Noname manuscript No.
(will be inserted by the editor)

Multi-ego-centered communities in practice

Maximilien Danisch · Jean-Loup Guillaume ·
Bénédicte Le Grand

Received: date / Accepted: date

Abstract We propose here a framework to unfold the ego-centered community structure of a given node in a network. The framework is not based on the optimization of a quality function, but on the study of the irregularity of the decrease of a proximity measure. It is a practical use of the notion of multi-ego-centered community and we validate the pertinence of the approach on benchmarks and a real-world network of wikipedia pages.

Keywords ego-centered community · multi-ego-centered community · proximity measure

1 Context and related work

Many real-world complex systems, such as social networks or computer networks can be modeled as large graphs, called complex networks. Because of the increasing volume of data available and the need to understand such huge systems, complex networks have been extensively studied these last ten years. Due to its applications, notably in market research and classification, and its intriguing nature, the notion of communities of nodes[1] and their detection has been at the center of this research. For an extensive survey on community detection, we refer to the 2010 review by Fortunato [FOR10].

Maximilien Danisch
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France
E-mail: maximilien.danisch@gmail.com

Jean-Loup Guillaume
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France

Bénédicte Le Grand
CRI, Université Paris 1 Panthéon-Sorbonne. 90 rue de Tolbiac, 75013 Paris, France

[1] The idea that there are groups of nodes which are very connected to one-another, but loosely connected to the outside.

Communities are clearly overlapping in real world systems, especially in social networks, where every individual belongs to various communities: its family, its colleagues, various groups of friends, etc. Finding all these overlapping communities in a huge graph is very complex: in a graph of n nodes there are 2^n such possible communities and 2^{2^n} such possible community structures. Even if these communities could be efficiently computed, it may lead to uninterpretable results. However, some studies have still tackled this problem, such as [PAL05] and [EVA09].

Because of the complexity of overlapping communities detection, most studies have restricted the community structure to a partition, where each node belongs to one and only one community. This problem, also very complex, does not have a perfect solution for now, however several algorithms with very satisfying results exist. The most popular one [BLO08] optimizes a quality function, called *modularity* [GIR02], in an agglomerative fashion.

Another approach, to keep the realism of overlapping communities, but without making the problem too complex, is to focus on a single node and try to find all the communities it belongs to, which we call ego-centered communities. This has been extensively studied following a quality function approach: starting from a group where only the given node is included and optimizing step by step (by adding or removing a node from the group) a given quality function. For instance in [CLA05] the quality function depends on the inner and outer degrees of nodes at the border of the community. In [LUO06] the quality function takes the inner and outer degrees of the nodes in the core of the community as variables. In [CHE09] the quality function depends on the density of intra-community links and the density of external links, while [NGO12] is an improvement of it. In [FRI11] the quality function, called *Cohesion*, depends on the density of inner and outer triangles.

This concept of ego-centered community has many applications: every time we are looking for nodes similar to a node of interest. For instance, the user is on a webpage (or wikipedia) and wants to find all pages dealing with the same topic. The problem is that a node often belongs to many communities (and thus topics for the example). To refine the search it is often useful to use the concept of multi-ego-centered communities. In that problem, we have a few nodes belonging to many communities, but hopefully they are all sharing only one of them and the goal is to find all nodes in the shared community. This problem has been investigated with a proximity approach in [DAN12] and with a quality function approach in [SOZ10] with the aim of organizing a successful cocktail party and in [TAT13] with the aim of finding several communities of different scales containing the set of nodes. Other relevant literature are [TON06] and [KOR07] where authors connect the query nodes with the k (input) most relevant nodes.

In this paper we are interested in a slightly different problem: given a node, we want to unfold all the communities it is part of.

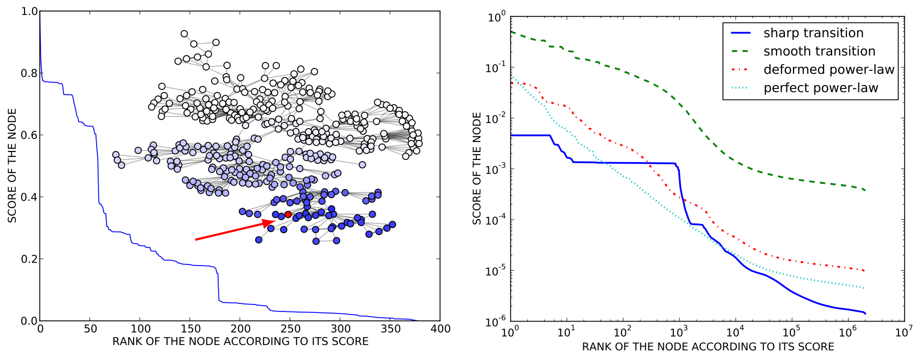
2 How to build multi-ego-centered communities?

The quality function approach suffers from two important drawbacks: (i) designing a good cost function is very difficult, particularly because of a problem of hidden scale parameters. For instance in [FRI11], the quality function, *cohesion*, incorporates a term measuring the density of triangle, which decreases in $O(s^3)$ (where s is the number of selected nodes) in sparse graphs. This thus leads to very small communities in sparse graphs. This problem could be coped by decreasing the effect of this density term, for instance by taking its power a ($a \leq 1$), which is a hidden scale parameter set to one in *cohesion.* (ii) Optimizing the quality function is also very hard because of the highly non-convex nature of the optimization landscape, which leads again to small communities. Indeed, as the optimization is conducted in a greedy way (any other method leading to very slow algorithms), it is thus missing large communities if the algorithms needs to go through lower values of the quality function to reach higher values corresponding to large communities.

In this article we propose a transversal approach to tackle this problem of finding ego-centered communities of a given node. Given a specific node u, we measure the proximity[2] of all nodes in the graph to node u and then try to find irregularities in the decrease of these proximity values, instead of optimizing a quality function. Such irregularities can reflect the presence of one or more communities. More precisely, if there exist a group of nodes that are equally similar to the node of interest, while all other nodes are less similar to it, then this group constitutes a community of the node of interest. Figure 1a exibits three plateaus separated with sharp decreases. When comparing with the graph, we see that the nodes on the two first plateaus correspond to two communities of the selected nodes at different scales. A same behaviour is obtained on figure 1b (for the sharp transition curve) for the large wikipedia network.

However a node often belongs to numerous communities and such a succession of plateaus and decreases is only occasionally observed: given randomly chosen nodes from the Wikipedia network, figure 1b shows the plots of the proximity for all nodes as a function of their ranking. Sharp transitions are seen when communities are well defined, while Smooth transitions occurs when communities are not well defined. Deformed power laws are obtained when several communities are overlapping and clean power laws, i.e., scale free laws, are seen when no scale can be extracted, i.e., lots of communities of various size are overlapping or no community exist at all. Thus this approach to unfold ego-centered communities often fails in the case where the node of interest belongs to many communities: The idea of multi-ego-centered communities solves this problem: although one node generally belongs to numerous communities, two appropriate nodes often fully characterize a single community. For instance, while two colleagues individually belong to many communities

[2] For all experiments we used the proximity introduced in [DAN12] called carryover opinion, however the framework is independent of the chosen proximity



(a) (b)

Fig. 1: Figure 1a shows the results for a co-authorship network [NEW06]: on the drawing of the network, an arrow points to the selected node, and the higher the score, the darker the node. On small graphs a simple linear scale for the plot of the proximities can be used.
Figure 1b shows the result for the wikipedia network, [PAL08], for such a large graph, a logarithmic scale is needed to observe irregularities in the decrease. The curve for 4 randomly chosen nodes are presented in order to show the different observed behaviors.

(colleagues, friends, family, ...), two of them could characterize their professional community (more precisely their community "at work").

To unfold such multi-ego-centered communities, the main idea is that a node belonging to both a community of node1 AND a community of node2 has to be near node1 AND to node2. The example in figure 2 shows how to proceed in two steps: (i) Evaluate for all nodes the proximity to node1 and to node2. (ii) The proximity to the set {node1,node2} can then be given by the minimum of the proximities to node1 and the proximities to node2 [3]. Then, again, if a plateau followed by a decrease is obtained, nodes before the decrease constitute a community. Note that doing this sometimes leads to the identification of a community which does not contain node1 and/or node2, for instance on figure 2, if we had chosen node2 only in a border community and not at the overlap of a border community and the central community, roughly the same result would have been obtained excluding node2. This is not a problem: if we are interested in communities which contain node1 AND node2, we can ignore the unfolded community if it does not contain both nodes.

At the same time, if we are interested in communities containing only node1, we can use node2 as an artifact and keep a community only if it contains node1, regardless of node2.

[3] This quantity measures to what extent a node is near from node1 AND node2. Doing the maximum of the proximities is not relevant for our problem, since this would unfold nodes that are part of a community of node1 OR node2, but doing the product of the scores could work too.

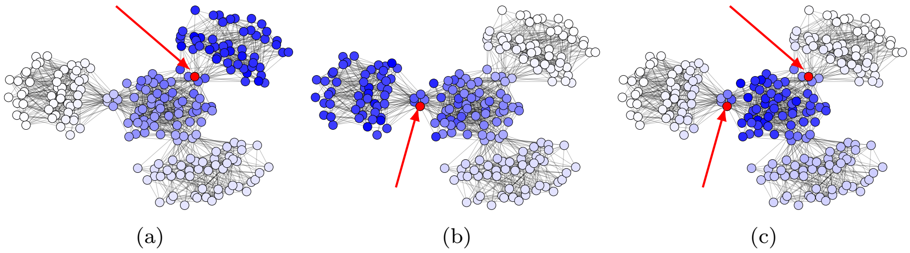


Fig. 2: Result for 4 overlapping Erdos-Renyi graphs of 50 nodes with an edge probability of 0.2 overlapping on 5 nodes. The darker a node, the higher its score. Arrows point to selected nodes on each figure. (2c) gives the (rescaled) minimum of the scores on the experiments presented on (2a) and (2b).

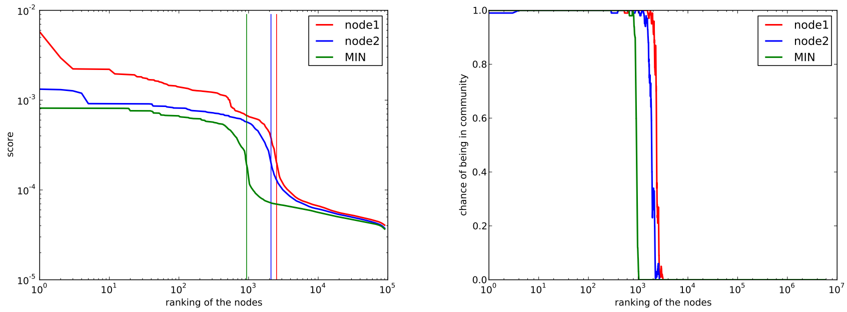
To validate the technique presented here, we extensively tested it and obtained good results on various homemade visualizable networks and on the Lancichinetti and Fortunato's benchmark for overlapping communities [LAN09]. We present here the results for a particular trial on the benchmark: we built a network of 100,000 nodes with 10,000 nodes belonging to 3 communities and the others belonging to only one community, we used a mixing parameter of 0.2 and kept default values of power law coefficients for the degrees distribution and sizes of communities distribution. We picked two nodes belonging to three communities, one of each common to both of them. The results are presented on figure 3: as we can see the unions of the three communities for both nodes is identified almost perfectly as is the community shared by both nodes. Indeed the Jaccard coefficient between the real communities and the one unfolded by the framework is always greater than 0.9.

Even thought any proximity measure can be used, we use for all experiments the carryover opinion which is a proximity measure based on opinion dynamics and first introduced in [DAN12]. Its value for a given node and all other nodes in the graph can be obtained very efficiently (in empirical linear time in real-world graphs) using a fix point algorithm. It consists in repeating the following three steps until convergence, for a given node of interest u:

$$\begin{array}{rcll} C^t(u) & = & MC^{t-1}(u) & \text{AVERAGING} \\ C^t(u) & = & \frac{C^t(u)-min(C^t(u))}{1-min(C^t(u))} & \text{RESCALING} \\ C^t_u(u) & = & 1 & \text{RESETTING} \end{array}$$

where,

- $C^t(u)$ is the score vector after t iterations for a starting node u and the component j of the vector $C^t(u)$ is noted $C^t_j(u)$.
- $C^0(u)$ is set to the null vector, except for the node of interest, u, which is set to one: $C^0_u(u) = 1$.



(a) (b)

Fig. 3: Figure 3a shows the proximities of all nodes as a function of their ranking for the two nodes having three communities while sharing one (node 1 and node 2). It also shows the minimum of these two scores for all nodes as a function of the ranking (MIN). The highest slope of each curve is identified by a vertical bar. Figure 3b shows the proportion of nodes (on a sliding window containing 100 nodes) actually in one of the three communities, as well as the proportion of nodes actually in the shared community, as a function of the same rankings.

- M is the averaging matrix, i.e., the transposed of the transition matrix: $M_{kl} = \frac{l_{kl}}{d_k}$, where l_{kl} is the weight of the link between the nodes k and l, and d_k is the degree of node k.

The vector $C^{\infty}(u)$ is called the carryover opinion of node u, since repeating the steps without rescaling is a simple opinion dynamics process and results in a value of 1 for all nodes. The rescaling allows to capture the proximity of nodes to the node of interest, which is carried over the whole process.

Concerning the computation time of the carryover opinion (i.e. the proximity between a given node in the graph and all other nodes), the AVERAGING, RESCALING and RESETTING steps need $O(m)$ time, $O(n)$ time and $O(1)$ time respectively. This leads to a total computation time of $O(tm)$ where t corresponds to the number of iterations to obtain convergence. In practice, we expect t to be small. For instance, for the wikipedia dataset which contains more than 2 million nodes and 40 million edges, t is about 300, leading to a running time of few seconds on an average laptop.

3 Framework

We use this notion of multi-ego-centered community and propose here an algorithm that, given a graph and a node (noted node1) in the graph will automatically unfold communities ego-centered on that node and label them.

Calculating the proximity from node1 can lead to a curve with sharp transitions and thus the identification of well defined communities containing node1, as seen on figure 1a and 1b (for the sharp transition curve). However, most of the time a power law is obtained meaning that node1 is part of many different and overlapping communities. In that case, we suggest to choose a set of candidate nodes and then see, for each one of them, if there is a well defined bi-ego-centered community containing node1 that emerges , i.e., if there is a sharp transition by doing the minimum of the proximity scores obtained for node1 and one of the candidate.

3.1 How to chose the candidates for node2?

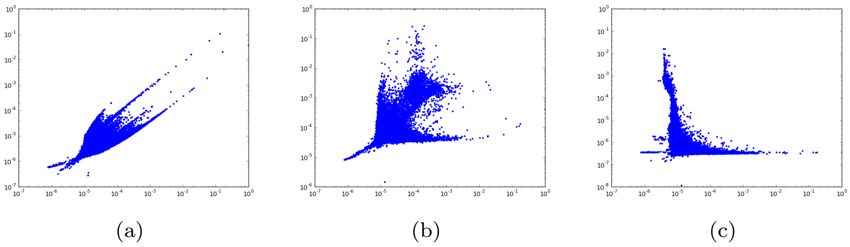
First, the proximities to node1 has to be computed. This gives a real value for each node present in the graph. Sorting these obtained values and plotting them as a function of their ranking leads to the proximity curve. If the outcome is a power-law, there is no relevant scale and node1 certainly belongs to several communities of various sizes.

We then want to pick node2 such that node1 and node2 roughly share only one community. If node2 is very dissimilar to node1 then it is very unlikely that node1 and node2 will share a common community: computing the minimum of the proximities to node1 and the proximities to node2 will lead to very small values. Indeed if the two nodes share no community, at least one of the scores will be very low.

Conversely if node2 is extremely near to node1 then the two nodes will share many communities. The proximity values obtained from node1 and node2 will be roughly the same and doing the minimum will not give more information leading to also roughly the same values.

Thus node2 must be near enough to node1, but not too near. Thus, it has to have a proximity obtained from node1 not too high and not too low. A low and high proximity threshold can be manually tunned to select all nodes at the right distance. Figure 4, shows the correlations between the proximities obtained for two nodes that are very near, figure 4a, two nodes that are at the right distance, figure 4b and two nodes that are very far, figure 4c. We can see that if the two nodes are two near, the scores are very correlated: they are almost the same aligning on the plot, while if the two nodes are too far every node in the graph is far from at least one of the two nodes. On the contrary, if the nodes are at the right distance, a majority of nodes are far from at least one of the two nodes, but some nodes are near to both nodes and can be isolated.

It is quite likely that many of these nodes at the right distance will lead to the identification of the same community, therefore not all of them need to be candidates, a random selection of them can be used if the running time of the algorithm matters. More precise selection strategies could be imagined and we will discuss this point in the future work section.



(a) (b) (c)

Fig. 4: These plots shows the correlation of the proximities obtained for two nodes that are very near, figure 4a, two nodes that are at the right distance, figure 4b and two nodes that are very far, figure 4c.

3.2 How to identify the ego-centered community of node1 and node2?

In order to identify the potential community centered on both node1 and node2, the proximity for node2 has to be computed. Then, for each node in the graph we must compute the minimum of the proximity values obtained from node1 and from node2. For a given node, noted node3, the minimum of the two scores is therefore a measure of the belonging of node3 to the community of both node1 and node2. We can then sort these minimum values and plot the minimum proximity curve. As before, an irregularity in the decrease, i.e., a plateau followed by a strong decrease, indicates that all nodes before the decrease constitute a community.

Detecting a plateau followed by a strong decrease can be done automatically. For instance, if the maximum slope is higher than a given threshold then a plateau/decrease structure is detected and the nodes before this maximum slope constitute a community. This threshold should be manually tuned. If there are several sharp decreases, we only detect the sharpest, this could be improved in the future.

In addition, if node1 is before the decrease then node1 is in the community. In that case, these nodes before the decrease constitute a community of node1. Note that node2 does not need to belong to this community since we are trying to find communities around node1 and that node2 is only a node that we use to find such communities.

As such this method is not very efficient when the proximity is computed from a very high degree node connected to a very large number of communities. In that case, the proximity tends to give high values to every node in the graph and doing the minimum with the scores obtained from a less popular node, which gives lower values to the nodes, will simply result in the values obtained with this second node. A rescaling before doing the minimum can fix the problem. In fact the lowest values reached by the proximities results in a plateau, rescaling (in logarithmic scale) the values such that these plateaus are at the same level solves this problem.

3.3 Cleaning the output and labeling the communities

The output of the two previous steps is a set of communities (where each node is scored), each candidate node can yield a community if the minimum exhibits a plateau. These communities need to be post-processed, since many of them are very similar.

We propose to clean the output as follows: if the Jaccard similarity [4] between two communities (or any other similarity measure between sets) is too large, it means that the communities are actually the same, they appear to be different because of the noise. In that case we only keep the intersection of these two communities. For each node in this new community (the intersection), the score is given by the sum of the scores in the removed communities.

We perform a final cleaning step, which is optional but gives better results: if a community is dissimilar to all other communities, we simply remove it. Indeed, a good community should appear for different candidate nodes. We observed that in general such communities come from the detection of a plateau/decrease structure which actually does not exist (it can happen if for instance the threshold is not set to a proper value).

We then label the community with the label of the best ranked node in the community, i.e., the node whose sum of values is the highest. If two communities have the same label we suggest to keep them adding an index (it can correspond to community at different scale).

We finally obtain a set of labeled communities which are not too similar. We will now show some results on a real network.

4 Results on Wikipedia

Because of size limitation, we will detail here the result for a single node, the wikipedia page entitled *Chess Boxing* [5]. This page exhibits good results which are easily interpretable and can be easily validated by hand.

For the node "Chess Boxing", the algorithm detailed in the previous part, iterated over 3000 nodes chosen randomly from the nodes between the 100^{th} and the 10.000^{th} best ranked nodes leads to the identification of 770 groups of nodes, figure 5 shows 3 examples of trials leading to the identification of a group along with an unsuccessful trial.

Figures 6a shows the jaccard similarity matrix of the 770 unfolded communities before performing intersections. The columns (and lines) of the matrix have been rearranged (using kmeans, considering the columns as vectors) so that columns corresponding to similar groups are next to each-other. We see that there are 716 communities very similar to each-other, while not similar to the other ones (the big white square). The intersection of these communities gave a final community labeled "Queen's Gambit" (labels and content will be

[4] For two sets A and B, the Jaccard similarity is given by $Jac(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

[5] ChessBoxing is a sport mixing Chess and Boxing in alternated rounds.

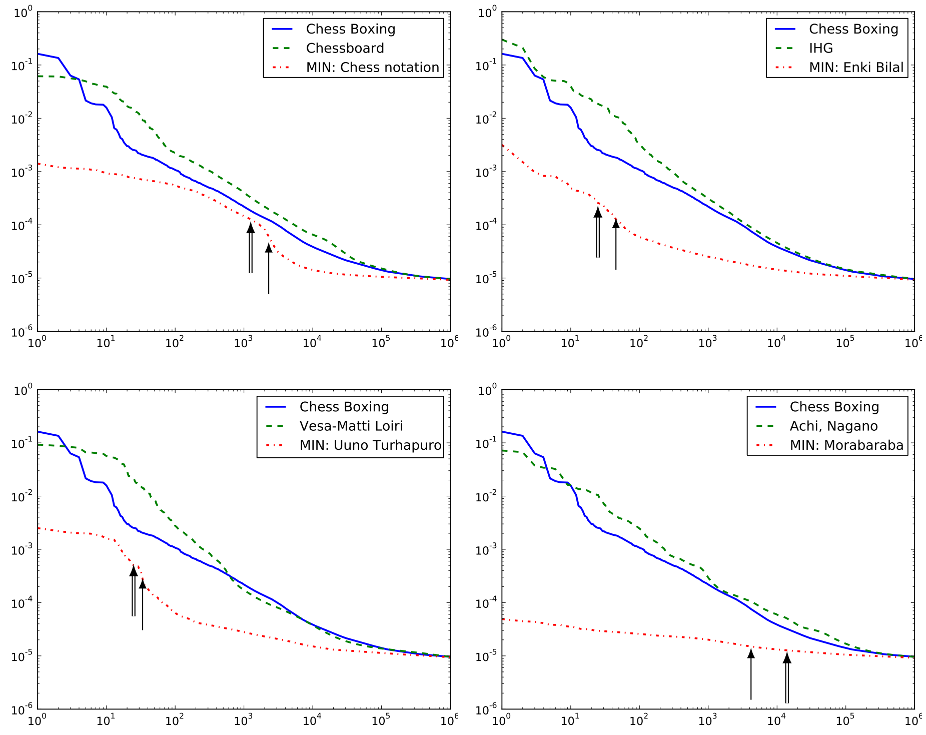


Fig. 5: Each figure shows the curves corresponding to a trial: the y axis represents the scores of the nodes and the x axis represents the ranking of the nodes according to their scores. The first (resp. second) curve represents the proximities from the node Chess Boxing (resp. a candidate for node2, the legend shows the label of the candidate), while the third curve shows the minimum, the label of the first classed node is in the legend. The double arrow shows the position of the node Chess Boxing, while the simple arrow shows the position of the sharpest detected slope.

explained hereafter, see table 1). If the candidate for node2 is in or around a large communities, we will have chance to unfold it, and it increases with the size of the community. A problem of the algorithm is that if very large communities exist, the algorithm can have some difficulty to unfold other small communities. We will come back to that problem in the future work section.

When zooming on the rest of the matrix, figure 6b we see 4 medium size groups of communities, the communities within each of these groups are very similar, but not similar to the rest of the communities, they correspond to the groups leading to communities labeled “Enki Bilal”, “Uuno Turhapuro”, “Da Mystery of Chessboxin’ ” and “Gloria” (see table 1). We also see 6 groups containing only a single community and not similar to any other communities, these are actually mistakes of the plateau/decrease detection part of the algorithm and are automatically deleted during the cleaning step.

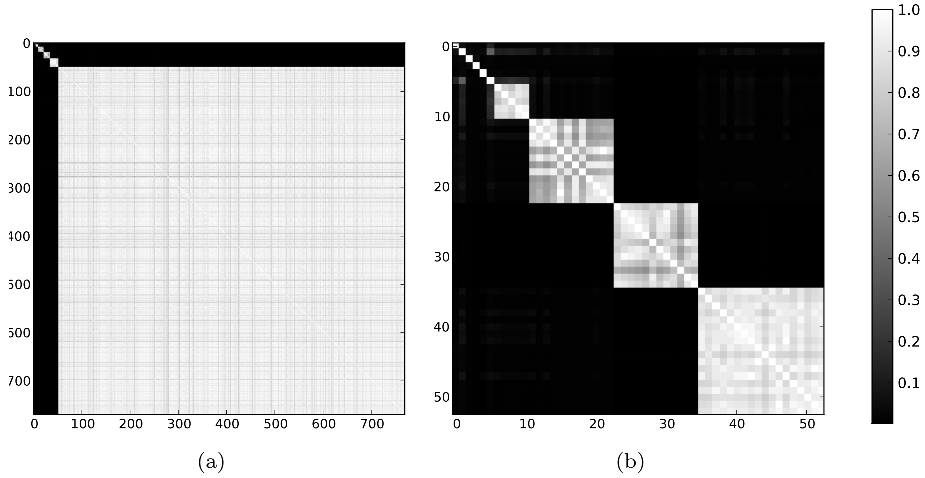


Fig. 6: Figure 6a is the jaccard similarity matrix of these 770 communities (the columns and lines of the matrix have been rearranged so that columns corresponding to similar groups are next to each-other). Figure 6b shows a zoom on the top left corner of the matrix, i.e., after removing the communities in the big group.

This decomposition in 5 main groups is easily obtained by intersecting similar groups (we used a jaccard similarity threshold of 0.7 [6]), while the other six groups are simply deleted. The labels and sizes of the groups, together with example nodes within these groups are presented in table 1. As we can see the algorithm identifies groups with very different sizes (from 26 nodes to 1.619 nodes on this example) which is a positive feature since other approaches are quite often limited to small sized communities.

Some labels are intriguing, however by checking their meaning on wikipedia on-line, all of them can be justified very easily:

- Enki Bilal is a French cartoonist, and its wikipedia page explains that "Bilal wrote [...] Froid Équateur [...] acknowledged by the inventor of chess boxing, Iepe Rubingh as the inspiration for the sport". The nodes in this group are mostly composed of its other cartoons.
- Uuno Turhapuro, is a Finnish movie character, while we cannot see anything about Chess Boxing on its wikipedia page, we can learn on the wikipedia page of Chess Boxing that Uuno Turhapuro is, as Enki Bilal,

[6] if A and B have the same size then, $jac(A, B) > 0.7$ iff A and B overlap at more than 82.3%. If $A \subset B$ then, $jac(A, B) > 0.7$ iff $|A| > 0.7|B|$

Label	Size	Examples of nodes
Enki Bilal	35	Rendez-vous à Paris, Exterminateur 17, Iepe Rubingh, Le Sommeil du monstre, White Birds, The Black Order Brigade, Froid-Équateur, La Foire aux immortels, Fabrice Giger, Goran Vejvoda
Uuno Turhapuro	26	Uuno Turhapuro kaksoisagentti, Uuno Turhapuro (film), Professori Uuno D.G. Turhapuro, Simo Salminen, Jori Olkkonen, Funny-Films Oy, Uuno Epsanjassa, Marjatta Raita, Chess boxing
Da Mystery of Chessboxin´	254	Wu-Tang Clan, Protect Ya Neck, Legend of the Wu-Tang Clan, Grandmasters (album), Gravel Pit, Wu-Tang Clan discography, Shame on a Nigga, Mr. Xcitement, U-God, Masta Killa, N-Tyce
Gloria	55	Gloria (Poulenc), Mass in E Flat Major, Gloria D. Miklowitz, Desiree Casado, Gloria (1999 film), Gloria (Disillusion album), Gloria, Oriental Mindoro, Gloria (singer), Chess boxing, Mass in F Minor
Queen's Gambit	1619	Checkmate, Fast chess, Baltic Defense, Symmetrical Defense, Closed Game, Marshall Defense, Tarrasch Defense, Torre Attack, Chigorin Defense, Chess handicap, Blindfold chess, Chess notation

Table 1: Labels of the communities, sizes of the communities and examples of nodes in the communities (best ranked nodes avoiding the one with a too long label) for the framework run on the "Chess Boxing" Wikipedia page. The meaning, relevance of the labels and the examples can be checked directly on wikipedia.

also acknowledged as the inspiration of the sport, with a scene "where the hero plays blindfold chess against one person using a hands-free telephone headset while boxing another person".

– "Da Mystery of Chessboxin' " is a song by an American rap group: "The Wu-Tang Clan". The nodes in the communities are related to the group and rap musik. It is therefore also relevant.
– "Gloria" is a page of disambiguation linking to many pages containing Gloria in their title. The current wikipedia page of "Chess Boxing" contains the sentence "On April 21, 2006, 400 spectators paid to watch two chess boxing matches in the Gloria Theatre, Cologne". However there is no hyperlink to the page "Gloria Theatre, Cologne" which is a stub. Looking at the records of wikipedia, we found that a link towards the page Gloria was added to the page "Chess Boxing" on May, the 3 2006 and then removed on January, the 31 2008. Due to the central nature of the page "Gloria" within the Gloria community, "Chess Boxing" was part of the Gloria community between these two dates, i.e., when the dataset was compiled!
– Finally, "Queen's Gambit" is a famous Chess opening, the community is composed of Chess related nodes. Even though we would have liked to label this community "Chess", "Queens' Gambit" is very specific to chess and thus characterizes this community very well.

Surprisingly, the algorithm did not find any community related to boxing. This could be a mistake due to the algorithm itself, however the wikipedia page of "Chess Boxing" explains that most chess boxers come from a chess background and learn boxing afterwards. They could thus be important within the community of Chess, but less important within the boxing community. Therefore this could explain that the node "Chess Boxing" lies within the community of Chess, but is at the limit of the boxing community.

5 Comparison to other approaches

5.1 Baselines

We compared our results to two baselines:

1. We have considered all vertices at a distance inferior to 2 from the "Chess Boxing" node; we have removed the "Chess Boxing" node and have run Louvain on the induced subgraph: this induced subgraph contains 0.5 million nodes and more than 10 million edges. The 15 communities obtained by Louvain algorithm are huge and do not seem relevant in the context of Chess Boxing. Even when looking at the 610 communities obtained at the lowest hierarchical scale of Louvain partition, many communities are irrelevant to Chess Boxing. Using a threshold on the distance appears to be not discriminative enought and leads to irrelevant communities.
2. In order to decrease the size of the induced subgraph and select only more relevant nodes, we have computed the carryover opinion from the "Chess Boxing" node and selected only the 5000 first nodes. Even though the threshold of 5000 nodes is a bit random, the selection of relevant nodes is more discriminative than in the previous baseline and the nodes are thus more related to chess boxing. This selection of 5000 nodes has led to better results with more relevant communities: among the 15 unfolded communities, some of them are similar to the ones detected by our framework, for instance, we obtained communities of pages related to chess, dealing with comics or dedicated to rap music. While this can seem interesting, two problem remains: (i) the way the communities are organized, for instance, pages related to chess are splitted into several communities, while we may expect only one and (ii) even if some communities seem relevant, many of the 15 communities do not.

5.2 Quality function

As stated in the related work section, there are other methods to find ego-centered communities, all of them based on the optimization of a quality function. We compare here shortly our results to the one of [NGO12] which, we believe, is the most advanced quality function approach since it corrects many drawbacks of previous methods.

Quality function techniques, due to the non-convexity of the optimization problem often lead to small communities, while our approach does not suffer from this drawback. We can indeed check this on the previous example for which the approach of [NGO12] finds only two small communities:

- The first one contains 7 nodes: Comic book, Enki Bilal, Cartoonist, La Foire aux immortels, La Femme Piège, Froid-équateur and Chess boxing. This community is strikingly similar to our community labeled "Enki Bilal" and is very relevant.
- The second one contains 5 nodes: Germany, Netherlands, 1991, International Arctic Science Committee and Chess boxing. This second community is not similar to any of the communities we found and does not seem to be particularly relevant.

6 Conclusion and perspectives

We introduced an algorithm which, given a node, finds communities ego-centered on that node. Contrary to other existing algorithms our algorithm does not follow an "optimization of a quality function approach", but rather searches for irregularities in the decrease on the values of a proximity measure and leads to the detection of communities of various sizes. It also finds a practical use of the concept of multi-ego-centered communities. The algorithm is time efficient and is able to deal with very large graphs, we validated the results on a practical example using a real very large graph of wikipedia pages.

The algorithm is already very good, however many features can be improved. For instance the detection of irregularities finds only the sharpest decrease, it would be good to use a better detection algorithm to find all relevant irregularities, which would give multi-scale communities.

Furthermore, the algorithm is only looking for bi-centered communities, and maybe some communities can appear only when centered on 3 or more nodes, it would be good to incorporate this feature, however it will increase the running time of the algorithm, especially because of unsuccessful trials. More advanced selection of candidates needs thus to be developed. We could for instance add the following selection feature: if a candidate is chosen for node2, nodes too similar to this candidate might be neglected since they would probably lead to the same result. The speed of the algorithm is, in fact, a very important feature: it is central to make it practical for dynamical communities which exhibit rich behaviors as we saw for the Gloria community.

As we saw the algorithm can have some difficulties to find very small communities if there exist very big communities. This might be the reason why when applied on a globally popular node, like "Biology" or "Europe", the algorithm is only returning one very big community, while we expect to have the communities of various sub-field of Biology or European country related topics. This is a feature of the algorithm that should be improved: relaunching the algorithm again on the induced subgraph of the nodes belonging to the big communities detected, or removing the nodes belonging to the big communities from the graph and running the algorithm again are to be investigated.

Acknoledgement

This work is supported in part by the French National Research Agency contract CODDDE ANR-13-CORD-0017-01.

References

[FOR10] Santo Fortunato. Community detection in graphs. Physics Reports 486, 75-174 (2010)

[PAL05] Palla, G., I. Derenyi, I. Farkas and T. Vicsek. 'Uncovering the overlapping community structure of complex networks in nature and society'. Nature 2005.

[EVA09] T.S. Evans and R. Lambiotte. 'Line Graphs, Link Partitions and Overlapping Communities'. Phys.Rev.E 80 (2009) 016105, DOI: 10.1103/PhysRevE.80.016105.

[BLO08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte and Etienne Lefebvre. 'Fast unfolding of communities in large networks'. J. Stat. Mech. (2008).

[GIR02] M. Girvan and M. E. J. Newman. ‘Community structure in social and biological networks’. PNAS June 11, 2002, *Biometrika*, vol. 99 no. 12, pp. 7821-7826.

[CLA05] Aaron Clauset. 'Finding local community structure in networks'. PHYSICAL REVIEW E 72, 026132, 2005.

[LUO06] F. Luo, J. Z. Wang, and E. Promislow. ‘Exploring local community structure in large networks’. In WI06., pages 233239, 2006.

[CHE09] Jiyang Chen, Osmar R. Zaiane and Randy Goebel. 'Community Identification in Social Networks'. Local 2009 Advances in Social Network Analysis and Mining.

[NGO12] Blaise Ngonmang, Maurice Tchuente, and Emmanuel Viennet. 'Local communities identification in social networks'. Parallel Processing Letters, 22(1), March 2012.

[FRI11] Adrien Friggeri, Guillaume Chelius, Eric Fleury. 'Triangles to Capture Social Cohesion'. IEEE (2011).

[SOZ10] Sozio, Mauro and Gionis, Aristides. 'The community-search problem and how to plan a successful cocktail party'. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 939–948, 2010, ACM.

[NEW06] MEJ Newman. 'Finding community structure in networks using the eigenvectors of matrices'. Physical Review E, 2006, APS.

[PAL08] Gergely Palla, Illes J. Farkas1, Peter Pollner, Imre Derenyi and Tamas Vicsek. ‘Fundamental statistical features and self-similar properties of tagged networks’. New J. Phys. 10 123026 (2008).

[LAN09] Lancichinetti, Andrea and Fortunato, Santo. 'Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities'. Physical Review E, 80, 1, 016118, 2009, APS.

[DAN12] M. Danisch, J.-L. Guillaume and B. Le Grand. Towards multi-ego-centered communities: a node similarity approach. Int. J. of Web Based Communities (2012).

[DAN13] M. Danisch, J.-L. Guillaume and B. Le Grand. Unfolding Ego-Centered Community Structures with A Similarity Approach. Complex Networks IV, 2013, pages 145153, Springer.

[TON06] Tong, Hanghang and Faloutsos, Christos. Center-piece subgraphs: problem definition and fast solutions. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 404–413, 2006, ACM.

[KOR07] Koren, Yehuda and North, Stephen C and Volinsky, Chris. Measuring and extracting proximity graphs in networks. ACM Transactions on Knowledge Discovery from Data (TKDD), 1, 3, 12, 2007, ACM.

[TAT13] Tatti, Nikolaj and Gionis, Aristides. Discovering Nested Communities. Machine Learning and Knowledge Discovery in Databases, 32–47, 2013, Springer.