Visualizing communities in dynamic networks

Mariano G. Beiro, Jorge Rodolfo Busch, José Ignacio Alvarez-Hamelin

Visualizing communities in dynamic networks

Mariano G. Beiró
Facultad de Ingeniería
Universidad de Buenos Aires
Email: mbeiro@fi.uba.ar

Jorge R. Busch
Facultad de Ingeniería
Universidad de Buenos Aires
Email: jbusch@fi.uba.ar

J. Ignacio Alvarez-Hamelin
INTECIN
Facultad de Ingeniería
Universidad de Buenos Aires
Email: ignacio.alvarez-hamelin@cnet.fi.uba.ar

***Abstract*—Community structure is relevant to understand the properties of social networks and predict their behavior. But when this study includes the dynamic evolution, finding these communities and following them through time can be even more useful: it may help us to understand how social networks grow and to develop constructive models. In this article we analyze a dynamic blog dataset with a static community detection algorithm based on modularity, and then we use a similarity measure in order to follow the communities through time. Finally we develop a tool to visualize the dynamics of the network. This tool provides a fast intuition about the evolution of the community structure.**

I. Introduction

In the context of social networks, the membership of an individual to a certain community is one of the main parameters that characterizes him. These memberships give raise to a community structure, which is typical for each particular network. Several methods have been proposed to find this structure, of which the most famous is called *modularity optimization*. Modularity is a standard metric for finding and evaluating communities [12].

In a similar way, when dealing with dynamic networks, we are interested in analyzing the *evolution* of its communities through time. This information may be used, e.g, to develop models of network growth, and also to characterize a particular instance as in the static case. There is no generalization of the modularity to dynamic networks, and some alternatives have been proposed to match the communities from two consecutive snapshots [8] [13].

In this work we study the evolution of the community structure for a real dynamic network with an algorithm based on modularity, and we track the communities with the help of the k-core decomposition [1]. We also develop a visualization tool to show the results.

The paper is organized as follows. We present our static community detection algorithm in Section 1. Then in Section 2 we explain how the community tracking works. The next section makes a brief description of the visualization algorithm, introduced in [11]. In Section 4 we show the results for a dynamic blog dataset, concluding our work in Section 5.

II. Communities detection and tracking

A. Static community structure

The concept of community is crucial in social networks. Starting from the idea of community as a subset whose vertices are densely connected between them but are scarcely connected to other subsets, Newman proposed a global metric called *modularity* in order to evaluate community structures.

Given a graph $G = (V, E)$ and a certain partition $\mathcal{C}$, modularity is defined as

$$Q(\mathcal{C}) = \sum_{C \in \mathcal{C}} \left(\frac{n(C)}{k(V)} - \frac{k^2(C)}{k^2(V)} \right) \tag{1}$$

where $n(C)$ is the number of edges internal to C, and $k(C)$ is the sum of the degrees of the vertices in C.

The first term for each community considers its internal connections, while the second term presents the average of that quantity for a random graph whose vertices have the same expected degree as the ones in G.

Fortunato [6] showed the existence of a scaling limit, which takes the following expression: if the real communities are small enough with respect to the graph size, then the modularity optimization will join them until all small communities are connected to a big one. Based on this observation, we have proposed the inclusion of a *resolution parameter* in the modularity definition:

$$Q(\mathcal{C}, t) = \sum_{C \in \mathcal{C}} \left(\frac{n(C)}{k(V)} - t\frac{k^2(C)}{k^2(V)} \right) \tag{2}$$

By varying t we can control the scaling limit. A value of t big enough will allow the process to obtain a partition $\mathcal{C}_t$ with as many communities as vertices. Decreasing its value, the smaller communities would merge to get bigger ones. A small enough t would form a partition $\mathcal{C}_t$ with a single community comprising the whole vertex set. This idea is related to the works by Arenas *et al.* [2] and Reichardt and Bornholdt [15], but our processes for different t's are not independent in the sense that given $t' > t$, $\mathcal{C}_t$ can be obtained by joining some communities in $\mathcal{C}'_t$.

In [9] we develop from this an agglomerative algorithm called *submodularity*. Initially, we take a t big enough in which all vertices belong to different communities. Then, we optimize $Q(\mathcal{C}, t)$ until arriving to a local maxima. At this point we decrease t, getting a smaller resolution and allowing for some communities to join. The process ends when t reaches 1 (modularity). The final solution $\mathcal{C}$ is a local maxima, in the sense that joining two communities would decrease the $Q(\mathcal{C})$. In order to obtain a higher resolution, the algorithm might be stopped in a bigger t.

To conclude this subsection, we shall mention that other definitions of community have been proposed. Some of them are local, like the concept of web community in [5] and the fitness optimization in [10]. An up to date survey can be found in [7].

B. Dynamic communities

As [3] states, the dynamics of communities can be analyzed in two different ways:

- By optimizing a quality function that includes temporal information, or
- By obtaining them statically, and then tracking each community through time, e.g. see [8].

Here we use the second approach. The static submodular algorithm is applied for each snapshot, and to evaluate the similarity between communities in successive snapshots we count the coincident vertices in the *central hub* of both communities. This is supported by the idea that the central vertices are probably the more representative of each community, and they should not move through time. Then, the tracking shall be more stable.

To find a community's central hub we used the *k-core decomposition* [16], [4] on the subgraph induced by the community vertices. A k-core has the property that all its nodes have degree equal or bigger than k.

Given $\mathcal{C}_t$ and $\mathcal{C}_{t+1}$, the partitions at time t and $(t+1)$ respectively, and two communities $C_i \in \mathcal{C}_t$ and $C_j \in \mathcal{C}_{t+1}$, we define a similarity s as

$$s(C_i, C_j) = K_{max}(C_i) \cap K_{max}(C_j) \tag{3}$$

where $K_{max}(C)$ represents the top core of the k-core decomposition of the subgraph induced by C.

We proceed as follows: we choose community pairs (C_i, C_j) from $\mathcal{C}_t$ and $\mathcal{C}_{t+1}$ starting with the ones with bigger similarity. Then we assign to C_j the same community identifier than C_i. This will allow us to track its evolution later. It may happen that $\mathcal{C}_{t+1}$ brings a new community (because it does not match with any previous one), or because a community in $\mathcal{C}_t$ was split. And, of course, it may also happen that a community from t dies and disappears in $t+1$.

III. Visualization

In [11] we introduced SnailVis: a tool for the visualization of static community structures. The algorithm in SnailVis represents each community as a circle, whose ratio is proportional to the number of internal connections. Cuts between communities (number of connections between them) are drawn as single edges, their thickness being proportional to the cut that they represent. In this way, looking at the relation between a community and the thickness of the edges which emerge from it will show *at a glance* how strong or well-defined that community is.

The set of communities is deployed following a spiral curve, starting with the biggest communities in the center, and placing the smaller ones in the periphery. The spiral equation is:

$$\rho = K \cdot \theta^{\beta}, \beta \in R \ , \tag{4}$$

with $\beta = 0.5$, which is the *Fermat's spiral*. This curve has a lower radius of curvature than others, e.g. the Archimedean spiral, and the revolutions are close to each other. This implies a better use of the drawing space.

Each community is placed subject to the constraint that its distance to the previous one is some value related to their radii. This gives raise to a non-linear equation which can be solved by the Newton-Raphson iterative method. After running the algorithm for many networks, we verified that an amount of 10 iterations is usually more than enough to guarantee an error of less than 1%.

In dynamic networks, communities may be born in a certain snapshot, live and develop themselves for some time and then die. So in the visualization, they should vary their size and the thickness of their edges through time, without changing their position. In order to do this, we modified SnailVis: to place each community we do not consider its size in a certain snapshot, but the biggest size that it will reach throughout its life. In this way we assure that communities are never drawn overlapped.

IV. Obtained results

The evolving network that we analyzed consists on a network of blogs obtained from the ANR WebFluence project [14]. Each node represents a blog, and an edge represents a link (uni- or bi-directional) between them. The network is incremental: links between blogs persist on time, and this implies that the vertices and edges are never removed.

The data was analyzed for a period of 4 months and snapshots were taken on a daily basis, so that the dataset contains 120 snapshots. See [3] for an analysis of communities in this network.

Figure 1 displays the modularity as a function of the snapshot number. It is worth to remark that the community division in each snapshot was done using the submodular algorithm [9] and stopping at $t = 1$, which corresponds to the modularity resolution. The algorithm yielded similar results to those obtained in [3], and the presented behavior is a characteristic of this dataset.

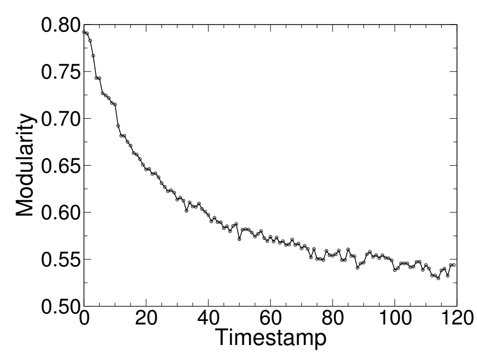
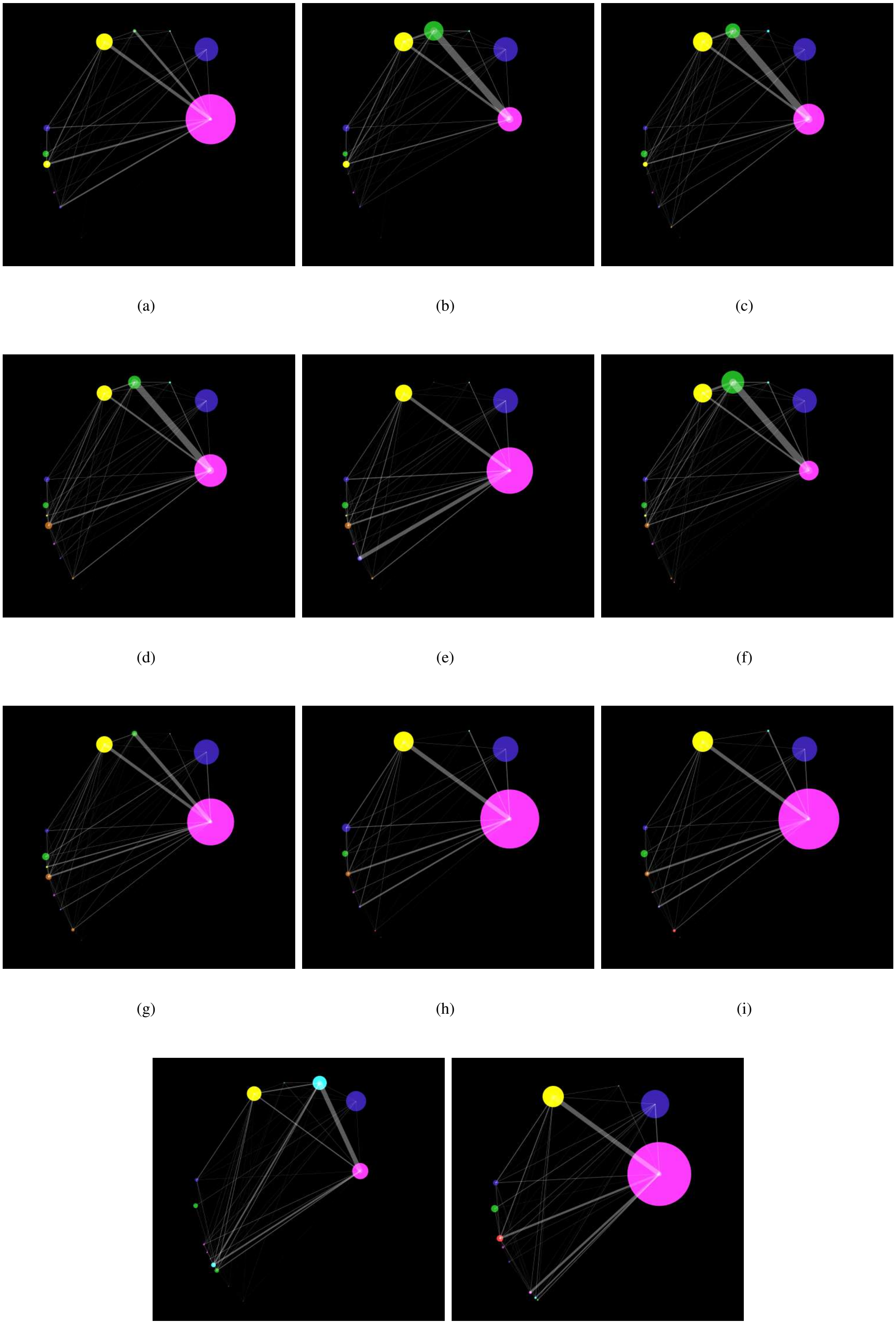


Fig. 1. Modularity evolution through time for the blogs network.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Fig. 2. Visualization of 30 snapshots (from 41 to 70) using SnalVis. Figures from (a) to (i) correspond to snapshots from 52 to 60, (j) is the 41 and (k) the 70 snapshots.

Figure 2 shows the visualization of this dataset using SnailVis [11]. In particular we analyze snapshots 41 to 70, in order to visualize more clearly the 33 communities living in this period. For the complete dataset, we found 90 communities instead. This big amount of communities is possibly due to a weakness in the tracking algorithm.

In 2(a) a small green community is represented. In the following snapshot, this community takes many nodes from the pink one (see 2(b)). These two communities alternatively share nodes that will be finally absorbed by the pink one. We noticed that this often occurs when two communities share a large number of edges.

The dynamics of some small communities can also be observed; e.g., Figure 2(g) displays some small communities just created. Finally, Figures 2(j) and 2(k) displays the initial (snapshot 41) and final (snapshot 70) community structure. In the latter, the light blue community disappears, as well as some small ones. Meanwhile, other small communities appear and the pink community seems to absorb the others. It is also important to notice that sizes in snapshot 41 are similar, whereas the opposite is observed in snapshot 70; both facts are according to Figure 1, i.e., the modularity decreases in higher snapshots. This happens because a high modularity is related to partitions whose subsets have similar sizes.

V. Conclusions

The discovery of dynamic communities is an emergent study which states many questions regarding, e.g., the definition of community, their similarity, and the processes of arousal and death. In this paper we proposed using a static community detection algorithm based on modularity and tracked the communities by studying their central hub. In our future work we will fucus on improving the tracking algorithm, analyzing the method stability and comparing it with similar ones.

We expanded our software SnailVis, developing the first dynamic communities visualization tool, as far as we know. Then, we applied it to generate images for a blogs network dataset. We think that tools like this one may help to understand the processes of evolution in community structure in the future research.

Acknowledgment

This work was partially funded by UBACyT I413/2008 project, INTECIN, and a Peruilh grant of Facultad de Ingeniería (UBA).

The authors would like to thank Jean-Loup Guillaume (LIP6) and the ANR WebFluence project for the provision of the blog dataset.

References

[1] J I Alvarez-Hamelin, L Dall'Asta, A Barrat, and A Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In Y Weiss, B Schölkopf, and J Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 41–50, Cambridge, MA, 2006. MIT Press.

[2] A Arenas, A Fernández, and S Gómez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008.

[3] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In 8^o *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 513–519, 2010.

[4] Béla Bollobás. The evolution of sparse graphs. *Graph Theory and Combinatorics*, pages 35–57, 1984.

[5] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160, New York, NY, USA, 2000. ACM.

[6] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings National Academy of Sciences*, 104(1):36–41, 2007.

[7] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[8] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Tracking evolving communities in large linked networks. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5249–5253, April 2004.

[9] J. I. Alvarez-Hamelin J. R. Busch, M. G. Beiró. On weakly optimal partitions in modular networks. Technical report, 2010. `http://arxiv.org/abs/1008.3443`.

[10] Andrea Lancichinetti, Santo Fortunato, and Jnos Kertsz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[11] J I Alvarez-Hamelin M G Beiró, J R Busch. Snailvis: a paradigm to visualize complex networks. *Proceedings of the Argentine Symposium on Technology*, pages 1682–1693, 2010.

[12] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb 2004.

[13] Gergely Palla, Albert-Laszlo Barabasi, and Tamas Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, April 2007.

[14] Webfluence Project, 2010. `http://webfluence.csregistry.org/`.

[15] J. Reichardt and S. Bornholdt. Statistichal mechanics of community detection. *Physical Review E*, 74(1):016110(14), 2006.

[16] S B Seidman. Network structure and minimum degree. *Social Networks*, 5:269–287, 1983.