Department of IT, NITK Surathkal [Dec-Jun 2020]

WSC (IT752) - Mini-project Evaluation - I

Leader-driven community detection in complex networks

Bavya Balakrishnan -192IT003
Prajwal M P -192IT015

16-Apr-2020

Paper chosen for implementation

LICOD: A Leader-driven algorithm for community detection in complex networks

Zied Yakoubi · Rushed Kanawati

Agenda

- Abstract
- Introduction
- Literature Survey
- Methodology
- Experimentation and Results
- Conclusion and Future Work
- Reference

Abstract

- Leader-driven community detection algorithms (LdCD) is a new trend in community detection in large scale complex networks.
- The basic idea is to identify leader nodes, around which local communities can be computed.
- In this project we describe a framework for implementing LdCD algorithm
- We also evaluate the performance of community detection algorithms against LdCD
- The results we obtained show that LdCD outperforms other algorithms

Introduction

- Complex Networks share a set of non-trivial characteristics that distinguish them from pure random graphs such as
  - Low separation degree (or what is better known as small-world feature)
  - Power law distribution of node's degrees
  - High clustering coefficient
- Real world complex networks exhibit a level of organization, called communities.
- A community is defined as a connected subgraph whose nodes are much linked with one each other than with nodes outside the subgraph.

  eg. In web as a complex network, a community would be a set of pages dealing with a same topic

- Useful in computation distribution,huge graph visualization and large-scale graph compression
- Different types of community detection algorithms are there:
  - *Disjoint communities detection*: compute a partition of the graph node's set where one node can belong to only one community at once
  - *Overlapping communities detection* : A node can belongs to several communities at once
  - *Local community identification* : to compute the community of a given node rather than partitioning the whole graph into communities

Introduction

- Disjoint and overlapping community detection problems are NP-hard
- Most applied graph partitioning criteria is the modularity.
- Serious limitations of modularity optimization-based approaches boosted the alternative approaches
- Emergent approaches include label propagation approaches and seed-centric ones
- The basic idea of seed-centric approaches is to select a set of nodes (i.e. seeds) around which communities are constructed.
- Special case of seeds is to select nodes that are likely to act as leaders of their communities
- since LdCD algorithms are not based on maximizing an objective function (i.e. the modularity),we use Task-oriented evaluation.
- We implement a method to transform classical clustering benchmarks into a community detection problem

Literature Review

Brief survey on community detection approaches and Community evaluation approaches.

Community detection approaches

1. Group-based approaches
2. Network-based approaches
3. Propagation-based approaches
4. seed-centric ones.

Literature Review

Authors	Methodology	Advantages	Limitations
Tsironis, S., Sozio, M., Vazirgiannis, M	Using Network based approach considering whole connection patterns in the network.	Spectral clustering approaches and hierarchical clustering approaches can be used.	• Number of clusters to be found should be provided as an input for the algorithm. • High computation complexity which might be cubic on the size of the input dataset • All modularity optimization approaches make implicit assumptions which do not hold,
Raghavan, U.N., Albert, R., Kumara, S.	Label propagation approach.All nodes update in a synchronous way their assigned labels by selecting the most frequent label in the direct neighborhood.	A low complexity incremental approaches for community detection	number of iterations grows in a logarithmic way with the growth of target network size
Ovelgönne, M.: ASONAM. In: Rokne, J.G., Faloutsos, C.	distributed label propagation	Fast algorithm handling very large-scale networks	• First, there is no formal guarantee of the convergence to a stable state. • Lastly, it lacks for robustness, since different runs produce different partitions due to random tie breaking.

Literature Review

Authors	Methodology	Advantages	Limitations
Cordasco, G., Gargano, L	Asynchronous, and semi-synchronous label updating	hinder the problem of oscillation and improve convergence conditions	• these approaches harden the parallelization of the algorithm by creating dependencies among nodes • they increase the randomness in the algorithm making the robustness even worse.
Khorasgani, R.R., Chen, J., Zaiane, O.R	inspired from the K-means clustering algorithm Leader driven seed centric approach	Most relevant communities are obtained	• algorithm requires as input the number k of communities to identify. • The convergence speed depends on the quality of initially selected k leaders.

Literature Review

- All modularity optimization approaches make implicitly the following assumptions:
  - 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.
- Leader-driven algorithms constitute a special case of seed-centric approaches.

Literature Review

Community evaluation approaches

Existing approaches can be divided into three main types

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

Ground-truth comparison approaches

Annotation by experts : experts in the system field have been able to define the community structure

Network generators : generate artificial networks with predefined community structure.

Implicit community definition : based on semantics

Different clustering evaluation methods

- **Adjusted Rand Index (ARI)** : This index has expected value zero for independent clusterings and maximum value 1 for identical clusterings.
- **Normalized Mutual Information (NMI) :**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

Methodology

Seed-centric approaches

- The basic idea underlying seed-centric approaches is to identify some particular nodes in the target network, called seed nodes, around which local communities can be computed

Algorithm 1 General seed-centric community detection algorithm

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

$\mathcal{C} \leftarrow \emptyset$
$S \leftarrow$ **compute_seeds(G)**
for $s \in S$ **do**
  $C_s \leftarrow$ **compute_local_com(s,G)**
  $\mathcal{C} \leftarrow \mathcal{C} + C_s$
end for
return compute_community($\mathcal{C}$)

- Leader-driven algorithms constitute a special case of seed centric approaches
- Nodes of a network are classified into two (eventually overlapping) categories: leaders and followers.

Methodology

- Leaders represent communities.

Algorithm 2 LICOD algorithm

```
Require: G =< V, E > a connected graph
L ← ∅ {set of leaders}
for v ∈ V do
        if is Leader (v) then
                L ← L ∪ {v}
        end if
end for
C ← computeComumunities Leader ( 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
```

Methodology

- **Algorithm is implemented using the igraph graph analysis toolkit**

Function is Leader () :

Based on nodes centralities

- Degree centrality
- Betweenness centrality
- Eigenvector centrality

A node is identified as a leader if its centrality is greater or equal to $\sigma \in [0, 1]$ percent of its neighbors centralities.

Function computeCommunitiesLeaders

Two leaders are grouped in the same community if the ratio of common neighbors to the total number of neighbors is above a given threshold $\delta \in [0, 1]$.

Methodology

Function membership(v, c)

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

Rank aggregation approaches

- Requirement:minimum number of pairwise disagreements
- Borda's method(position-based approach)

$B_{L_k}(i) = \{\text{count}(j) | L_k(j) < L_k(i) \,\&\, j \in L_k\}$. The total Borda's score for an element is then: $B(i) = \sum_{t=1}^{k} B_{L_t}(i)$.

- Kemeny optimal aggregation(order based approach)

si is preferred to sj ,if the majority of rankers ranks si before sj

Community assignment

- Each node will be assigned to top-ranked communities in its final obtained membership preference list.
- Threshold epsilon controls the degree of desired overlapping

Proposed enhancements/novelty

- For each network we apply the proposed algorithm by changing the configuration parameters.
- We apply new centrality measure Eigenvector centrality for leader identification and analyse its impact in the proposed algorithm.
- Try with different approximate Kemeny aggregation approaches for rank aggregation
- Test the algorithm with different rank aggregation approaches
- Test the algorithm algorithm on large-scale networks

Experimentation and Results

Datasets

The proposed approach is evaluated on a set of four widely used benchmark networks for which a ground-truth decomposition into communities is known.

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

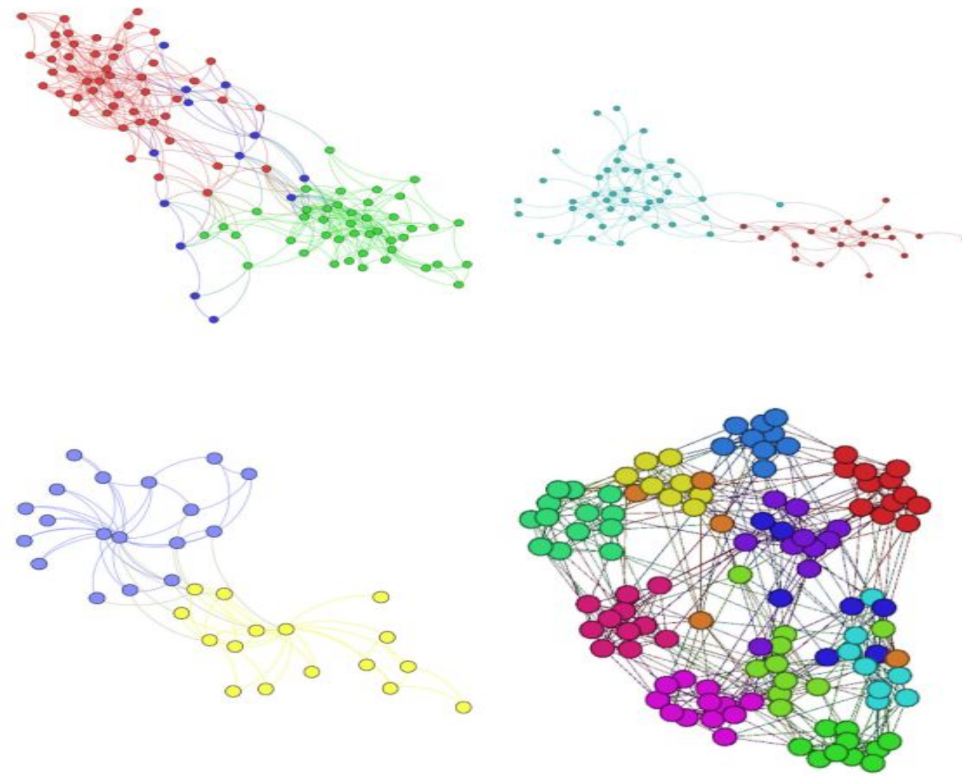
Configuration parameters

- 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 $\in [0.0, 0.1, 0.2]$

Experimentation and Results

- We evaluated the proposed approach on a set of four widely used benchmark networks for which a ground-truth decomposition into communities is known.
- We went through the basic topological characteristics and real community structure of four selected networks.

1. Zachary Karate Club
2. College football network
3. US Politics books network
4. Dolphins social network



Experimentation and Results

- For each network we have applied the proposed algorithm by changing the configuration parameters.
- We evaluated the impact of variations in the parameters σ,δ and Epsilon
- For each configuration, we computed the NMI, ARI and the modularity Q.
- We found out which configuration accelerate slightly the convergence for the right value to obtain.

Experimentation and Results

- Variation in σ has significant impact on the results obtained.
- The best results are obtained for σ around 0.8, 0.9.
- Epsilon has negligible impact on obtained results.
- Increasing Epsilon results in diminishing the NMI and ARI.
- 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.
- Best values of δ are 0.9 and 0.5
- 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.
- Results show that the use of the betweenness centrality accelerate slightly the convergence for the right value to obtain.
- Borda out performs Local Kemeny in all the networks.
- Performance of each configuration differ from one network to another and it has close relation with topology and speciality of network

Experimentation and Results

Comparison of performances of applying LICOD to different configuration parameters

Dataset	Algorithm	NMI	ARI	Modularity	#Communities
	Sigma=0.9,delta=0.9				
Zachary	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
	Sigma=0.9,delta=0.9				
US Politics	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

Experimentation and Results

Comparison of performances of applying LICOD to different configuration parameters

Dataset	Algorithm	NMI	ARI	Modularity	#Communities
	Sigma=0.9,delta=0.9				
Football	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**
	Sigma=0.9,delta=0.9				
Dolphins	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

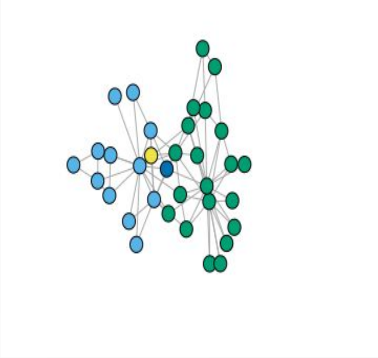
Bold values indicate the best score by LICOD

Experimentation and Results

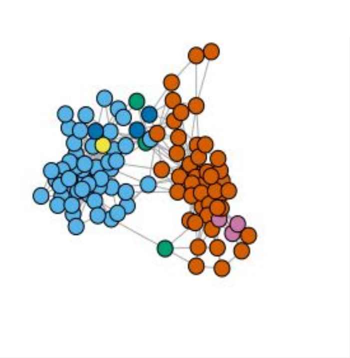
- We also compared the results of our algorithm with results obtained by well-known algorithms:
    - **Newman–Girvan algorithm**
    - **WalkTrap algorithm**
    - **Louvain algorithm**
- The configuration adopted for proposed algorithm is the following: Centrality metric is betweenness centrality, Voting method is Borda, $\sigma = 0.9$ $\delta = 0.9/0.5$, and epsilon = 0
- Results show that LICOD performs better than the other algorithms for Zachary,Dolphin and US Politics networks.
- This could be explained by the absence of leaders in other networks, which makes the communities detection task more difficult.

Experimentation and Results

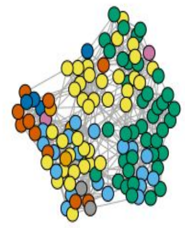
Community structure of the four selected benchmark networks obtained by LICOD (Centrality metric:betweenness centrality, Voting method:Borda, σ =0.9, δ = 0.9,and Epsilon = 0)



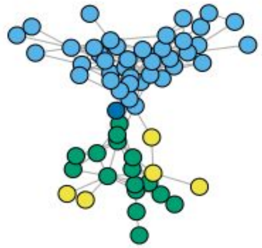
Zachary Karate Club Network



US Politics books network



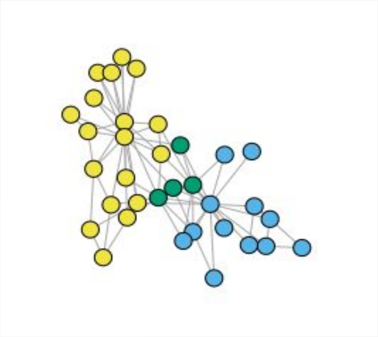
College football network



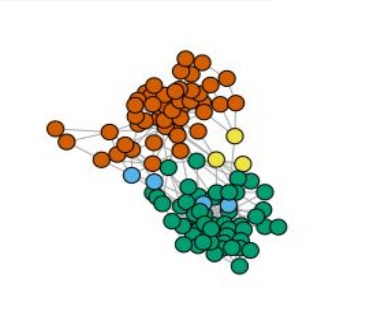
Dolphins social network

Experimentation and Results

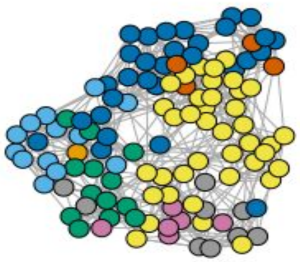
Community structure of the four selected benchmark networks obtained by LICOD (Centrality metric:betweenness centrality, Voting method:Borda, σ =0.9, δ = 0.5,and Epsilon = 0)



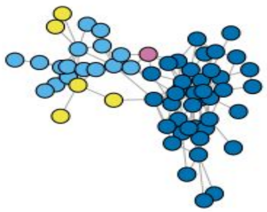
Zachary Karate Club Network



US Politics books network



College football network



Dolphins social network

Experimentation and Results

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

- We can conclude that modularity metric does not correspond to the best decomposition into communities as measured by both NMI and ARI
- 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.

Experimentation and Results

- We also tested the algorithm on a large scale Email network
- There are 1005 nodes and 25571 edges in the network and 42 ground-truth communities
- we could only obtain average performance for it in comparison to other state of the art algorithms.

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

Experimentation and Results

Task Driven Evaluation using Data clustering

- Step 1. Choose some sparse classical datasets for experiment. We have selected 5 public datasets from UCI website

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

- Step 2. Compute an n*n similarity/distance matrix for the n data points. We can use various measures for estimating distance.

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

TABLE V: Distance measures used.

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

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

- 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 evaluation criteria compare the performances of algorithms

Experimentation and Results

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

- We observe some real world network characteristics such as small diameter and low density on these graphs
- Graphs obtained by using Chebyshev distance shows high density. However, graphs has high clustering coefficient. Graphs shows high transitivity as well.
- We applied the community detection algorithms on the graphs obtained by cosine distance as they are relatively closer to real world networks.

Experimentation and Results

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	0.308765	0.1534895	0.8134986	13
	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

For wine the LdCd outperforms other algorithms. It shows good results for other datasets as well.

Results of Innovative Work

- Though Eigenvalue centrality is expected to give good results it doesn't work best for the network we chose as the performance of each parameter configuration has a close relation to the specialities of the network.
- Borda out performs all the different Kemeny aggregation approaches including Local Kemeny in all the networks.
- We tested the proposed algorithm on large scale networks.

Conclusion and Future Work

- 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.
- Uses data clustering to apply a task-driven evaluation of community detection algorithms.
- Develop a full distributed self-stabilizing version for large scale networks.
- Adapt the approach for K-partite and for multiplex networks

Individual Contribution

- Both the team members contributed equally in the code development of **licod.R** and **main.R**

- **IdentifyCommunityLeaders.R** and **IdentifyLeaders.R** were coded by the team member Bavya Balakrishnan

- **BordaRanking.R, kemeny.R and epsilon_threshold_graph**.R were coded by team member Prajwal M P
- Both the team members contributed equally in developing data clustering driven evaluation(**rng_graph.R**)

Reference

[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)