Detecting Communities in Complex Networks using a Hybrid Immunological Algorithm

Rocco Alessandro Scollo

Complex Intelligent System
Department of Mathematics and Computer Science
University of Catania, Italy

January 22, 2021

Outline

Complex Networks

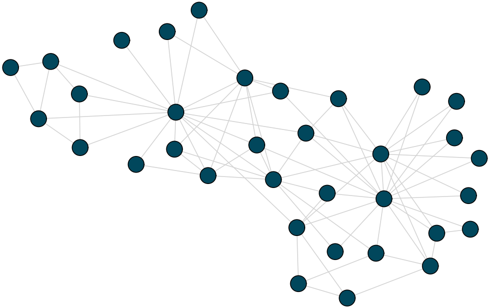
Real networks have complex and non-trivial topological features:

- social science
- biology
- computer science
- ...

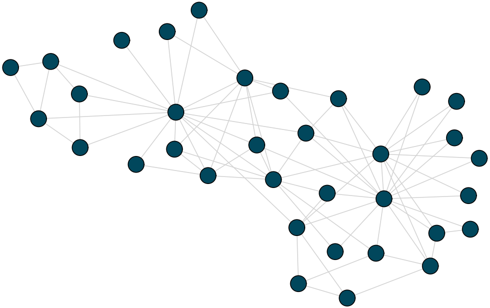
Nowadays, with Internet, these networks have reached millions or billions of vertices and interactions (e.g. Facebook, Twitter, ...).

Graphs are useful tools for representing and analysing real networks.

Complex Networks



Complex Networks



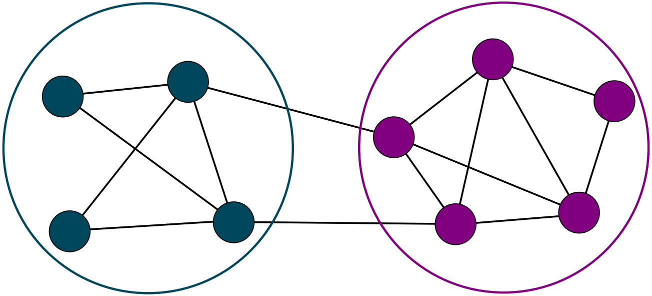
Community structure

High number of edges within groups of vertices and low number between these groups.

Community Structure

Communities are groups of vertices with similar properties and/or role.

Many real networks exhibit a community structure.



Community Detection: Applications

- **Classification** of individuals that belong to the same communities.
- **Clustering** the clients of an e-commerce with similar interests of from the same country.
- **Identifying** groups of proteins with the same function in biological networks to understand the physiology of a cell.

Community Detection: Problem

The aim of **community detection** is to identifying **clusters** of nodes and their hierarchical organization.

The concept of community or cluster is not well-defined.

A graph with a community structure is different from a **random graph**, that usually does not have a community structure.

Modularity

The **modularity** is the most popular evaluation function for the community detection, proposed by *Newman* and *Girvan* in 2004 [7].

Uses a randomized version of the original graph, called **null model**, with edges rewired at random and with the same expected degree for each vertices.

$$Q = \frac{1}{2m}\left[\sum_{i=1}^{N}\sum_{j=1}^{N}\left(A_{ij} - P_{ij}\right)\delta(i,j)\right]$$

where P_{ij} is the expected number of edges between vertices i and j in the null model.

Modularity

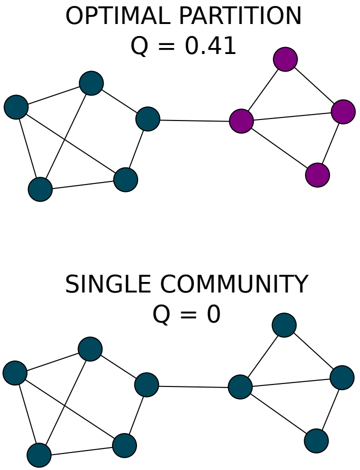
Using a null model with the **same degree distribution** of the original graph, P_{ij} is:

$$P_{ij} = \frac{k_i k_j}{2m}$$

Only pairs of vertices belonging to the same cluster contribute to the sum:

$$Q(C) = \sum_{c \in C} \left[\frac{\ell_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right]$$

Modularity



OPTIMAL PARTITION
Q = 0.41
SUBOPTIMAL PARTITION
Q = 0.22
SINGLE COMMUNITY
Q = 0
NEGATIVE MODULARITY
Q = -0.12

Hybrid-IA

HYBRID-IA is an **immunological algorithm** inspired by the clonal selection principle of the immune system.

It is a population-based **metaheuristic** that uses a local search procedure to improves the search.

It is based on three main **immune operator**:

- cloning
- hypermutation
- aging

Hybrid-IA

The **cloning operator** create a population copying the current set of solutions:

- static cloning
- dynamic cloning

The **aging operator** removes old solutions for maintaining diversity within the population.

Hybrid-IA

The **hypermutation operator** modifies the cloned solutions in order to explore their neighbourhood.

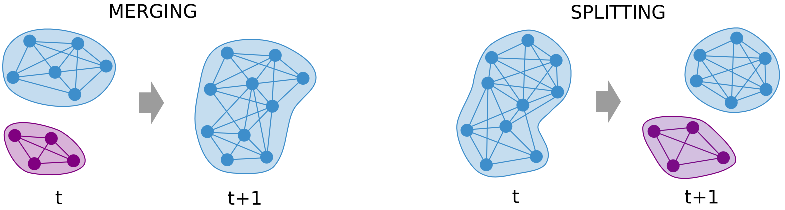
The **mutation rate** (number of mutations) can be

- static: $M = k$
- dynamic: $M = \lfloor (\alpha \times \ell) + 1 \rfloor$, with $\alpha = e^{-\rho \hat{f}(\vec{x})}$

Hybrid-IA

For the community detection problem, α is the **probability** to move a node from a community to another one.

The operator chooses two communities C_i and C_j, with $C_i \neq C_j$, and moves all vertices from C_i to C_j with a probability of α.



Hybrid-IA: Local Search

The **local search operator** trying to improve the modularity value of the selected solutions moving a node from its community to nearby community.

It is based on the **move vertex** operator of *Kernighan* and *Lin* [5] for the clustering problem.

Hybrid-IA: Local Search

The **move gain** is defined as the variation in modularity when moving the node u from the community C_i to the community C_j:

$$\Delta Q_u(C_i, C_j) = \frac{l_{C_j}(u) - l_{C_i}(u)}{m} + d_V(u)\left[\frac{d_{C_i} - d_V(u) - d_{C_j}}{2m^2}\right]$$

where C_j is a community in the neighbourhood of node u.

Hybrid-IA: Local Search

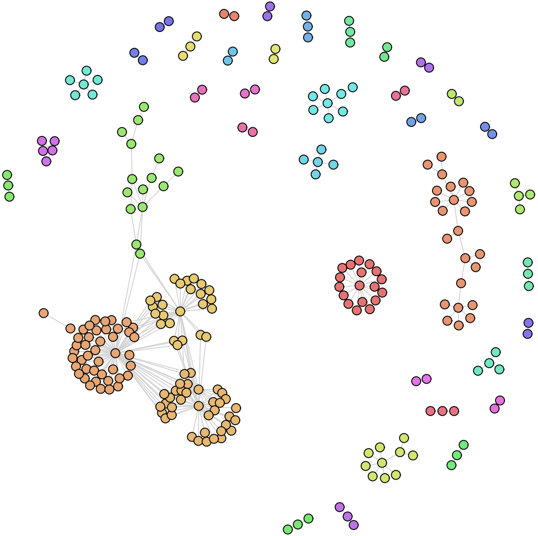
The local search tries to repair **not well-formed** communities, i.e. communities with a low ratio between internal links and sum of degrees.

Then the move operator is applied to nodes that **lies on the boundaries** of the communities and share links with other modules.

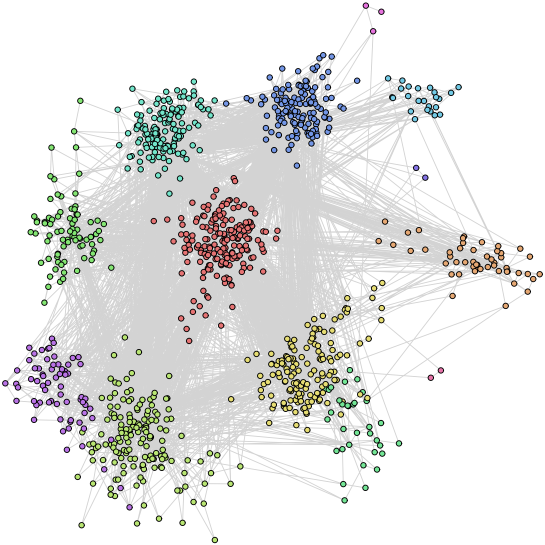
Hybrid-IA: Results

Networks		Hybrid-IA	Louvain [1]	HDSA	BADE	SSGA	BB-BC	BA	GSA
Cattle PPI	Q	**0.7195**	**0.7195**	**0.7195**	0.7183	0.7118	0.7095	0.7143	0.7053
	N_C	40	40	40	41	40	48	42	43
E.coli TRN	Q	0.7785	0.7793	**0.7822**	0.7680	0.7507	0.7520	0.7629	0.7416
	N_C	43	41	47	58	61	71	56	61
C.elegans MRN	Q	**0.4506**	0.4490	0.4185	0.3473	0.3336	0.3374	0.3514	0.3063
	N_C	10	10	13	25	22	21	22	24
Yeast TRN	Q	0.7668	**0.7683**	-	-	-	-	-	-
	N_C	33	26	-	-	-	-	-	-
Helicobacter pylori PPI	Q	0.5359	**0.5462**	0.5086	0.4926	0.4726	0.4681	0.4900	0.4600
	N_C	51	24	52	69	70	75	62	77
E.coli MRN	Q	**0.3817**	0.3734	-	-	-	-	-	-
	N_C	13	8	-	-	-	-	-	-
Yeast PPI (2)	Q	0.5796	**0.5961**	-	-	-	-	-	-
	N_C	159	46	-	-	-	-	-	-
Yeast PPI (1)	Q	0.7002	**0.7648**	-	-	-	-	-	-
	N_C	353	213	-	-	-	-	-	-

Hybrid-IA: Community Structure



Hybrid-IA: Community Structure



Benchmarks

Modularity is a good evaluation function but does not reflects *true* partition.

LFR Benchmarks [6] generates weighted undirected and directed synthetic networks, with simple or overlapping communities.

Normalized Mutual Information

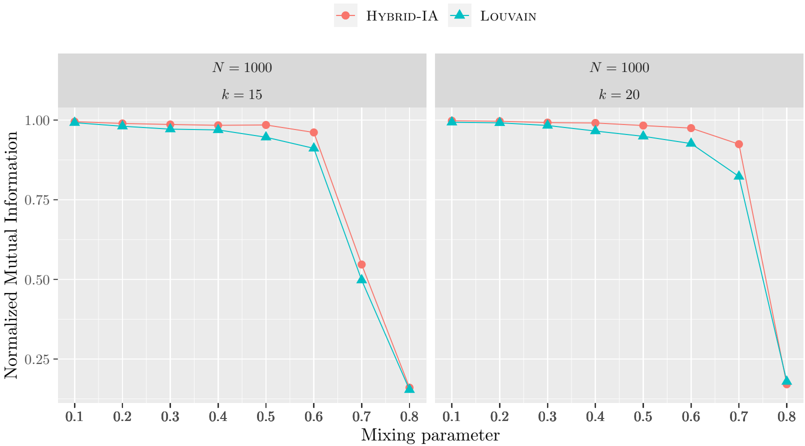
Normalized Mutual Information [3] measures the similarity between to partition using the **information theory**:

$$NMI(X,Y) = \frac{2(H(X) - H(X|Y))}{H(X) + H(Y)}$$

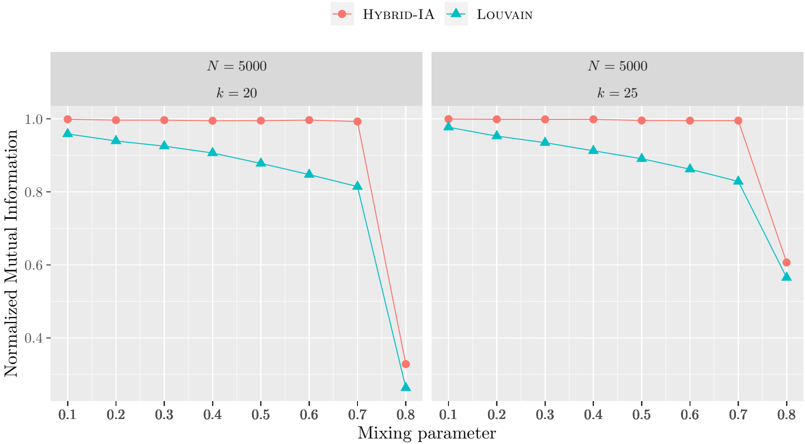
where

$$H(X) = -\sum_{x} P(x)\log P(x)$$
$$H(X|Y) = -\sum_{x,y} P(x,y)\log P(x|y)$$

Normalized Mutual Information



Normalized Mutual Information



Local Search

A study to investigate on how the position of the **local search operator** can influence HYBRID-IA.

Three different positions within the evolutionary cycle:

A) after the selection operator

B) after the hypermutation operator replacing mutated individuals

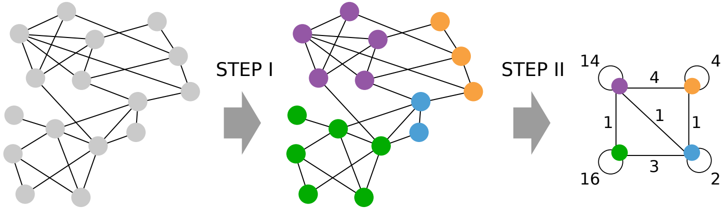
C) after the hypermutation operator producing new individuals

Local Search: Results

	A			B			C		
Instance	N_C	Q	Time	N_C	Q	Time	N_C	Q	Time
$(300, 15, 0.5)$	11.6	**0.392061**	8.3	11.6	**0.392061**	14.5	11.6	**0.392061**	15.5
$(300, 20, 0.5)$	11.2	**0.386560**	9.4	11.2	**0.386560**	16.9	11.2	**0.386560**	17.9
$(500, 15, 0.5)$	17.6	0.436989	13.7	16.8	**0.437168**	24.6	16.8	**0.437168**	26.1
$(500, 20, 0.5)$	17.0	**0.430526**	16.3	17.0	**0.430526**	29.7	17.0	**0.430526**	31.3
$(1000, 15, 0.5)$	34.4	0.467073	28.0	30.0	0.468122	51.2	30.4	**0.468205**	53.9
$(1000, 20, 0.5)$	37.0	0.468532	33.8	33.0	**0.469451**	62.6	32.2	0.469442	65.2
$(5000, 20, 0.5)$	196.4	0.493532	182.4	190.2	0.493985	346.0	189.4	**0.493994**	353.5
$(5000, 25, 0.5)$	193.0	0.493379	228.8	185.4	**0.493741**	438.1	184.6	0.493740	427.1

Future Works

Adapt HYBRID-IA with a **multi level** approach for very large networks.



Use **Data Mining** techniques, like frequent pattern, or probability learning to gather useful information to guide the search.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. DOI: 10.1088/1742-5468/2008/10/P10008.

[2] V. Cutello, G. Fargetta, M. Pavone, and R. A. Scollo. "Optimization Algorithms for Detection of Social Interactions". In: *Algorithms* 13.6 (2020), p. 139. DOI: 10.3390/a13060139.

[3] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. "Comparing community structure identification". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.09 (Sept. 2005), P09008–P09008. DOI: 10.1088/1742-5468/2005/09/p09008.

[4] S. Fortunato. "Community detection in graphs". In: *Physics Reports* 486.3-5 (2010), pp. 75–174. DOI: j.physrep.2009.11.002.

[5] B. W. Kernighan and S. Lin. "An efficient heuristic procedure for partitioning graphs". In: *The Bell System Technical Journal* 49.2 (Feb. 1970), pp. 291–307. DOI: 10.1002/j.1538-7305.1970.tb01770.x.

References

[6] A. Lancichinetti and S. Fortunato. “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities”. In: *Physical Review E* 80 (1 July 2009), p. 016118. DOI: 10.1103/PhysRevE.80.016118.

[7] M. E. J. Newman and M. Girvan. “Finding and evaluating community structure in networks”. In: *Physical Review E* 69 (2 Feb. 2004), p. 026113. DOI: 10.1103/PhysRevE.69.026113.

Thanks for your attention!