The Third International Symposium on Optimization and Systems Biology (OSB'09)
Zhangjiajie, China, September 20–22, 2009

Extracting Community Structure of Complex networks by Self-Organizing Maps

Zhenping Li[1] Rui-Sheng Wang[2] Luonan Chen[3]

[1]School of Information, Beijing Wuzi University, Beijing 101149,China
[2]School of Information, Renmin University of China, Beijing 100872, China.
[3]Department of Electrical Engineering and Electronics,
Osaka Sangyo University, Osaka 574-8530, Japan

Abstract Identifying community structure is an important issue in network science and has attracted attention of researchers in many fields. It is relevant for social tasks, biological inquires, and technological problems. In this paper, we proposed a new approach based on self-organizing map to community detection. By using a proper weight-updating scheme, a network can be organized into dense subgraphs according to the topological connection of each node. Besides unweighted undirected networks, our method can also be used to detect communities in both weighted and bipartite networks.

Keywords Complex network; Community detection; Self-organizing map; Neural networks

1 Introduction

It has been shown in the past that many real systems can be represented as networks, in which nodes denote the objects of interest and edges that connect nodes describe the relationships between them. Such systems include social systems, ecological systems, and cellular systems [1]. The networks in these systems have been revealed to have many interesting topological properties, such as the small-world property and power-law degree distribution [2]. One topic of current interests in network science is the detection of community structure in complex networks. A community in a network can be qualitatively described as a collection of vertices within a graph that are densely connected among themselves while being loosely connected to the rest of the network. Many social and biological networks exhibit such a community or modular structure [3]. Uncovering such community structure not only helps us understanding the topological structure of large-scale networks, but also revealing the functionality of each component.

A number of methods have been proposed to detect community structure underlying a network, which can be roughly divided into two classes. One class is based on the clustering or aggregation principles, such as betweenness-based methods [3], random walk methods [5], machine learning methods [6, 7]. The second class is to build an optimization model to maximize certain modularity measures or criteria, such as modularity function Q [8, 9, 12, 11, 10], modularity density D [13, 14], information-theoretical method [4].

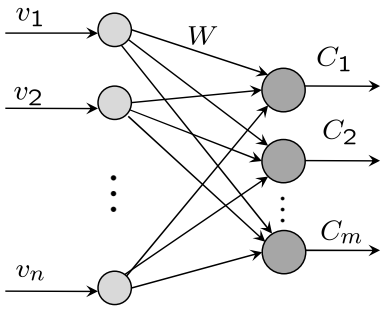


Figure 1: A two-layer self-organizing map for community detection.

The underlying difficulty in community detection is that there is no unique definition for 'community' and thus evaluation of network partitions is not straightforward. On the other hand, optimization of modularity measures has been proved to be NP-hard, which means that there is no polynomial-time exact algorithm for the community detection problem unless NP=P. In this paper, we design a self-organizing map approach [15] for community detection in complex networks. According to the topological connection of each node, our approach automatically organizes a network into dense subgraphs without any heuristic manipulation. Besides unweighted undirected networks, our method can also be used to detect communities in both weighted and bipartite networks.

2 Methods

Consider a network G with n nodes and m putative communities (here m can be an upper bound of the number of communities). Let $A = [a_{ij}]_{n\times n}$ be the adjacency matrix of the network G, and $B = A + I$ (I is a unit matrix). The scheme of the self-organizing map for detecting community structure, shown in Figure 1, has n input neurons corresponding to the nodes $v_1, v_2, \cdots, v_n$ in G, and m output neurons representing putative communities $C_1, C_2, \cdots, C_m$. For each node i, the learning input $B_i \in R^n$ is a vector such that $b_{i,i} = 1$ and $b_{j,i} = 1$ if and only if v_j is adjacent to v_i in G, $b_{j,i} = 0$, otherwise. The connection weight matrix between input neurons and output neurons is $W = [w_{ij}]_{n\times m}$, where the weight w_{ij} expresses the possibility or membership degree that node v_i belongs to community C_j. If the input neuron v_i is mapped into the output neuron C_j, then the connection between v_i and C_j should be reinforced. Moreover, all other weights of the winner output neuron C_j are also modified according to the adjacency relationship between the corresponding nodes and the input node v_i, since two adjacent nodes are more likely to be in a same community.

The input vectors used in the learning phase are columns of the matrix B. For the node v_i, the corresponding input vector is $x = B_i$, where B_i is the i-th column of the matrix B, i.e. $x_i = 1$, and $x_j = 1$ if and only if v_i is adjacent to v_j. The discriminant function is the normalized correlation

$$\eta(W^j, x) = (W^j \cdot x)^T B (W^j \cdot x) \tag{1}$$

where W^j is the jth column of the weighted matrix W, and '$\cdot$' denotes the inner product between two vectors. The winner neuron $\bar{C}_j$ with respect to the input vector x is selected by the following rule:

$$\bar{j} = \arg\max_j \eta(W^j, x). \tag{2}$$

The weights associated with the winner neuron are updated as follows

$$W^{\bar{j}}(k+1) = \frac{W^{\bar{j}}(k) + \alpha B_i}{\|W^{\bar{j}}(k) + \alpha B_i\|_\infty}, \tag{3}$$

and the weights associated with the non-winner neuron are updated as follows

$$W^j(k+1) = \frac{W^j(k) + \alpha(1 - B_i)}{\|W^j(k) + \alpha(1 - B_i)\|_\infty}, \tag{4}$$

where α is the learning rate, and $j \neq \bar{j}$.

After the training phase, the nodes are mapped into no more than m communities. The communities are constructed according to the final connection weight matrix W. For the i-th node, we define that it belongs to the $\bar{j}$-th community if

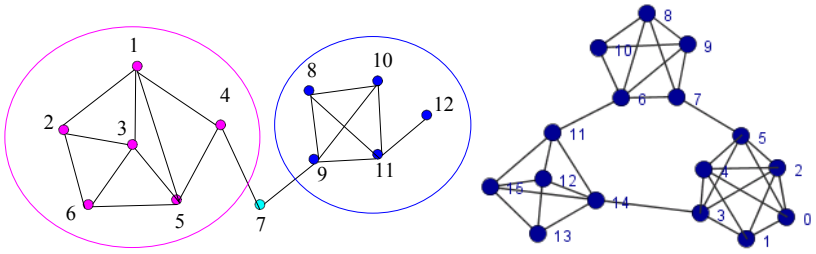
$$\bar{j} = \arg\max_j w_{ij}.$$

The details of our implementation of the SOM algorithm are described as follows:

- **Step 1. Initialization**
  Set the initial learning rate α_0 and the maximum number of iterations $MaxIter$. Randomly initiate $W(0)_{n\times m}$ and let $k = 0$. Compute the input matrix $B_{n\times n} = A + I$.
- **Step 2. Learning**
  Substep 2.1. Among all nodes in the network, randomly select a node v_i with the input vector $x = B_i$.
  Substep 2.2. For $j = 1$ to m, calculate $W^j(k)$, the connection between node v_i to the output neuron C_j. Calculate the discriminant function $\eta(W^j(k), x)$ by equation (1). Then determine a winner neuron $\bar{j}$ according to (2).
  Substep 2.3. Update the connection weight matrix $W(k+1)$ by the formulae (3) and (4).
  Substep 2.4. Repeat Substep 2.1 to Substep 2.2 until all nodes are learned.
  Substep 2.5. Update the learning rate parameter α if adaptive learning rates are used. If $\|W(k+1) - W(k)\| < \varepsilon$ or $k > MaxIter$, go to Step 3; otherwise, let $k = k + 1$ and go to Substep 2.1.
- **Step 3. Output**
  Classify all nodes into no more than m groups according to their final winner neurons. Return the corresponding communities of the network.

For those real world networks with known community structure, we introduce a normalized mutual information index as a measure of similarity between the real partition P and the identified partition P' [16]:

$$I_{NMI}(P', P) = \frac{-2\sum_{i=1}^{|P'|}\sum_{j=1}^{|P|} n_{ij}^{p'_i p_j} \log(n_{ij}^{p'_i p_j} n / (n_i^{p'_i} n_j^{p_j}))}{\sum_{i=1}^{|P'|} n_i^{p'_i} \log(n_i^{p'_i}/n) + \sum_{j=1}^{|P|} n_j^{p_j} \log(n_j^{p_j}/n)}$$



(a) a (b) b

Figure 2: Two simple examples of modular complex networks.

where $n_i^{p'_i}$ represents the number of nodes in cluster p'_i and $n_{ij}^{p'_i p_j}$ denotes the number of the shared elements between clusters p'_i and p_j. Obviously, $0 \leq I_{NMI} \leq 1$ with $I_{NMI}(P,P) = 1$. Note that this measure can compare two partitions with different number of communities.

3 Computational results

In this section, we do numerical experiments both on artificial networks and real networks. The algorithm is implemented in C++ and run on a 2.4G Hz Pentium 4 processor using Microsoft Visual C++ compiler 6. The software is available upon request. In the following experiments, the learning rate λ is initially set as 0.5 and linearly decreases from 0.5 to 0.1. Actually, we observe that our approach is very robust to this parameter.

3.1 Experiment on simulated networks

We first test our method on two small networks depicted in Figure 2. For the small network in Figure 2(a), our method can detect the two dense subgraphs as communities (denoted by circles). Node 7 can belong to either the left community or the right community. In fact, the connection weight of node 7 to either community is not distinctly larger. Such a case is very common in complex networks since some nodes are sparsely connected with other nodes and do not form a community. For the small modular network in Figure 2(b), the three apparent communities can be easily detected by our algorithm. From these two small networks, we can see that our algorithm can efficiently identify the underlying community structure, and especially is able to discard some sparsely connected nodes according to their connection weights.

Then, we test our method on a set of benchmark computer-generated networks designed in [3]. In this network set, each network has 128 nodes, which are divided into 4 communities of size 32 each. Edges are placed randomly with two fixed expectation values k_{in} and k_{out} so as to keep the average degree of a node to be 16 and the average k_{out} of each node's edge connecting to nodes in other communities. Figure 3 shows the fraction of nodes that are classified into their correct communities with respect to k_{out} by our method, the optimization of modularity density D [13], and the spectral algorithm [9] respectively. The performance of our method is a little better than that of spectral algorithm and marginally worse than that of optimization of D. But our method is very fast and its running time for each network is no more than 20 seconds, while the running time

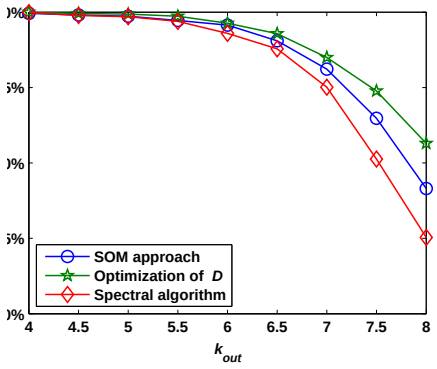


Figure 3: Comparison of the three methods on the computer-generated networks.

of solving the integer programming for optimization D is more than 1 minute [13]. This indicates that our method can be used for large scale networks.

3.2 Experiment on real world networks

We also test our method on several famous real world networks. For example, for the famous karate club network analyzed by Zachary [17], the self-organizing map can detect a partition identical to the two friend groups (data not shown). The journal index network constructed by Rosvall and Bergstrom [4] consists of 40 journals as nodes from 4 different fields: physics, chemistry, biology, and ecology and 189 links connecting nodes if at least one article from one journal cites an article in the other journal during 2004. Ten journals with the highest impact factor in the 4 different fields were selected. By using the self-organizing map method, we can partition the network into 4 communities correctly (see Fig 4). The method can also partition the network into two, or three modules if a small upper bound m is used and the result is consistent with that in [4] and [13].

The college football network of the United States has also been widely used as a benchmark test example in network science due to its natural community structure [3, 6, 13, 7]. It is the network representation of the schedule of Division I games for the 2000 season: The nodes in the network represent the 115 teams, while the edges represent 613 games played in the course of the year. The teams are divided into 13 conferences containing around 8-12 teams each. The proposed self-organizing approach can partition the network into conferences with a high degree of success, which is shown in Figure 5, where the nodes with the same shapes and colors are teams in a same conference, and the dense subgraphs in the layout are communities detected by our method. We can see that the correct rate of our method is more than 91%. The detected partition is very near to the real one since $I_{NMI}(P',P) = 0.9279$. The modularity degrees characterized by Q (D) on the original partition and on the detected partition are 0.5371 and 0.5811 (23.6900 and 38.9892), respectively, which means that the partition detected by our method is more reasonable from the topological view point. It is worth noting that even the upper bound m is set to be more than 13, for example, letting $m = 14, 15, 16, 17, 18$ and so on, we still can obtain the same community structure, with some empty modules, which means that

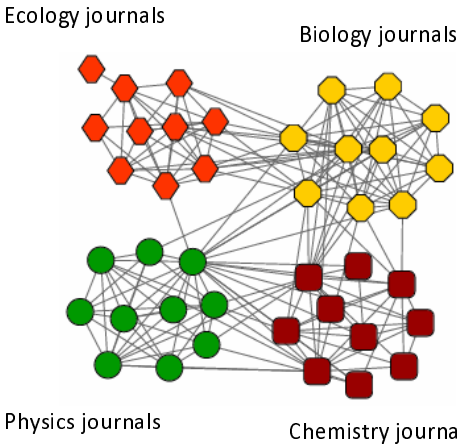


Figure 4: The detected community structure of the journal index network.

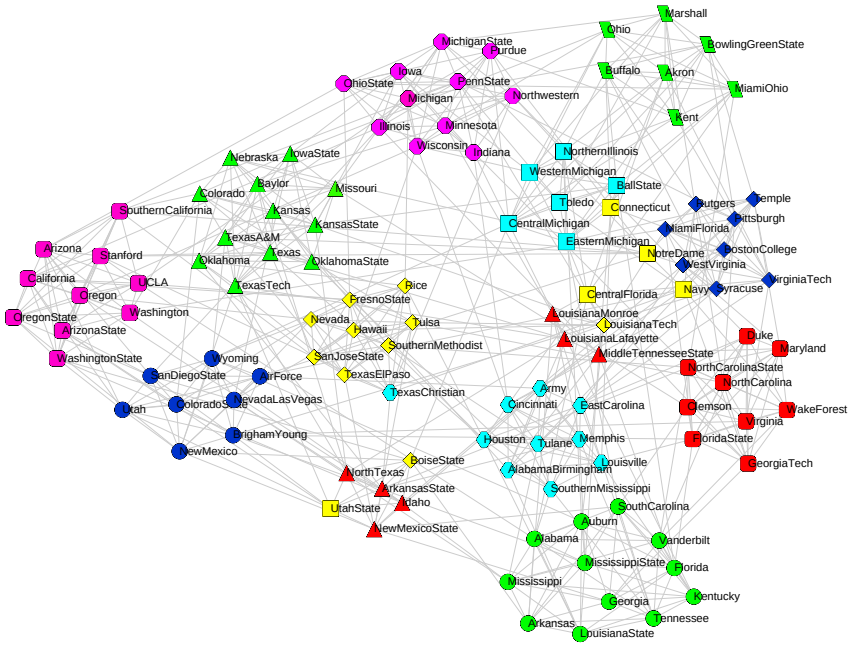


Figure 5: The detected communities in the football network.

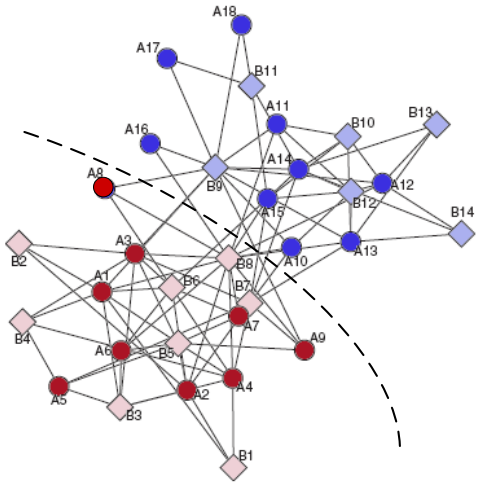


Figure 6: The detected community structure of the Southern women network.

our method is not so dependent on the prior knowledge and initial value about the number of communities.

3.3 Experiment on bipartite networks

Since there are many systems that are more suitable to represented as bipartite networks such as plant-animal mutualistic networks, scientific publication networks, and artistic collaboration networks. Hence, it is needed to identify the communities in bipartite networks. We observe that the self-organizing method can also detect communities in such bipartite networks. During the 1930s, some ethnographers collected data on social stratification in the town of Natchez, Mississippi [18]. They collected data on women's attendance to social events in the town and analyzed the resulting women-event bipartite network in light of other social and ethnographic variables. In [19], the authors analyzed the modules of this bipartite network by considering the projection of the bipartite network into the women's space and into the events' space. In this paper, we used our method to detect the communities in this bipartite network without projection. The partition result is shown in Figure 6, where circles represent women and diamonds represent social events. All the women and events in the same side of the dash line consist of a community. The partition captures the two-module structure of the network which coincides with the original subjective partition proposed by the ethnographers who collected the data.

4 Conclusion and discussion

In this paper, we proposed a new community detection algorithm based on self-organizing map. It can automatically organize a network into dense subgraphs without any heuristic manipulation. Besides the efficiency and effectiveness both on weighted and undirected networks, the self-organizing approach can also identify communities in bipartite networks. In addition, this community detection algorithm is suitable for very large networks without knowing the number of communities. In the future research, we

will explore the application of this method to detect communities in biological networks and general directed networks.

In this study, we simply use $A+I$ as the input matrix for the self-organizing map to see if the underlying community structure of a complex network can be uncovered according to such basic local connection information. The results indicate that the local connection of nodes indeed can tell much if mined properly. As a future research topic, it is interesting to examine other input data and give a systematic comparison, such as Laplasian matrices $A-D$, $D^{-1}A$, $D^{-1/2}AD^{-1/2}$ (D is the diagonal degree matrix), or any other graph kernels. It is noted that although self-organizing maps have a similar scheme with k-means, k-means cannot be directly applied to detect community structure of complex networks [20]. It, like many other methods, needs to first map the network topology into Euclidean vectors or other similarity vectors through some techniques, whereas this is not prerequisite in our method. In addition, validation of our method on large-scale inhomegenous benchmark networks is worth further exploration [11].

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant No.10631070, No.60873205, No.10701080 and Beijing Natural Science Foundation under Grant No. 1092011. It is also partially supported by Foundation of Beijing Education Commission under Grant No. SM200910037005, the Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (PHR(IHLB)), and Scientific Research Base foundation of Beijing Wuzi University.

References

[1] Chen, L., Wang, R.S., Zhang, X.S. *Biomolecular Networks: Methods and Applications in Systems Biology*. John Wiley and Sons, 2009.

[2] Albert, R., Barabási, A. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74, 47-97, 2002.

[3] Girvan, M., Newman, M. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99, 7821, 2002.

[4] Rosvall, M., Bergstrom, C. An information-theoretic framework for resolving community structure in complex networks. *Proc. Natl. Acad. Sci.*, 104, 7327, 2007.

[5] Rosvall, M., Bergstrom, C. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.*, 105, 1118, 2008.

[6] Zhang, S., Wang, R.S., Zhang, X.S. Uncovering fuzzy community structure in complex networks. *Phys. Rev. E*, 76:046103, 2007

[7] Wang, R.S., Zhang, S., Wang, Y., Zhang, X.S., Chen, L. Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures. *Neurocomputing*, 72, 134-141, 2008

[8] Newman, M., Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 26113, 2004.

[9] Newman, M. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*. 103, 8577-8582, 2006.

[10] Schuetz, P., Caflisch, A. Multistep greedy algorithm identifies community structure. *Phys. Rev. E*, 78, 026112, 2008.

[11] Lancichinetti, A., Fortunato, S., Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* 78, 046110, 2008.

[12] Guimerà, R., Amaral, L. Functional cartography of complex metabolic networks. *Nature*, 433, 895, 2005.

[13] Li, Z., Zhang, S., Wang, R.S., Zhang, X., Chen, L. Quantitative function for community detection. *Phys. Rev. E*, 77, 36109, 2008.

[14] Zhang, X.S., Wang, R.S. Optimization analysis of modularity measures for network community detection. *Lecture Notes in Operations Research*, 9:13-20, 2008.

[15] Zhang, X.S. *Neural Networks in Optimization*. Kluwer Academic Publishers, 2000.

[16] Danon, L., Daz-Guilera, A., Duch, J., Arenas, A. Comparing community structure identification. *J. Statist. Mech.: Theory and Experiment*, 09, P09008, 2005.

[17] Zachary, W.W. An informal flow model for conflict and fission in small groups. *J. Anthropol. Res.*, 33,452-473, 1977.

[18] Davis, A., Gardner, B. B., Gardner, M. R. *Deep South*, University of Chicago Press, Chicago, 1941.

[19] Guimerà,R., Sales-Pardo, M., Amaral, L.A.N. Module identification in bipartite and directed networks. *Phys. Rev. E*, 76, 036102, 2007.

[20] Gustafsson, M., Hörnquista, M. and Lombardi, A. Comparison and validation of community structures in complex networks. *Physica A*. 367, 559-576, 2006.