Unveiling Hidden Communities Through Cascading Detection on Network Structures

Jean-Gabriel Young, Antoine Allard, Laurent Hébert-Dufresne, and Louis J. Dubé
Université Laval, Québec (QC) G1V 0A6, Canada,
jean-gabriel.young.1@ulaval.ca
WWW home page: http://www.dynamica.phy.ulaval.ca

Abstract. Community detection is the process of assigning nodes and links in significant communities (e.g. clusters, function modules) and its development has led to a better understanding of complex networks. When applied to sizable networks, we argue that most detection algorithms correctly identify prominent communities, but fail to do so across multiple scales. As a result, a significant fraction of the network is left uncharted. We show that this problem stems from larger or denser communities overshadowing smaller or sparser ones, and that this effect accounts for most of the undetected communities and unassigned links. We propose a generic cascading approach to community detection that circumvents the problem. Using real network datasets with two widely used community detection algorithms, we show how cascading detection allows for the detection of the missing communities and results in a significant drop of the fraction of unassigned links.

Key words: networks, community structure, community detection algorithms, resolution limit

1 Introduction

Over the course of the last decade, network science has attracted an ever growing interest since it provides important insights on a large class of interacting complex systems. One of the features that has drawn much attention is the structure of interactions highlighted by the network representation. Indeed, it has become increasingly clear that global structural patterns emerge in most real networks [1]. One such pattern, where links and nodes are aggregated into larger groups, is called the community structure of a network.

While the exact definition of communities is still not agreed upon [2], the general consensus is that these groups should be denser than the rest of the network. The notion that communities form some sort of independent units (families, friend circles, coworkers, protein complexes, etc.) within the network is thus embedded in that broader definition. It follows that communities represent functional modules, and that understanding their layout as well as their organization on a global level is crucial to a fuller understanding of the system under scrutiny [3] [4].
By developing techniques to extract this organization, one assumes that communities are encoded in the way nodes are interconnected, and that their structure may be recovered from limited, incomplete topological information. Various algorithms and models have been proposed to tackle the problem, each featuring a different definition of the community structure while sharing the same general objective. Although these tools have been used with success in several different contexts \cite{2, 5, 6}, a number of shortcomings are still to be addressed.

In this report, we show how to improve existing algorithms independent of the procedure or the definitions they use. More precisely, we first show that present algorithms tend to overlook small communities found in the neighborhood of larger, denser ones. Then, we propose and develop a \textit{cascading} approach to community detection that greatly enhance their performance.

\section{Resolution limit due to shadowing}

It is known that a resolution limit exists for a large class of community detection algorithms that rely on the optimization of a quality function (e.g., modularity \cite{1}) over non-overlapping partitions of the network \cite{7}. Indeed, it appears that the size of the smallest detectable community is related to the size of the network. This leads to counterintuitive cases where clearly separated clusters of nodes are considered as one larger community because they are too small to be resolved by the detecting algorithm. A possible solution could be to conduct a second analysis on all detected communities in order to verify that no smaller modules can be identified.

However, the optimal partition of a network should include overlapping communities, as they capture the multiplicity of functions that a node might fulfill since nodes can then be shared between many communities. We argue that a different resolution limit, due to an effect that we refer to as \textit{shadowing}, arises in detection algorithms that:

1. allow such overlapping communities;
2. rely on some global resolution parameter.

Shadowing typically occurs when large/dense communities act as screens hence preventing the detection of smaller/sparser adjacent communities. To illustrate this phenomenon, we study two families of detection algorithms based on two different paradigms of community structure, namely nodes and links communities.

\subsection{Clique percolation algorithm}

The clique percolation algorithm (CPA) \cite{6} defines communities as maximal \textit{k-clique} percolation chains, where a \textit{k-clique} is a fully connected subgraphs of \textit{k} nodes, and where a percolation chain is a group of cliques that can be reached from one adjacent \textit{k}-clique to another \cite{8}. The complete community structure

\footnote{Two \textit{k}-cliques are said to be adjacent if they share \textit{k} – 1 nodes.}
Fig. 1. Shadowing effect for the CPA. (a) The yellow region is the sole detectable community with \( k = 4, 5 \), while its union with the black region corresponds to the community detected with \( k = 3 \). This pathological example illustrates the two undesirable extreme effects mentioned in the main text: either most of the network is detected as a single community, or only large and dense clusters are detected. No optimal value of \( k \) can be found in this case. (b) The structure of this subgraph nevertheless suggests that it could be decomposed in a dense community in the middle, surrounded by smaller communities. We see that if the links involved in the dense community (detected with \( k = 4 \) or 5) were removed, a second iteration of the algorithm with \( k = 3 \) would lead to the detection of several smaller communities that were overshadowed by the larger one. This illustrates the essence of the cascading detection method discussed in Sec. 3.

is obtained by detecting every maximal percolation chains for a given value of \( k \).

It is noteworthy that the definition of a community in this context is consistent with the general description of communities outlined in Sec. 1. Indeed, \( k \)-clique percolation chains are dense by definition, and a sparser neighboring region is required to stop a \( k \)-clique percolation chain, ensuring that communities are denser than their surroundings. We expect shadowing as both conditions listed in Sec. 1 are met:

1. since percolation chains—communities—consist of \( k \)-cliques sharing \( k - 1 \) nodes, overlapping communities occur whenever two cliques share less than \( k - 1 \) nodes;
2. the size of the cliques, \( k \), acts as a global resolution parameter.

Let us explain this last point. In principle, low values of \( k \) lead to a more flexible detection of communities as a smaller clique size allows a wider range of configurations. However, low values of \( k \) often yield an excessively coarse-grained community structure of the network since percolation chains may grow almost unhindered and include a significant fraction of the nodes. In contrast, large values of \( k \) may leave most of the network uncharted as only large and dense
clusters of nodes are then detected as communities. An optimal value corresponding to a compromise between these two extreme outcomes must therefore be chosen. As this value of $k$ attempts to balance these two unwanted effects for the entire network as a whole, a shadowing effect is expected to arise causing the algorithm to overlook smaller communities, or to merge them with larger ones. See Fig. 1 for an illustration of this effect.

2.2 Link clustering algorithm

The link clustering algorithm (LCA) aggregates links—and hence the nodes they connect—into communities based on the similarity of their respective neighborhood. Denoting $e_{ab}$ the link between nodes $a$ and $b$, the similarity of two adjacent links $e_{ik}$ and $e_{jk}$ (attached to a same node $k$ called the keystone) is quantified through a Jaccard index

$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|},$$

(1)

where $n_+(q)$ is the set of node $q$ and its neighbors, and $|n_+(q)|$ is the cardinality of the set. Figure 2 illustrates the calculation of $S(e_{ik}, e_{jk})$. Once the similarity has been calculated for all adjacent pair of links, communities are built by iteratively aggregating adjacent links whose similarity exceeds a given threshold $S_c$. We refer to links that are left after this process (i.e., communities consisting of one single link) as unassigned.

For the purpose of this study, we use the lowest value of $k$ such that no extensive community is detected. As suggested in [6], the largest community is considered extensive if it contains about twice as many nodes as the second largest community. In networks with large unbreakable cliques (Internet and Protein networks), a lower ratio is used (0.25 and 0.31, respectively).
The pairwise unions of the three sets \( n_+ (a) \), \( n_+ (b) \) and \( n_+ (c) \) contain considerably more elements that their corresponding intersections since nodes \( a \), \( b \) and \( c \) all have high degrees. According to Eq. 1 this implies that \( e_{ab} \), \( e_{bc} \) and \( e_{ac} \) share lower similarities—namely \( S(e_{ac}, e_{bc}) = S(e_{ab}, e_{bc}) = 3/22 \) and \( S(e_{ab}, e_{ac}) = 3/17 \)—than if the triangle had been isolated (see Sec. 3). It is therefore likely that these three links will be left unassigned.

Again, a shadowing effect is expected, as the two aforementioned conditions are fulfilled:

1. because communities are built by aggregating links, this algorithm naturally allows communities to overlap (to share nodes) since a node can belong to as many communities as its degree (number of links it is attached to) can allow;
2. the similarity threshold \( S_c \) acts as a global resolution parameter as it dictates whether two links belong to the same community or not.

To elucidate the global aspect of \( S_c \), we describe how its value is chosen (as proposed in [5]). Let us first define the density \( \rho_j \) of community \( j \) as

\[
\rho_j = \frac{d_j - \left( n_j - 1 \right)}{\frac{n_j(n_j-1)}{2} - \left( n_j - 1 \right)},
\]

where \( d_j \) and \( n_j \) are the number of links and nodes in community \( j \), respectively. Considering that a community of \( n \) nodes must at least include \( n - 1 \) links, \( \rho_j \) computes the fraction of potential “excess links” that are present in the community. The similarity threshold \( S_c \) is then chosen such that it maximizes the overall density of the communities

\[
\rho(S_c) = \frac{1}{D} \sum_{j \in C(S_c)} d_j \rho_j
\]

where \( C(S_c) \) is the set of communities detected for a given \( S_c \), \( D \) is the total number of links assigned to communities of more than one link (i.e., \( d_j > 1 \)). Note that \( \rho(S_c) \) is typically a well-behaved function of \( S_c \) and normally displays a single maximal plateau [5]. The value of \( S_c \) corresponding to this plateau is then selected as it leads, on average, to the denser set of communities, hence its global nature.
Following an analysis similar to Sec. 2.1, we expect small communities to be left undetected as they are eclipsed by larger and denser ones. This is mainly due to the use of a resolution parameter \( S_c \) that cannot be adjusted locally. For instance, links in a small community could exhibit vanishing similarities because some of the associated nodes are hubs (nodes of high degree). This is especially true in the vicinity of large and dense clusters whose nodes are typically of high degree (see Fig. 3 for an illustration).

3 Cascading detection

Figures 1 and 3 suggest that the inability to detect small or sparse communities in the vicinity of larger or denser ones—the shadowing effect—could be circumvented by removing these structures from the networks. We formalize this idea and propose a cascading approach to community detection that proceeds as follows:

1. identify large or dense communities—by tuning the resolution parameter accordingly—using a given community detection algorithm;
2. remove the internal links of the communities identified in step 1;
3. repeat until no new significant communities are found.

The first iteration of this algorithm detects the communities that are normally targeted by detection algorithms, thus ensuring that the cascading approach retains the main features of the “canonical” community structure. After removal of links involved in the detected communities, a new iteration of the detection algorithm is then performed on a sparser network in which previously hidden communities are now apparent. This process is repeated until a final and more thorough partition of the network into overlapping communities is eventually obtained.

For example, in the case of the CPA, a high value of \( k \) (which leads to the traditional community structure) is selected for the first iteration of the algorithm. The network then becomes significantly sparser since all cliques of size \( k' > k \) are destroyed by the removal of internal links in step 2. Subsequent iterations of the detection algorithm can thus be conducted at lower \( k \), unveiling finer structures, as the pathways formed by dense cluster are no longer available. The process naturally comes to a halt at \( k = 3 \), since \( k = 2 \) only detects the disjoint components of the network. In the case of the LCA, the detection is stopped before the partition density reaches zero, for \( \rho(S_c) \approx 0 \) only detects chains of links (the keystone ensures a non-vanishing similarity), which in general are not classified as significant communities.

It is worth mentioning that conducting this repeated analysis does not increase the computational cost significantly, since the cascading algorithm scales exactly like the community detection algorithm used at each iteration, and since

\(^3\) Internal links are defined as links that join two nodes belonging to the same community.
Fig. 4. Fraction of remaining assignable links for real networks using the cascading approach. (left) The number of unassigned links after one iteration of the CPA—corresponding to a typical use—is shown in yellow, and the final state is shown in black. Whenever more than 2 iterations were performed, the intermediate results are shown in orange. For the Gnutella network, the optimal value was $k = 3$ at the first iteration, leading to an immediate complete detection of the community structure. (right) Results of a canonical use of the LCA are shown in white and shades of blue correspond to subsequent iterations. The final state is again shown in black. Note that all results are normalized to the number of assignable links in the original network. For the CPA, this corresponds to the number of links that belong to at least one 3-clique. For the LCA, a link is considered assignable if at least one of the two nodes it joins have a degree greater than one.

the number of iterations that can be carried is small (typically less than 10). Moreover, the size of the networks (number of links and nodes) effectively decreases after each iteration, further reducing the cost.

4 Results and discussions

To investigate the efficiency and the behavior of the cascading detection, we have applied our approach to 10 network datasets: arXiv cond-mat circa 2004 (arXiv) [6], Brightkite online (Brightkite) [9], university Rovira i Virgili email exchanges (Email) [10], Gnutella peer-to-peer data (Gnutella) [11], internet autonomous systems (Internet) [4], MathSciNet co-authorship (Mathsci) [12], Pretty-Good-Privacy data exchange (PGP) [13], Western States Power Grid (Power) [14], Protein-protein interactions (Protein) [6] and word associations (Words) [6].

First and foremost, our results show that cascading detection always improves the thoroughness of the community structure detection. Indeed, Fig. 4 shows that while a traditional use of the algorithms yields partitions with high fractions of
unassigned links, the cascading approach leads to community structures where this fraction is significantly reduced. More precisely, the percentage of remaining assignable links drops from 54.1% to 26.3% on average in the case of CPA, and from 41.0% to 5.3% in the case of LCA. Note how cascading detection is more efficient when applied to the LCA. This is due to the fact that the effective network gets increasingly sparser with each iteration, and that link clustering works equally well on sparse and dense networks, whereas clique percolation requires a high level of clustering to yield any results.

Figure 5 confirms that as the cascading detection proceeds, smaller—and previously masked—communities are detected, regardless of the algorithm used. For instance, Fig. 5(c) clearly shows how a significant number of 3-cliques are overlooked by “traditional” use of the CPA. However, large communities are also found after many iterations, suggesting that the shadowing effect is not restricted to small communities.

Visual inspection of the detected communities not only verifies the quality of the hidden communities, but also confirms our intuition of the shadowing effect.

---

Footnote: Links that are not part of any triangles cannot be assigned to a community by the CPA since they cannot take part in any k-clique, whereas the LCA can potentially assign every link to a community, since isolated links were removed from the datasets.
Fig. 6. Sample of the communities detected with the cascading approach on the Words network. The detected communities are shown (red) as well as their neighboring nodes (grey). Red and grey labels identify respectively semantic fields and individual words.
effect. For instance, Fig. 6(a) shows a triangle detected at the third iteration (out of five) of the LCA on the Words network. This structure was most likely missed during the initial detection due to the high degree of its three nodes, as speculated in Fig. 5. Similarly, although \( k = 4 \) was initially chosen (according to the criterion discussed in Sec. 2.1) for the CPA on the Words network, a second iteration using \( k = 3 \) has permitted the detection of other significant communities such as the one shown in Fig. 6(c).

More complex structures and correlations are also brought to light using this approach. Figure 6(b) presents a star of high-degree nodes detected at the third iteration of the LCA on the word association network. None of these nodes are directly connected to each other, but they share many neighbors. Hence, once the main communities were removed—here semantic fields related to toys, theatre and music—the shadow was lifted such that this correlated, but unconnected structure could be detected. Whether this particular structure should be defined as a relevant community is up for debate. Keeping in mind that there are no consensus on the definition of a proper community in complex networks, the role of algorithms, and consequently of the cascading method, is to infer plausible significant structures.

Internal link removal is destructive in the sense that information about shadowed communities is lost in the process, as some of the internal links are shared by more than one community [5]. Leaving these links untouched would certainly enhance the quality of the detected communities while further reducing the uncharted portion of the network. Nevertheless, without using refined algorithm and by only resorting to our simple idea, we obtain surprisingly good results. This suggest that shadowing is not necessarily due to the density of the prominent communities but rather to the stiffness of the resolution parameter. Indeed, by using a cascading approach, we allow this parameter to vary artificially from a region of the network to the other, as the algorithm is effectively applied to a new network – partially retaining the structure of the original network – at each iteration. A once rigid global parameter can now flexibly adapt to small changes in the topology of the network to better reveal subtle structures.

5 Conclusion and perspective

In conclusion, we have managed to significantly reduce the uncharted portion of a network by assigning an important fraction of seemingly random links to relevant communities. This significant improvement in community detection will help shrink the gap between analytical models and their real network counterparts. The difficult problem of accurately modeling the dynamical properties of real networks might be better tackled if one includes complex community structure through comprehensive distributions or solved motifs [15, 16], two applications for which a reliable and complete partition is fundamental.

Moreover, this work opens the way to more subtle cascading approach, as envisioned at the end of the previous section. For instance, we could build an extreme version of the algorithm where communities are detected one by one.
Such an approach would enable a perfect adaptation of the resolution parameter to the situation at hand. And while it would certainly come to significant computational cost, it could lead to the mapping of the community detection problem unto simpler problems. If we accept to detect communities one at a time, the detection of the most significant ones can be done through well optimized methods, such as modularity optimization [17], which would otherwise be incapable of detecting overlapping communities. Finally, perhaps the most significant observation that emerges from our work could be simply stated: since community structure occurs at all scales, global partitioning of overlapping communities must be done sequentially, cascading through the organizational layers of the network.

Appendix A: Summary of the cascading detection results

| Network | CPA_n | CPA_f | # iterations | LCA_n | LCA_f | # iterations |
|---------|-------|-------|--------------|-------|-------|--------------|
| arXiv   | 41.4  | 7.6   | 3            | 22.3  | 2.3   | 7            |
| Email   | 73.4  | 64.5  | 2            | 47.7  | 1.3   | 5            |
| Gnutella| 0     | 0     | 1            | 35.3  | 18.7  | 2            |
| Internet| 92.9  | 78.6  | 2            | 28.2  | 2.7   | 5            |
| PGP     | 46.2  | 6.4   | 2            | 46.3  | 4.5   | 6            |
| Power   | 79.2  | 5.5   | 2            | 63.6  | 4.4   | 3            |
| Protein | 50.5  | 26.4  | 2            | 38.5  | 6.2   | 3            |
| Words   | 49.5  | 21.2  | 2            | 46.1  | 2.6   | 5            |

Table 1. Summary of the results presented in Fig. 4. CPA_n and LCA_n are the percentages of remaining assignable links for a normal use of the CPA and LCA, respectively, whereas CPA_f and LCA_f are the final percentages using a cascading approach. The number of iterations to reach the final state is also given.

Acknowledgments

The authors wish to thank the Gephi [18] development team for their visualization tool; Yong-Yeol Ahn et al. for the link community algorithm; Gergely Palla et al. for the clique percolation algorithm; all the authors of the cited papers for providing the network data; and Calcul Québec for computing facilities. This research was funded by CIHR, NSERC and FRQNT.

References

1. M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” PNAS, 99, 7821–6, (2002).
2. S. Fortunato, “Community detection in graphs,” Phys. Rep., 486, 75–174, (2010).
3. L. Hébert-Dufresne, P.-A. Noël, V. Marceau, A. Allard, and L. J. Dubé, “Propagation dynamics on networks featuring complex topologies,” Phys. Rev. E, 82, 36115, (2010).
4. L. Hébert-Dufresne, A. Allard, V. Marceau, P.-A. Noël, and L. J. Dubé, “Structural Preferential Attachment: Network Organization beyond the Link,” Phys. Rev. Lett. 107, 158702, (2011).
5. Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link Communities Reveal Multiscale Complexity in Networks,” Nature, 466, 761, (2010).
6. G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” Nature, 435, 814–8, (2005).
7. S. Fortunato and M. Barthélemy, “Resolution limit in community detection,” PNAS, 104, 36–41, (2007).
8. I. Derényi, G. Palla, and T. Vicsek, “Clique Percolation in Random Networks,” Phys. Rev. Lett., 94, 160202, (2005).
9. E. Cho, S.A. Myers, & J. Leskovec, "Friendship and Mobility: User Movement in Location-Based Social Networks," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), (2011).
10. R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, & A. Arenas, "Self-similar community structure in a network of human interactions," Phys. Rev. E, 68, 065103, (2003).
11. M. Ripeanu, I. Foster, & A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," IEEE Internet Computing Journal, 6, 50-57, (2002).
12. G. Palla, I.J. Farkas, P. Pollner, I. Derényi, & T. Vicsek, "Fundamental statistical features and self-similar properties of tagged networks," New J. Phys., 10, 123026, (2008).
13. M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, & A. Arenas, "Models of social networks based on social distance attachment," Phys. Rev. E, 70, 056122, (2004).
14. D.J. Watts, & S.H. Strogatz, "Collective dynamics of small-world networks," Nature, 393, 440–2, (1998).
15. A. Allard, L. Hébert-Dufresne, P.-A. Noël, V. Marceau & L. J. Dubé, “Bond percolation on a class of correlated and clustered random graphs,” J. Phys. A, 45, 405005, (2012).
16. B. Karrer and M. E. J. Newman, “Random graphs containing arbitrary distributions of subgraphs,” Phys. Rev. E, 82, 66118, (2010).
17. M. E. J. Newman, “Modularity and community structure in networks,” PNAS, 103, 8577–82, (2006).
18. M. Bastian, S. Heymann & M. Jacomy, “Gephi: An Open Source Software for Exploring and Manipulating Networks,” International AAAI Conference on Weblogs and Social Media, (2009).