March 4, 2008

Fast unfolding of community hierarchies in large networks

Vincent D Blondel, *Université catholique de Louvain*
Jean-Loup Guillaume
Renaud Lambiotte, *Université catholique de Louvain*
Etienne Lefebre, *Université catholique de Louvain*

Fast unfolding of community hierarchies in large networks

Vincent D. Blondel[1], Jean-Loup Guillaume[2], Renaud Lambiotte[1] and Etienne Lefebvre[1]
[1] *Department of Mathematical Engineering, Université catholique de Louvain,*
4 avenue Georges Lemaitre, B-1348 Louvain-la-Neuve, Belgium
[2] *LIP6, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France*

(Dated: March 4, 2008)

Social, technological and information systems can often be described in terms of complex networks that have a topology of interconnected nodes that combines organization and randomness [1, 2, 3, 4, 5]. The typical size of large networks such as social network services, mobile phone networks or the web now counts in millions when not billions of nodes and these scales demand new methods to retrieve comprehensive information from their structure [6]. A promising approach consists in decomposing the networks into sub-units or communities, which are sets of highly connected nodes [7]. The identification of these communities is of crucial importance as they may help to uncover a-priori unknown functional modules such as topics in information networks or cyber-communities in social networks. Moreover, the resulting meta-network, whose nodes are the communities, may then be used to visualize the original network structure. Here we propose a simple community detection method that reveals the hierarchical community structure of networks and that outperforms all other known community detection methods. We use our method to identify language communities and analyze community interactions in a Belgian mobile phone network of 2.6 million customers and we apply it to a web network of 118 million nodes and more than one billion links.

PACS numbers: 89.75.-k, 02.50.Le, 05.50.+q, 75.10.Hk

I. INTRODUCTION

The problem of community detection [8] requires the partition of a network into communities of strongly connected nodes, with the nodes belonging to different communities only sparsely connected. Finding exact optimal partitions in networks is known to be computationally intractable, mainly due to the explosion of the number of possible partitions as the number of nodes increases. Several algorithms have therefore been proposed to find reasonably good partitions in a reasonably fast way. This search for fast algorithms has attracted much interest in recent years due to the increasing availability of large network data sets and the impact of networks on every day life. As an example, the identification of the place of an individual in a partition –at the heart of a community or at the interface between communities– is of crucial importance in order to optimize viral methods of diffusion. One can distinguish several types of community detection algorithms: divisive algorithms detect inter-community links and remove them from the network [7, 9, 10], agglomerative algorithms merge similar nodes/communities recursively [11], spectral methods are based on the eigenvectors of the Laplacian matrix [12], and optimization methods are based on the maximisation of a benefit function [13, 14]. The quality of the partitions resulting from these methods is often measured by the so-called modularity of the partition. The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities [15]. This measure has been used to compare the quality of the partitions obtained by different methods, but also as an objective function to optimize [16]. Unfortunately, exact modularity optimization is a problem that is computationally hard [17] and so approximation algorithms are necessary when dealing with large networks. The fastest approximation algorithm for optimizing modularity on large networks was proposed by Clauset et al. [13]. That method consists in recurrently merging communities that optimize the production of modularity. Unfortunately, this greedy algorithm has a tendency to produce super-communities that contain a large fraction of the nodes, even on synthetic networks that have no significant community structure. This artefact also has the disadvantage to slow down the algorithm considerably and makes it inapplicable to networks of more than a million nodes. This undesired effect has been circumvented by introducing tricks in order to balance the size of the communities being merged, thereby speeding up the running time and making it possible to deal with networks that have a few million nodes [18]. The largest networks that have been dealt with so far in the literature are a protein-protein interaction network of 30739 nodes [6], a network of about 400000 items on sale on the website of a large on-line retailer [13], and a Japanese social networking systems of about 5.5 million users [18]. These sizes still leave considerable room for improvement considering that, as of today, the social networking service Facebook has about 64 million active users, the mobile network operator Vodaphone has about 200 million customers, Google indexes several billion web-pages and the human brain is estimated to have about a hundred billion neurons. Let us also notice that in most large networks such as those listed above there are several natural organization levels –communities divide themselves into sub-communities– and it is thus desirable to obtain community detection methods that reveal this hierarchical structure.

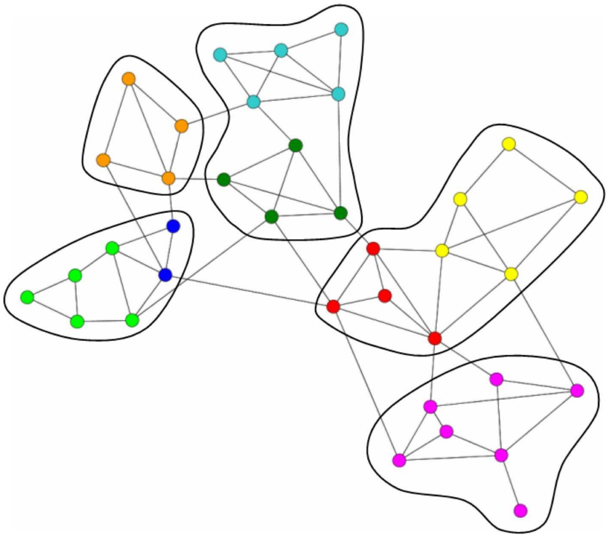

FIG. 1: On this simple illustrative example the algorithm produces two hierarchy levels. The colours show the first level partition and the surrounded clusters of nodes correspond to the partition at the second level. Although the latter partition has higher modularity, both partitions make sense.

II. METHOD

We now introduce our algorithm that finds high modularity partitions of large networks in short time and that unfolds a complete hierarchical community structure for the network, thereby giving access to different resolutions of community detection. Contrary to most of the other community detection algorithms, the network size limits that we are facing with our algorithm are due to limited storage capacity rather than limited computation time: identifying communities in a 118 million nodes network took only 152 minutes [19]. Our algorithm is divided in two phases that are repeated iteratively. Assume that we start with a weighted network of N nodes (weighted networks are networks that have weights on their links, such as the number of communications between two mobile phone users). First, we assign a different community to each node of the network. So, in this initial partition there are as many communities as there are nodes. Then, for each node i we consider the neighbours j of i and we evaluate the gain of modularity that would take place by placing i in the community of j. The node i is then placed in one of the communities for which this gain is maximum, but only if this gain is positive. If no positive gain is possible, i stays in its original community. This process is applied repeatedly and sequentially for all nodes until no further improvement can be achieved and the first phase is then complete. Part of the algorithm efficiency results from the fact that the gain in modularity ΔQ obtained by moving i in the community C of j can easily be computed by:

$$\begin{aligned}\Delta Q = &\left[\frac{\sum_{in} + k_{i,in}}{m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] \\ &- \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right], \qquad (1)\end{aligned}$$

where $\sum_{in}$ is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C and m is the sum of the weights of all the links in the network.

The second phase of the algorithm consists in building a new network whose nodes are now the communities found during the first phase. To do so, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. It is then possible to reapply the first phase of the algorithm to the resulting weighted network and to iterate. The algorithm naturally incorporates the notion of hierarchy, as communities of communities are built during the process (see Figure 1). By construction, the number of meta-communities decreases at each time step, until there are no more changes and a local maximum is attained.

This simple algorithm has several advantages. First, its steps are intuitive and easy to implement, and the outcome is unsupervised. Moreover, the algorithm is extremely fast. This is due to the fact that the possible gains in modularity are easy to compute with the above formula and that the number of communities decreases drastically after just a few iterations so that most of the running time is concentrated on the first iterations. Our algorithm is also unaffected by the so-called resolution limit problem of modularity. It is shown in Fortunato and Barthélemy [20] that modularity optimization may fail to identify communities smaller than a certain size and this induces a resolution limit on the community detected by a pure modularity optimization approach. Our approach provides instead a hierarchical tree of communities, i.e. a decomposition of the network into communities for different levels of organization. This flexibility allows the end-user to zoom in the network and to observe its structure with the desired resolution (see Figure 2).

III. APPLICATION TO LARGE NETWORKS

In order to verify the validity of our algorithm, we have applied it on a number of test-case networks that are commonly used for efficiency comparison (see Table 1). The networks that we consider include a small social network [21], a network of 9000 scientific paper and their citations [22], a sub-network of the internet [23] and a webpage network of a few hundred thousands webpages (the nd.edu domain, see [24]). In all cases, one

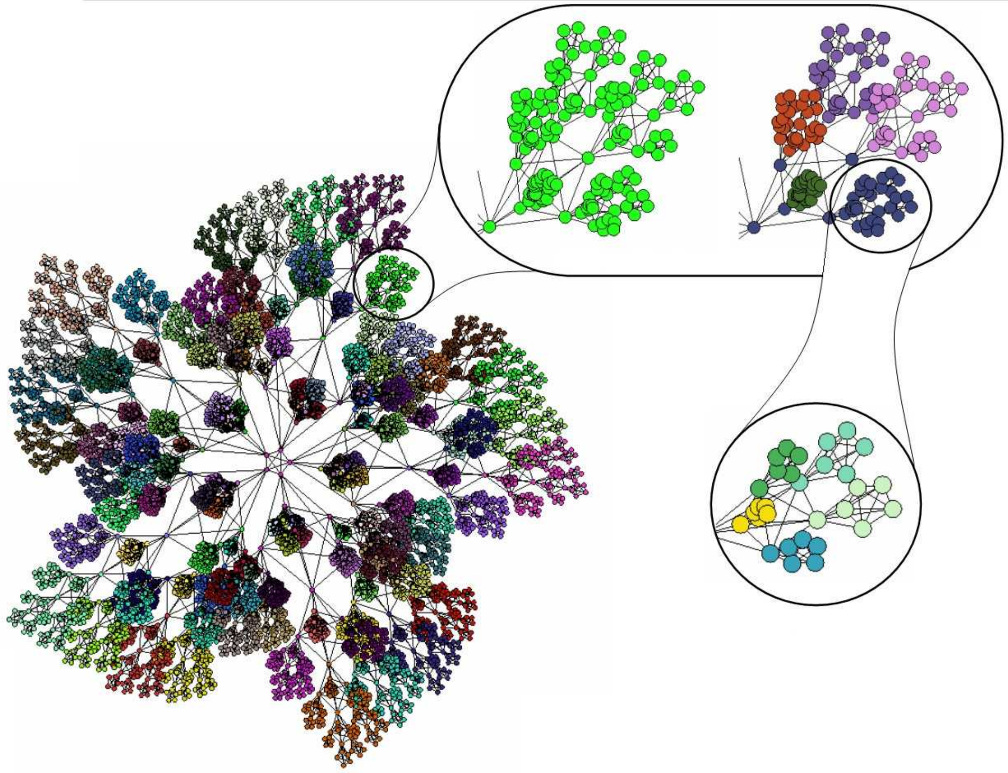

FIG. 2: We use our method for partitioning this modular, fractal-like network that is made of approximately 20000 nodes and has a multi-level community structure. The communities are plotted with different colours and at different resolution levels. The partition with the highest resolution is found after the first step of the algorithm and corresponds to the bottom of the hierarchical tree of communities. At further steps, the merging of small communities leads to the uncovering of larger meta-communities.

can observe both the rapidity and the large values of the modularity that are obtained. Our method outperforms all the other methods to which it is compared. We also have applied our method on two web networks of unprecedented sizes: a sub-network of the .uk domain of 39 million nodes and 783 million links [25] and a network of 118 million nodes and 1 billion links obtained by the Stanford WebBase crawler [25, 26]. Even for these very large networks, the computation time is small (12 minutes and 152 minutes respectively) and makes networks of still larger size, perhaps a billion nodes, accessible to computational analysis.

To validate the communities obtained we have applied our algorithm to a large network constructed from the records of a Belgian mobile phone company. This network is described in details in [27] where it is shown to exhibits typical features of social networks, such as a high clustering coefficient and a fat-tailed degree distribution. The network is composed of 2.6 million customers, between whom weighted links are drawn that account for their total number of phone calls during a 6 month period. This large social network is exceptional due to the particular situation of Belgium where two main linguistic communities (French and Dutch) coexist and which provides an excellent way to test the validity of our community detection method by looking at the linguistic homogeneity of communities [28]. From a more sociological point of view, the possibility to highlight the linguistic, religious or ethnic homogeneity of communities opens perspectives for describing the social cohesion and the potential fragility of a country [29].

On this particular network, our community detection algorithm has identified a hierarchy of six levels. At the bottom level every customer is a community of its own and at the top-level there are 261 communities that have more than 100 customers. These communities account for about 75% of all customers. We have performed a language analysis of these 261 communities (see Figure 3). The homogeneity of a community is characterized by the percentage of those speaking the dominant language in that community; this quantity goes to 1 when the community tends to be monolingual. Our analysis reveals that the network is strongly segregated, with most communities almost monolingual. There are 36 communities with more than 10000 customers and, except for one community at the interface between the two language clusters, all these communities have more than 85% of its members speaking the same language (see Figure 4 for a complete distribution). It is interesting to analyse more closely the only community that has a more

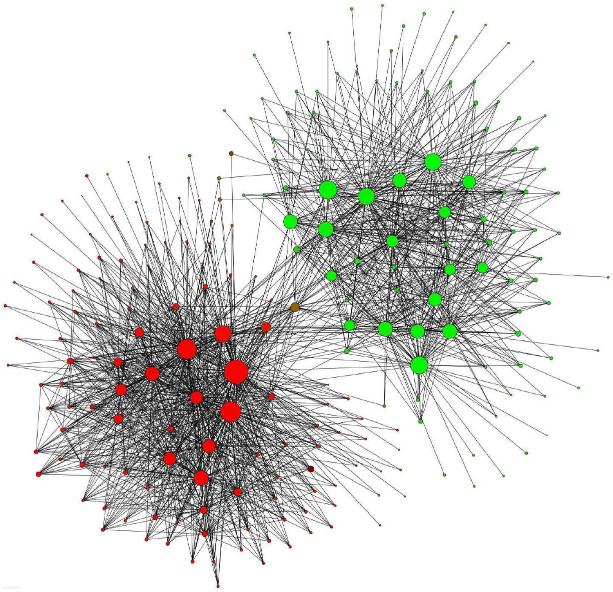

FIG. 3: Graphical representation of the network of communities extracted from a Belgian mobile phone network. About 2M customers are represented on this network. The size of a node is proportional to the number of individuals in the corresponding community and its colour on a red-green scale represents the main language spoken in the community (red for French and green for Dutch). Only the communities composed of more than 100 customers have been plotted. Notice the intermediate community of mixed colours between the two main language clusters.

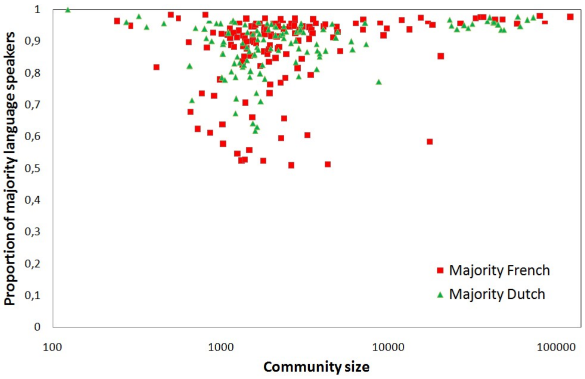

FIG. 4: For the largest communities in the Belgian mobile phone network we represent the size of the community and the proportion of customers in the community that speak the dominant language of the community. For all but one community of more than 10000 members the dominant language is spoken by more than 85% of the community members.

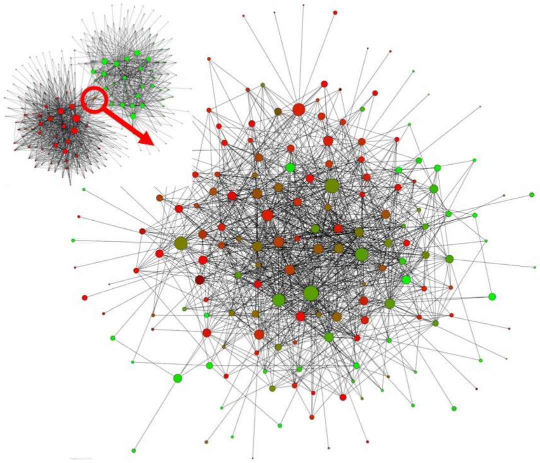

FIG. 5: Representation of the community at the interface between the two main language clusters of Figure 3. The community is made of several sub-communities with no apparent language separation.

equilibrate distribution of languages. Our hierarchy revealing algorithm allows us to do this by considering the sub-communities provided by the algorithm at the lower level. As shown on Figure 5, these sub-communities are closely connected to each other and are themselves composed of heterogeneous groups of people. These groups of people, where language ceases to be a discriminating factor, might possibly play a crucial role for the integration of the country and for the emergence of consensus between the communities [30]. One may indeed wonder what would happen if the community at the interface between the two language clusters on Figure 3 was to be removed.

Another interesting observation is related to the presence of other languages. There are actually four possible language declarations for the customers of this particular mobile phone operator: French, Dutch, English or German. It is interesting to note that, whereas English speaking customers disperse themselves quite evenly in all communities, more than 60% of the German speaking customers are concentrated in just one community. Let us finally observe that, as can be visually noticed on Figure 3, French speaking communities are much more densely connected than their Dutch speaking counterparts: on average, the strength of the links between French speaking communities is 54% stronger than those between Dutch speaking communities. This difference of structure between the two sub-networks seems to indicate that the two linguistic communities are characterized by different social behaviours and therefore suggests to search other topological characteristics for the communities.

Acknowledgments

This research was supported by the Communauté Française de Belgique through a grant ARC and by the Belgian Network DYSCO, funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. J.-L. G. is also supported by the project MAPE (ANR France) and MAPAP (Safer Internet Plus Programme, European Union).

[1] R. Albert and A.-L. Barabási (2002) Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 4797.

[2] A.-L. Barabási and R. Albert (1999) Emergence of scaling in random networks. *Science* **286**, 509512.

[3] J.F.F. Mendes and S.N. Dorogovtsev, *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, Oxford, 2003).

[4] M.E.J. Newman, A.-L. Barabási and D.J. Watts, *The Structure and Dynamics of Networks* (Princeton University Press, Princeton, 2006).

[5] D.J. Watts and S.H. Strogatz (1998) Collective dynamics of small-world networks. *Nature* **393**, 440442.

[6] G. Palla, I. Derényi, I. Farkas and T. Vicsek (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814-818.

[7] M. Girvan and M.E.J. Newman (2002) Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **99**, 7821-7826.

[8] S. Fortunato and C. Castellano (2007) Community structure in graphs. *arXiv:0712.2716*

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

[10] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi (2004) Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA* **101**, 26582663.

[11] P. Pons and M. Latapy (2006) Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications* **10**, 191-218.

[12] M.E.J. Newman (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104.

[13] A. Clauset, M.E.J. Newman and C. Moore (2004) Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111.

[14] F. Wu and B.A. Huberman (2004) Finding communities in linear time: a physics approach. *Eur. Phys. J. B* **38**, 331-338.

[15] M.E.J. Newman (2006) Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **103**, 8577-8582.

[16] M.E.J. Newman (2004) Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**, 066133.

[17] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski and D. Wagner (2006) Maximizing Modularity is hard. *physics/0608255*

[18] K. Wakita and T. Tsurumi (2007) Finding community structure in a mega-scale social networking service. *Proceedings of IADIS international conference on WWW/Internet 2007*, 153-162.

[19] All methods described here have been compiled and tested on the same machine: a bi-opteron 2.2k with 24GB of memory.

[20] S. Fortunato and M. Barthélemy (2007) Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **104**, 36-41.

[21] W.W. Zachary (1977) An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**, 452-473.

[22] http://www.cs.cornell.edu/projects/kddcup/ (Cornell KDD Cup)

[23] M. Hoerdt and D. Magoni (2003) Completeness of the internet core topology collected by a fast mapping software. *Proceedings of the 11th International Conference on Software, Telecommunications and Computer Networks*, 257-261.

[24] R. Albert, H. Jeong and A.-L. Barabási (1999) Diameter of the World Wide Web. *Nature* **401**, 130.

[25] http://law.dsi.unimi.it/ (Laboratory for Web Algorithmics)

[26] http://dbpubs.stanford.edu:8091/∼testbed/doc2/WebBase/ (Stanford WebBase Project)

[27] R. Lambiotte, V.D. Blondel, C. de Kerchove, E. Huens, C. Prieur, Z. Smoreda and P. Van Dooren (2008) A gravity model for the geographical dispersal of mobile communication networks. *arXiv:0802.2178*

[28] G. Palla, A.-L. Barabási and T. Vicsek (2007) Quantifying social group evolution. *Nature* **446**, 664-667.

[29] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási (2007) Structure and tie strengths in mobile communication networks. *Proc. Natl. Acad. Sci. USA* **104**, 7332-7336.

[30] R. Lambiotte, M. Ausloos and J. A. Hołyst (2007) Majority Model on a network with communities. *Phys. Rev. E* **75** 030101(R).

TABLE I: Summary of numerical results. This table give the performances of the algorithm of Newman and Girvan [9], of Pons and Latapy [11] and of our algorithm for community detection in networks of various sizes. For each method/network, the table displays the modularity that is achieved and the computation time. Empty cells correspond to a computation time over 24 hours. The source code for the algorithm by Wakita and Tsurumi [18] is not available for comparison and we have therefore not been able to run their algorithm on the same data. However, the largest network that they are able to treat has 5.5M nodes and they do not expect their method to scale beyond 10M nodes. By extrapolation on the basis of the results and computation times that they provide for various network sizes, we expect their method to take 3 hours on our phone network and several days on the web networks of 39M and 118M nodes.

	Karate	Arxiv	Internet	Web nd.edu	Phone	Web uk-2005	Web WebBase 2001
Nodes/links	34/77	9k/24k	70k/351k	325k/1M	2.6M/6.3M	39M/783M	118M/1B
Newman-Girvan	.404/0s	.772/3.6s	.692/799s	.927/5034s	-/-	-/-	-/-
Pons-Latapy	.43/0s	.757/3.3s	.729/575s	.895/6666s	-/-	-/-	-/-
Our algorithm	.43/0s	.813/0s	.781/1s	.935/3s	.769/134s	.979/738s	.984/152mn