Journal of Statistical Mechanics:
Theory and Experiment

SISSA

Fast unfolding of communities in large networks

To cite this article: Vincent D Blondel *et al J. Stat. Mech.* (2008) P10008

* Modularity is used to measure 'goodness' of communities. It is ...

* Three types of community detection
① Divisive
② Agglomerative
③ Optimization.
Louvain appears to be agglomerative, but is also optimizing for the config until the change in modularity is zero.

* "Our algorithm is limited by available storage, not computation" ⇒ assuming they mean 'memory', and if by "storage" it's a good example of an algorithm where PIM, or other mechanisms to reduce memory cost would be beneficial, justifying it as a case study for HIVE.

* Algo is in 2 phases:
① Assign each node to a different community + compute modularity.
② Rebuild graph in config that maximizes modularity.
Iterative until no change in modularity.

* The order of processing nodes impacts execution time but not really modularity; they aren't sure why.

* They claim linear complexity.

* Their results show superior modularity scores + compute time compared to other approaches
↳ Metrics for a benchmark?

* Additional heuristics for early stopping, like adding a "good enough" threshold for modularity allow it to increase execution speed further.

Open Questions:

① Why is precise modularity intractable?
② Why does the ordering of nodes impact the processing time?
③ What is the optimal ordering of nodes? Does their representation in memory impact performance
④ What is optimal way to store a graph in memory? Do different configurations work better for different problems? Would treating it spatially - eg build Adj. Matrix with highest degree @ the centre and less connected outwards improve / degrade?
⑤ Is the OS responsible for assigning to memory, or could the programmer control this?
⑥ What is the 'Resolution Limit Problem'
⑦ Since this paper, what is the biggest dataset it's been applied to?

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Fast unfolding of communities in large networks

Vincent D Blondel[1], **Jean-Loup Guillaume**[1,2], **Renaud Lambiotte**[1,3] **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, F-75005 Paris, France
[3] Institute for Mathematical Sciences, Imperial College London, 53 Prince's Gate, South Kensington Campus, London SW7 2PG, UK
E-mail: vincent.blondel@uclouvain.be, jean-loup.guillaume@lip6.fr, r.lambiotte@imperial.ac.uk and pixetus@hotmail.com

Received 18 April 2008
Accepted 3 September 2008
Published 9 October 2008

Online at stacks.iop.org/JSTAT/2008/P10008
doi:10.1088/1742-5468/2008/10/P10008

Abstract. We propose a simple method to extract the community structure of large networks. Our method is a heuristic method that is based on modularity optimization. It is shown to outperform all other known community detection methods in terms of computation time. Moreover, the quality of the communities detected is very good, as measured by the so-called modularity. This is shown first by identifying language communities in a Belgian mobile phone network of 2 million customers and by analysing a web graph of 118 million nodes and more than one billion links. The accuracy of our algorithm is also verified on ad hoc modular networks.

Keywords: random graphs, networks, new applications of statistical mechanics

ArXiv ePrint: 0803.0476

Contents

1. Introduction 2
2. Method 3
3. Application to large networks 6
4. Conclusion and discussion 10
Acknowledgments 11
References 11

1. Introduction

Social, technological and information systems can often be described in terms of complex networks that have a topology of interconnected nodes combining organization and randomness [1, 2]. The typical size of large networks such as social network services, mobile phone networks or the web is now counted in millions, if not billions, of nodes and these scales demand new methods to retrieve comprehensive information from their structure. A promising approach consists in decomposing the networks into sub-units or communities, which are sets of highly interconnected nodes [3]. 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.

The problem of community detection requires the partition of a network into communities of densely connected nodes, with the nodes belonging to different communities being only sparsely connected. Precise formulations of this optimization problem are known to be computationally intractable. 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 datasets and the impact of networks on everyday life. One can distinguish several types of community detection algorithms: divisive algorithms detect inter-community links and remove them from the network [4]–[6], agglomerative algorithms merge similar nodes/communities recursively [7] and optimization methods are based on the maximization of an objective function [8]–[10]. 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 [5, 11]. In the case of weighted networks (weighted networks are networks that have weights on their links, such as the number of communications between two mobile phone users), it is defined as [12]

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{1}$$

where A_{ij} represents the weight of the edge between i and j, $k_i = \sum_j A_{ij}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, the δ function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise and $m = \frac{1}{2}\sum_{ij} A_{ij}$.

Modularity has been used to compare the quality of the partitions obtained by different methods, but also as an objective function to optimize [13]. Unfortunately, exact modularity optimization is a problem that is computationally hard [14] 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* [8]. That method consists in recurrently merging communities that optimize the production of modularity. Unfortunately, this greedy algorithm may produce values of modularity that are significantly lower than what can be found by using, for instance, simulated annealing [15]. Moreover, the method proposed in [8] 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 of slowing 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 [16].

The largest networks that have been dealt with so far in the literature are a protein–protein interaction network of 30 739 nodes [17], a network of about 400 000 items on sale on the website of a large on-line retailer [8] and a Japanese social networking system of about 5.5 million users [16]. These sizes still leave considerable room for improvement [18] considering that, as of today, the social networking service Facebook has about 64 million active users, the mobile network operator Vodafone has about 200 million customers and Google indexes several billion web-pages. 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 [19].

2. Method

We now introduce our algorithm that finds high modularity partitions of large networks in a short time and that unfolds a complete hierarchical community structure for the network, thereby giving access to different resolutions of community detection. Contrary to all 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 min[4].

Our algorithm is divided into two phases that are repeated iteratively. Assume that we start with a weighted network of N nodes. 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 removing i from its community and by

[4] All methods described here have been compiled and tested on the same machine: a bi-opteron 2.2k with 24 GB of memory. The code is freely available for download on the web-page http://findcommunities.googlepages.com.

placing it in the community of j. The node i is then placed in the community for which this gain is maximum (in the case of a tie we use a breaking rule), 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. Let us insist on the fact that a node may be, and often is, considered several times. This first phase stops when a local maxima of the modularity is attained, i.e. when no individual move can improve the modularity[5]. One should also note that the output of the algorithm depends on the order in which the nodes are considered. Preliminary results on several test cases seem to indicate that the ordering of the nodes does not have a significant influence on the modularity that is obtained, while it may affect the computation time, but the reasons for this dependence are not clear. In particular, taking the nodes in a natural order related to the community structure itself (e.g. the order given by a previous community computation, or the postcode) does not give clear improvement (see section 3). The problem of choosing an order is thus worth studying since it could give good heuristics to enhance the computation time.

Part of the algorithm's efficiency results from the fact that the gain in modularity ΔQ obtained by moving an isolated node i into a community C can easily be computed by

$$\Delta Q = \left[\frac{\sum_{\text{in}} + 2k_{i,\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right], \quad (2)$$

where $\sum_{\text{in}}$ is the sum of the weights of the links inside C, $\sum_{\text{tot}}$ 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, $k_{i,\text{in}}$ 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. A similar expression is used in order to evaluate the change of modularity when i is removed from its community. In practice, one therefore evaluates the change of modularity by removing i from its community and then by moving it into a neighbouring community.

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 [20]. Links between nodes of the same community lead to self-loops for this community in the new network. Once this second phase is completed, it is then possible to reapply the first phase of the algorithm to the resulting weighted network and to iterate. Let us denote by 'pass' a combination of these two phases. By construction, the number of meta-communities decreases at each pass, and as a consequence most of the computing time is used in the first pass. The passes are iterated (see figure 1) until there are no more changes and a maximum of modularity is attained. The algorithm is reminiscent of the self-similar nature of complex networks [21] and naturally incorporates a notion of hierarchy, as communities of communities are built during the process. The height of the hierarchy that is constructed is determined by the number of passes and is generally a small number, as will be shown in some examples below.

[5] In order to decrease the overall running time of the method it is possible to introduce a threshold and then stop the first phase as soon as the relative gain in modularity does not exceed this threshold. The numerical results reported here have been obtained with this minor modification.

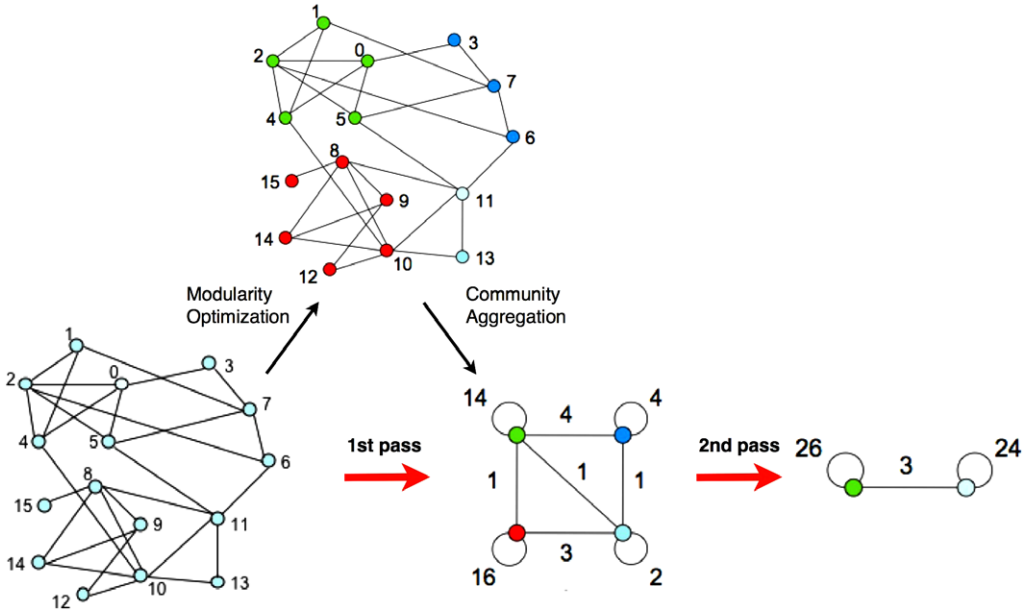

Figure 1. Visualization of the steps of our algorithm. Each pass is made of two phases: one where modularity is optimized by allowing only local changes of communities; one where the communities found are aggregated in order to build a new network of communities. The passes are repeated iteratively until no increase of modularity is possible.

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, i.e. computer simulations on large ad hoc modular networks suggest that its complexity is linear on typical and sparse data. 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 passes so that most of the running time is concentrated on the first iterations. The so-called resolution limit problem of modularity also seems to be circumvented thanks to the intrinsic multi-level nature of our algorithm. Indeed, it is well known [22] that modularity optimization fails to identify communities smaller than a certain scale, thereby inducing a resolution limit on the community detected by a pure modularity optimization approach. This observation is only partially relevant in our case because the first phase of our algorithm involves the displacement of single nodes from one community to another. Consequently, the probability that two distinct communities can be merged by moving nodes one by one is very low. These communities may possibly be merged in the later passes, after blocks of nodes have been aggregated. However, our algorithm provides a decomposition of the network into communities for different levels of organization. In order to illustrate this feature, let us focus on the ring of 30 cliques discussed in [22], where the cliques are composed of 5 nodes and are interconnected through single links. The first pass of the algorithm finds the natural partition of the network, where each community corresponds to one clique. The second pass finds the global maximum of modularity where cliques are combined into groups of 2. Consequently, if the cliques are indeed merged in the final

Table 1. Summary of numerical results. This table gives the performances of the algorithm of Clauset *et al* [8], of Pons and Latapy [7], of Wakita and Tsurumi [16] 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 h. Our method clearly performs better in terms of computer time and modularity. It is also interesting to note the small value of Q found by WT for the mobile phone network. This bad modularity result may originate from their heuristic which creates balanced communities, while our approach gives unbalanced communities in this specific network.

	Karate	Arxiv	Internet	Web nd.edu	Phone	Web uk-2005	Web WebBase 2001
Nodes/ links	34/77	9k/24k	70k/351k	325k/1M	2.04M/5.4M	39M/783M	118M/1B
CNM	0.38/0 s	0.772/3.6 s	0.692/799 s	0.927/5034 s	—/—	—/—	—/—
PL	0.42/0 s	0.757/3.3 s	0.729/575 s	0.895/6666 s	—/—	—/—	—/—
WT	0.42/0 s	0.761/0.7 s	0.667/62 s	0.898/248 s	0.553/367 s	—/—	—/—
Our algorithm	0.42/0 s	0.813/0 s	0.781/1 s	0.935/3 s	0.76/44 s	0.979/738 s	0.984/152 mn

partition due to the resolution limit, they are distinct after the first pass. This result suggests that the intermediate solutions found by our algorithm may also be meaningful and that the uncovered hierarchical structure may allow the end-user to zoom into the network and to observe its structure with the desired resolution.

3. 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 and we have compared it with three other community detection algorithms (see table 1). The networks that we consider include a small social network [23], a network of 9000 scientific papers and their citations [24], a sub-network of the internet [25] and a web-page network of a few hundred thousand web-pages (the nd.edu domain, see [26]). In all cases, one 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 [27] and a network of 118 million nodes and 1 billion links obtained by the Stanford WebBase crawler [27, 28]. Even for these very large networks, the computation time is small (12 min and 152 min, respectively) and makes networks of still larger size, perhaps a billion nodes, accessible to computational analysis. It is also interesting to note that the number of passes is usually very small. In the case of the Karate Club [23], for instance, there are only 3 passes: during the first one, the 34 nodes of the network are partitioned into 6 communities; after the second one, only four communities remain; during the third one, nothing happens and the algorithm therefore stops. In the above examples, the number of passes is always smaller than 5.

J. Stat. Mech. (2008) P10008

We have also tested the sensitivity of our algorithm by applying it on ad hoc networks that have a known community structure. To do so, we have used networks composed of 128 nodes which are split into 4 communities of 32 nodes each [29]. Pairs of nodes belonging to the same community are linked with probability $p_{\rm in}$ while pairs belonging to different communities are linked with probability $p_{\rm out}$. The accuracy of the method is evaluated by measuring the fraction of correctly identified nodes and the normalized mutual information. In the benchmark proposed in [29], the fraction of correctly identified nodes is 0.67 for $z_{\rm out} = 8$, 0.92 for $z_{\rm out} = 7$ and 0.98 for $z_{\rm out} = 6$, i.e. an accuracy similar to that of the algorithm of Pons and Latapy [7] and of the algorithm of Reichardt and Bornholdt [30]. To our knowledge, only two algorithms have a better accuracy than ours, the algorithm of Duch and Arenas [31] and the simulated annealing method first proposed in [15], but their computational cost limits their applicability to much smaller networks than the ones considered here. Our algorithm has also been successfully tested on other benchmarks, such as the ones proposed in [19, 32]. In the case [32], the normalized mutual information is nearly 1 for the macro-communities with a mixing parameter k_3 up to 35. It reaches 0.5 when the mixing parameter is around 55.

To validate the communities obtained we have also applied our algorithm to a large network constructed from the records of a Belgian mobile phone company. This network is described in detail in [33] 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. In this paper, we have focused on a subset of 2.04 million customers for whom several entries are associated, such as their age, sex, language and the postcode of the place where they live. 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 [34]. 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 [35].

On this particular network, our community detection algorithm has identified a hierarchy of 6 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 2). 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 10 000 customers and, except for one community at the interface between the two language clusters, all these communities have more than 85% of their members speaking the same language (see figure 3 for a complete distribution). It is interesting to analyse more closely the only community that has a more 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 in figure 3, these sub-communities are closely connected to each other and are themselves composed of heterogeneous groups of people. These

J. Stat. Mech. (2008) P10008

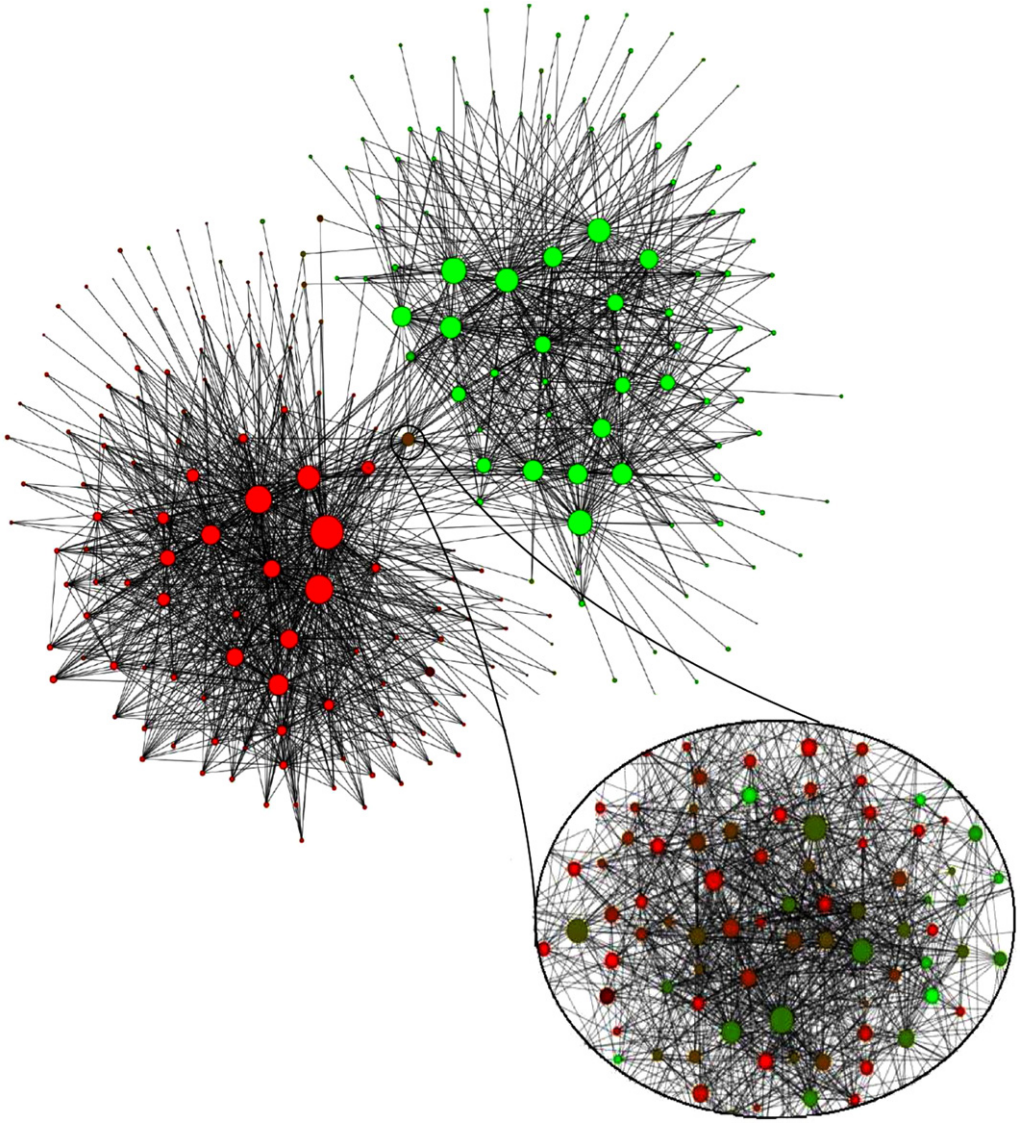

Figure 2. Graphical representation of the network of communities extracted from a Belgian mobile phone network. About 2 million customers are represented on this network. The size of a node is proportional to the number of individuals in the corresponding community and its colour on a red–green scale 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. A zoom at higher resolution reveals that it is made of several sub-communities with less apparent language separation.

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 [36]. One may indeed wonder what would happen if the community at the interface between the two language clusters in figure 2 was to be removed.

J. Stat. Mech. (2008) P10008

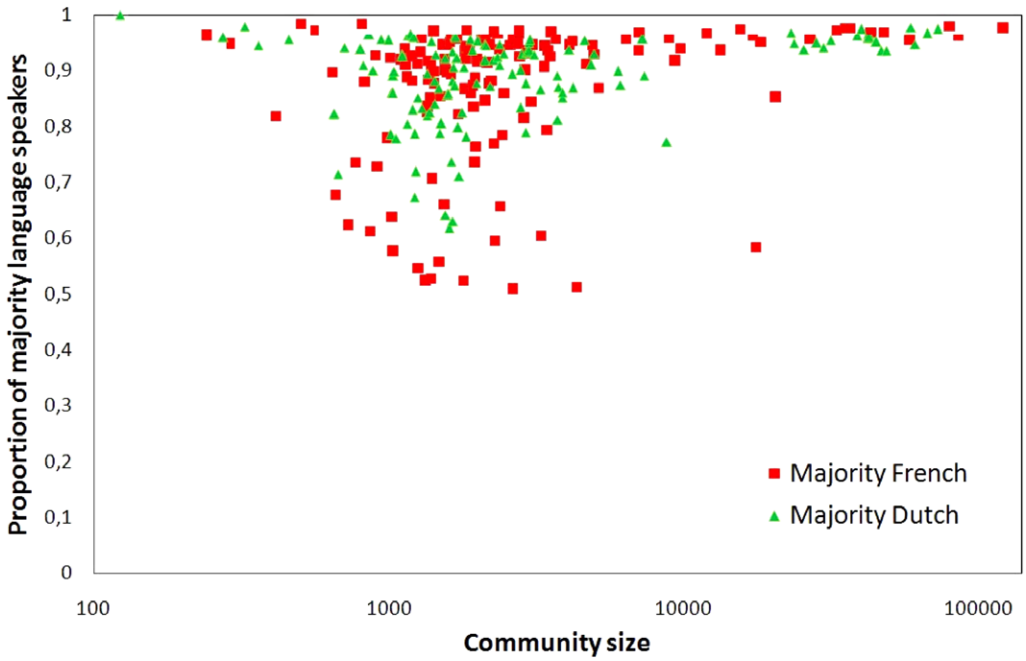

Figure 3. For the largest communities in the Belgian mobile phone network we represent the size of the community and the proportion of customers in the community that speak the dominant language of the community. For all but one community of more than 10 000 members the dominant language is spoken by more than 85% of the community members.

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. This is probably due to the fact that German-speaking people are mainly concentrated in a small region close to Germany, while English-speaking people are spread over the whole country. Let us finally observe that, as can be visually noticed in figure 2, 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.

We have also focused on this mobile phone network in order to elucidate the role played by the ordering of the nodes on the output of the method. To do so, we have first performed 100 analyses where the ordering of the nodes is chosen randomly. From the modularity Q_i and computation time T_i evaluated at each run i, we have computed the average and variance of these variables. In the case of modularity, the value is almost constant over all the runs with $\langle Q \rangle = 0.76$ and a deviation of $\sigma_Q \equiv \sqrt{\langle Q^2 \rangle - \langle Q \rangle^2} = 10^{-2}$. The fluctuations of the computation time are more important, as $\langle T \rangle = 44.2$ s and

$\sigma_T = 3.2$ s but remain reasonably small as this is only a 7% variation. The smallest and largest values of T among the 100 runs are 39 and 55 s. This interval suggests that a good choice for the ordering of the nodes may substantially accelerate the dynamics. We have therefore checked if an order related to the community structure would accelerate the computation time. To do so, we have ordered the nodes by their postcodes, but this choice did not lead to any improvement as compared to a random ordering, as $Q_{\rm zip} = 0.76$ and $T_{\rm zip} = 44$ s.

4. Conclusion and discussion

We have introduced an algorithm for optimizing modularity that allows us to study networks of unprecedented size. The limitation of the method for the experiments that we performed was the storage of the network in main memory rather than the computation time. This change of scales, i.e. from around 5 millions nodes for previous methods to more than 100 million nodes in our case, opens exciting perspectives as the modular structure of complex systems such as whole countries or huge parts of the Internet can now be unravelled. The accuracy of our method has also been tested on ad hoc modular networks and is shown to be excellent in comparison with other (much slower) community detection methods. It is interesting to note that the speed of our algorithm can still be substantially improved by using some simple heuristics, for instance by stopping the first phase of our algorithm when the gain of modularity is below a given threshold or by removing the nodes of degree 1 (leaves) from the original network and adding them back after the community computation. The impact of these heuristics on the final partition of the network should be studied further, as well as the role played by the ordering of the nodes during the first phase of the algorithm.

By construction, our algorithm unfolds a complete hierarchical community structure for the network, each level of the hierarchy being given by the intermediate partitions found at each pass. In this paper, however, we have only verified the accuracy of the top level of this hierarchy, namely the final partition found by our algorithm, and the accuracy of the intermediate partitions has still to be shown. Several points suggest, however, that these intermediate partitions make sense. First, intermediate partitions correspond to local maxima of modularity, maxima in the sense that it is not possible to increase modularity by moving one single 'entity' from one community to a neighbouring one. In the first pass of the algorithm, these entities are nodes, but at subsequent passes, they correspond to larger and larger sets of nodes. Intermediate partitions may therefore be viewed as local maxima of modularity at different scales. It is the agglomeration of nodes during the second phase of the algorithm which allows us to uncover larger and larger communities, thereby taking advantage of the self-similar structure of many complex networks. Second, the final partition found by our algorithm has a very high value of modularity for a broad range of system sizes (for instance, as shown in table 1, our algorithm performs better in terms of modularity than those of Clauset *et al* [8], of Pons and Latapy [7] and of Wakita and Tsurumi [16]). Finally, it is instructive to consider a community C found at the last pass of our algorithm. In order to test the validity of the sub-communities found at the penultimate pass, it is tempting to look at community C as a new network, thereby neglecting links going from C to the rest of the network. By reapplying our algorithm on the isolated community C, one expects to find very similar sub-communities due to the

J. Stat. Mech. (2008) P10008

local optimization involved at each step. These are, however, very qualitative arguments and the multi-resolution of our algorithm will only be confirmed after looking in detail at the hierarchies found in ad hoc networks with known hierarchical structure [19] or without community structure (e.g. Erdös–Renyi random graphs), or after comparing with other methods incorporating a tunable resolution [32, 37, 38].

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-LG is also supported in part by MAPAP SIP-2006-PP-221003 and ANR MAPE projects.

References

[1] Albert R and Barabási A-L, 2002 *Rev. Mod. Phys.* **74** 4797
[2] Newman M E J, Barabási A-L and Watts D J, 2006 *The Structure and Dynamics of Networks* (Princeton, NJ: Princeton University Press)
[3] Fortunato S and Castellano C, 2007 arXiv:0712.2716
[4] Girvan M and Newman M E J, 2002 *Proc. Nat. Acad. Sci.* **99** 7821
[5] Newman M E J and Girvan M, 2004 *Phys. Rev.* E **69** 026113
[6] Radicchi F, Castellano C, Cecconi F, Loreto V and Parisi D, 2004 *Proc. Nat. Acad. Sci.* **101** 2658
[7] Pons P and Latapy M, 2006 *J. Graph Algorithms Appl.* **10** 191
[8] Clauset A, Newman M E J and Moore C, 2004 *Phys. Rev.* E **70** 066111
[9] Wu F and Huberman B A, 2004 *Eur. Phys. J.* B **38** 331
[10] Newman M E J, 2006 *Phys. Rev.* E **74** 036104
[11] Newman M E J, 2006 *Proc. Nat. Acad. Sci.* **103** 8577
[12] Newman M E J, 2004 *Phys. Rev.* E **70** 056131
[13] Newman M E J, 2004 *Phys. Rev.* E **69** 066133
[14] Brandes U, Delling D, Gaertler M, Goerke R, Hoefer M, Nikoloski Z and Wagner D, 2006 arXiv:physics/0608255
[15] Guimera R, Sales M and Amaral L A N, 2004 *Phys. Rev.* E **70** 025101
[16] Wakita K and Tsurumi T, 2007 *Proc. IADIS Int. Conf. on WWW/Internet 2007* p 153
[17] Palla G, Derényi I, Farkas I and Vicsek T, 2005 *Nature* **435** 814
[18] Raghavan U N, Albert R and Kumara S, 2007 *Phys. Rev.* E **76** 036106
[19] Sales-Pardo M, Guimera R, Moreira A A and Amaral L A N, 2007 *Proc. Nat. Acad. Sci.* **104** 15224
[20] Arenas A, Duch J, Fernández A and Gómez S, 2007 *New J. Phys.* **9** 176
[21] Song C, Havlin S and Makse H A, 2005 *Nature* **433** 392
[22] Fortunato S and Barthélemy M, 2007 *Proc. Nat. Acad. Sci.* **104** 36
[23] Zachary WW, 1977 *J. Anthropol. Res.* **33** 452
[24] http://www.cs.cornell.edu/projects/kddcup/ (Cornell KDD Cup)
[25] Hoerdt M and Magoni D, 2003 *Proc. 11th Int. Conf. on Software, Telecommunications and Computer Networks* p 257
[26] Albert R, Jeong H and Barabási A-L, 1999 *Nature* **401** 130
[27] http://law.dsi.unimi.it/ (Laboratory for Web Algorithmics)
[28] http://dbpubs.stanford.edu:8091/~testbed/doc2/WebBase/ (Stanford WebBase Project)
[29] Danon L, Díaz-Guilera A, Duch J and Arenas A, 2005 *J. Stat. Mech.* P09008
[30] Reichardt J and Bornholdt S, 2004 *Phys. Rev. Lett.* **93** 218701
[31] Duch J and Arenas A, 2005 *Phys. Rev.* E **72** 027104
[32] Lancichinetti A, Fortunato S and Kertesz J, 2008 arXiv:0802.1218
[33] Lambiotte R, Blondel V D, de Kerchove C, Huens E, Prieur C, Smoreda Z and Van Dooren P, 2008 *Physica* A **387** 5317
[34] Palla G, Barabási A-L and Vicsek T, 2007 *Nature* **446** 664
[35] Onnela J-P, Saramäki J, Hyvönen J, Szabó G, Lazer D, Kaski K, Kertész J and Barabási A-L, 2007 *Proc. Nat. Acad. Sci.* **104** 7332

[36] Lambiotte R, Ausloos M and Hołyst J A, 2007 *Phys. Rev.* E **75** 030101(R)
[37] Arenas A, Fernández A and Gómez S, 2008 *New J. Phys.* **10** 053039
[38] Delvenne J-C, Yaliraki S and Barahona M, 2008 in preparation