Long range community detection

Thomas Aynaud
LIP6 – CNRS – Université Pierre et Marie Curie
4 place Jussieu
75005 Paris, France
thomas.aynaud@lip6.fr

Jean-Loup Guillaume
LIP6 – CNRS – Université Pierre et Marie Curie
4 place Jussieu
75005 Paris, France
jean-loup.guillaume@lip6.fr

***Abstract*—Complex networks can usually be divided in dense subnetworks called communities. In evolving networks, the usual way to detect communities is to find several partitions independently, one for each time step. However, this generally causes troubles when trying to track communities from one time step to the next. We propose here a new method to detect only one decomposition in communities that is good for (almost) every time step. We show that this unique partition can be computed with a modification of the Louvain method and that the loss of quality at each time step is generally low despite the constraint of global maximization. We also show that some specific modifications of the networks topology can be identified using this unique partition in the case of the Internet topology.**

***Index Terms*—complex networks, dynamics, communities, long term communities, modularity, Internet topology**

Introduction

Complex networks are nowadays one major tool in modeling. They arise naturally to describe interactions between objects. For instance, the web is a network of web pages linked through hyperlinks and understanding the shape and the structure of this network may have many applications to search engines or recommendation tools. Various properties of these networks have been described. For example, they often have a low density but behave locally almost like a clique.

A good way to study this particular structure is to consider these networks as composed of several communities. The definition and the detection of these communities can be used in graph visualization [1] and to mine various kind of networks: they can be groups of interest on a social network [2], groups of web pages dealing with the same subject on the web, proteins that share a common function in a metabolic network [3] or modules in a software source code [4] for instance. Formalizing the notion of community and finding them automatically is a difficult task which has attracted many attentions in the recent years. In particular, many algorithms based on the network topology have been proposed (see [5]–[7] for surveys).

However, most of these studies focus on the study of static networks. The usual method is to collect data during a long period and then to aggregate them in a big static network to study. But real data are almost always dynamic: pages appear or disappear on the web or people move and change their relationships for instance, and thus much information which is crucial in understanding phenomena in the networks is lost.

Thus, methods to study evolving networks have been proposed. Most of them see the network as a succession of snapshots representing the state of the network at a given time. Then, communities can be computed on each snapshot but due to the instabilities of the algorithms, tracking the communities across time steps is an important matter [8], [9]. Thus several techniques have been developed to stabilize the algorithms [10], to define communities differently [11]–[13] and to analyze the results [14], [15]. One idea proposed in [16] is to compute communities that are good both at time t and at time $t-1$. We extend this idea here by computing one unique community decomposition that is almost always good, instead of finding good partitions on every snapshot. We will thus try to compute long term communities and while our partition will not be the best at every snapshot, it will be unique and good on average to describe the overall structure of the network. The partition will be fixed but, since not all nodes exist all the time, the actual communities at a given time step will not always be the same.

This paper is organized as follows: in the first section we will present our long range communities definition and one way to effectively find them. Then, we will propose validation techniques and experimental results on a real network and we will finally conclude.

I. Long term communities, definition and detection

A. Definition

We consider an evolving graph as a succession of static graphs, each of them being a snapshot of the state of the network at a given time. To simplify, we will consider that time steps are in the set $\{1, 2, ..., T\}$. We will note $G_i = (V_i, E_i)$ the snapshot that represents the network at time i with V_i its set of nodes and E_i its set of edges.

The classic approach here is to detect T partitions Π_i representing each the best communities for each snapshot. However, most algorithms provide partitions which are drastically different even if the network has been only slightly modified. Instead, we are looking for only one partition of the union of the nodes of every time step that is good in average. As a consequence the quality of this partition at one particular time will be lower than the maximal reachable but we aim to detect one partition that is almost always good instead of being optimal at just one time step.

More formally, the communities will be a partition of $V = \cup_i V_i$. Let's $Q(\Pi, G_t)$ be the classic modularity of the partition Π considering only the nodes of the graph G_t as defined in [17]. Then, to find a partition of globally good quality, we will try to optimize the *global modularity* which is the sum of the classic modularity for each time step:

$$Q_{global}(\Pi, G = (G_1, ..., G_T)) = \sum_{i=1}^{i=T} Q(\Pi, G_i)$$

As the classic modularity ranges between -1 and 1, this metric ranges between $-T$ and T. To optimize this, we can use a modification of the Louvain method.

B. Optimization algorithm

The classic Louvain method is a hierarchical greedy algorithm which is composed of two phases, executed alternatively. During the first phase, nodes are moved one by one to maximize the quality gain and during the second phase the nodes are grouped into their communities to build the network between communities. Then the algorithm continues on this network until no gain is possible. Refer to [18] for more details.

The efficiency of the Louvain Method to optimize the modularity comes from the fact that the gain can be easily computed locally.

To adapt the Louvain Method to optimize the global modularity, one must change two elements: the computation of the quality gain in the first phase and how to build the network between the communities in the second one.

To change the computation of the quality gain, one must remark that the difference between two decompositions globally is the sum of the differences of the classic modularity for each snapshot. Thus, the global modularity gain can also be easily computed locally: it is the sum of the classic gains for each snapshot. The transformation of the network into the network between communities can also be modified to take into account the dynamic graphs. The dynamic graph is composed of several static graphs and we will transform these static graphs independently. Each snapshot is changed in the snapshot between communities by applying the same transformation than for the Louvain method on every snapshot.

A different way to understand the global algorithm is to apply the Louvain Method on each snapshot in parallel and to select the changes globally to obtain the maximal global modularity gain.

The adaptation of the Louvain algorithm takes a time that is faster than the sum of the running time of the classic Louvain algorithm on each snapshot. As the Louvain algorithm is really fast, it remains applicable to huge datasets composed of thousands of nodes and thousands of time steps.

II. Validation

In this section we will validate our approach using a strongly dynamic dataset. The studied network represents the topology of multicast routers obtained with the `mrinfo` tool. This tool allows to ask the neighbors of a given multicast router. Every day, `mrinfo` was run on a first router and then recursively on every neighbor, neighbors of neighbors and so on to obtain a graph of multicast routers. This measurement has been performed during several years giving a dynamic map of the multicast router topology. For more details on the measurement see [19]. Here we focus on data of 2005 and there are thus 365 snapshots, one for each day, of 3100 nodes on average.

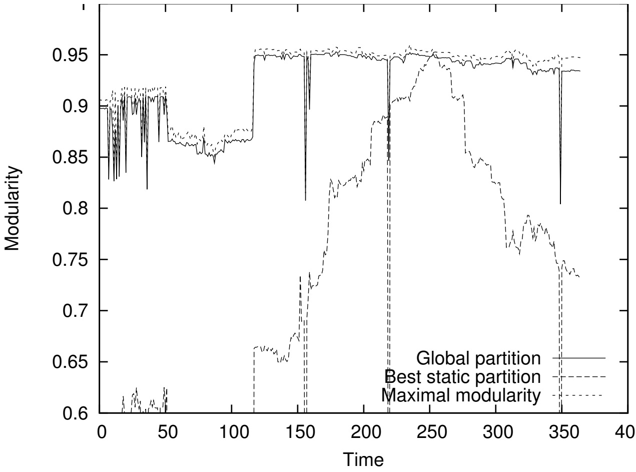

Figure 1. Instantaneous modularities

A. Classic modularity of global partitions

To study the efficiency of the algorithm we compare two partitions. The global partition is the result of the optimization with the modified Louvain algorithm designed to optimize the global modularity. Then, we have used as if it was global one of the partitions found by the classic Louvain algorithm on the snapshots taken independently[1]. Finally, the maximum global modularity that can be expected is the sum of the maximum classic modularity achieved on each snapshot.

We present on figure 1 the classic modularity of these partitions at every time step. Obviously the classic modularity is an optimum of the algorithm at each time step, and the maximum global modularity is the area under the curve. As we can see on the figure, the global partition is almost always very close to the optimal one. Thus, with only one partition we are able to capture the overall structure of this network. It works even during the event between days 50 and 110. It seems that the set of nodes has changed during this period, and thus the partition of the other nodes is not affected by this event. The best static partition here clearly fails in discovering a global structure since it is good only during a few snapshots (around time 250). This shows the interest of looking for a globally good partition rather than just selecting one snapshot.

[1] As nodes may not exist in every snapshots, all the unclassified nodes are put together. We have selected the one which has the best global modularity

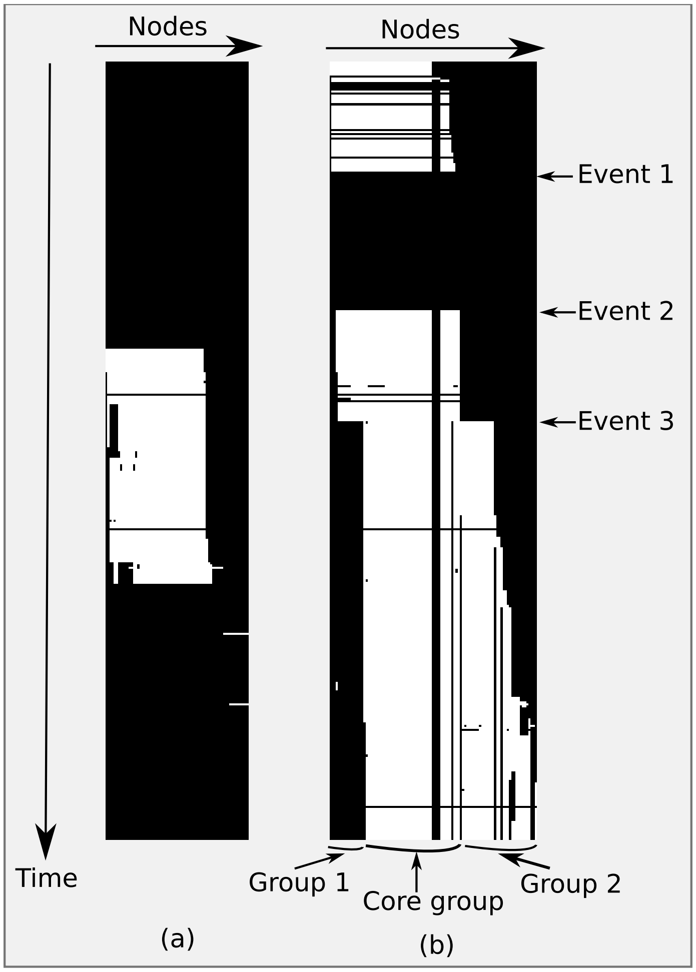

Figure 2. Pictures representing the presence of nodes of two community in mrinfo

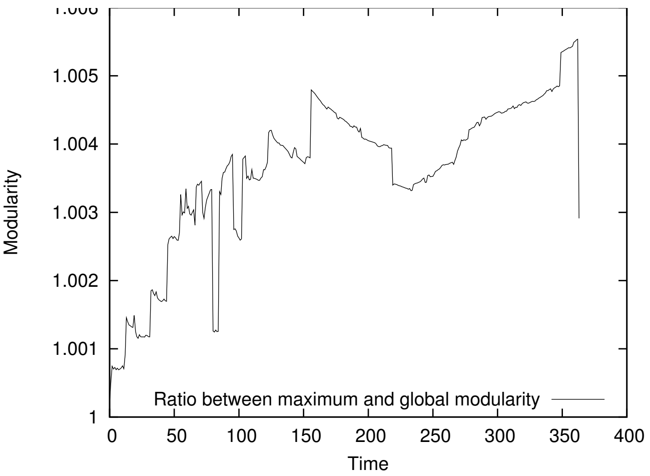

Figure 3. Global modularities depending on the time window size

B. Nodes existence during the community life

Using our definition, a community contains a set of nodes which may exist at different times. Two nodes of a given community may even never exist both at the same time but be strongly connected to a very stable core. To represent this, we have drawn one picture for every community where each column corresponds to a node and each line to a time. If the $n{-}th$ node of the community exists at time t, the point (n, t) is white and conversely the point is black if it does not exist. The choice of the order here is difficult. If you do not put together nodes that exist at the same moments, pictures seem noisy and are difficult to understand. We have chosen one simple order: nodes are first ordered according to their first appearance time since nodes that appear together are often similar, and then sorted by the number of snapshots where they exist in order to group nodes involved in the same events.

We selected two representative communities in figure 2. Group of nodes that appear or disappear together inside the community can be identified, showing that the selected grouping is relevant. The picture 2(a) is the simplest case. The community found contains one group of nodes that exist almost during the same time except a few outliers. It is the white square in the middle.

Figure 2(b) is composed of three groups with different behaviors. First, from the beginning, the group 1 and the core group exist and after a few days all of them disappear, which correspond to the first event. Then, after a few days, the two previous groups appear again and stay for a few time steps. Then, there is a new event when the group 2 on the right appears and the group 1 on the left disappears. A few nodes are connected after that. Thus, the life of this community is articulated around one central group, the core group which is the white vertical line in the middle that exists all the time except during a few rounds, and two border groups that change during time. They are grouped all together because of the connection to the core group.

C. Divergence of global and instantaneous structure

Until now, we have only optimized the global modularity over the whole period, but another possibility is to optimize the sum of the modularity on a given time interval. This allows the detection of time steps where the global structure changes. Indeed, if the global modularity starts to dissociate from the optimal, the global structure is changing and this moment is important.

Figure 3 presents the ratio of the maximal global modularity and the global modularity computed from time 0 to a given time t. There is first a small stabilization period. When there are not many snapshots, one new snapshot changes the overall structure. After a given period, this effect disappears because one new snapshot becomes less important.

After this period, the ratio takes values in a small range. But some variations can be detected, for example at day 225 where the ratio stops decreasing and starts increasing. This may reflect a possible change in the network topology or in the average network behavior. The fact that the ratio is low means that the network contains an average structure and when the ratio is raising, the network is losing its global structure, thus the new snapshots are significantly different from the previous.

Thus, this allows measuring how the average structure changes. If it is very close to the optimal, it is representative of the overall structure of the network. If the ratio is increasing, it means that the structure is changing and thus it allows to detect events.

III. Conclusions

We have seen that it is possible to detect one unique community decomposition that is good almost every time, depending on how the network evolves. Such partitions are globally good and not just optimized for one snapshot and thus give a more meaningful insight in the global structure of the network and may be more stable in case of small perturbations localized in time.

These partitions can be computed with one efficient algorithm and are thus computable on large datasets. We have presented results on one dataset but the behavior can vary on other networks and depends on their dynamics. For example, on an aggregated network, where there is only links and nodes additions without removal (for example a web graph, where links tend to never disappear), the last snapshots contain almost all the information and thus their decompositions have high global modularity. On a highly dynamic network with only a few nodes but many link transformations, there is a gap between the optimal quality and the quality of the global partition.

Many things remain unknown. For example, those partitions, by regrouping nodes from different snapshots, may place some nodes that never exist at the same time in the same community. This requires new analysis techniques like the node-existence picture representing which node is present at a given time.

Thanks to the ratio of the average structure quality and the maximum, we can measure how representative the global partition is and how the network is evolving. Thus, it could be used to determine interesting time windows of observations.

Finally, one major drawback of this new metric is due to the summation itself. The sum is commutative and thus the order of the snapshots has no effect. This means that there is no causality in how the snapshots are studied and that, for instance, the succession of snapshots G_1, G_2, G_3 is strictly equivalent to the succession of snapshots G_3, G_2, G_1. As a first approximation, it is all right, but it would be great to adapt the metric to deal with the causality of events.

References

[1] D. Auber, Y. Chiricota, F. Jourdan, and G. Melancon, "Multiscale visualization of small world networks," in *Proc. IEEE Symposium on Information Visualization*, vol. pages, 2003, pp. 75–81.

[2] M. L. Wallace, Y. Gingras, and R. Duhon, "A new approach for detecting scientific specialties from raw cocitation networks," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, pp. 240–246, 2009.

[3] J. Zhao, H. Yu, J. Luo, Z. Cao, and Y. Li, "Hierarchical modularity of nested bow-ties in metabolic networks," *BMC bioinformatics*, vol. 7, no. 1, p. 386, 2006.

[4] S. Mancoridis, B. Mitchell, C. Rorres, Y. Chen, and E. Gansner, "Using automatic clustering to produce high-level system organizations of source code," in *Proc. 6th Intl. Workshop on Program Comprehension*, 1998, pp. 45–53.

[5] M. A. Porter, P. J. Mucha, and J.-p. Onnela, "Communities in Networks," *World Wide Web Internet And Web Information Systems*, pp. 0–26.

[6] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, pp. 27–64, 2007.

[7] S. Fortunato, "Community detection in graphs," *Physics Reports*, 2009.

[8] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," in *National Academy of Sciences of the United States of America*, vol. 101, no. Suppl 1. National Acad Sciences, 2004, p. 5249.

[9] G. Palla, A.-L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, pp. 664–667, 2007.

[10] R. Kumar, A. Tomkins, and D. Chakrabarti, "Evolutionary clustering," in *In Proc. of the 12th ACM SIGKDD Conference*, 2006.

[11] M. B. Jdidia, C. Robardet, and E. Fleury, "Communities detection and analysis of their dynamics in collaborative networks." in *ICDIM*. IEEE, 2007, pp. 744–749.

[12] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe, "A framework for community identification in dynamic social networks," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, p. 717, 2007.

[13] B. L. Tseng, Y.-R. Lin, Y. Chi, S. Zhu, and H. Sundaram, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 2, pp. 1–31, 2009.

[14] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult, "Monic: modeling and monitoring cluster transitions," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2006, pp. 706–711.

[15] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu, "Graphscope: parameter-free mining of large time-evolving graphs," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2007, pp. 687–696.

[16] X. Song, Y. Chi, B. L. Tseng, D. Zhou, and K. Hino, "Evolutionary spectral clustering by incorporating temporal smoothness," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2007, pp. 153–162.

[17] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 26113, 2004.

[18] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech*, vol. 10008, pp. 1–12, 2008.

[19] J. Pansiot, P. Mérindol, B. Donnet, and O. Bonaventure, "Extracting Intra-Domain Topology from mrinfo Probing," in *Passive and Active Measurement*, 2009.