EISTI
École Internationale des Sciences du Traitement de l'Information
www.eisti.fr

Local modularity selection for local community identification

Gustavo Fleury Soares, Induraj R. Ramamurthy, Quoc Viet Pham

CY-Tech / EISTI – École Internationale des Sciences du Traitement de L'Information, Cergy, France.
E-mails: [fleurysoar, pudhupattu, phamquocvi] @eisti.eu

Abstract

The Local Community from one specific node depends largely on the modularity function applied. To determine the best local modularity to apply, we create a model using a vector of centralities and trained using the ground-truth of benchmark networks. We applied the tests on artificial networks generated by LRF generator to predict the best modularity that can be applied to each node of the network. Using the best modularity that the ML models predicted, we constructed consensus matrix and used ensemble clustering by CSPA on the constructed consensus matrix. To check the efficiency of the ensemble clustering, we also Performed ensemble clustering on the LRF's known ground truth and compared the clustering efficiency. The results of which are communicated in terms of accuracy and NMI.

Keywords: Graphs, Modularity, Local Community Quality, Centrality, LRF Artificial Networks, Machine Learn

1. Introduction

A classical approach for local community identification relays on applying a greedy optimization approach that explore the network starting from the target node (for which we want to identify the community) and adding progressively nodes that maximize a given quality function (ex. a local modularity function). One major short-come of this approach is that the computed community depends largely on the applied quality function and that all quality functions can suffer from a problem of sticking into a local maximum.

In this project we want to explore the possibility of learning the best local modularity to apply in function of topological features of the target node. The proposed approach is the following:

1. Given a set of networks for which a ground-truth information about the community structure is available, we compute for each node its local community applying different quality functions.
2. Using the community ground truth information, we can readily select the best quality function that yields the best result.
3. Each node can be then being described by a vector of attributes given its different centralities values in the network.
4. The problem of selecting the best quality function to apply can then be reformulated as multi-label supervised classification problem.

The goal of this project is to implement this approach using benchmark networks (Zachary Karate Club, Football, Dolphins and Polbooks) and also generated artificial networks (LRF). [1]

We developed most of the solutions in R. For example, the tables will show results of Zachary Karate Club network.

2. Best Local Communities Quality

For Local Community detection we start with one node and we try to add vertices neighbours while a community quality function (Q) is enhanced.

We used basic 3 quality functions:

Local Modularity R (mod_R):

$$R = \frac{B_{in}}{B_{in} + B_{out}}$$

Local Modularity M (mod_M):

$$M = \frac{D_{in}}{D_{out}}$$

Local Modularity L (mod_L):

$$L = \frac{L_{in}}{L_{ex}}$$

For create a label which is the modularity that has the best quality compared with the ground-truth, we developed the following table. The modularity are labels in 1-mod_R, 2-mod_M and 3-mod_L. The modularity that gives the best Local Community Quality was assigned in best_mod. (We will present just 3 nodes, for example).

	1	2	3	
nodes	**mod_R**	**mod_M**	**mod_L**	**best_mod**
1	0.454	0.214	0.454	3
2	0.267	0.214	0.267	3
6	0.144	0.226	0.144	2

3. Centralities of Graph

The analysis of a complex network could be done with the characteristics of each node or of all network. In nodes, we can apply ranking of the importance or influence or centrality functions.

For example, the degree centrality consists in the number of neighbors of a node divided by max number of neighbor tested for all nodes.

$$C_d(\mathrm{v}) = \frac{|\, neigh(v)\,|}{max_u\,|\, neigh(v)\,|}$$

We used the library CINNA – "Deciphering Central Informative Nodes in Network Analysis". That give us 49 options of graph centralities measures. [2]

To select few for use in model training we selected the most informative centrality measure, using Principal Component Analysis (PCA). [3] The result plot for the Zachary Network are presented in the Appendix I.

We used the following centralities:

1	Stress Centrality
2	Group Centrality
3	Eccentricity Centrality
4	Harary Centrality
5	Geodesic K-Path Centrality
6	Shortest-Paths Betweenness Centrality
7	Entropy Centrality
8	Flow Betweenness Centrality
9	MNC - Maximum Neighborhood Component
10	Degree Centrality [1]
11	Laplacian Centrality
12	Closeness Centrality (Freeman) [1]

To create the data frame to train the model, we calculate each given centrality for each node and associate with best modularity. The following table exemplify this table. The appendix 2 shows this table for Zachary network.

Node	**C1**	**(…)**	**Cn**	**Best_Mod**

4. Machine Learn Models

We applied the Random Forest algorithm for create a model using the given centralities to decide which is the modularity will give us the best community quality.

Like our training graphs does not have a lot of nodes, it was necessary to use kfold cross validation with repetition to the model get a good accuracy [4].

For compare the performance of different type of models, we create an Artifical Neural Network model using the package "neuralnet" [5]. We used 3 hidden layers and logistic for activation function. The follow figure exemplifies the neural network, with the centralities for the input and type of modularity for output.

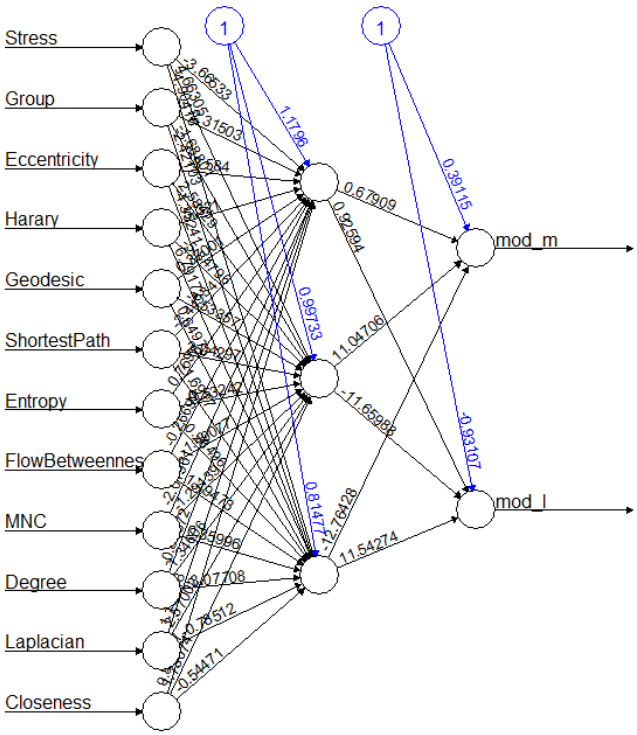


***Figura 1**. Artificial Neural Network example.*

5. LRF Artificial Network

Graphs generated using the Lancichinetti-Fortunato-Radicchi (LFR) [4] model are widely used for assessing the performance of network community detection algorithms. In this work we try to apply a trained model to verify the quality of the model.

For create the graph we used the implementation of NetworkX in Python [5] . The graphs were saved in *.gml* files and passed to R scripts to verify the quality of the models. The follow figure exemplifies a LRF artificial network.

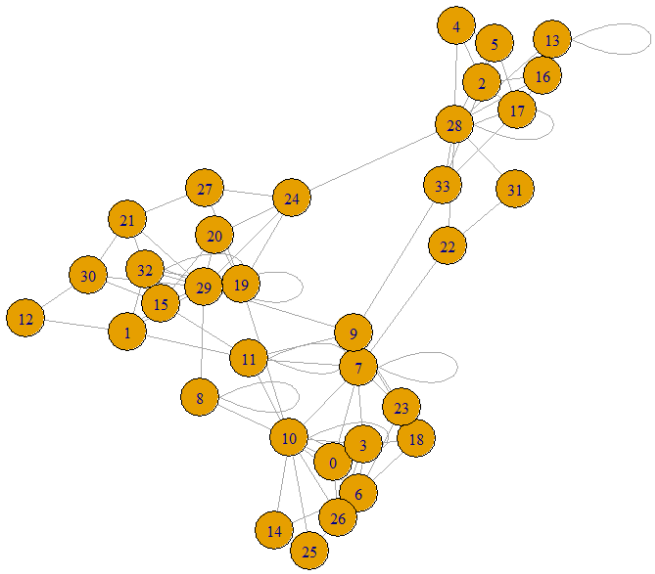


***Figura 2**. LRF Graph - 34 Vertices.*

6. Results and Discussion

We tested three different models with the benchmark networks. The following table present the accuracy for each model.

Dataset	RF-kfold	RF-kfold-R	ANN
Karate	0.778	0.889	0.958
Dolphins	0.765	0.765	0.958
PolBooks	0.742	0.710	0.986

We can verify that the ANN gives best accuracies, but we need to verify if it is not overfitted.

Using the LRF artificial network for test, we applied the models to decide the best modularity for each node. The following table exemplifies the accuracy result, for a model using Karate Graph.

Dataset	RF-kfold	RF-kfold-R	ANN
LRF-34v	0.618	0.471	0.618

The above depicts the result of the CSPA cluster compared to the predicted clusters. The Results of karate, dolphin and the LRF generated clusters comparison are shown above.

Karate ground truth vs Karate clustered(CSPA)								
RI	**ARI**	**MI**	**AMI**	**VI**	**NVI**	**ID**	**NID**	**NMI**
0.59	0.20	0.21	0.30	0.84	0.80	0.480	0.69	0.30
Dolphin ground truth vs Dolphin clustered(CSPA)								
0.49	-0.06	0.06	0.11	0.93	0.93	0.55	0.88	0.11
LRF ground truth vs LRF clustered(CSPA)								
0.60	0.09	0.15	0.13	1.88	0.92	0.94	0.86	0.13

For next steps, we can indicate create new models using bigger benchmark graphs. It will take a lot of computer power to calculate all the modularity's for each node, but could result in best model for be applied in other bigger graphs.

References

[1] R. Kanawati, *Social Network Analysis: Community Detection,* Paris, 2019.

[2] M. Ashtiani, M. Mirzaie and M. Jafari, "CINNA - Deciphering Central Informative Nodes in Network Analysis," [Online]. Available: https://cran.r-project.org/web/packages/CINNA/index.html. [Accessed 2020].

[3] M. Ashtiani, "Network Analysis in R: Centrality Measures," 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/centrality-network-analysis-R. [Accessed 2020].

[4] J. Brownlee, "How to estimate model accuracy," [Online]. Available: https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/. [Accessed 2020].

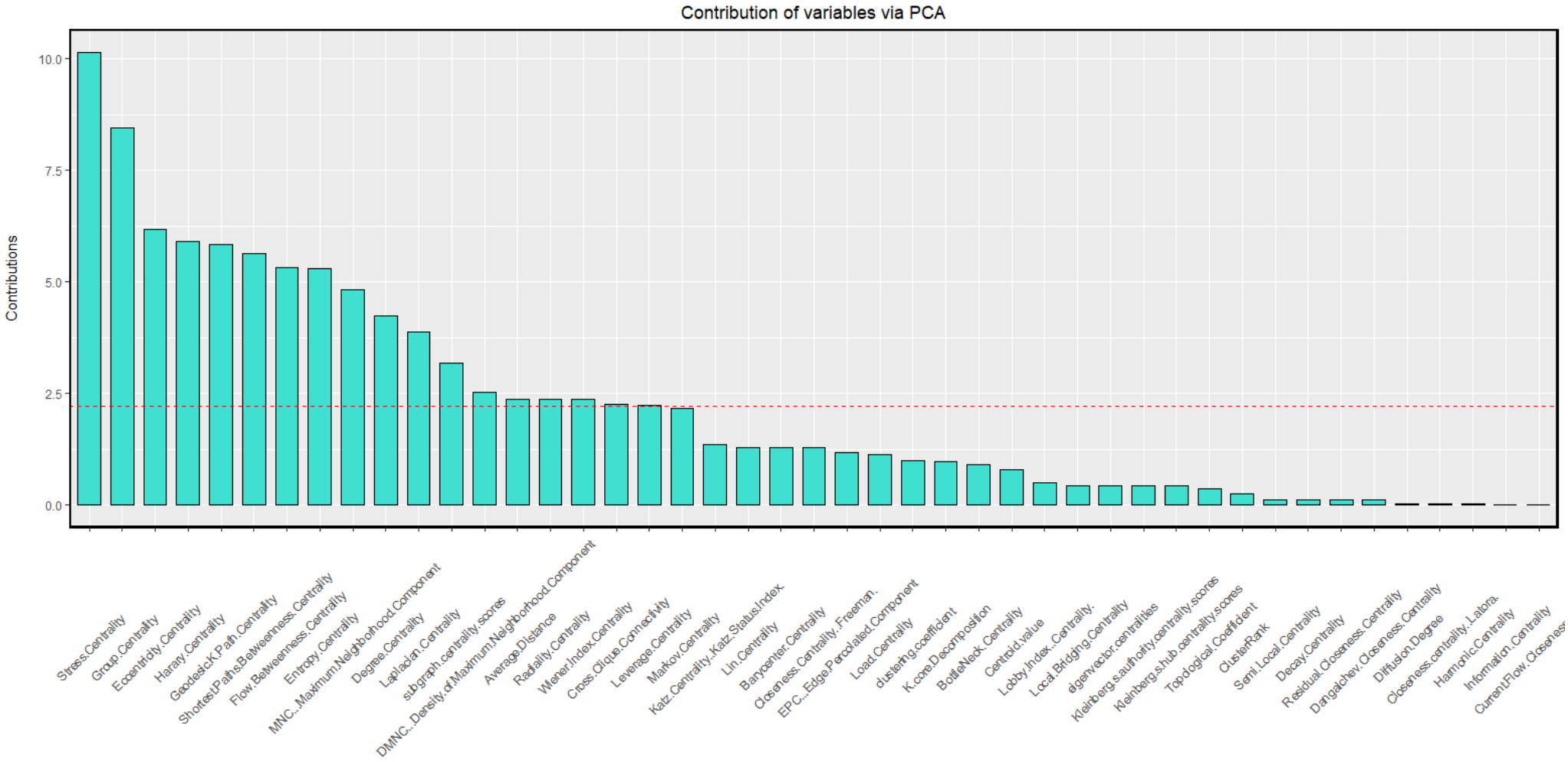
[5] M. Alice, "Multilable Classification with Neuralnet Package," [Online]. Available: https://www.r-bloggers.com/multilabel-classification-with-neuralnet-package/. [Accessed 2020].

[6] A. Lancichinetti, S. Fortunato and F. Radicchi, "Benchmark graphs for testing community detection algorithms," 30 October 2008.

[7] N. Developers, "LFR_benchmark_graph," [Online]. Available: https://networkx.github.io/documentation/networkx-2.1/reference/algorithms/generated/networkx.algorithms.community.community_generators.LFR_benchmark_graph.html. [Accessed January 2020].

[8] UiPath, "The Forrester Wave," [Online]. Available: https://www.uipath.com/company/rpa-analyst-reports/forrester-wave-rpa. [Accessed 2019].

Appendix I


Contribution of variables via PCA
Contributions
10.0
7.5
5.0
2.5
0.0
Stress.Centrality
Group.Centrality
Eccentricity.Centrality
Harary.Centrality
Geodesic.K.Path.Centrality
Shortest.Paths.Betweenness.Centrality
Flow.Betweenness.Centrality
Entropy.Centrality
MNC...Maximum.Neighborhood.Component
Degree.Centrality
Laplacian.Centrality
subgraph.centrality.scores
DMNC...Density.of.Maximum.Neighborhood.Component
Average.Distance
Radiality.Centrality
Wiener.Index.Centrality
Cross.Clique.Connectivity
Leverage.Centrality
Markov.Centrality
Katz.Centrality..Katz.Status.Index.
Lin.Centrality
Barycenter.Centrality
Closeness..Centrality..Freeman.
EPC...Edge.Percolated.Component
Load.Centrality
clustering.coefficient
K.core.Decomposition
BottleNeck..Centrality
Centroid.value
Lobby.Index...Centrality.
Local.Bridging.Centrality
eigenvector.centralities
Kleinberg.s.authority.centrality.scores
Kleinberg.s.hub.centrality.scores
Topological.Coefficient
ClusterRank
Semi.Local.Centrality
Decay.Centrality
Residual.Closeness.Centrality
Dangalchev.Closeness.Centrality
Diffusion.Degree
Closeness.centrality..Latora.
Harmonic.Centrality
Information.Centrality
Current.Flow.Closeness

Appendix II – Centralities and Best Modularity - Zachary Karate Club Network

Node	lbest_Mod	Stress	Group	Eccentricity	Harary	Geodesic	ShortestPath	Entropy	FlowBetweenness	MNC	Degree	Laplacian	Closeness
1	3	0	3.719	3	0.333	33	231.0714	7.740439	0	10	16	410	0.0172
2	3	1	1.876	3	0.333	33	28.47857	8.088788	10	8	9	194	0.0147
3	3	10	1.786	3	0.333	33	75.85079	8.088788	21	7	10	242	0.0169
4	3	2	1.093	3	0.333	33	6.288095	8.088788	9	6	6	134	0.0141
5	3	0	0.759	4	0.250	25	0.333333	8.088788	2	3	3	58	0.0115
6	2	1	1.042	4	0.250	25	15.83333	8.088788	3	4	4	70	0.0116
7	2	2	1.042	4	0.250	25	15.83333	8.088788	3	4	4	70	0.0116
8	3	0	0.593	4	0.250	32	0	8.088788	0	4	4	102	0.0133
9	2	5	0.649	3	0.333	33	29.52937	8.088788	7	5	5	148	0.0156
10	3	1	0.480	4	0.250	32	0.447619	8.088788	1	1	2	60	0.0132
11	3	0	0.759	4	0.250	25	0.333333	8.088788	0	3	3	58	0.0115
12	3	0	0.559	4	0.250	25	0	8.088788	0	1	1	34	0.0111
13	3	0	0.535	4	0.250	25	0	8.088788	0	2	2	50	0.0112
14	3	4	0.615	3	0.333	33	24.21587	8.088788	4	4	5	146	0.0156
15	2	0	0.466	5	0.200	24	0	8.088788	0	2	2	64	0.0112
16	2	0	0.466	5	0.200	24	0	8.088788	0	2	2	64	0.0112
17	2	0	0.733	5	0.200	17	0	8.088788	0	2	2	22	0.0086
18	3	0	0.492	4	0.250	25	0	8.088788	0	2	2	56	0.0114
19	2	0	0.466	5	0.200	24	0	8.088788	0	2	2	64	0.0112
20	3	2	0.464	3	0.333	33	17.14683	8.088788	2	2	3	96	0.0152
21	2	0	0.466	5	0.200	24	0	8.088788	0	2	2	64	0.0112
22	3	0	0.492	4	0.250	25	0	8.088788	0	2	2	56	0.0114
23	2	0	0.466	5	0.200	24	0	8.088788	0	2	2	64	0.0112
24	2	0	0.949	5	0.200	24	9.3	8.088788	0	4	5	110	0.0119
25	3	0	0.843	4	0.250	32	1.166667	8.088788	0	2	3	38	0.0114
26	3	1	0.810	4	0.250	32	2.027778	8.088788	5	2	3	40	0.0114
27	2	0	0.589	5	0.200	23	0	8.088788	0	2	2	48	0.0110
28	2	2	0.763	4	0.250	32	11.79206	8.088788	3	2	4	90	0.0139
29	3	3	0.539	4	0.250	32	0.947619	8.088788	5	2	3	78	0.0137

30	2	1	0.832	5	0.200	24	1.542857	8.088788	4	4	4	92	0.0116
31	2	2	0.599	4	0.250	32	7.609524	8.088788	5	3	4	106	0.0139
32	3	7	1.084	3	0.333	33	73.00952	8.088788	12	3	6	150	0.0164
33	2	1	2.766	4	0.250	32	76.69048	8.088788	12	12	12	278	0.0156
34	2	0	4.209	4	0.250	32	160.5516	8.088788	0	14	17	436	0.0167