Cluster detection algorithm in neural networks*

David Meunier and Hélène Paugam-Moisy

Institute for Cognitive Science, UMR CNRS 5015
67, boulevard Pinel F-69675 BRON - France
E-mail: {dmeunier,hpaugam}@isc.cnrs.fr

May 16, 2006

Abstract

Complex networks have received much attention in the last few years, and reveal global properties of interacting systems in domains like biology, social sciences and technology. One of the key feature of complex networks is their clusterized structure. Most methods applied to study complex networks are based on undirected graphs. However, when considering neural networks, the directionality of links is fundamental. In this article, a method of cluster detection is extended for directed graphs. We show how the extended method is more efficient to detect a clusterized structure in neural networks, without significant increase of the computational cost.

1 Introduction

Many systems of interacting entities are represented as graphs, where actors are described by nodes, and interactions between actors can be formalized as edges between the nodes. Studies on complex networks have shown that social, biological and technological networks share a clusterized structure [1]. A definition of a "cluster" (also called "community", or "module") in a "weak sense" [2] could be that the elements inside a cluster interact more strongly with each others than with the other elements of the graph. Recently, Girvan & Newman [3, 4] have proposed an algorithm (GN method) to determine clusters in a given undirected graph, without any restriction on the size and the number of clusters.

The connectivity of neural networks is often either complete (all neurons are connected to all the others) or random. However, it has been shown that natural neural network topologies are neither random nor complete, but display an intermediate clusterized structure [5, 6]. In a neural network, the edges between nodes are directed, due to the unidirectionality property of synapses.

*Published in Proceedings of 14^{th} European Symposium on Artificial Neural Networks (ESANN 2006), M. Verleysen (ed.), p 19-24, Bruges, Belgium

We describe an extension of the GN method for directed graphs, where clusters consist of strongly connected components of the graphs [7]. As the topology constrains the dynamics, it is useful to determine how the nodes are linked at the topological level for understanding how they interact at the dynamical level [8]. We show, as illustration, a topology underlying a dynamical neural network obtained after an evolutionary optimisation [9]. The topology has a clusterized structure, and the extended cluster detection method is more efficient than the GN method for discovering clusters in the network directed graph.

2 Cluster detection for undirected graph

2.1 Betweenness centrality

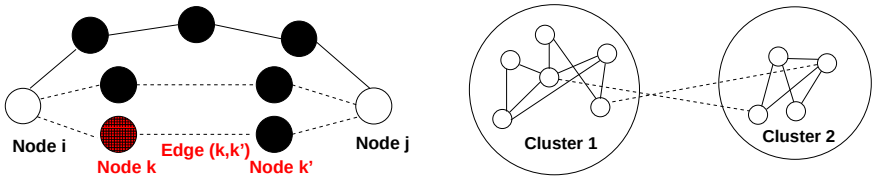

Figure 1: *Left:* Node-betweenness and edge-betweenness centralities. *Right:* Edges between nodes of two clusters have higher edge-betweenness centralities (dashed lines) than edges between nodes of the same cluster (solid lines).

The notion of *node-betweenness centrality* has first appeared in the field of social science [10] to determine the role of each actor (node) in a social network. This measure is based on the notion of shortest path in a graph. If two nodes are connected, there might exist several paths between them. Shortest paths are those with the smaller number of nodes along the path. There might exist several shortest paths between two nodes, if some paths have the same (minimal) length. Betweenness centrality for a node k is the number of shortest paths between two other nodes i and j that go through k (noted $\sigma_{ij}(k)$), divided by the total number of shortest paths that go from i to j (σ_{ij}). On Figure 1 (left), two shortest paths exist between i and j (dotted links), and only one crosses node k, thus $\sigma_{ij}(k) = 1/2$. The betweenness centrality of node k is given by the sum of all different pairs of nodes i and j in the graph:

$$C_B(k) = \sum_{i \neq j} \frac{\sigma_{ij}(k)}{\sigma_{ij}} \text{ with } i \neq k \text{ and } j \neq k \tag{1}$$

Girvan and Newman extend the definition to edges by defining the *edge-betweenness centrality*. On Figure 1 again, edge-betweenness centrality of edge $\{k, k'\}$ for nodes i and j is $\sigma_{ij}(\{k, k'\}) = 1/2$. Edge-betweenness centrality for a given edge $\{k, k'\}$ is the sum on all pairs i and j (different from k and k'):

$$C_B(\{k,k'\}) = \sum_{i \neq j} \frac{\sigma_{ij}(\{k,k'\})}{\sigma_{ij}} \tag{2}$$

2.2 Girvan and Newman cluster detection algorithm

Consider a graph with two clusters of highly interconnected nodes and few edges between them (Figure 1, right). For the shortest paths between nodes that belong to different clusters (dashed lines), it can be seen that most of the shortest paths go through the few edges that join the clusters together (solid lines). Hence these edges have a strong edge-betweenness centrality. GN method is based on this property. By removing iteratively the edges with highest edge-betweenness centrality, the clusters of the graphs are disconnected. The cluster detection method is a two-pass algorithm.

- The first pass consists in computing the edge-betweenness centrality for all the edges in the graph, removing the edge with the highest edge-betweenness centrality, then computing again the edge-betweenness centrality for all the edges in the resulting graph, etc... until all the edges have been removed.
- The second pass is the cluster building. At the beginning, each cluster consists of one single node. The edges between clusters are added by reading in reverse order the edges removed in the first pass. Whenever an edge joins two nodes that are part of two different clusters, all the nodes are grouped in a single cluster. Whenever an edge joins two nodes that are part of the same cluster, the set of clusters remains unchanged.

2.3 Modularity

The iterative method corresponding to the second pass gives successively several sets of clusters. By considering the reverse order of edge removal, the first added edges are more likely to be inside densily connected clusters, and the last added edges join together these densely connected clusters. However, how to determine when the algorithm switches between these two extreme behaviours, i.e. when to stop the cluster building algorithm for relevance? Newman & Girvan [4] introduce a measure called *modularity* (see also [11]):

$$\mathcal{M}(C) = \sum_{c=0}^{N_C} \left[\frac{d_c}{L} - \left(\frac{l_c}{L} \right)^2 \right] \tag{3}$$

where N_C is the total number of clusters in a given set C, d_c is the number of edges between nodes of a given cluster c, l_c is the total degree (i.e. the total number of edges) of nodes in cluster c and L the total number of edges in the whole network. The modularity is based on the edges in the initial network,

without edge removal. This measure equals 1 if all the edges are inside the clusters of C, and 0 if there is an equal proportion between edges inside and outside the clusters of C. The optimal set of clusters is the one with the highest modularity during the cluster building.

3 Extension to directed graphs

3.1 Algorithm

GN method has been mostly applied to undirected graphs. However, in the case of neural networks, the directionality of the edges is a key feature. A directed edge is classically referred as "arc". Consider two clusters A and B. After the addition of an arc from a node in A to a node in B, the nodes of cluster A are able to reach the nodes of cluster B, but there is no reason that the nodes of cluster B are able to reach the nodes of cluster A. If A and B were grouped, the new cluster will contain nodes that cannot reach some other nodes in the cluster via a directed path.

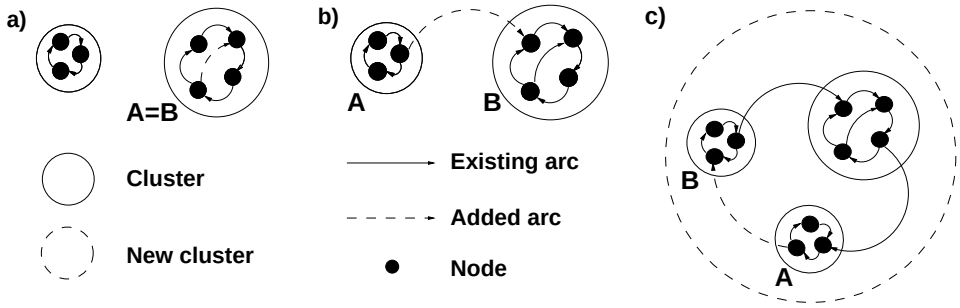

Figure 2: Description of the construction of arc-linked clusters.

We extend the cluster formation process by introducing the notion of *arc-linked cluster*. In this case, we use betweenness centrality definitions extended to the case of directed graphs [3]. Clusters with an arc between them are not always grouped to form a single cluster as in the GN method. When an arc is added from a node in cluster A to a node in cluster B, we check with a depth-first search whether there exists a directed path that goes from B to A:

- If cluster A is already the same as cluster B (case a in Figure 2), the set of clusters is unchanged.
- If such a path does not exist (b in Figure 2), an arc from A to B is stored, without modification of the cluster set.
- Otherwise (c in Figure 2), all the clusters along the path are grouped into a single cluster.

The present algorithm has the effect of building incrementally clusters that are strongly connected components of the directed graph.

3.2 Complexity

Consider a network with n nodes and m edges. In a sparse network, $n \sim m$. Edge-betweenness centralities for all the edges are computed with the Brandes algorithm [12], with complexity $O(mn)$, at each edge deletion, and $O(nm^2)$ for the complete removal phase. For the GN method, the building phase complexity is $O(m)$. In the extended algorithm, the complexity is $O(m+n)$ for each edge addition, since we compute a depth-first search at each step. Hence the overall complexity for the cluster building phase is $O(m(n+m))$, higher than in GN method, but still negligible compared to the complexity of the removal phase.

4 Application of the extended cluster detection algorithm

For illustration, we compute the clusters in an example neural network topology for comparing the GN method and the arc-linked cluster detection method.

The neural network is composed of dynamical objects (spiking neurons and temporally varying synapses) and controls the behaviour of a virtual robot whose performance in a predator-prey environment is optimized by an evolutionary algorithm [9]. For evaluating the computational gain from one generation to the next, we need to know the clusters of the neural network. The network topology is described by projections, i.e. arcs between groups of neurons that define all-to-all synaptic links between neurons of two connected groups. The network is composed of 100 internal, 9 input and 4 output groups of neurons. The two cluster detection algorithms are compared on the graph of a network with 426 projections (Figure 3). The nodes of the graph correspond to the internal groups of neurons, and the arcs correspond to projections between groups.

Figure 3 displays the variations of the maximal edge-betweenness centrality during the edge removal process and the modularity, for the GN cluster building algorithm (dashed line) and for the extended algorithm (solid line). Arrows clarify the order of edge deletion (left) and the reverse cluster building order (right). The cluster building graph is presented in reverse order, since the number of removed links is related to the number of added links. The vertical bar indicate the highest maximal edge-betweenness centrality on both graphs.

The optimal set of clusters should correspond to the peak of the maximal edge-beweenness centrality achieved during the edge removal process, because the strong drop that follows the peak corresponds to a maximal cluster disconnection (see section 2.2). However, with the GN method, the peak of modularity happens sooner (128 added links) in the cluster building process than the peak of maximal edge-beweenness centrality. Conversely, with the extended method, the peak of modularity happens (390 added links) close to the drop of maximal

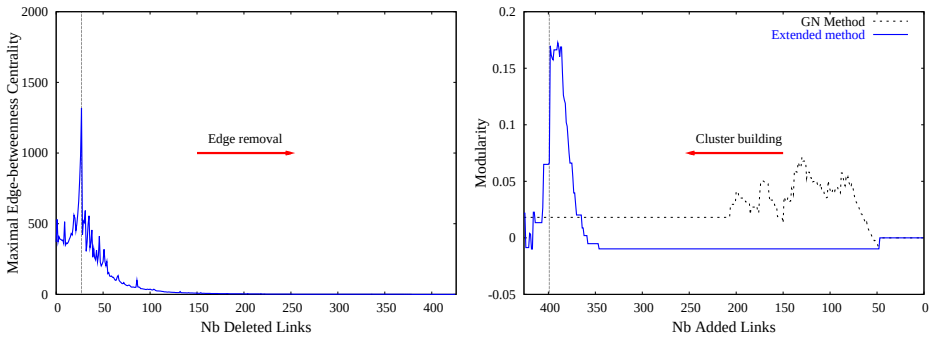

Figure 3: *Left:* Maximal edge-betweenness centrality during the edge removal process. *Right:* Modularity during cluster building with the GN method and the arc-linked cluster detection.

edge-beweenness centrality ($27 = 426 - 399$ deleted links). Moreover, the peak is narrower and much higher (0.17 compared to 0.07) than with GN method. Last, the optimal set of clusters obtained by arc-linked cluster detection is more pertinent in the context of the neural network considered as example.

5 Conclusion

We have extended a cluster detection method to the case of directed graphs, well suited to study neural networks since the directionality of edges is a key feature. We have shown that the arc-linked cluster detection method achieves a narrower and higher modularity peak than the GN method and results in a more pertinent optimal set of clusters. The identification of clusters is particularly suitable to show the emergence of a modular topology in neural networks and provides a framework to study the dynamics of functional modules [8].

References

[1] M.E.J. Newman. The structure and function of complex networks. *SIAM Rev.*, 45:167–256, 2003.

[2] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. U.S.A.*, 101(9):2658–2663, 2004.

[3] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.*, 99(12):7821–7826, 2002.

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

[5] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(440-442), 1998.

[6] O. Sporns and G. Tononi. Classes of network connectivity and dynamics. *Complexity*, 7(1):28–38, 2002.

[7] J.T. Gross and J. Yellen. *Graph Theory and Its Applications*. CRC Press, Boca Raton, FL, 2nd edition, 2005.

[8] S.H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[9] D. Meunier and H. Paugam-Moisy. Evolutionary supervision of a dynamical neural network allows to learn with on-going weights. In *Proc. of IJCNN'05*, pages 1493–1498, 2005.

[10] L.C. Freeman. Centrality in networks: I. Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[11] R. Guimerà and L.A.N. Amaral. Cartography of complex networks: modules and universal roles. *J. Stat. Mech.: Theory Exp.*, pages 1–13, 2005.

[12] U. Brandes. A faster algorithm for betweenness centrality. *J. Math. Sociol.*, 25:163–177, 2001.