Complex and Social Networks: Lab session 5

Finding community structure

Miguel Alcon & Egon

Contents

TASK 1

Introduction

In this session we will run and compare different community finding algorithms. We will use implementations from the igraph package.

Given an undirected graph, we want to analyze, for each available community finding algorithm the value achieved by the output partition for each of the following criteria: 'Triangle Partition Ratio' (TPT), 'expansion', 'conductance' and 'modularity'.

- Triangle partition ratio: fraction of nodes in C that belong to a triad

$$TPT = \frac{|\{u : u \in C \text{ and } \{(w,v) \in E : w,v \in C, (u,w),(u,v) \in E\} \neq \emptyset\}|}{n_c}$$

- Expansion: number of edges per node leaving the cluster

$$\frac{f_c}{n_c}$$

- Conductance: fraction of total edge volume that points outside the cluster

$$\frac{f_c}{2m_c + f_c}$$

- Modularity: difference between numebr of edges in C and the expected nr. of edges $E[mc]$ of a random graph with the same degree distribution

$$\frac{1}{4m}\left(m_c - E\left[m_c\right]\right)$$

Where:

$$f_c = C = |\{(u, v)|u \in C, v \notin C\}|$$

$$m_c = C = |\{(u, v)|u, v \in C\}|$$

and n_c is just the number of nodes.

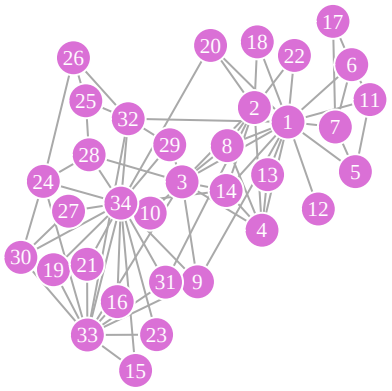
To calculate TPT of a community division we decided to calculate a weighted average of the TPT evaluated on each cluster. For the other metrics this is not necessary.

Test on the Zachary Karate Graph

This is a social network of friendships between 34 members of a karate club at a US university in the 1970s.

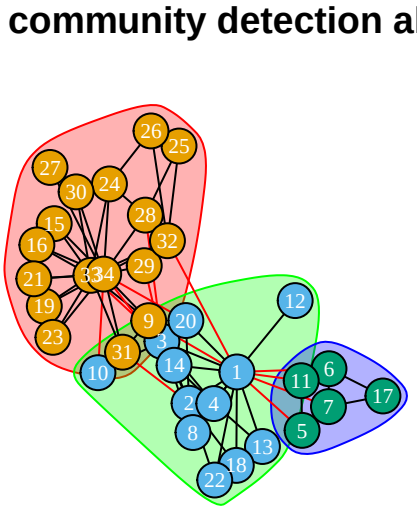
[See W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977)]

This network became a popular example of community structure.

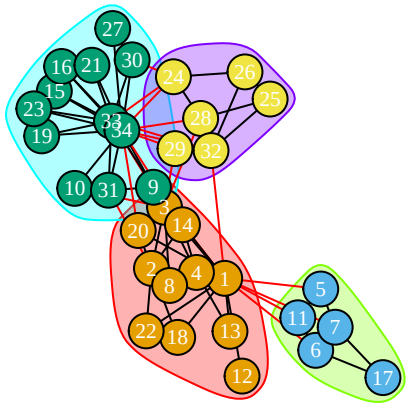


	Ncluster	TPT	Expansion	Conductance	Modularity
Edge Betweennes	5	8.500000	0.7058824	0.1818182	0.4012985
Fastgreedy	3	12.029412	0.5588235	0.1386861	0.3806706
Label propagation	2	21.500000	0.2941176	0.0684932	0.3717949
Leading eigenvector	4	6.941177	0.7647059	0.2000000	0.3934089
Multilevel	4	10.941176	0.6176471	0.1555556	0.4188034
Optimal	4	10.676471	0.6176471	0.1555556	0.4197896
Spinglass	4	10.676471	0.6176471	0.1555556	0.4197896
Walktrap	5	4.117647	0.9411765	0.2580645	0.3532216
Infomap	3	16.205882	0.4117647	0.0985915	0.4020381

The infomap algorithm seems to work very well: with only three clusters we got quite high TPT and modularity (as we want), and low conductance and expansion (as we want).



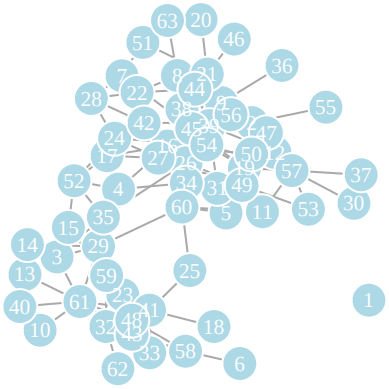
With four cluster instead we have three algoritms that hold similar results, for example:



Dolphin social network

All members of a school were assumed associated. Half-weight index (HWI) was used to quantify the frequency of association among individuals. Data source

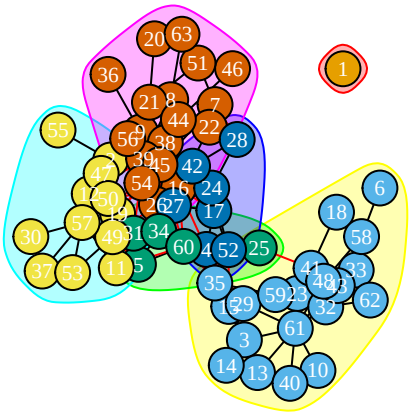
Lusseau, David, et al. "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations." Behavioral Ecology and Sociobiology 54.4 (2003): 396-405.



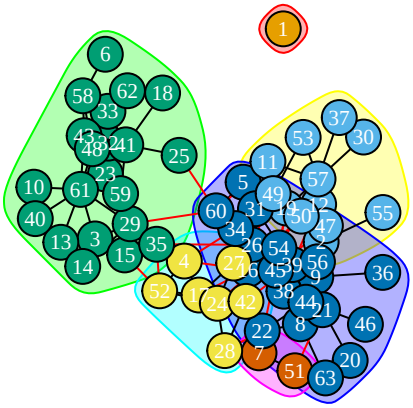
	Ncluster	TPT	Expansion	Conductance	Modularity
Edge Betweennes	6	23.14286	0.5079365	0.1118881	0.5193821
Fastgreedy	5	28.04762	0.4444444	0.0965517	0.4923263
Label propagation	6	23.85714	0.5238095	0.1157895	0.5122622
Leading eigenvector	6	14.98413	0.7301587	0.1691176	0.4911989
Multilevel	6	21.30159	0.5714286	0.1276596	0.5277283
Optimal	6	20.07937	0.6031746	0.1357143	0.5285194
Walktrap	5	27.19048	0.4444444	0.0965517	0.4888454
Infomap	6	21.30159	0.5714286	0.1276596	0.5277283

Spinglass is omitted because the algorithm does not work with unconnected graphs.

Multilevel community detection algorithm



Edge betweenness community detection algorithm

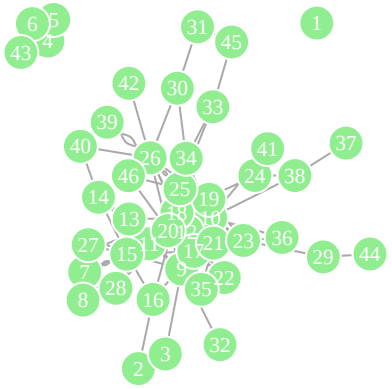


Both multilevel and edge betweenness seems to be good choice for our dolphin school.

Tortoises

A bipartite network was first constructed based on burrow use - an edge connecting a tortoise node to a burrow node indicated burrow use by the individual. Social networks of desert tortoises were then constructed by the bipartite network into a single-mode projection of tortoise nodes.

Sah, Pratha, et al. "Inferring social structure and its drivers from refuge use in the desert tortoise, a relatively solitary species." Behavioral Ecology and Sociobiology 70.8 (2016): 1277-1289.

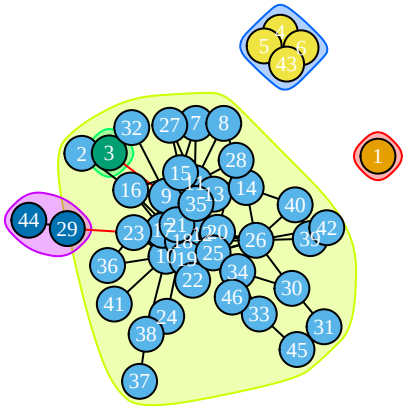


	Ncluster	TPT	Expansion	Conductance	Modularity
Edge Betweennes	15	19.565217	0.7173913	0.1843575	0.3062478
Fastgreedy	7	6.565217	0.8695652	0.2325581	0.4022339
Label propagation	5	64.782609	0.0652174	0.0143541	0.1221520
Leading eigenvector	7	5.108696	1.0000000	0.2771084	0.3584906
Multilevel	7	7.434783	0.8260870	0.2183908	0.4272428
Optimal	7	6.521739	0.8478261	0.2254335	0.4311588
Walktrap	9	23.804348	0.5434783	0.1336898	0.3528836
Infomap	9	15.391304	0.6739130	0.1712707	0.3921324

This time we see that, given the difficult disconnetted graph, the different community algorithms performs really differently.

The only algorithm that find a small number of cluster is the Label propagation algorithm.

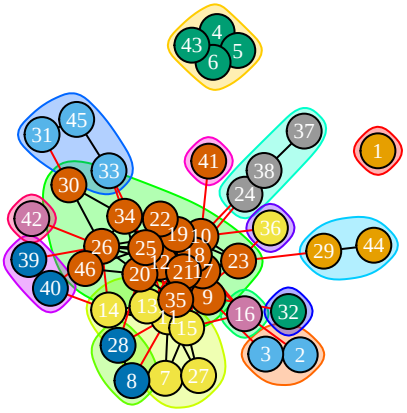
Label propagation community detection algorithm



Unfortunatly, a glance is enough to say that this division is not satisfyng. The algorithm mainly divid in different clusters the disconnected components.

On the other extreme, the edge betweenness algorithm finds 15 clusters.

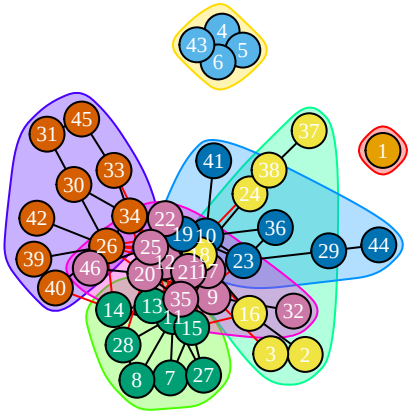
Label propagation community detection algorithm



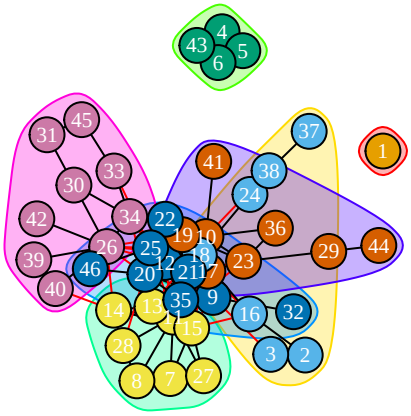
Does this makes any sense?

We can try to see partitions computed by other algorithms (but we still have to question ourselves if it makes sense to us having 7/9 cluster on only 46 individuals). This two algorithms return an almost identical result, but the second one takes a little bit of computational time more.

Multilevel community detection algorithm

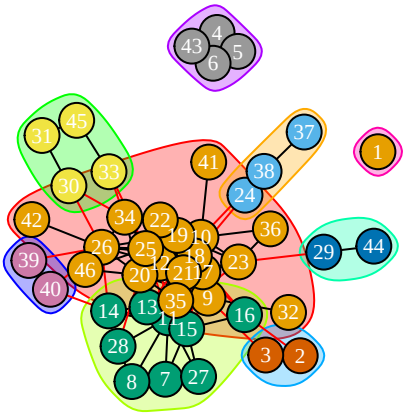


Optimal community detection algorithm

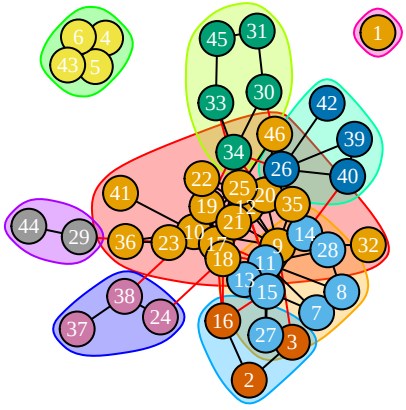


With 9 clusters, we have again two algorithms that finds similar result. The first one is a little bit better on the TPT metric, the other one is slightly better in all the other metrics.

Walktrap community detection algorithm



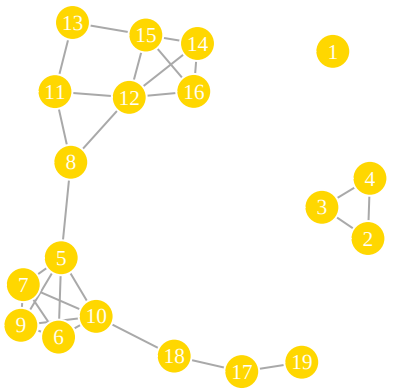
Infomap community detection algorithm



Weaver aves

A network edge was drawn between individuals that used the same nest chambers either for roosting or nest-building at any given time within a series of observations at the same colony in the same year, either together in the nest chamber at the same time or at different times.

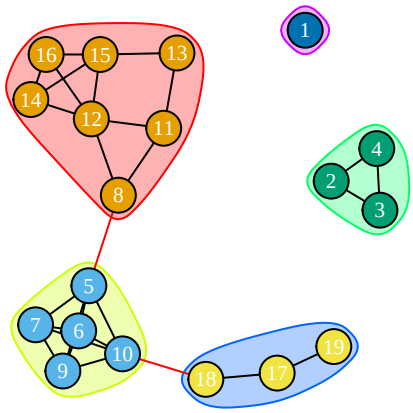
van Dijk, Rene E., et al., "Cooperative investment in public goods is kin directed in communal nests of social birds." Ecology letters 17.9 (2014): 1141-1148.



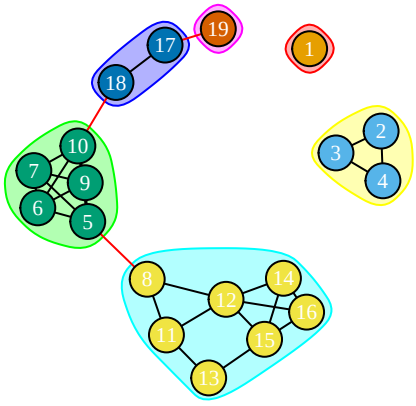
	Ncluster	TPT	Expansion	Conductance	Modularity
Edge Betweennes	5	4.631579	0.1052632	0.0370370	0.5860969
Fastgreedy	5	4.578947	0.1578947	0.0566038	0.5465561
Label propagation	5	4.631579	0.1052632	0.0370370	0.5860969
Leading eigenvector	6	4.631579	0.1578947	0.0566038	0.5529337
Multilevel	5	4.631579	0.1052632	0.0370370	0.5860969
Optimal	5	4.631579	0.1052632	0.0370370	0.5860969
Walktrap	5	4.631579	0.1052632	0.0370370	0.5860969
Infomap	5	4.631579	0.1052632	0.0370370	0.5860969

This time we have a more "easy" graph to partition (communities are easy to see even just by looking at it), and results algotrithms seems indeed more consistent. All the algorithm finds the similiar solution, with the exception that one of them divides one of the communities in two.

Infomap community detection algorithm



Leading eigenvector community detection algorithm



TASK 2

Introduction

In this section, we are going to analyze the resultant communities of applying one community detection algorithm to a huge network, the Wikipedia network.

Algorithm selection

We did an experiment to select the algorithm we are going to use in the analysis. Time of computation, modularity, and number of communities are the values we focus on to make the decision. The following table shows the results of this experiment.

Algorithm	Time	Modularity	Community length
edge betweenness	565.5369680	0.8518692	3778
fast greedy	5.6315761	0.7452734	1037
label propagation	0.8565390	0.7083109	1800
leading eigenvector	10.3263278	0.2018541	850
multi level	0.5176749	0.7702816	886
walktrap	36.7464299	0.6723655	2197

We had to discard some of the algorithms:

- `edge.betweenness.community`. Long computation time. We applied this algorithm to a 30% sample of the network because with a larger sample it did not finish in a reasonable time.
- `leading.eigenvector.community` Low modularity compared with the other ones.
- `optimal.community`. It crashes.
- `spinglass.community`. It crashes.

Finally, we selected the `fastgreedy.community` algotithm because it has more centered values for modularity and community length.

Analysis of the communities