complexity analysis of proposed algorithm is presented in this section. The paper include 3 algorithms namely, *MDCD, Diffuse* and *MembershipEval*. Algorithm *Diffuse* takes $O(1)$ time as it updates attributes of node u. For an average degree k of network G, algorithm *MembershipEval* takes $O(k^2)$ time. Main algorithm *MDCD* is called for every new edge of graph $G(V,E)$ and take $O(E)$ time. If clauses in Line 1-5 and 6-9 take $O(1)$ time. Line 10-26 take $O(k^2)$ time in worst case. Hence, time complexity of the proposed algorithm is computed as $O(Ek^2)$.

5.3 Performance Assessment

5.3.1 State-Of-The-Art Algorithms

- RandW - The concept of using metadata of network is used by authors in [152]. Random walk algorithm is improvised by considering attributes of edges and nodes for selecting next node in a random walk.
- LeadF - Authors in [146] consider internal structure of communities as an important factor for overall community structure of network. They proposed a leader-follower based approach. Leader selection is followed by follower distribution in this approach.

Algorithm 5: MDCD

Input: New Edge: (u,v), Dynamic Graph: $G(V,E)$
Output: $G.communities$: Community Structure

```
if (Is.New.u) ∧ (Is.New.v) then
    i ← NewCommunityID
    u.community ← i
    u.membership ← 1
    AddToCommunity(v, i, 1)
else if (!Is.New.u) ∧ (!Is.New.v) then
    if u.community ≠ v.community then
        Diffuse(u)
        Diffuse(v)
else
    if (Is.New.u) ∧ (!Is.New.v) then
        (c, μ) ← MembershipEval(u)
        if μ ≥ θ then
            AddToCommunity(u, c, μ)
        else
            i ← NewCommunityID
            u.community ← i
            u.membership ← 1
    else
        (c, μ) ← MembershipEval(v)
        if μ ≥ θ then
            AddToCommunity(v, c, μ)
        else
            i ← NewCommunityID
            v.community ← i
            v.membership ← 1
Return G.communities
```

- Tiles - This approach [138] is also based on internal structure of communities. Authors classify nodes as core and peripheral members. For every new network interaction label propagation algorithm is applied on core members of participating community.

- TCD2 - Tree structure is used by authors in [112] for identifying community structure in network. Initially height of tree is decided based on few well-established social

Algorithm 6: Diffuse

Input: Node: u Dynamic Graph: $G(V,E)$
Output: $G(V,E)$: Updated Graph

```
c_old ← u.community
μ_old ← u.membership
(c, μ) ← MembershipEval(u)
if c ≠ c_old then
    u.community ← c
    u.membership ← μ
    AddToCommunity(u, c, μ)
else
    if μ ≠ μ_old then
        u.membership ← μ
Return G(V,E)
```

Algorithm 7: MembershipEval

Input: Node: u, Dynamic Graph: $G(V,E)$
Output: c: Community label, μ: Membership

```
n ← G.neighbors.u
comCandidate ← {}
forall j ∈ neighbor do
    n_n ← G.neighbors.n
    forall k ∈ G.n_n do
        i ← n_n.j.community
        comCandidate ← comCandidate ∪ i
forall C_i ∈ comCandidate do
    μ_{C_i}(u) ← MembershipInC_i          ▷ Refer equation 5.1
(μ_max, C) ← {μ_{C_i}, C_i : μ_{C_i} > μ_{C_j} ∀ j ≠ i}
Return (C, μ_max)
```

concepts. After that a set of rules govern the nodes classification into different communities.

- DAMNF - Authors in [172] uses encoder-decoder model on non-negative matrix factorization. It state that the inherent hierarchal network information can be better learned from encoders and provide efficient classification.

- MNMF - It [165] uses network embedding for finding its existing community structure. They propose a model by optimizing NMF using modularity based

optimization.

- NNSED - The concept of NMF and encoder-decoder is amalgamated by authors [156] to create a model that can learn cluster membership distribution of nodes in network.
- GEMSEC - Authors in [142] generate a mutual information matrix by using random walks on adjacency matrix. Further factorization is performed on obtained matrix. Node embedding and classification is obtained simultaneously.
- EdMot - This method [89] aims to form a graph structure which is composed of higher order motifs. Lovain method is used for generating non-overlapping clusters.
- SCD - It [132] uses an optimization approach to find communities in network. Weighted community clustering (WCC) metric is greedily maximized by algorithm.

5.3.2 Datasets

The proposed work is evaluated on real and synthetic datasets whose ground truth is available. Three real and five synthetic datasets are used in the paper. This section presents a brief description of these datasets.

5.3.2.1 Real Datasets

- Karate- [177] generated this social network dataset after studying a university's karate club. It was initially gathered in order to create a model of an anthropological problem. The club consisted of 34 members. Three years are spent monitoring members and their interactions. Due to disputes between the administration and the instructor, the club exhibits group disintegration. We used the version [48] of the Karate dataset.
- Dolphin- Lusseau et al. [99] gathered this information from 62 bottlenose dolphins inhabiting in New Zealand's Doubtful Sound. Their conduct was observed between 1994 and 2001.

- Football- It refers to the games of American football between Division IA institutions. The graph is made up of vertices representing collegiate teams and edges representing season matches between the participating teams. During a particular season, each of these teams is separated into conferences of eight to twelve teams. Teams from the same conference are more likely to compete against each other than teams from other conferences.

5.3.2.2 Synthetic datasets

Comparative analysis of different algorithms is performed on five synthetic datasets. They are generated using benchmark proposed by Lancichinetti et al. in [81]. It is commonly known as LFR benchmark and widely used for testing in community detection field. This class of graphs follow power law distribution for node degree and community size. Graphs with 500 nodes are generated for the experiment with various value of μ in range $[0, 1]$.

5.3.3 Metrics

The literature of community detection problem incorporates various performance metrics. A set of metrics are selected from them to evaluate quality as well as accuracy of results obtained from algorithm. Section 2.4 presents description of evaluation metrics used in the paper.

5.3.4 Results

This section presents comparative analysis of *MDCD* with state-of-the-art algorithms over real and synthetic datasets. Results of quality and accuracy metrics are shown in different subsections. Quality metric results are compared with ten state-of-the-art algorithms, including overlapping and non-overlapping algorithms. Meanwhile, accuracy results are compared for only non-overlapping algorithms.

5.3.4.1 Quality

In order to evaluate the performance in terms of quality measures, an empirical analysis of the proposed algorithm among ten state-of-the-art algorithms over eight (three real and five synthetic datasets) well-established datasets have been conducted. The findings of this empirical analysis has been shown in table 5.1. From the results, it is clear that our proposed algorithm offers better quality in all the quality evaluation metrics. The overall performance of our proposed algorithm for synthetic datasets (LFR1 to LFR5) achieve better Modularity, Coverage, and External Density over state-of-the-art methods, but lags behind in Average Isolability in LFR2, LFR3, LFR4, and LRF5 datasets. For real datasets, Karate and Football, our proposed algorithm performs better than other algorithms in all quality measures, whereas it lags behind TCD2 in all the quality measures. For football dataset, *MDCD* shows better results for modularity and external density compared to all algorithms. It also presents better metric values of Coverage and Average Isolability than all algorithms except TCD2. Overall, our proposed algorithm attains better quality not only with the real datasets but also with synthetic datasets.

TABLE 5.1: Quality metrics results' comparison of the algorithm with state of the art algorithms

Network Type	Dataset	Algorithm	Quality Metrics Results				
			Modularity	Coverage	External Density	Average Isolability	NOC
Real	Karate	RandW	0.6341	0.8718	0.0351	0.7684	2
		LeadF	0.3540	0.4615	0.0911	0.2314	8
		TILES	0.1289	0.2949	0.0557	0.1818	11
		TCD2	1.0000	1.0000	0.0000	0.7029	2
		DANMF	0.0592	0.0897	0.0749	0.0945	8
		MNMF	0.2486	0.2949	0.053	0.3092	10
		NNSED	0.164	0.2436	0.0636	0.1436	7
		GEMSEC	0.2935	0.3974	0.0474	0.2954	10
		EdMot	0.5215	0.6923	0.0292	0.654	4
		SCD	0.3134	0.4872	0.0402	0.1252	17
		Proposed	0.6656	0.8846	0.0125	0.4902	2

Table 5.1 – continued from previous page

Network Type	Dataset	Algorithm	Quality Metrics Results				
			Modularity	Coverage	External Density	Average Isolability	NOC
	Dolphin	RandW	0.4313	0.4906	0.0488	0.2825	9
		LeadF	0.3777	0.4214	0.0523	0.2196	17
		TILES	0.2969	0.3522	0.0283	0.1147	32
		TCD2	0.8396	0.8553	0.0108	0.7859	6
		DANMF	0.3624	0.4214	0.0281	0.3477	8
		MNMF	0.446	0.5283	0.0218	0.4336	10
		NNSED	0.1955	0.239	0.0387	0.0853	13
		GEMSEC	0.4689	0.5472	0.0223	0.4599	10
		EdMot	0.648	0.7547	0.0132	0.6986	5
		SCD	0.4932	0.5912	0.0188	0.1699	24
		Proposed	0.6672	0.5786	0.0123	0.4293	6
	Football	RandW	0.5629	0.6215	0.0401	0.4000	10
		LeadF	0.4993	0.5514	0.0476	0.3420	9
		TILES	0.4772	0.5041	0.0246	0.3299	22
		TCD2	0.7453	0.7896	0.0203	0.7814	8
		DANMF	0.647	0.7145	0.0155	0.6871	8
		MNMF	0.6382	0.7047	0.0152	0.704	10
		NNSED	0.3474	0.3834	0.032	0.2252	16
		GEMSEC	0.6116	0.6754	0.0172	0.6806	10
		EdMot	0.6413	0.708	0.0151	0.7059	10
		SCD	0.5979	0.6607	0.0171	0.5895	14
		Proposed	0.8085	0.7374	0.0116	0.779	2
Synthetic	LFR1	RandW	0.6019	0.561	0.0024	0.4422	18
		LeadF	0.6143	0.5798	0.0023	0.4753	19
		Tiles	0.5969	0.5677	0.0023	0.4404	16
		TCD2	0.6238	0.5998	0.0023	0.4724	14
		DANMF	0.5612	0.5713	0.0025	0.5642	18
		MNMF	0.7107	0.722	0.0015	0.719	10
		NNSED	0.5009	0.5088	0.0036	0.1714	12

Table 5.1 – continued from previous page

Network Type	Dataset	Algorithm	Quality Metrics Results				
			Modularity	Coverage	External Density	Average Isolability	NOC
		GEMSEC	0.6055	0.3772	0.0021	0.2527	10
		EdMot	0.7009	0.7119	0.0015	0.7068	32
		SCD	0.4753	0.4832	0.0026	0.1962	176
		Proposed	0.8123	0.8245	0.0021	0.6968	49
	LFR2	RandW	0.5666	0.5873	0.003	0.49	8
		LeadF	0.5702	0.5896	0.0029	0.5035	9
		Tiles	0.5439	0.5617	0.0029	0.4622	4
		TCD2	0.5681	0.6358	0.0026	0.5074	7
		DANMF	0.5505	0.5614	0.0027	0.5314	8
		MNMF	0.5936	0.6042	0.0023	0.5998	10
		NNSED	0.4181	0.4255	0.0042	0.2465	8
		GEMSEC	0.7215	0.7328	0.0033	0.5793	10
		EdMot	0.5956	0.6061	0.0022	0.5981	22
		SCD	0.3603	0.3662	0.0033	0.1732	182
		Proposed	0.6688	0.2759	0.0019	0.0998	45
	LFR3	RandW	0.3504	0.3568	0.0041	0.252	16
		LeadF	0.3477	0.3541	0.0037	0.2416	32
		Tiles	0.3579	0.3643	0.0037	0.262	18
		TCD2	0.3301	0.336	0.0038	0.236	15
		DANMF	0.3389	0.3458	0.0043	0.291	8
		MNMF	0.3458	0.3508	0.0039	0.35	10
		NNSED	0.3822	0.3896	0.0051	0.1312	8
		GEMSEC	0.3315	0.338	0.0017	0.1797	10
		EdMot	0.4187	0.4257	0.0035	0.3843	10
		SCD	0.1637	0.1661	0.0045	0.0801	346
		Proposed	0.5676	0.1648	0.0024	0.0905	46
	LFR4	RandW	0.3122	0.3172	0.004	0.227	6
		LeadF	0.5517	0.5591	0.0054	0.1068	105
		Tiles	0.335	0.3404	0.0039	0.2477	5

Table 5.1 – continued from previous page

Network Type	Dataset	Algorithm	Quality Metrics Results				
			Modularity	Coverage	External Density	Average Isolability	NOC
		TCD2	0.3089	0.3139	0.004	0.224	7
		DANMF	0.2488	0.2533	0.0048	0.1955	8
		MNMF	0.3419	0.3466	0.0039	0.3464	10
		NNSED	0.4131	0.42	0.0051	0.2111	5
		GEMSEC	0.361	0.3675	0.0016	0.1787	10
		EdMot	0.3364	0.3413	0.0038	0.3302	16
		SCD	0.1524	0.1545	0.0046	0.0821	346
		Proposed	0.6448	0.2257	0.0022	0.0984	30
	LFR5	RandW	0.3629	0.3693	0.0044	0.2513	13
		LeadF	0.4967	0.5029	0.0059	0.1022	105
		Tiles	0.3931	0.4002	0.0044	0.2764	12
		TCD2	0.3586	0.3649	0.0044	0.2477	11
		DANMF	0.1543	0.1571	0.0053	0.1563	8
		MNMF	0.3364	0.3412	0.004	0.3409	10
		NNSED	0.4835	0.4928	0.0053	0.2352	4
		GEMSEC	0.6816	0.6953	0.0034	0.3402	10
		EdMot	0.3445	0.3498	0.0038	0.3383	13
		SCD	0.1512	0.1533	0.0046	0.0753	353
		Proposed	0.7018	0.1803	0.0018	0.0817	44

5.3.4.2 Accuracy

To evaluate the accuracy of our proposed algorithm, a rigorous comparison among six state-of-the-art methods over the eight (three real and five synthetic datasets) well-established datasets has been performed. The result of the accuracy analysis is presented in table 5.2. From the results, it is clear that our proposed algorithm offers better accuracy without affecting the quality of clusters. For synthetic datasets (LFR1 to LFR5), our proposed algorithm achieve higher F-measure, NMI, and ARI over state-of-the-art methods. For

real datasets, its performance suffers only in the Karate dataset compared to the RandW algorithm. But for the other real-time datasets such as Dolphin and Football, our proposed algorithm achieve higher F-measure, NMI, and ARI, over the state-of-the-art methods. Overall, our proposed algorithm attains higher accuracy not only with the real datasets but also with synthetic datasets without sacrificing the cluster quality.

TABLE 5.2: Accuracy metrics results' comparison of the algorithm with state of the art algorithms

Network Type	Dataset	Algorithm	Accuracy Metrics Results			
			F-measure	NMI	ARI	NOC
Real	Karate	RandW	0.9704	0.8365	0.8823	2
		LeadF	0.1310	0.4522	0.3362	8
		TCD2	0.6257	0.1552	0.0168	2
		GEMSEC	0.3924	0.3852	0.1625	10
		EdMot	0.5684	0.4739	0.3679	4
		SCD	0.4366	0.448	0.2354	17
		Proposed	0.6914	0.5969	0.4909	2
	Dolphin	RandW	0.2222	0.4157	0.1846	9
		LeadF	0.0361	0.3226	0.0680	17
		TCD2	0.8321	0.4613	0.6820	6
		GEMSEC	0.3506	0.3217	0.1156	10
		EdMot	0.5693	0.4837	0.3443	5
		SCD	0.31	0.3678	0.1429	24
		Proposed	0.7661	0.5929	0.5794	6
	Football	RandW	0.3045	0.7292	0.5151	10
		LeadF	0.2544	0.6411	0.4306	9
		TCD2	0.2594	0.6632	0.1150	8
		GEMSEC	0.7503	0.8789	0.6978	10
		EdMot	0.8268	0.8786	0.7875	10
		SCD	0.746	0.7825	0.7168	14
		Proposed	0.869	0.8905	0.7398	8

Table 5.2 – continued from previous page

Network Type	Dataset	Algorithm	Accuracy Metrics Results			
			F-measure	NMI	ARI	NOC
Synthetic	LFR1	RandW	0.7425	0.8661	0.7297	18
		LeadF	0.7454	0.8533	0.7341	19
		TCD2	0.7186	0.8601	0.7040	14
		GEMSEC	0.7635	0.8848	0.7509	10
		EdMot	0.7542	0.815	0.7475	32
		SCD	0.6383	0.8806	0.6138	176
		Proposed	0.8982	0.959	0.8915	49
	LFR2	RandW	0.5940	0.8063	0.5717	8
		LeadF	0.5944	0.8123	0.5717	9
		TCD2	0.5935	0.8004	0.5718	7
		GEMSEC	0.5954	0.8242	0.5715	10
		EdMot	0.5917	0.7767	0.5722	22
		SCD	0.5991	0.8717	0.5709	182
		Proposed	0.6927	0.8777	0.6732	45
	LFR3	RandW	0.0841	0.4183	0.0205	16
		LeadF	0.06	0.2796	0.0012	32
		TCD2	0.0798	0.4169	0.0205	15
		GEMSEC	0.0628	0.1181	0.0039	10
		EdMot	0.0843	0.1594	0.0249	10
		SCD	0.0619	0.1172	0.003	346
		Proposed	0.1082	0.557	0.0398	46
	LFR4	RandW	0.0623	0.5012	0.0028	6
		LeadF	0.0548	0.2609	0.0006	105
		TCD2	0.0623	0.4255	0.0027	7
		GEMSEC	0.0549	0.0823	0.0008	10
		EdMot	0.0609	0.1216	0.0029	16
		SCD	0.0698	0.5144	0.005	346
		Proposed	0.0798	0.4169	0.0205	30

Table 5.2 – continued from previous page

Network Type	Dataset	Algorithm	Accuracy Metrics Results			
			F-measure	NMI	ARI	NOC
		RandW	0.0585	0.3645	0.0015	13
		LeadF	0.0542	0.2627	0.0007	105
	LFR5	TCD2	0.0569	0.3756	0.0032	11
		GEMSEC	0.0545	0.0729	0.0008	10
		EdMot	0.0575	0.0901	0.0052	13
		SCD	0.0571	0.1603	0.0024	353
		Proposed	0.0629	0.4998	0.0074	44

5.4 Conclusion and Future Work

The paper has proposed an algorithm *MDCD* for finding disjoint communities in dynamic networks along with a novel membership function for calculating the belongingness of a node to a community. *MDCD* takes $O(Ek^2)$ time in worst case. Importance of 2-hop neighbours in node's existing community label is exploited for formulating membership function. Community life cycle and its components are also discussed in the paper. Performance of *MDCD* is supported by presenting an empirical analysis in table 5.1 and 5.2. Seven metrics are used for analysis on eight datasets (3 real and 5 synthetic). Obtained results are compared with ten state-of-the-art algorithms selected from literature. Result shows that *MDCD* is more aligned towards accuracy. It also performs well for quality metrics, where it lags on few steps only.

Paper aims to identify communities in dynamic network using fuzzy membership value. There are still a few aspects left for future researchers to explore. The most obvious aspect is to further increase the performance of algorithm. As the paper focuses on disjoint communities, it is also possible to compute overlapping communities with certain changes in framework. Experiments are performed for a fixed value of θ which can be further explored.