A Leader-driven algorithm for community detection in complex networks

Bavya Balakrishnan
Prajwal MP

Abstract: A new trend in community detection in large-scale complex networks is Leader-driven community detection algorithms. Identifying some particular nodes in the target network, called leader nodes, around which local communities can be computed is the basic idea here. Through the results we demonstrate that our approach outperforms top state of the art algorithms for community detection in complex networks.

Introduction

Large-scale complex networks share a set of non-trivial topological characteristics that distinguish them from pure random graphs. They are: low separation degree (or what is better known as small-world feature), power law distribution of node's degrees, and high clustering coefficient. Almost all real-world complex networks exhibit an organization called Communities as a consequence of this. A community is loosely defined as a connected subgraph whose nodes are much linked with one another than with nodes outside the subgraph. Nodes in a community within the network are generally supposed to share common properties or play similar roles. This suggests that we can understand more about the complex networked systems by discovering and examining their underlaying communities. For example,in a network of transactions in an e-commerce site, this would express a set of similar customers. A community would be a set of pages dealing with a same topic in Web as a complex network.

Since he community-level structure is exhibited by almost all studied real-world complex networks,to implement a pre-treatment step for a number of general complex operations such as computation distribution, huge graph visualization and large-scale graph compression an efficient approach for detecting communities would be useful[1].

Many algorithms have been proposed for detecting communities in complex networks.

Disjoint communities detection:
Computing a partition of the graph node's set is the goal here. One node can be a part of only one community. This is the problem most of the work in the area of community detection deals with.

Overlapping communities detection:
Computing soft clustering of the graph node's set is the goal where a node can belongs to several communities at once.

Local community identification:
Here we compute the community of a given node rather than partitioning the whole graph into communities.

Both problems, disjoint and overlapping community detection are NP-hard.The most applied graph partition criteria are the modularity initially introduced in [2]. Some serious limitations of modularity optimization-based approaches have been pointed out by many researchers. This boosted the research for alternative approaches for community detection.Emergent approaches include label propagation approaches [3] and seed-centric ones [4].

Literature Survey

We provide a brief survey on various community detection algorithms and community evaluation approaches.

1.Community detection approaches
We classify existing approaches into four classes: Group-based approaches, network-based approaches, propagation-based approaches and seed-centric ones.

1.1 Group-based approaches
These are approaches based on identifying groups of nodes that are highly connected or share some strong connection patterns like the following ones,

- High mutual connectivity:
  A community can be approximated to a maximal clique or to a γ -quasi-clique. A subgraph G is said to be γ -quasi-clique if d(G) ≤ γ . This is a NP hard problem.
- High internal reachability:
  A community core can be approximated by a maximal k-clique, k-club or k-core subgraph. A k-clique (resp.k-club) is a maximal subgraph in which the longest shortest path between any nodes (resp. the diameter) is ≤k. A maximal connected subgraph in which each node has a degree ≥k is a k-core. In [5], Authors introduce the method of k-community which is defined as a connected subgraph G ‘ = (V ‘ ⊂ V, E ‘ ⊂ E) of a graph G in which for every pair of nodes u, v∈V’ the following constraint holds: Neighbours (v)∩ Neighbours (u)| ≥ k. Computational complexity is polynomial.

1.2 Network-based approaches
The whole connection patterns in the network is considered in this approach. Drawback of this type of algorithm is that usually the number of clusters to be found should be provided as an input. Those based on optimizing a quality metric of graph partition are the most popular network-based approaches. Among various quality metric, modularity is the most widely used 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

$$Q(\mathcal{P}) = \sum_{c\in\mathcal{P}} e(\mathcal{C}) - a(\mathcal{C})^2$$

where

$e(\mathcal{C}) = \frac{\sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{C}} A_{ij}}{2\times m_G}$ is the fraction of links inside the community C, and

$a(\mathcal{C}) = \frac{\sum_{i\in\mathcal{C}}\sum_{j\in V} A_{ij}}{2.m_G}$ is the fraction of links incident to a node in C.

For computing partitions that maximize the modularity different heuristic approaches have been proposed.

- Agglomerative approaches:
  Algorithm starts by considering each single node as a community in this bottom up approach. Then, it iterates by merging some communities based on some quality criteria. eg.Louvain algorithm [6]. The algorithm is composed of two phases. First, by optimizing modularity in a local way it looks for small communities. Second, it builds a new network whose nodes are the communities by aggregating nodes of the same community . If the overall modularity of the obtained partition can be enhanced then two adjacent communities are merged. Until a maximum of modularity is reached the steps are repeated iteratively.The computing complexity of the approach is evaluated to be O(nlog(n)).

- Separative approaches:
  This is a top-down approach, where an algorithm starts by considering the whole network as a community. To split the network into communities it iterates to find the ties based on different criteria. The most known representative of this class of approaches is Newman Girvan algorithm[7]. The algorithm is based on the simple idea that a tie linking two communities should have a high betweenness centrality since an inter-community tie would be traversed by a high fraction of shortest paths between nodes belonging to these different communities. Cutting at each iteration the tie with the highest betweenness centrality, the algorithm iterates for m times. Output will be the partition of highest modularity. Since the computation complexity is high: $O(n^2 \cdot m + (n)^3 \log(n))$ this is prohibitive to apply to large-scale networks.

Other classical optimization approaches such as genetic algorithms and evolutionary algorithms can also be used for modularity optimizations.

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

- The best partition of a graph is the one that maximize the modularity.
- It is possible to find a precise partition with maximal modularity if a network has a community structure
- Partitions inducing high modularity values are structurally similar if a network has a community structure.

All three above-mentioned assumptions do not hold necessarily in complex networks according to recent studies.
These serious drawbacks of modularity-guided algorithms have boosted the research for alternative approaches.

1.3 Propagation-based approaches
The underlaying idea is simple: each node $v \in V$ in the network is assigned a specific label l_v . In a synchronous way all nodes update their labels by selecting the most frequent label in the direct neighborhood. Until reaching a stable state where no more nodes change their labels the algorithm iterates.The complexity of each iteration is O(m). Hence, with k the number of iterations before convergence, overall computation complexity is O(km) .

Drawbacks
- Convergence to a stable state is not guaranteed
- Lastly, it lacks for robustness, since due to random tie breaking different partitions are produced by different runs.

1.4 Seed-centric approaches
Identifying some particular nodes in the target network, called seed nodes, around which local communities can be computed is the basic idea here[8,9,10]. Three principle steps are as given below.
1. Seed computation.
2. Seed local community computation.
3. Community computation out from the set of local communities computed in the previous step.

Algorithm 1 General seed-centric community detection algorithm
Require: G =< V, E > a connected graph,
1: C ← ∅
2: S ← compute_seeds(G)
3: for s ∈ S do
4: C s ← compute_local_com(s,G)
5: C ← C + C s
6: end for
7: return compute_community(C)

Leader-driven algorithms constitute a special case of seed-centric approaches. There are 2 overlapping categories to which nodes of a network are classified into: leaders and followers. Leaders represent communities. Followers nodes are assigned to most relevant communities in the assignment step. Different algorithm follow different node classification approaches and different node assignment strategies.

In [11], an approach directly inspired from the K-means clustering algorithm is proposed by authors. Number of communities to identify is the input to algorithm. A major disadvantage of the approach is clearly this. Randomly select k nodes. Others are labelled as followers. Each leader node represents a community. We assign each follower node to the most nearby leader node. Different levels of neighborhood are allowed. The follower node is labeled as outlier if no nearby leader is found. Algorithm compute a set of new leader nodes again. The most central node is selected as a leader for each community. The process is repeated until stabilization. The quality of initially selected k leaders determines the speed 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] . Any node whose closeness centrality is less than at least one of its neighbors becomes a leader. The inverse of the average distance to all other nodes in the network gives the closeness centrality of a node v. In decreasing order of closeness centrality the list of leaders is sorted. Then each leader is assigned direct followers that are not already assigned to another leader. Leaders that are not followed by any node are assigned to the community to which the majority of its direct neighbors belong.

2.Community evaluation approaches

Three main types of existing approaches:
1. Evaluation on networks for which a ground-truth decomposition into communities is known.
2. Evaluation in function of the topological features of computed communities.
3. Task-driven evaluation.

2.1 Ground-truth comparison approaches

Networks with ground-truth partitions can be obtained by one of the following ways:

Annotation by experts: For some networks representing real systems, experts in the system field have been able to define the community structure. These networks are generally small allowing them to be handled by experts. Different parameters of the network including the 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 use: Generate artificial networks with predefined community structure is the idea here.

Implicit community definition : Infer the community structure in a graph applying simple rules taking usually the semantic of ties into account. 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 external clustering evaluation indices to evaluate and compare community detection algorithms. In this work, we apply two widely used indices: the Adjusted Rand Index (ARI) [14] and the Normalized Mutual Information (NMI) [15].

The ARI index is based on counting the number of pairs of elements that are clustered in the same clusters in both compared partitions. Let $P_i = \{P_{i_1}, \ldots, P_{i_l}\}$, $P_j = \{P_{j_1}, \ldots, P_{j_k}\}$ be two partitions of a set of nodes V . The set of all (unordered) pairs of nodes of V can be partitioned into the following four disjoint sets:

• S_{11} = {pairs that are in the same cluster under P_i and P_j }

• S_{00} = {pairs that are in different clusters under P_i and P_j }

• S_{10} = {pairs that are in the same cluster under P_i but in different ones under P_j }

• S_{01} = {pairs that are in different clusters under P_i but in

The rand index is given by:

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

ARI is the normalized difference of the Rand Index.

$$\mathrm{ARI}(P_i, P_j) = \frac{\sum_{x=1}^{l} \sum_{y=1}^{k} \binom{|P_i^x \cap P_j^y|}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3}$$

where:

$$t_1 = \sum_{x=1}^{l} \binom{|P_i^x|}{2}, \quad t_2 = \sum_{y=1}^{k} \binom{|P_j^y|}{2}, \quad t_3 = \frac{2 t_1 t_2}{n(n-1)}$$

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

Based on the notion of mutual information there exist another family of partitions comparisons functions.We seek to quantify how much we reduce the uncertainty of the clustering of randomly picked element from V in a partition Pj if we know Pi .

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

,

is the Shannon’s entropy of a partition Pi

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

gives the mutual information between two random variables X , Y

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

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

The LICOD approach

Community composed of 2 types of nodes Leaders and Followers. We used igraph graph analysis toolkit in R to implement the algorithm.

Algorithm 2 LICOD algorithm

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

$\mathcal{L} \leftarrow \emptyset$ {set of leaders}
for $v \in V$ **do**
  if $isLeader(v)$ **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{v\}$
  end if
end for
$\mathcal{C} \leftarrow computeComumunitiesLeader(\mathcal{L})$
for $v \in V$ **do**
  for $c \in \mathcal{C}$ **do**
    $M[v,c] \leftarrow membership(v,c)$ {see equation 6}
  end for
  $P[v] = \textbf{sortAndRank}(M[v])$
end for
repeat
  for $v \in V$ **do**
    $P^*[v] \leftarrow \textbf{rankAggregate}_{x \in \{v\} \cap \Gamma_G(v)} \mathbf{P[x]}$
    $P[v] \leftarrow P^*[v]$
  end for
until Stabilization of $P^*[v] \forall v$
for $v \in V$ **do**
  / assigning v to communities */*
  for $c \in P[v]$ **do**
    if $|M[v,c] - M[v, P[0]]| \leq \epsilon$ **then**
      $COM(c) \leftarrow COM(c) \cup \{v\}$
    end if
  end for
end for
return $\mathcal{C}$

Details about each of the main functions is given below.

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

Degree centrality This is given by the proportion of nodes directly connected to the target node
Betweenness centrality For every pair of the network, count how many times a node can interrupt the shortest paths between the two nodes in the pair.
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 σ ∈ [0, 1] percent of its neighbors centralities it is identified as a leader. High value of σ means fewer leaders.

2.Function computecommunitiesleaders

If the ratio of common neighbors to the total number of neighbors is above a given threshold δ ∈ [0,1], then two leaders are grouped to same community. So σ and δ determines the number of communities.

3.Function memebership(v, c)

Membership degree of a node v to community c is given by,

$$membership(v, c) = \frac{1}{(min_{x \in COM(c)} SPath(v, x)) + 1} \quad (6)$$

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

4.Rank aggregation approaches

If we have a set of ranklists provided by nodes,each node will adjust its community membership preference list by merging this with preference lists of its direct neighbors in the network. Different strategies can be used for this aggregation. Approaches can be divided to 2 types.

position-based approaches: Borda's Method [20]. For a set of complete ranked lists L = [L_1 , L_2 , L_3 , . . . , L_k], the Borda's score of an element i and a list L_k is given by:

BL_k (i) = {count(j)|L_k (j) <L_k (i) & j ∈ L_k }.

B(i) = $\Sigma_{t=1}$ Bl_t(i) ,is the total Borda's score

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, but using a non-transitive comparison relationship between candidates. Si is preferred to Sj if majority of rankers rank Si before Sj.

5.Community assignment

Each node will be assigned to top-ranked communities in its final obtained membership preference list. A node is also assigned to communities for which its membership is epsilon-far from the membership degree to the top-ranked community. This parameter decides the degree of overlapping.

Experimentation

Evaluation on benchmark networks

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 network is a social network of friendships between 34 members of a karate club at a US university in 1970 [16]. Following a dispute, the network was divided into two groups between the club's administrator and the club's instructor. The dispute ended in the instructor creating his own club and taking about half of the initial club with him. The network can hence be divided into two main communities.

American college football dataset This dataset contains the network of American football games [17]. The 115 nodes represent teams and the edges represent games between 2 teams. The teams are divided into 12 groups containing around 8–12 teams each and games are more frequent between members of the same group. Also teams that are geographically close but belong to different groups are more likely to play one another than teams separated by a large distance. Therefore, in this dataset groups can be considered as known communities.

American political books [22]This is a political books copurchasing network. Nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the “customers who bought this book also bought these other books” feature on Amazon. Books are classified into three disjoint classes: liberal, neutral or conservative. The classification was made separately by Mark Newman based on a reading of the descriptions and reviews of the books posted on Amazon

Dolphins social network This network is an undirected social network resulting from observations of a community of 62 dolphins over a period of 7 years [18]. Nodes represent dolphins and edges represent frequent associations between dolphin pairs occurring more often than expected by chance. Analysis of the data revealed two main groups.

Table 1 Basic topological characteristics of selected 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 selected networks with real communities indicated by the color code is shown in Fig 1. Table 1 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 Table 2.

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.Epsilon has negligible impact on obtained results. Increasing Epsilon results in diminishing the NMI and ARI. This can be explained by the fact that high value of Epsilon 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.

These results show that the use of the betweenness centrality accelerate slightly the convergence for the right value to obtain. 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.

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 choice of configuration as a property of target network constitutes one interesting topic to cope with.

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 is the following: Centrality metric is betweenness centrality, Voting method is Borda, $\sigma = 0.9$ $\delta = 0.9/0.5$, and epsilon = 0.Table 3 gives obtained results on the three datasets.

These results show that LICOD performs better than the other algorithms for both Zachary and US Politics networks. It also gives competitive results in the other two networks. 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.

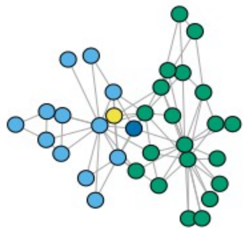
Table 2 Comparison of performances of applying LICOD to different configuration parameters

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

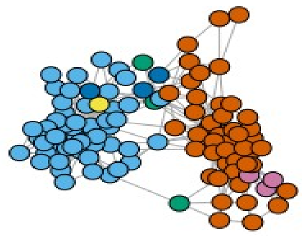
Bold values indicate best values of LICOD

Fig. 1 Community structure of the four selected benchmark networks obtained by LICOD (Centrality metric:betweenness centrality, Voting method:Borda, σ = δ = 0.9,and Epsilon = 0)
1.Zachary Karate Club Network [16]
2.US Politics books network [22]
3.College football network [2]
4.Dolphins social network [18]

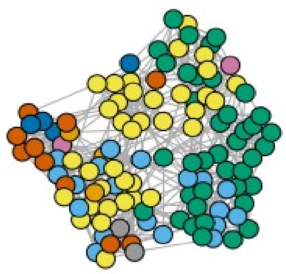
1



2



3



4

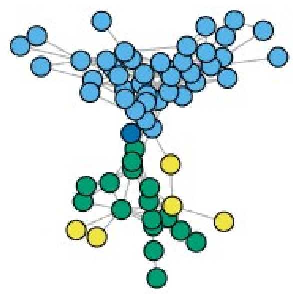


Fig. 2 Community structure of the four selected benchmark networks obtained by LICOD (Centrality metric:betweenness centrality, Voting method:Borda, σ = 0.9, δ = 0.5,and Epsilon = 0)

1.Zachary Karate Club Network [16]

2.US Politics books network [22]

3.College football network [2]

4.Dolphins social network [18]

1 2

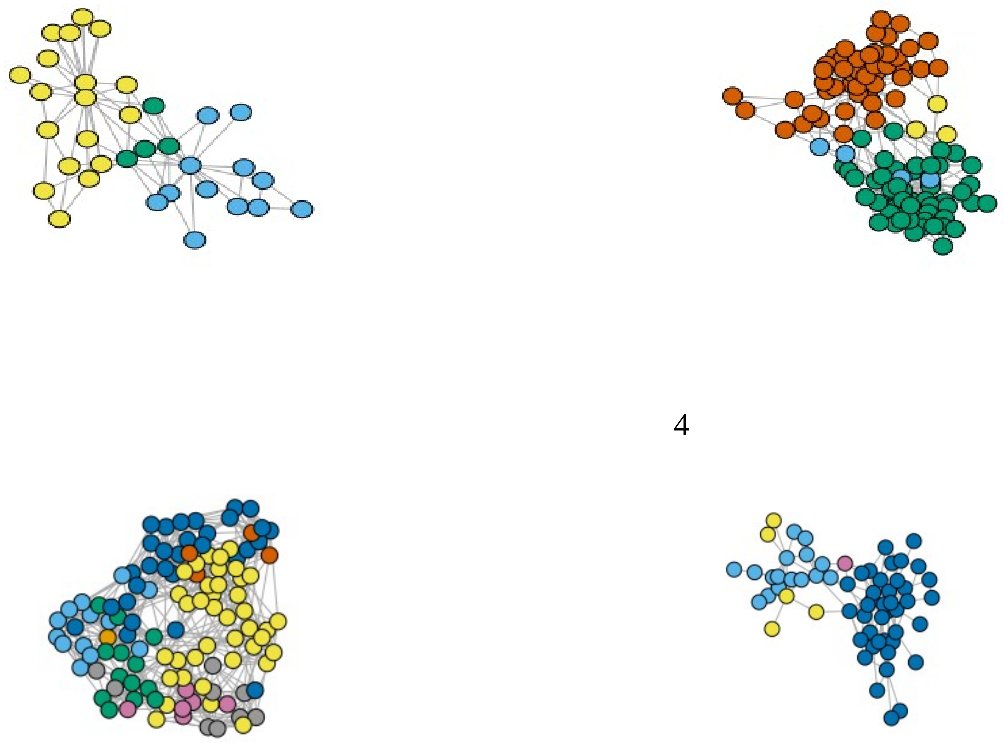


Table 3 Comparison of performances of different community detection algorithms

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

Bold values indicate the best score by LICOD

Conclusion

We contribute to the state of the art on community detection in complex networks by providing an efficient community detection algorithm. Its capacity to detect communities is demonstrated by small benchmark social 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