A Leader-driven algorithm for Community Detection in Complex networks

Dr. Sowmya Kamath S
Faculty, Information Technology
National Institute of Technology, Karnataka
Surathkal, India
sowmyakamath@nitk.edu.in

Prajwal M
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
prajwalmp.192it015@nitk.edu.in

Bavya Balakrishnan
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
bavyabalakrishnan.192it003@nitk.edu.in

***Abstract*—A new trending direction to the research of community detection in large-scale complex networks is Leader-driven community detection algorithms. First we identify some particular nodes in the target network, called leader nodes, around which local communities can be computed. Through the results we demonstrate that the proposed algorithm perform better than top state of the art algorithms for community detection in complex networks. We also explore a new task-driven evaluation criteria for community detection using data clustering.**

***Index Terms*—Community detection,Leader-driven algorithms, Task-based evaluation**

I. Introduction

There is a set of topological characteristics that differentiate Large-scale complex networks from pure random graphs. They are: low degree of separation (small-world feature), power law distribution of node's degrees, and high clustering coefficient. In almost all real-world complex networks the nodes tend to be organized to modules called Communities as a result of this. A community can be defined as a connected subgraph whose nodes are related with one another than with rest of the the subgraph. As the vertices within in a community have some behavioural similarities we can understand more about the networks by uncovering and analysing their primary community structure. For example, in an e-commerce site with network of transactions, a community can be considered as set of similar customers. A community is the group of pages on the same topic in Web as a complex network.

To perform computation distribution, huge graph visualization and large-scale graph compression in real world complex networks we can efficiently make use of their underlaying community structure. [1].

Many approaches have been put foward for community detection real-world networks.

Disjoint communities detection: Partitioning the network in such a way one that one node belongs to only one community. This is the task most of the research in community detection deals with.

Overlapping communities detection: Soft clustering of the graph where a node can belongs to multiple communities at once.

Local community identification: Instead of clustering the whole graph into communities we compute the community of a given node.

Disjoint and overlapping community detection algorithms are NP-hard problems. Modularity which was initially proposed in [2] is the most popular graph partition criteria in research today. Although few critical drawbacks of approaches based on modularity optimization have been discovered by many researchers. This led to more research for other new emerging techniques for community detection such as label propagation [3] and seed-centric approaches [4].

II. Literature review

We have gone through different emerging community detection algorithms and community evaluation approaches as part of survey.

A. Community detection Algorithms

There are mainly 4 classes of Community detection Algorithms: Approaches based on Group, network, propagation and seed-centric approaches.

1) Group-based approaches: Here we identify the set of highly connected nodes that share some strong connectivity similar to the below ones,

- **High mutual connectivity:** This is an NP hard problem. We can consider maximal clique or to a γ -quasi-clique as a community. A γ -quasi-clique is a subgraph where d(G) $\leq \gamma$.
- **High internal reachability:** Here a maximal k-clique, k-club or k-core subgraph community can be considered as a community structure. A k-clique is a maximal subgraph with the diamteter $\leq$ k and k-core is a maximal connected subgraph with every node having degree $\geq$ k. In [5], researchers explore the approach of k-community where a community is given as G ' = (V ' $\subset$ V, E ' $\subset$ E) of a graph G where all u, v $\in$ V' has the restriction given as : Neighbours(v) $\subset$ Neighbours(u) $\geq$ k. All the above mentioned approaches have polynomial Computational complexity.

2) Network-based approaches: The complete connection patterns in the network is considered in this approach. Drawback of this type of algorithm is that usually the number of communities to be found should be provided as an input. The partitioning based on quality metric optimization are the most popular network-based approaches. Among various quality metric, modularity is the widely accepted one.

This is defined as follows. Let $P = \{C_1, ..., C_k\}$ a partition of the node's set V of a graph. The modularity of the partition P is given by

$$Q(P) = \sum_{c \epsilon P} e(C) - a(C)^2 \tag{1}$$

where

$$e(C) = \frac{\sum_{i \epsilon C} \sum_{j \epsilon C} A_{ij}}{2 \times m_G} \tag{2}$$

is the fraction of edges connecting vertices within the community C, and

$$a(C) = \frac{\sum_{i \epsilon C} \sum_{j \epsilon V} A_{ij}}{2.m_G} \tag{3}$$

is the fraction of edges incident to a vertex in Community C.

For improving modularity-optimisation different heuristic approaches have been discovered.

- **Agglomerative approaches:** Algorithm begins by taking each node as a community in this bottom up approach. Then, it progresses by combining some communities based on some quality metric through many iterations. eg.Louvain algorithm [6]. The algorithm has two phases. First, by local modularity-optimization it identify small communities. Second, it aggregates nodes of the communities to form a new network where each node is a community and then start merging those nodes(communities) to form new communities. If the modularity can be improved then two adjacent communities are merged. Until a threshold is reached the steps are repeated.The computing complexity of the approach is O(nlog(n)).
- **Separative approaches:** This is a top-down approach. Algorithm begins by considering the whole network as a single community. To partition the network into clusters it iterates to find the ties based on different metric. The most known algorithm of this type is Newman-Girvan algorithm [7]. The algorithm is based on the relatively high betweenness centrality shown by an inter-community tie as it would be connecting high fraction of shortest paths between nodes in different communities. At the end of every iteration we choose the tie with the highest betweenness centrality. The algorithm iterates for m times and gives an output of highest modularity. Since the computation complexity is high: $O(n^2m + (n)^3log(n))$ this is not suitable for large-scale networks.

Other methods as genetic algorithms and evolutionary algorithms can also be used for modularity optimization based approaches

Implicit assumptions made by all modularity optimization approaches are as follows:

- The partition that gives maximum modularity is the best one.
- It is possible to computes a partition with maximal modularity if a network has an underlaying community structure
- There can be structurally similar Partitions with high modularity values if a network has an underlaying community structure.

All three above-mentioned assumptions do not hold necessarily in complex networks according to recent studies.

These serious drawbacks of modularity-optimization algorithms have led to the research for more better methods.

3) Propagation-based approaches: A label lv is given to each node v $\in$ V in the graph. Synchronously all nodes update their labels to the most common label in their neighborhood in many iterations. A stable state is reached when no node change its label. Overall computation complexity is O(km) with k iterations for convergence and complexity of each iteration O(m).

Drawbacks

- Convergence to a stable state is not guaranteed
- Less robust, since due to random tie breaking different clusterings output for different runs.

4) Seed-centric approaches: We identify seed nodes around which local communities can be computed in the input network [8], [9], [10]. Algorithm is composed of 3 main steps.

1. Computing Seed nodes.
2. Computing local community for each seed.

Algorithm 2 LICOD algorithm

Require: G =< V, E > a connected graph

```
L ← ∅ {set of leaders}
for v ∈ V do
    if isLeader (v) then
        L ← L ∪ {v}
    end if
end for
C ← computeComumunitiesLeader ( L )
for v ∈ V do
    for c ∈ C do
        M[v, c] ← membership(v, c) #membership degree of a node v to a community c
    end for
    P[v] = sortAndRank(M[v]) #sorted ranklist for each vetex
end for
repeat
for v ∈ V do
        #adjust its community membership preference list by merging with pereference
        #list of neighbours
        P ∗ [v] ← rankAggregate x∈{v}∩# G (v) P[x]
        P[v] ← P ∗ [v]
end for
until Stabilization of P ∗ [v]∀v
for v ∈ V do
    /* assigning v to communities */
    for c ∈ P[v] do
        if |M[v, c] − M[v, P[0]]| ≤ epsilon then
            COM(c) ← COM(c) ∪ {v}
        end if
    end for
end for
return C
```

Fig. 1: Leader driven community detection Algorithm

3. Computing a new clustering out from the set of local communities estimated in the last step.

Algorithm 1 Basic seed-centric community detection algorithm

Input: A connected graph G with set of vertices V and set of edges E,

1: Initialize (C) to be an empty set
2: Seed ← Seed_computation(G)
3: for s ϵ Seed do
4: C s ← local_com_computation(s,G)
5: C ← C + C s //Finding set of local communities
6: end for
7: return community_computation(C)

Leader-driven algorithms is a seed-centric method. There are 2 categories to which nodes of a network are classified into: leaders and followers. Leaders are representitives of communities. Followers are assigned to most fitting communities in the assignment step. There are different node classification approaches and node assignment strategies.

In [11], an approach inspired from the K-means clustering algorithm is proposed by authors. Number of partitions is the input to algorithm. A major drawback of the approach is clearly this. Algorithm starts by randomly selecting k nodes as learders. Others are labelled as followers. Each leader node represents a community. We assign each follower node to the most nearby leader node. The follower is identified as outlier if no community is found. Algorithm forms a set of new leader nodes again by selecting the central node as a leader for each community. The process is repeated until stabilization. The quality of initially selected k leaders determines the pace of convergence. Selecting the top k nodes that have the top degree centrality and that share little common neighbors is considered as the best approach for selection of initial set.

An algorithm similar to our approach is proposed in [10] . A node with less closeness centrality than at least one neighbor can be chosen as a leader. The reciprocal of the sum of the topolgical distance from the vertex to all vertices in the graph gives closeness centrality. In decreasing order of closeness centrality the list of leaders is sorted. Then each leader is given followers that are free and not already assigned to any other leader. Any leader with no followers can be assigned to the most frequent community chosen by most of its neighbours.

B. Community evaluation approaches

Three main classes of approaches for evaluation:

1. Approach for the networks with ground-truth communities information.

2. Approach considering the topological properties of communities estimated.

3. Task-driven evaluation.

1) Ground-truth comparison approaches: Inoder to obtain networks with ground-truth community structure we have many ways.

Experts' Interpretation: Experts in the similar area of work can define the ground-truth community for some small real world networks. Different properties of the network such as size, the density, the degree distribution law, the clustering coefficient, the distribution of communities size as well as the separability of the obtained communities can be decided by user in [12]

Network generators : Constructing artificial networks for a given community structure is the idea here.

Using Implicit community structure : With the help some implicit characteristics and semantics of community structure. For example in [13] authors define a community in the Live journal social network as groups of fans of a given artist.

When a ground-truth community structure is available, we can use classical clustering evaluation indices to evaluate and compare community detection algorithms. In this work, we apply two popular indices: the Adjusted Rand Index (ARI) [14] and the Normalized Mutual Information (NMI) [15].

The ARI index is based on count of pairs of elements that are clustered in the same groups in both partitions in comparison. If there are 2 partitions of vertex set V given as P i = P i 1 , . . . , P i l , P j =P j 1 , . . . , P j k then we can divide the node pairs to exclusive groups as given below

- Set 11 = pairs that are in the same groups in P i and P j
- Set 00 = pairs that are in different groups in P i and P j
- Set 10 = pairs that are in the same groups in P i but in different groups in P j
- Set 01 = pairs that are in different groups in P i but in same groups in P j

The rand index is given by:

$$R(P_i, P_j) = \frac{2 \times (n_{11} + n_{00})}{n \times (n-1)} \tag{4}$$

Normalized difference of the Rand Index is defined as ARI

$$ARI(P_i, P_j) = \frac{\sum_{x=1}^{i} \sum_{y=1}^{k} \left(\begin{array}{c} \mid P_i^x \cap P_j^y \mid \\ 2 \end{array} \right) - t_3}{\frac{1}{2}(t_1 + t_2) - t_3} \tag{5}$$

where:

$$t_1 = \sum_{x=1}^{l} \left(\begin{array}{c} \mid P_i^x \mid \\ 2 \end{array} \right), t_2 = \sum_{y=1}^{k} \left(\begin{array}{c} \mid P_i^y \mid \\ 2 \end{array} \right), t_3 = \frac{2t_1 t_2}{n(n-1)} \tag{6}$$

This index has expected value zero for disjoint clusterings and maximum value 1 for identical clusterings.

With the given knowledge about partition Pj how successfully we cluster a randomly picked element from V in a partition Pj is quantified to mutual information.

$$H(P_i) = -\sum_{x=1}^{l} \frac{\mid P_i^x \mid}{n} log_2 \left(\frac{\mid P_i^x \mid}{n} \right) \tag{7}$$

,

the Shannon's entropy of a partition Pi

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \tag{8}$$

gives the mutual information between two random variables X , Y

In [15], authors propose a normalized version given by:

$$NMI(X, Y) = \frac{MI(X, Y)}{\sqrt{H(X)H(Y)}} \tag{9}$$

III. Methodology

A. The proposed Leader Driven community Detection

The algorithm of the proposed approach is given in (1)

Details about each of the main functions is given below.

1.Function is Leader () : Leader nodes are supposed to have higher centrality. In our experiments we have explored 3 centrality measures.

Degree centrality:This is given by the proportion of nodes directly connected to the target node

Betweenness centrality: For a given node its betweenness centrality is computed by counting fraction of shortest paths passing through the node for every other node pairs out of total possible shortest paths connecting them.

Eigenvector centrality :Tries to generalize degree centrality by incorporating the importance of the neighbours (or incoming links in directed graphs). It computes the centrality of a node as a function of the centralities of its neighbours.

If a node's centrality is greater or equal to $\sigma \in [0, 1]$ percent of its neighbors centralities it is identified as a leader. High value of σ means less number of leaders.

2.Function computecommunitiesleaders If the fraction of common members in total number of direct neighbors is above a given threshold $\delta \in [0,1]$, then two leaders are clustered to same community. Number of communities depends on σ and δ.

3.Function memebership(v, c) Degree of Membership a node v to community c is given by,

$$membership(v, c) = \frac{1}{(min_{x \epsilon COM(c)} SPath(v, x)) + 1} \tag{10}$$

Membership of all leaders of a community is 1 in that community.

4.Rank aggregation approaches

If we have a set of community membership preference list provided by nodes,each node will aggregate its list by merging

this with preference lists of its neighbors in the raph. Different strategies can be used for this aggregation. Approaches can be divided to 2 main types.

position-based approaches: Borda's Method [20]. If $L = [L_1, L_2, L_3, ..., L_k]$ is the set of ranked list then the Borda's score of an element i in list L_k is given by:

$$BL_k(i) = count(j) \mid L\_k(j) < L_k(i) \& j \in L_k. \quad (11)$$

$B(i) = \Sigma_{t=1} B_{L_t}(i)$,is the total Borda's score considering all lists order-based approaches: Kemeny optimal aggregation [21].The basic idea of all proposed approximate Kemeny aggregation is to sort the candidate list, using standard sorting algorithms based on non-transitive comparison. Si is preferred to Sj if majority of rankers prefer Si over Sj.

5.Community assignment Each node will be assigned to top-ranked communities in its final list of membership preference. Inorder to bring overlapping a node is also assigned to communities for which its membership is ϵ-different from the membership degree to the top-ranked community.

IV. Experimentation and Results

A. Evaluation on benchmark networks

We used igraph graph analysis toolkit in R to implement the algorithm.The proposed approach is evaluated on a set of four widely used benchmark networks for which a ground-truth decomposition into communities is known. These networks are the following:

Zachary's karate club This is a social network consists of 34 members of a karate club at a US university back in 1970 [16]. After the dispute between administrator and instructor of the club the club got divided into 2 communities and the instructor ended up creating a club all by himself with his set of followers.

American college football dataset This dataset contains the network of American football games [17]. Each node in the network represent teams. An edge is formed between teams playing games. Based on the frequency of the games played among them the teams are divided to 12 groups with each containing 8-12 members. This frequency of games will also depend upon the geographical distance.

American political books [22] This is a copurchasing network of online seller Amazon.com about the books related to US politiccs sold by them. Each node represents a book. Based on the co-purchasing by same buyer edges are formed between books. This has a close relation with recommendation feature on Amazon. Considering the reviews posted on Amazon about the books Mark Newman divided the books into 3 three disjoint classes such as liberal, neutral or conservative.

Dolphins social network [18]. This is an undirected social network created based on the behaviour of a community of 62 dolphins over a period of 7 years. If frequent assosications between pair of dolphins(node) is observed edges are formed between them. Based on the observations 2 communities of dolphins are formed.

TABLE I: Topological characteristics benchmark networks

Dataset	# Nodes	# Edges	# Real communities
Zachary	34	78	2
Football	115	616	12
US Politics	100	411	3
Dolphins	62	159	2

The structure of the chosen networks with Ground truth communities indicated by the color code is shown in (2). (I) gives Basic topological characteristics of selected benchmark networks.

By changing the configuration parameters as follows we applied the LdCd algorithm on each network.

- Centrality metrics = [Degree centrality (dc), Betweenness centrality (BC),Eigen Vector Centrality]
- Voting method = [Borda, Local Kemeny]
- $\sigma \in$ [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
- $\delta \in$ [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
- $\epsilon \in$ [0.0, 0.1, 0.2]

We compute the NMI, ARI and the modularity Q for each of the above given configurations. The results obtained from different configurations are shown in (II).

Variation in σ has significant impact on the results obtained. The best results are obtained for σ around 0.8, 0.9. This argues for the validity of the idea of introducing the σ threshold and not to consider extreme cases where a node is qualified as a leader if it has the highest centrality in its direct neighborhood. ϵ has negligible impact on obtained results. Increasing ϵ results in diminishing the NMI and ARI. This can be explained by the fact that high value of ϵ increases the overlapping degree of obtained communities while real communities we have here are all disjoint. Reducing δ decreased the number of communities obtained for certain datasets and give good values for NMI,ARI. Since it is not logically accepted for all complex networks we fix 0.9 and 0.5 as the best values for δ. We observe the variation of NMI, ARI and Q, for each of the possible configurations depending on the choice of the used centrality and the voting method.

In the experimets we conducted we observe that the betweenness centrality provides more convergence for towards the correct community structure to get. Though Eigen value centrality is expected to give good results it doesn't work best for the network we chose. Borda out performs Local Kemeny in all the networks.

Community structure of the four selected benchmark networks obtained by proposed LdCd by the best 2 parameter configurations are shown in (3) and (4)

We notice that the performance of each configuration differ from one network to another and this has close relation to the specialities and topology of network. The dependency between configuration parameters and speciality of network is topic to be explored deeper.

We also compared the results of our algorithm with results obtained by well-known algorithms: The Newman–Girvan algorithm [7], the WalkTrap algorithm [19] and the Louvain algorithm [6]. The configuration adopted for proposed algorithm

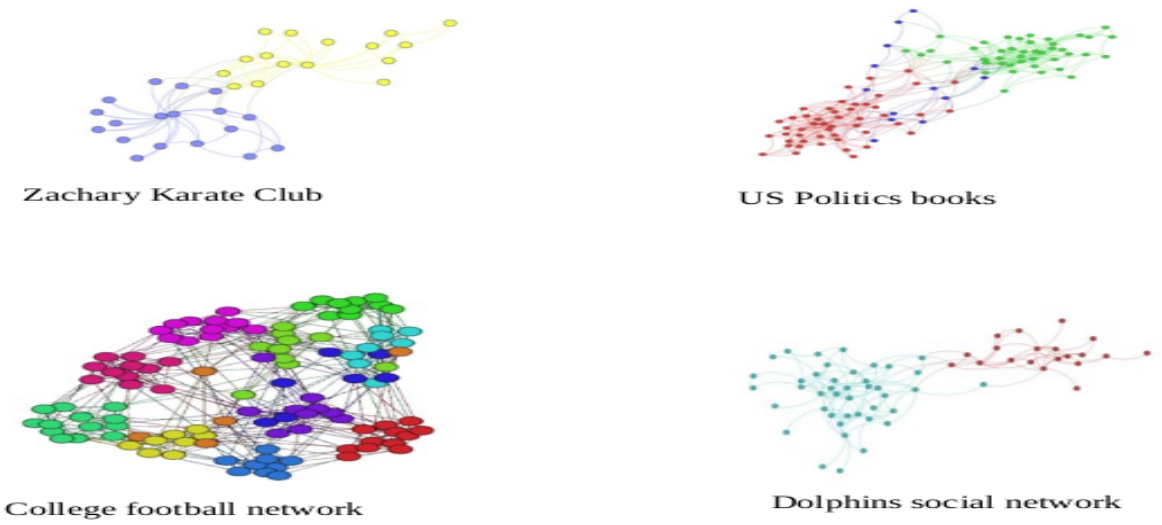

Fig. 2: Ground truth community structure of benchmark networks

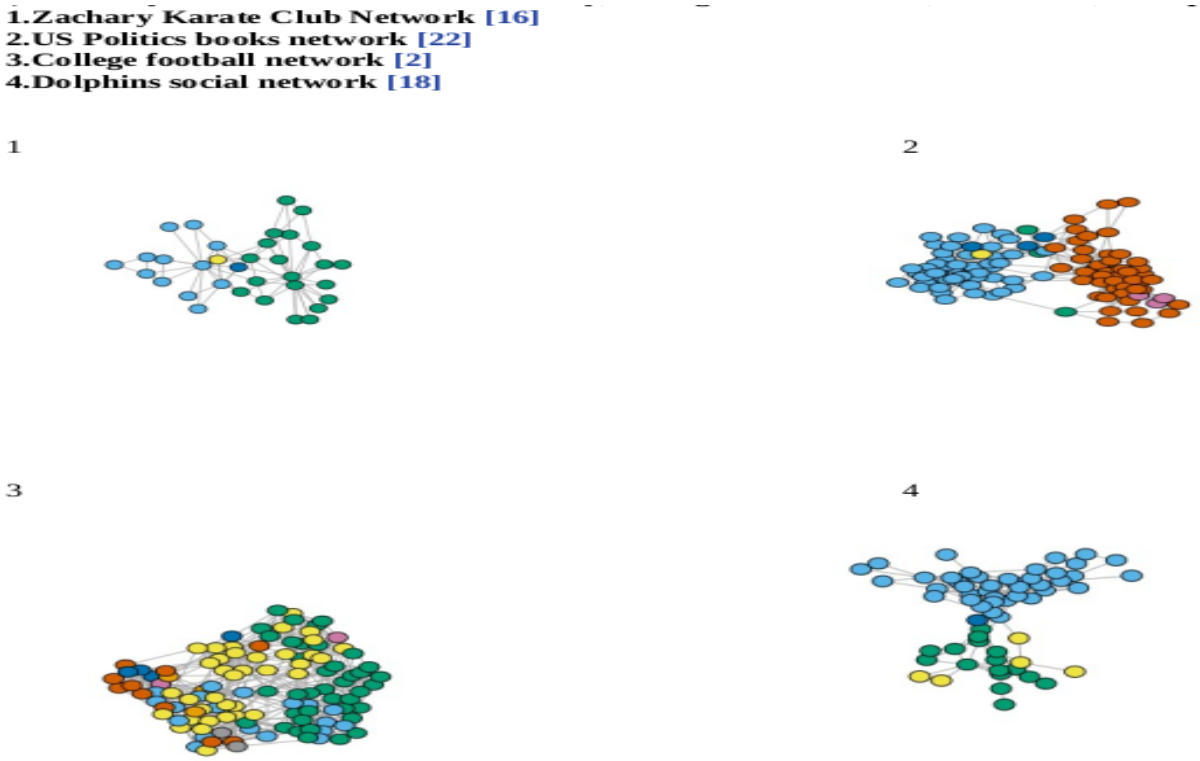

Fig. 3: Community structure of the four selected benchmark networks obtained by LdCd (Centrality metric:betweenness centrality, Voting method:Borda, $\sigma = \delta = 0.9$,and Epsilon = 0)

TABLE II: Comparison of performances of applying LdCd to different configuration parameters

Dataset	Algorithm	NMI	ARI	Modularity	#Communities
Zachary	Sigma=0.9,delta=0.9				
	LICOD(Borda_Betweenness)	0.6337324	0.6819667	0.3051446	4
	LICOD(Borda_EigenVector)	0.6337324	0.6819667	0.3051446	4
	LICOD(Borda_Degree)	0.6964905	0.745851	0.4222855	4
	LICOD(Kemeny_Betweenness)	0.6140008	0.523254	0.2635602	5
	LICOD(Kemeny_EigenVector)	0.3699431	0.2801601	0.1938692	6
	LICOD(Kemeny_Degree)	0.3699431	0.2801601	0.2048817	6
	Sigma=0.9,delta=0.5				
	LICOD(Borda_Betweenness)	0.839372	0.8279669	0.3391683	3
	LICOD(Borda_EigenVector)	0.7209632	0.7530574	0.3258547	3
	LICOD(Borda_Degree)	0.839372	0.8279669	0.3391683	3
US Politics	Sigma=0.9,delta=0.9				
	LICOD(Borda_Betweenness)	0.5181528	0.6076945	0.4228922	6
	LICOD(Borda_EigenVector)	0.5181528	0.6076945	0.4228922	6
	LICOD(Borda_Degree)	0.5135687	0.5956606	0.4222855	6
	LICOD(Kemeny_Betweenness)	0.3583441	0.2938903	0.244029	11
	LICOD(Kemeny_EigenVector)	0.3907821	0.2802048	0.1969498	17
	LICOD(Kemeny_Degree)	0.3844464	0.3493627	0.1854166	13
	Sigma=0.9,delta=0.5				
	LICOD(Borda_Betweenness)	0.5448607	0.6486401	0.4447247	4
	LICOD(Borda_EigenVector)	0.5448607	0.6486401	0.4447247	4
	LICOD(Borda_Degree)	0.5448607	0.6486401	0.4447247	4
Football	Sigma=0.9,delta=0.9				
	LICOD(Borda_Betweenness)	0.5393786	0.2122235	0.285424	14
	LICOD(Borda_EigenVector)	0.5185883	0.1987756	0.2740979	15
	LICOD(Borda_Degree)	0.5303016	0.240639	0.3555429	11
	LICOD(Kemeny_Betweenness)	0.5394264	0.1460565	0.08332247	38
	LICOD(Kemeny_EigenVector)	0.5165736	0.1244962	0.06052655	41
	LICOD(Kemeny_Degree)	0.5187954	0.156331	0.0945847	30
	Sigma=0.9,delta=0.5				
	LICOD(Borda_Betweenness)	0.5281761	0.221265	0.3019661	12
	LICOD(Borda_EigenVector)	0.4948808	0.2001475	0.3063052	13
	LICOD(Borda_Degree)	0.5479841	0.2649123	0.392897	9
Dolphins	Sigma=0.9,delta=0.9				
	LICOD(Borda_Betweenness)	0.7688508	0.8430689	0.3470393	4
	LICOD(Borda_EigenVector)	0.7688508	0.8430689	0.3470393	4
	LICOD(Borda_Degree)	0.8089225	0.8846558	0.3442506	3
	LICOD(Kemeny_Betweenness)	0.4107441	0.4223021	0.2359875	6
	LICOD(Kemeny_EigenVector)	0.6533707	0.5416906	0.2994541	5
	LICOD(Kemeny_Degree)	0.412597	0.4047125	0.2772833	6
	Sigma=0.9,delta=0.5				
	LICOD(Borda_Betweenness)	0.7688508	0.8430689	0.3470393	4
	LICOD(Borda_EigenVector)	0.7688508	0.8430689	0.3470393	4
	LICOD(Borda_Degree)	0.7174538	0.8258898	0.3517068	3

is the following: Centrality metric is betweenness centrality, Voting method is Borda, σ = 0.9 δ = 0.9/0.5 , and ϵ= 0. (III) gives obtained results on the 4 datasets by using different algorithms.

The results obtained clearly reveal that proposed algorithm outperforms other algorithms for Zachary,Dolphins and US Politics networks. We obtain competitive results in the football network as well. This could be explained by the absence of leaders in these two networks, which makes the communities detection task more difficult.

We can conclude that modularity metric does not correspond to the best decomposition into communities as measured by both NMI and ARI. To be precise, we always observe the best modularity for Louvain method(even better than the modularity of the ground-truth decomposition), however, it is not ranked first according to NMI. Parameter configurations with high values of σ gives the best results in our approach. We also tested the algorithm on a large scale Email network generated by a large European research institution. An edge is formed between person u and person v if u sent v atleast one mail. There are 1005 nodes and 25571 edges in the network. Each individual belongs to exactly one of 42 departments at the research institute. Due to topological property of graph we observed we could only obtain average performance for it in comparison to other state of the art algorithms. The results are tabulated in (IV)

B. Task Driven Evaluation using Data clustering

We explore the task driven evaluation of Community detection algorithms by converting data clustering task to community detection. There are some works already been published with the above idea [23]. The approach is as described below

Step 1. Choose some sparse classical datasets for experiment. We have selected 5 public datasets from UCI website [24]. Their properties are given in (VI) .

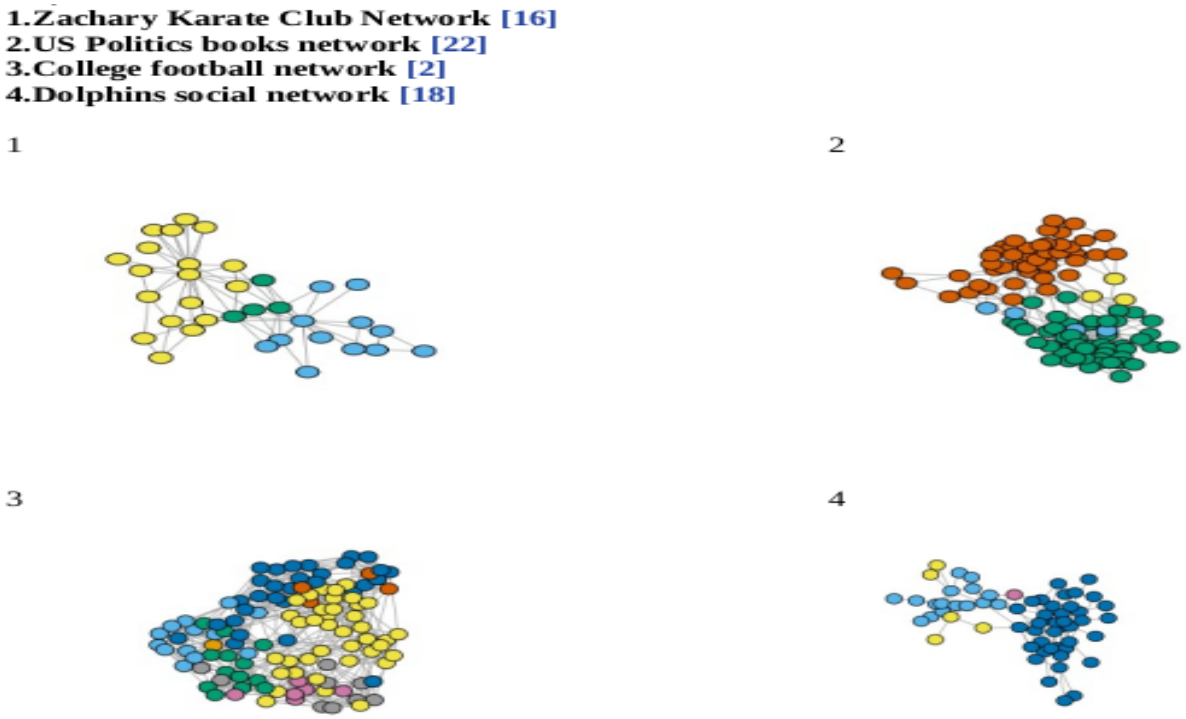

Fig. 4: Community structure of the four selected benchmark networks obtained by LdCd (Centrality metric:betweenness centrality, Voting method:Borda, σ = 0.9, δ = 0.5,and Epsilon= 0)

TABLE III: Comparison of performances of different community detection algorithms

Dataset	Algorithm	NMI	ARI	Modularity	#Communities
Zachary	Newman	0.5798278	0.4686165	0.4012985	5
	Louvain	0.5866348	0.4619069	0.4188034	4
	Walktrap	0.504178	0.3331266	0.3532216	5
	LICOD(Borda_Betweenness)	0.839372	0.8279669	0.3391683	3
US Politics	Newman	0.5584515	0.6823684	0.5168011	5
	Louvain	0.512133	0.5579848	0.5204853	4
	Walktrap	0.5427476	0.6534224	0.5069724	4
	LICOD(Borda_Betweenness)	0.5448607	0.6486401	0.4447247	4
Football	Newman	0.8788884	0.7781023	0.599629	10
	Louvain	0.8903166	0.8069409	0.6045696	10
	Walktrap	0.8873604	0.8154427	0.6029143	10
	LICOD(Borda_Betweenness)	0.5281761	0.221265	0.3019661	12
Dolphins	Newman	0.5541605	0.3949115	0.5193821	5
	Louvain	0.5108534	0.3274327	0.5185317	5
	Walktrap	0.53725	0.416739	0.4888454	4
	LICOD(Borda_Betweenness)	0.7688508	0.8430689	0.3470393	4

Step 2. Compute an n*n similarity/distance matrix for the n datapoints. We can use various measures as given in Table V for estimating distance.

Step 3. Construct Relative neighborhood graph (RNG) by connecting data items xi,xj if they satisfies the below rule.

$$d(x_i, x_j) \leq max_l\{d(x_i, x_l), d(x_j, x_l)\}, \forall l \neq i, j \qquad (12)$$

Step 4. Apply the proposed algorithm for community detection on the graph constructed. Apply other top state of the art algorithms as well.

Step 5. Using various evalution criteria compare the performances of algorithms

We have tabulated the features of obtained graphs in (VI).

Using different distance measures we constructed RNG graphs on all the selected datasets and results are shown in (VIII).

We observe some real world network characteristics such as small diameter and low density on these graphs. Unfortunately the graphs obtained by using Chebyshev distance shows high density. However, graphs has high clustering coefficient. Graphs shows high transitivity as well.

TABLE IV: Performance of LdCd on large scale network

Dataset	Algorithm	NMI	ARI	Modularity	#Communities
email-Eu-core network	Louvain	0.5361345	0.2505032	0.4211831	27
	Walktrap	0.580412	0.1974976	0.3501706	145
	LICOD(Borda_Betweenness)	0.2035174	0.01793908	0.06871058	46

Distance	Formula
Euclidean distance	$dist_{euc}(x,y) = \sqrt{\sum_{i=1}^{n} \mid xi - yi \mid^2 2}$
Cosine similarity	$dist_{cos}(x,y) = 1 - \frac{x.y}{\mid x \mid \mid y \mid}$
Chebyshev distance	$dist_{cheb}(x,y) = max_i(x_i - y_i)$

TABLE V: Distance measures used.

TABLE VI: Structural properties of used datasets

Dataset	**Glass**	**Iris**	**Wine**	**Vehicle**	**Abalone**
No of instances	214	150	178	846	4177
No of attributes	10	5	14	19	8
No of classes	7	3	3	4	29

We applied the community detection algorithms on the graphs obtained by cosine distance as they are relatively closer to real world networks. Results obtained on all 5 RNG graphs applying 4 different community detection algorithms are evaluated using NMI,ARI and modularity and they are tabulated in (VII). The actual class information for each data item is already available in the dataset. We observe that modularity is not a good criteria for quality measurement. The highlighted values indicate the better performance of LdCd over other algorithms. For wine and Abalone the LICOD outperforms other algorithms. It shows good results for other datasets as well.

V. Conclusion and Future work

We contribute to the state of the art on community detection in complex networks by providing an efficient leader driven community detection algorithm. Its capacity to detect communities is demonstrated by small benchmark social networks. We also proposed a new task-driven community evaluation approach using data clustering. As a future development to the proposed approach we think about improving the speed of algorithm to perform better in multiplex and bipartite networks.

References

[1] Grabowski, S., Bieniecki, W.: Tight and simple web graph compression. CoRR, abs/1006.0 (2010)

[2] Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS 99(12), 7821–7826 (2002)

[3] Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorith to detect community structures in large-scale networks. Phys. Rev.E 76, 1–12 (2007)

[4] Kanawati, R.. Seed-centric approaches for community detection in complex networks. In: Meiselwitz, G., (ed.) 6th International Conference on Social Computing and Social Media, volume LNCS 8531, pp. 197–208, Crete, Greece. Springer, New York (2014)

[5] Verma, A., Butenko, S.: Graph partitioning and graph clustering. In: Bader, D.A., Meyerhenke, H., Sanders, P., Wagner, D. (eds.) Network clustering via clique relaxations: a community based approach. Contemporary Mathematics, pp. 129–140. American Mathematical Society, Providence (2012)

[6] Blondel, V.D., Guillaume, J.-L., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. 2008, P10008 (2008)

[7] Newman, M.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69, 02613:1–022613:15 (2004)

[8] Kanawati, R.: LICOD: Leaders identification for community detection in complex networks. In: SocialCom/PASSAT, pp. 577–582 (2011)

[9] Papadopoulos, S., Kompatsiaris, Y., Vakali, A.: A graph-based clustering scheme for identifying related tags in folksonomies. In: DaWak, pp. 65–76 (2010)

[10] Shah, D., Zaman, T.: Community detection in networks: The leader-follower algorithm. In: Workshop on Networks Across Disciplines in Theory and Applications, NIPS (2010)

[11] Khorasgani, R.R., Chen, J., Zaiane, O.R.: Top leaders community detection approach in information networks. In: 4th SNA-KDD Workshop on Social Network Mining and Analysis, Washington D.C. (2010)

[12] Lancichinetti, A., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E 4, 046110 (2008)

[13] Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. In: Zaki, M.J., Siebes, A., Yu, J.X., Goethals, B., Webb, G.I., Wu, X. (eds.) ICDM, pp 745–754. IEEE Computer Society (2012)

[14] Hubert, L., Arabie, P.: Comparing partitions. J. Classif. 2(1), 192–218 (1985)

[15] Strehl, A., Ghosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617 (2003)

[16] Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. 33, 452–473 (1977)

[17] Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS 99(12), 7821–7826 (2002)

[18] Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav. Ecol. Sociobiol. 54, 396–405 (2003)

[19] Pons, P., Latapy, M.: Computing communities in large networks using random walks. J. Graph Algorithms Appl. 10(2), 191–218 (2006)

[20] Borda, J.C.: Mémoire sur les élections au scrutin. Comptes rendus de l'Académie des sciences, traduit par Alfred de Grazia comme Mathematical Derivation of a election system, Isis, vol. 44, pp. 42–51 (1781)

[21] Kemeny, J.G.: Mathematics without numbers. Daedalus 88, 571–591 (1959)

[22] Krebs, V. Political books network. http://www.orgnet.com

[23] de Oliveira, T.B.S., Zhao, L., Faceli, K., de Carvalho, A.C.P.L.F.:Data clustering based on complex network community detection.In: IEEE Congress on Evolutionary Computation, 2008 (CEC2008), 1-6 June 2008, Hong Kong, pp. 2121–2126 (2008)

[24] http://archive.ics.uci.edu/ml/datasets.html.

TABLE VII: Performance comparison of LdCd with Louvain, Walktrap, Newman–Girvan algorithms

Dataset	Algorithm	NMI	ARI	Modularity	No of communities
Iris	Newman	0.6894212	0.4709471	0.7228967	8
	Louvain	0.5890717	0.3748021	0.7187097	9
	Walktrap	0.6562614	0.4987374	0.6972523	12
	LICOD	0.6565191	0.5399218	0.4503184	2
Glass	Newman	0.4575755	0.2138958	0.7670461	11
	Louvain	0.476197	0.217018	0.7563275	12
	Walktrap	0.4179286	0.1384216	0.7403119	16
	LICOD	0.4597846	0.3052357	0.3798834	5
Wine	Newman				
	Louvain	0.3159939	0.1362786	0.7944163	12
	Walktrap	0.3017275	0.09450665	0.7867851	17
	LICOD	0.3683113	0.3543734	0.4127416	2
Vehicle	Newman	0.2383895	0.11754	0.796356	14
	Louvain	0.2331711	0.1143421	0.7868221	14
	Walktrap	0.2467031	0.1158469	0.7698765	16
	LICOD	0.1360652	0.07975364	0.4266861	2

TABLE VIII: Topological properties of RNG graphs

Dataset	Feature	Euclidean	Chebyshev	Cosine
Iris	**# Edges**	404	2476	442
	Diameter	33	14	23
	Average degree	5.38666666666667	33.0133333333333	5.89333333333333
	Density	0.036152125279642	0.221565995525727	0.039552572706935
	Transitivity	0.08252688172043	0.348853912013634	0.033080808080808
Glass	**# Edges**	558	7786	552
	Diameter	21	8	24
	Average degree	5.21495327102804	72.7663551401869	5.1588785046729
	Density	0.024483348690273	0.341626080470361	0.024220086876399
	Transitivity	0.013966480446927	0.252269101340897	0.011970534069982
Wine	**# Edges**	380	514	438
	Diameter	102	84	59
	Average degree	4.26966292134831	5.7752808988764	4.92134831460674
	Density	0.024122389386149	0.03262870564337	0.02780422776614
	Transitivity	0	0.178455284552845	0
Vehicle	**# Edges**	2602	4072	2774
	Diameter	63	54	45
	Average degree	6.15130023640662	9.62647754137116	6.55791962174941
	Density	0.007279645250185	0.011392281114049	0.007760851623372
	Transitivity	0.004284490145673	0.091105407372458	0