ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

Performance evaluation of different methods to detect community structure in scale-free complex networks

José H. H. Grisi-Filho, Marcos Amaku, Raul Ossada, Fernando Ferreira
School of Veterinary Medicine and Animal Health – University of São Paulo
05508-270, São Paulo, SP
E-mail: grisi@vps.fmvz.usp.br

ABSTRACT

A property that seems to be common to many networks is community structure, the division of network nodes into groups within which the network connections are dense, but between which they are sparser [5].

Many algorithms were developed to identify the community structure of a network [5], [2]. These algorithms presents different characteristics that results in different accuracy and computational cost, and therefore can be used in different situations, regarding the objectives of the research.

A well-known method to generate a scale-free network is the preferential attachment [1], in which links are added to nodes based on their degree. We developed a method that takes the opposite way to generate a scale-free network, inspired by [4]. We start by setting the number of nodes in the network, N. For each node, we randomly sort a degree k, based on a power-law distribution $P(k) \sim k^{-\alpha}$, where α is the exponent. This sort is based on the following formula:

$$k = u^{-1/(\alpha-1)} \quad (1)$$

where *k* is the degree of each vertex, *u* is a random number from a uniform distribution between 0 and 1 and α is the exponent of the power-law, always greater than 1. This generates a random real number greater than 1, with a distribution of $P(k) \sim k^{-\alpha}$. The closest integer is taken as the node's degree.

We create a vector of size N, which is a list with all nodes degrees. For a node with a degree *k*, let us define *h* as the open connections of that node (which, in the initial state, equals *k*). The algorithm to create the networks is as follows: 1. Choose the node with the higher available degree (*h*) in the network; 2. Connect that node with *h* other nodes, randomly selected from a list with available nodes, thus exhausting the links of the chosen node; 3. Subtract one link from the available number of links (*h*) of the nodes connected; Repeat steps 1, 2 and 3 until there are no more nodes with open connections.

To create a community structure, we generate a number of networks equals to the desired number of communities, and then connect them to each other, thus creating a network with M communities (being M the amount of initial networks). The quantity of links between communities is very important to community structure detection [5], so let us define ξ as the fraction between intra-communities connections and inter-communities connections, or the number of edges between vertices of different communities divided per the number of edges between vertices that belongs to the same community. We set a value for ξ for each network, calculate the number of edges between communities for that ξ, and then select (randomly) pairs of vertices from different communities that will be connected, creating a network with well-defined (or not, if ξ is too high) community structure.

We have chosen to test the hereafter called Newman [5] and Clauset [2] methods to find communities within a network. Both algorithms were implemented in R statistical software [6] by [3]. We have tested those algorithms in scale-free networks generated by the algorithm described above.

Both methods were tested in undirected networks with 500 vertices, 5 communities (with 100 nodes each), $\alpha = 1.5$ (what creates, in general, networks with 10.000 edges), and ξ varying from 10 to 50%. For each case, we generate 100 networks, in order to determine execution time and

accuracy, defined here as the proportion between the nodes correctly assembled in their communities and all the nodes in the network.

As noted in Figure 1, the Newman method presents a better accuracy for networks with a clear community structure (ξ = 10%). However, this method is much more time-consuming, with a mean of 79,08 seconds per analyzed network, in comparison with the Clauset method, which presents a mean of only 0,02 seconds.

Figure 1 also shows that the performance of Newman method is very compromised by the addition of inter-communities edges, while this is not the case for the Clauset method, which has a lower decrease in accuracy for ξ = 50%. In this case, the mean time required for Newman and Clauset methods are 306,5s and 0,06s, respectively.

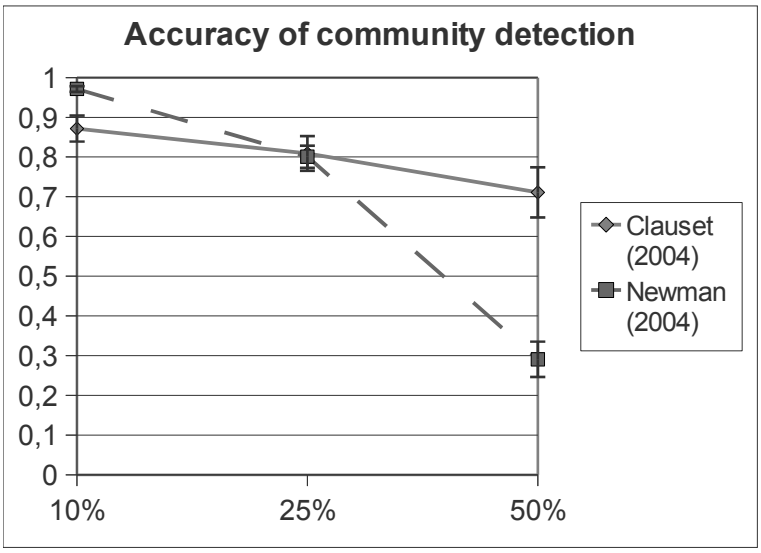

Figure 1

Mean accuracy of methods for three values of ξ. Error bars represents samples standard deviation.

We conclude that for the networks generated with the algorithm presented here, the Newman method will only bring advantage to research when analyzing small and with clear community structure networks. Otherwise, not just the Clauset method have similar or better accuracy, but its lower time-consuming enable the analysis of great order networks.

Keywords: *Complex Networks, Community Detection and Identification, Scale-Free Networks*

References

[1] A.L. Barabási, R. Albert, *Science* 286, 509-512 (1999).

[2] A. Clauset, M.E.J. Newman, C. Moore, *Physical Review E* 70, 1-6 (2004).

[3] G. Csardi, T. Nepusz: The igraph software package for complex network research, *InterJournal, Complex Systems 1695*. 2006. http://igraph.sf.net

[4] T. Kalisky, R. Cohen, S. Havlin, *Lecture Notes on Physics* 650, 3-34 (2004).

[5] M.E.J. Newman, M. Girvan, Finding and Evaluating Community Structure in Networks. *Physical Review E*, *vol*, 69pp026113-1 (2004).

[6] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. 2010. http://www.R-project.org.