Renormalization group on complex networks

Saurabh Gandhi

05/15/14

Abstract

In this article, I describe a renormalization group approach for complex networks, as applied by Rozenfeld et.al. in 1. A large number of naturally occuring networks are scale-free, i.e. display a powerlaw degree distribution 2. Moreover, many of these networks display both the small world property, and fractal characteristics. The RG approach demonstrates a transition between fractal and small world networks with increasing scale, and enables the classification of networks in one of the two universality classes. Assuming an underlying fractal structure, it also provides a method for determining the distribution of overlying shortcuts in the network.

1 Introduction

Many naturally occuring networks are found to be scalefree, i.e. they have a powerlaw degree distribution. However, these networks display properties of two seemingly contradictory structures - smallworldness, where the diameter of the network (longest node-node distance) increases logarithmatically with the number of nodes, and fractal, where the diameter is expected to increase much more rapidly, as a powerlaw.

In order to explain this, the authors assume an underlying fractal structure of the network, with some long distance shortcuts. Depending on the distribution of these shortcuts, the network, which appears locally fractal, may on larger scales, have a topology similar to either a fractal or a completely connected graph. For example, let r represent the distance between two nodes on a network. For shortcuts distributed according to $p(r) = r^{-\alpha}$, a small α indicates large number of long distance shortcuts. In this case, we may expect, with renormalization, that the shortcuts connect more of the renormalized nodes, and hence appear globally small world, ultimately leading to a complete graph. For a large α, the shortcuts are mostly local, and are lost during renormalization, so that the renormalized graph is similar to the underlying fractal graph.

The box covering algorithm, as described in 3, is used for renormalizing the network. The renormalization technique also gives a way of deducing the distribution of shortcuts in a naturally occuring network, which is a very useful metric for understanding information/energy flow across these networks. In the rest of this article, I explain the box covering method, the renormalization technique and the implementation algorithm as used by the authors. I also try to apply the technique to the network of neurons in C. *elegans*.

2 Network RG

Box covering The RG is performed using the box covering technique. All nodes of the network are boxed so that nodes within each box are at a distance of up to b from one another. Each of the boxes then acts as a 'super'node. Two supernodes in the renormalized network are connected if there is an edge in the original network connecting some of their constituent nodes.

Box covering is also used for defining the dimensionality of the network. The box dimension, d_B, is defined by $N_0/N_b = b^{d_B}$

Network structure We assume an underlying fractal network with N_0 nodes, denoted G_0. Shortcuts are added on top of this network, with probability distribution $p(r) = Ar^{-\alpha}$. The network with shortcuts is denoted by G'.

$G_b = R_b(G_0)$, the renormalized fractal network, is also a fractal network with the same structure (degree distribution) as the original network ($R_b(G_0) = G_0$, up to an overall scaling in the number of nodes). Note that fractal networks can be defined as those having the above property. Thus, it is sufficient to look at the shortcuts alone in order to understand the RG flow of the network G'.

RG flow We need to obtain the distribution of shortcuts, p_b, after the renormalization.

$$p_b(r) = 1 - P(\text{there are no shortcuts at distance } r \text{ in } G'_b)$$

Which implies, there are no shortcuts at distance br across the b^{d_B} nodes in G' corresponding to the nodes in G'_b, giving

$$P(\text{there are no shortcuts at distance } r \text{ in } G'_b) = (1 - p(br))^{b^{d_B} * b^{d_B}}$$

$$p_b(r) = 1 - (1 - A(br)^{-\alpha})^{b^{2d_B}} \tag{1}$$

In the limit of $b \to \infty$, we have

$$\lim_{b\to\infty} p_b(r) = 1 - exp(-Ar^{-\alpha}b^{-\alpha+2d_B}) \tag{2}$$

There is, thus, a critical value of $s \equiv \alpha/d_B = 2$, below which $p(b) \to 1$. In this phase, the network goes towards a fully connected state, implying that at larger scales, it shows the small world properties. For $s > 2$, $p(b) \to 0$, which implies that in this phase, the shortcuts disappear at large scales and the network returns to its underlying fractal structure. This is in accordance to the intuitive expectation that if the probability of long distance links is high (small α), the graph will show small world characteristics. For $s = 2$, the RG flows towards another stable fixed point consisting of a fractal network with overlying shortcuts given by $\tilde{p}(r) = 1 - \exp(-Ar^{-2d_B})$. As a consistency check, for large r, this probability goes as $Ar^{-2d_B} = p(r)$ with $\alpha = 2d_B$.

Average degree The average degree (number of connections per node), z, provides a rough estimate of the efficiancy of information flow on the network. We can look at the behaviour of z under RG as follows:
Let z_0 be the average degree of G_0, and correspondingly, z_b, z' and z'_b for $R_b(G_0)$, G', $R_b(G')$.
Let the number of shortcuts between length 1 and L is given by $M(L)$. Thus, $z' - z_0 = 2M(D)/N_0$, where D is the diameter of the network G_0. The factor of 2 is because each shortcut adds to the degree of two nodes.

$$\begin{aligned} M(L) &\approx \int_1^L p(r)(d_B r^{d_B - 1} dr) \\ &= d_B \int_1^L A r^{-\alpha} r^{d_B - 1} dr \\ &= \tfrac{A}{1-s}(L^{d_B(1-s)} - 1) \end{aligned} \tag{3}$$

Thus,

$$z' - z_0 = \frac{2A}{1-s}\left(\frac{D^{d_B(1-s)} - 1}{N_0}\right) \tag{4}$$

After renormalizing by b, shortcuts smaller than b will be within the supernodes, and will not add to the degree. Also, renormalization does not change average degree in the pure fractal network. Thus,

$$z'_b - z_b = z'_b - z_0 = \frac{2(M(D) - M(b))}{N_b} \tag{5}$$

In the limit of $D \to \infty$, we find $z'_b - z_0 = (z' - z_0)x_b^{\lambda}$, where $x_b = b^{d_B}$ and λ is given by:

$$\begin{aligned} \lambda &= 1 && \text{if } s \leq 1 \\ &= 2 - s && \text{if } s > 1 \end{aligned} \tag{6}$$

As observed before, if $s > 2$, $\lambda < 0$, and the difference in the degrees of the renormalized network and pure fractal network goes to 0, i.e. shortcuts are lost at larger scales. For $s < 2$, the average degree increases till we get a fully connected graph. Further, by fitting z'_b to the scaling function, the value of λ for a network can be obtained.

3 Analysis of a network

In this section, I implement some of the ideas described above for a natrually occuring network. The network is the neural network of C. *elegans*, and the data was obtained from the freely available source at 4.

There are multiple types of edges in the neural network, coresponding to distinct communication channels, such as electrical impulses, chemical signals etc. Also, the network is directed. However, in order to simplify the analysis, all directed edges have been replaced by undirected ones. Further, analysis of only one type of edge leads to a very small network where the renormalization does not work well. For that reason, I have treated all kinds of edges equivalently.

Under these assumptions, the network contains 281 nodes and 2405 edges, with a diameter of 4.

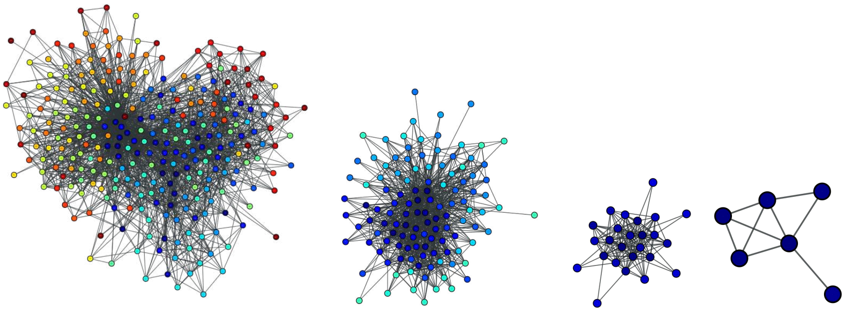
Box covering algorithm In order to compute the box dimensionality of a graph, it is covered with a minimum possible number of boxes, and the dimension is given by $N_0/N_b = b^{d_B}$. Song et.al. show that the optimal network covering problem can be mapped on to a vertex coloring problem as follows:

To cover a network, G, with boxes of size l, obtain the dual graph G' of G by removing all edges in G and connecting nodes that are separated by a distance greater than l in G. The vertex coloring problem is to color the vertices of G' using minimum possible colors in such a way that no edge connects two nodes of the same color. Clearly, under this scheme, all nodes separated by a distance larger than l in G will have a distinct color, and we define boxes as sets of nodes with the same color.

The vertex coloring, and hence box covering, algorithm is an NP hard problem, but the authors describe a 'Greedy' algorithm for obtaining an optimal solution. The following analysis is based on this algorithm.

Renormalization results Since $D = 4$, the largest box can be at most of size 4. Figure 1 shows the successive renormalized versions of the network. Figure 2 shows the fractal dimension, which turns out to be 1.426, by fitting a linear curve between $\log(x_b)$ and l_b.

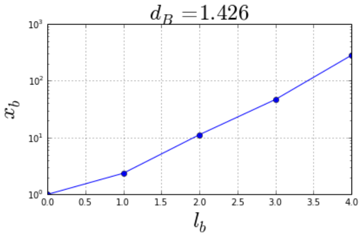
Figure 1: Successive renormalization of the neural network in C. *elegans*



(a) $N_0 = 281$ (b) $N_1 = 119$ (c) $N_2 = 25$ (d) $N_3 = 6$

This renormalization, however, gives only 3 points, from which it is not possible to make an accurate estimate of the scaling exponent λ, and hence α, the

Figure 2: Computing the dimensionality



distribution of overlying shortcuts. In the original paper, the authors carry out what they call (but do not describe) partial renormalization so as to obtain a more reliable estimate of λ

4 Conclusion

The RG procedure applied to networks can be used to extract important topological information about the network, namely, its dimensionality (and hence the underlying fractal structure) and the distribution of overlying shortcuts. Rozenfeld et.al. use this technique to analyze more real world networks such as the WWW, and biological networks like the metabolism network in E. *coli* and a yeast protein interaction network. They find that the WWW and the metabolic network have $s < 2$, i.e. they have many long distance shortcuts, and thus, have higher efficiency for information/energy flow as compared to the other two networks, which have $s > 2$, and have a strong fractal character at larger scales. This technique can thus provide useful insight towards understanding networks.

5 References

1. Phys. Rev. Lett. 104, 025701
2. Science 286 (5439), 509-512
3. J. Stat. Mech. (2007) P03006
4. http://www.wormatlas.org/neuronalwiring.html
5. http://graph-tool.skewed.de/ is a python library for analysing and visualizing graphs. It was used for generating the images, and computing metrics such as the average degree, diameter etc.