Unfolding ego-centered community structures with “a similarity approach”

Maximilien Danisch[1], Jean-Loup Guillaume[1] and Bénédicte Le Grand[2]

[1] LIP6, Université Pierre et Marie Curie, 4 Place Jussieu, 75005 Paris, France
[2] CRI, Université Paris 1 Panthéon-Sorbonne. 90 rue de Tolbiac, 75013 Paris, France

Abstract. We propose 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 similarity measure. It is a practical use of the notion of multi-ego-centered community and we validate the pertinence of the approach on a real-world network of wikipedia pages.

1 Context and related work

Many real-world complex systems, such as social or computer networks can be modeled as large graphs, called complex networks. Because of the increasing volume of data 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[3] and their detection has been at the center of this research. For an extensive survey on community detection, we refer to [FOR10].

Communities are clearly overlapping in real world systems, especially in social networks, where every individual belongs to various communities: family, colleagues, 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, in particular the Louvain method [BLO08] which optimizes the *modularity* [GIR02] in an agglomerative fashion, and Infomap [ROS08].

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

[3] Groups of nodes very connected to one-another, but loosely connected to the outside.

step (by adding or removing a node from the group) a given quality function, see [CLA05,CHE09,FRI11,NGO12].

However this 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 find ego-centered communities of a given node which we will detail next. We show the result of our method when applied to a real large graph: the whole wikipedia network containing more than 2 million labeled pages and 40 million edges hyperlinks [PAL08].

2 Framework

Given a specific node u, we measure the similarity[4] of all nodes in the graph to u and then try to find irregularities in the decrease of these similarity values, instead of optimizing a quality function. Such irregularities can reflect the presence of one or more communities. More precisely, if there exists a group of nodes that are equally similar to the node of interest, while all other nodes are less similar to it, then sorting in decreasing order and plotting these similarity values will lead to two plateaus separated by a strong decrease. The nodes before the strong decrease constitute a community of u. However this routine often leads to a power-law with no plateau and from which no scale can be extracted; this happens when lots of communities of various sizes are overlapping which is often the case. To cope with this problem, we use the notion of multi-ego-centered community, i.e., centered on a set of nodes instead of a single node. The key idea is that, although one node generally belongs to numerous communities, a small set of appropriate nodes can fully characterize a single community.

We thus need to smartly pick another node, v, evaluate the similarity of all nodes in the graph to v and then for each nodes in the graph, compute the minimum of the score obtained from u and the score obtained from v: this minimum evaluates how a node is similar to u AND v. Once again, if there exists a group of nodes that are equally similar to u AND v, while all other nodes are less similar to it, then this group can constitute a community. Note that doing

[4] Even though other similarity measures can be used, we use the carryover opinion introduced in [DAN12].

this sometimes leads to the identification of a community which does not contain u and/or v, however since we are interested only in communities containing u, we use v as an artifact and keep a community only if it contains u, regardless of v. The framework consists in doing this for enough candidate nodes v in order to obtain all communities of u. We will now detail the steps of the framework.

2.1 How to chose the candidates for v?

First, the Carryover opinion of u has to be computed [DAN12]. This gives a real value for each node present in the graph: its similarity to u. Sorting the obtained values and plotting them as a function of their ranking leads to the carryover curve. If the outcome is a power-law, there is no relevant scale and u certainly belongs to several communities of various sizes.

We then want to pick v such that v and u roughly share exactly one community. If v is very dissimilar from u then it is very unlikely that u and v will share a common community: computing the minimum of the scores obtained from running the carryover opinion from u and the scores obtained from running the carryover opinion from v 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 v is extremely similar to u then the two nodes will share many communities. The carryover opinion values obtained from u and v will be roughly the same and doing the minimum will not give more information.

Thus v must be similar enough to u, but not too similar: it has to have a carryover score obtained from u not too high and not too low. A low and high similarity thresholds can be manually tuned to select all nodes at the right distance in order to fasten the execution.

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 will be discussed in the future work section.

2.2 How to identify the ego-centered community of u and v?

In order to identify the potential community centered on both u and v, we must compute the minimum of the carryover values obtained from u and from v for each node, w, of the graph. The minimum of the two scores is therefore a measure of the belonging of w to the community of both u and v. We can then sort these minimum values and plot the minimum carryover 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: if the maximum slope is higher than a given threshold, 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 u is before the decrease then u is in the community. In that case, these nodes before the decrease constitute a community of u. Note that v does not need to belong to this community since we are trying to find communities around u and that v is only a node that we use to find such communities.

As such this method is not very efficient when the carryover opinion is run from a very high degree node connected to a very large number of communities. In that case, the carryover tends to give high values to every node in the graph and calculating 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. Indeed the lowest values obtained by running the carryover opinion result in a plateau, rescaling (in logarithmic scale) the values such that these plateaus are at the same level solves this problem.

2.3 Cleaning the output and labeling the communities

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

We propose to clean the output as follows: if the Jaccard similarity [5] between two communities (or any other similarity measure between sets) is too large, it means that although 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 (intersection) community, the score is given by the sum of the scores in the original communities.

We perform an optional cleaning step, which enhances the 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 such communities come from the detection of a plateau/decrease structure which does not exist (it often happens when the threshold is not set to a proper value).

We finally 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 both (it can be different scales of the same community).

This algorithm finally gives a set of distincts labeled communities. We now show some results on a real network.

3 Results and validation

Because of size limitation, we focus here on the result for a single node, the wikipedia page entitled *Chess Boxing* [6]. This page exhibits good results which are easily interpretable and can be validated by hand.

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

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

For the "Chess Boxing" node, the algorithm iterated over 3000 nodes chosen at random from the nodes between the 100^{th} and the 10.000^{th} best ranked nodes leads to 770 groups of nodes. Figure 1 shows a successful trial leading to the identification of a group along with an unsuccessful trial.

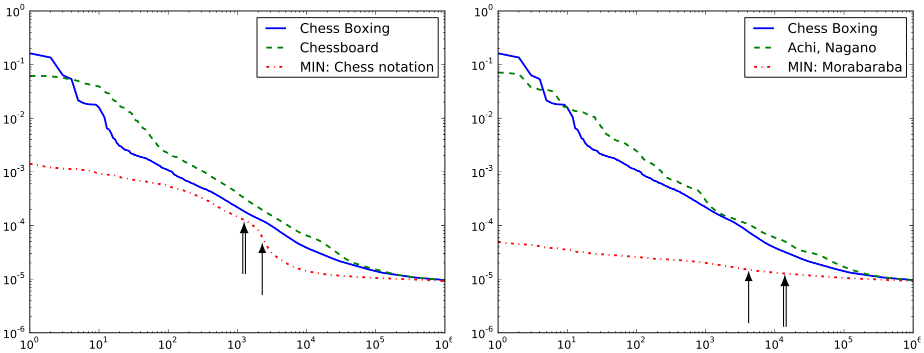
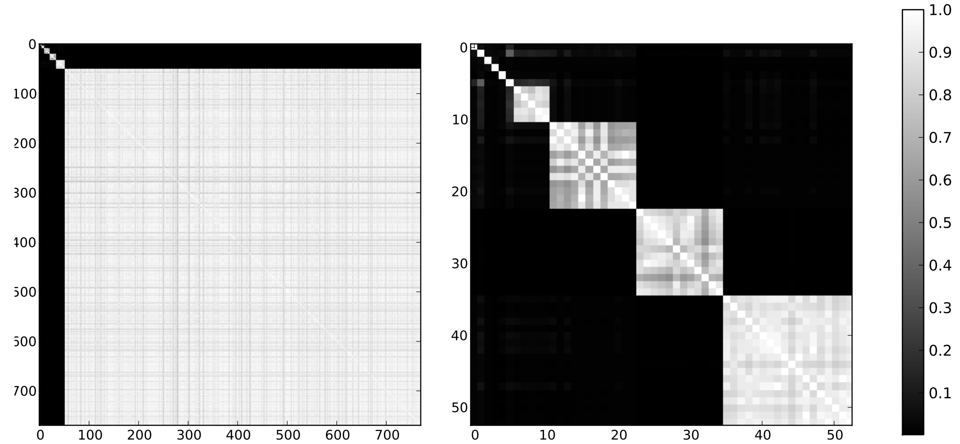


Fig. 1: Each figure shows the curves corresponding to a trial: the y axis represents the scores and the x axis represents the ranking of the nodes according to their scores. The first (resp. second) curve is the carryover opinion run from the node Chess Boxing (resp. a candidate for v, the legend shows the label of the candidate), while the third curve shows the minimum, the label of the first ranked node is in the legend. The first trial is successful, while the second is not (no plateau/decrease structure). The double arrow shows the position of the "Chess Boxing" node, while the simple arrow shows the position of the sharpest slope.

Figures 2a shows the Jaccard similarity matrix of the 770 unfolded communities before cleaning. The columns and lines of the matrix have been rearranged so that columns corresponding to similar groups are next to each other. We see that there are 716 communities very similar to one another, while not similar to the other ones. If u is in or around a large community, we have a high probability to unfold it, and this probability 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 2b, we see 4 medium size groups of communities and 6 groups containing only a single community, these are actually mistakes of the plateau/decrease detection part of the algorithm and these groups are automatically deleted during the cleaning step.

This decomposition into 5 main groups is easily obtained by intersecting similar groups (we used a Jaccard similarity threshold of 0.7, while the other six singleton groups are automatically deleted. The labels and sizes of the 5 groups



(a)

(b)

Fig. 2: Figure 2a is the rearranged Jaccard similarity matrix of these 770 communities. We see that there are 716 communities very similar to one another while not similar to the rest of the communities (the big white square). Figure 2b shows a zoom on the top left corner of the matrix.

are "Enki Bilal" (35 nodes), "Uuno Turhapuro" (26 nodes), "Da Mystery of Chessboxin' " (254 nodes), "Gloria" (55 nodes) and "Queen's Gambit" (1.619 nodes). 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. Wikipedia indicatess 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. It is, as Enki Bilal, 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". The nodes in this group are mostly other cartoons characters or actors in the movies or strongly related to finnish movies.
- "Da Mystery of Chessboxin' " is a song by an American rap band: "The Wu-Tang Clan". The nodes in the community are related to the band and rap music, which is 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 and the community is composed of Chess related nodes. Even though we could 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 "Chess Boxing" node lies within the community of Chess, but is at the limit of the boxing community.

4 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 similarity 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.

Some features of the algorithm can be improved. For instance the detection of irregularities finds only the sharpest decrease, it would be good to find all relevant irregularities, which would give multi-scale communities.

Furthermore, the algorithm is only looking for bi-centered communities, and some communities might 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 thus needs to be developed. We could for instance add the following selection feature: if a candidate is chosen for v, nodes too similar to this candidate might be neglected since they would probably lead to the same result. The speed of the algorithm is a very important feature and is central to make it practical for the study of evolving communities.

As we saw the algorithm can have some difficulties to find very small communities if there exist very large ones. 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-fields 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 should be investigated.

Acknowledgments

This work is supported in part by the French National Research Agency contract DynGraph ANR-10-JCJC-0202.

References

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

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

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

[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)

[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.

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

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

[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.

[GLE03] P. Gleiser and L. Danon. Adv. Complex Syst.6, 565 (2003).

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

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

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

[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)

[ROS08] Martin Rosvall and Carl T. Bergstrom/ Maps of information flow reveal community structure in complex networks PNAS 105, 1118 (2008).