Modern Physics Letters B, Vol. 22, No. 7 (2008) 459–466

World Scientific
www.worldscientific.com

ALGORITHMS FOR FRACTAL DIMENSION CALCULATION

O. SHANKER

Hewlett Packard Company,
16399 West Bernardo Drive, San Diego, CA 92130, USA
oshanker@gmail.com
http://<www.geocities.com/oshanker/graphdimension.htm>

Received 18 August 2007
Revised 29 August 2007

Algorithms to calculate the fractal dimension of a complex network are presented. One of the algorithms is applied to a parametrized class of models whose fractal dimension transitions from one to two. For the system size we considered here (16384 nodes), the transition takes place from one at $p = 0$ to essentially two at the small value $p = 0.03$. This seems to indicate that the transition is likely to become infinitely sharp and occur at $p = 0$ as the system size increases to infinity.

Keywords: Graph algorithms; complex networks; fractal dimension.

1. Introduction

Estimating the fractal dimension of a complex network or graph (in the large system limit) is an important problem in Statistical Mechanics. This work introduces algorithms to calculate the fractal dimension. We use one algorithm to study a class of graphs. Section 2 reviews the definition and properties of fractal dimension. Section 3 presents algortithms to calculate fractal dimension. Section 4 introduces a class of models whose dimension is fractal and applies one of the algorithms to calculate the fractal dimension. Section 5 studies the vertex degree distribution for the class of models. Regular lattices typically have a narrow spread of vertex degree. Finally, the conclusion is presented.

2. Fractal Dimension Definition

A mathematically elegant definition of dimension for complex networks has been given based on the complex network zeta function[1] and the graph surface function.[2] These functions have been applied to study patterns in Language Analysis.[3] The fractal dimension thus defined has been compared[4] with the properties of dimensions defined in other discrete contexts, e.g., complexity classes in theoretical computer science.[5–11] Reference 4 also derived results about the analytic structure of the complex network zeta function using the theory of Dirichlet series. In this section, we will briefly review the definition.

Let us denote by r_{ij} the distance from node i to node j of a complex network (the length of the shortest path connecting the first node to the second node). r_{ij} is ∞ if there is no path from node i to node j. The graph surface function, $S(r)$, is defined as the number of nodes which are exactly at a distance r from a given node, averaged over all nodes of the network. We define the volume[2] for a complex network as

$$V(r) = \sum_{i=1}^{r} S(i)\,. \tag{1}$$

The complex network zeta function $\zeta_G(\alpha)$ is defined as

$$\zeta_G(\alpha) := \frac{1}{N} \sum_{i} \sum_{j \neq i} r_{ij}^{-\alpha}\,, \tag{2}$$

where N is the graph size, measured by the number of nodes. The definition Eq. (2) can be expressed as a weighted sum over the node distances. This gives the Dirichlet series expression for the complex network zeta function:

$$\zeta_G(\alpha) = \sum_{r} S(r)/r^{\alpha}\,. \tag{3}$$

If the average degree of the nodes (the mean coordination number for the graph) is finite, then there is exactly one value of α, $\alpha_{\text{transition}}$, at which the complex network zeta function transitions from being infinite to being finite. This has been defined as the dimension of the complex network. For regular discrete d-dimensional lattices $\mathbf{Z}^d$ with distance defined using the L^1 norm

$$\|\mathbf{n}\|_1 = \|n_1\| + \cdots + \|n_d\|\,, \tag{4}$$

the transition occurs at $\alpha = d$. For a one-dimensional regular lattice, the graph surface function $S_1(r)$ is exactly two for all values of r. (There are two nearest neighbors, two next-nearest neighbors, etc.) Thus, the complex network zeta function $\zeta_G(\alpha)$ is equal to $2\zeta(\alpha)$, where $\zeta(\alpha)$ is the usual Riemann zeta function. The $S_d(r)$ satisfy the recursion relation,

$$S_{d+1}(r) = 2 + S_d(r) + 2\sum_{i=1}^{r-1} S_d(i)\,. \tag{5}$$

This result follows by choosing a given axis of the lattice and summing over cross-sections for the allowed range of distances along the chosen axis. Asymptotically, $S_d(r) \to 2^d r^{d-1}/\Gamma(d)$ and $\zeta_G(\alpha) \to 2^d\zeta(\alpha - d + 1)/\Gamma(d)$ as $r \to \infty$ ($r \to \infty$ corresponds to $\alpha \to \alpha_{\text{transition}}$). $\zeta_G(\alpha) \to 2d$ as $\alpha \to \infty$. In applying Eq. (5) to calculate the surface function for higher values of d, we need an expression for the sum of positive integers raised to a given power k. The following result was presented in an earlier work[1] (using Bernoulli numbers instead of the zeta function values used below):

$$\sum_{i=1}^{r} i^k = \frac{r^{k+1}}{(k+1)} + \frac{r^k}{2} + \sum_{j=1}^{(k+1)>2j} \frac{(-1)^{j+1} 2\zeta(2j) k! r^{k+1-2j}}{(2\pi)^{2j}(k+1-2j)!}\,. \tag{6}$$

Another formula which can be used recursively is

$$\sum_{k=1}^{n} \binom{n+1}{k} \sum_{i=1}^{r} i^k = (r+1)((r+1)^n - 1)\,. \tag{7}$$

Reference 4 studied the complex network zeta function as a function of the complex variable s instead of the real variable α, i.e.,

$$\zeta_G(s) = \sum_r S(r)/r^s\,, \tag{8}$$

and derived many interesting properties using the theory of Dirichlet series.[12,13] The results[4] are summarized here.

Let us define

$$\gamma = \limsup_{r\to\infty} \frac{\log V(r)}{\log r}\,. \tag{9}$$

Then the following result is true for the fractal dimension.

Theorem 1. *The transition of the series* 3 *from non-convergence to convergence occurs when* α *crosses the value* γ *defined in Eq.* (9) *from below, i.e.,* $\alpha_{\text{transition}} = \gamma$.

The region of convergence of the sum 8 is given as below.

Theorem 2. *The region of convergence for the series* 8 *is the half-plane lying to the right of the real point* $\alpha_{\text{transition}}$.

The analytic structure of the point of transition is described by the theorem below.

Theorem 3. *The real point of the line of convergence of the series* 8, $\alpha_{\text{transition}}$, *is a singular point of the function* $\zeta_G(s)$.

The following integral representation for the complex network zeta function can be used to derive many further analytic properties of the complex network zeta function.

$$\zeta_G(s) = \sum_{n=1}^{\infty} \frac{S(n)}{n^s} = \frac{1}{\Gamma(s)} \int_0^{\infty} x^{s-1} \left(\sum S(n) e^{-nx} \right) dx\,. \tag{10}$$

Another integral expression for the volume in terms of the complex network zeta function follows also from the theory of Dirichlet series:

$$V(r) = \sum_{n=1}^{r} S(n) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \zeta_G(s) e^{\omega s} \frac{ds}{s}\,, \tag{11}$$

where the integration is performed in the region of convergence of the series, $c > \alpha_{\text{transition}}$, and $\log r < \omega < \log(r+1)$.

3. Algorithm

To calculate the graph surface function, for each node we need information about all shortest paths of a given length to any other node in the graph. There are many algortithms to calculate all shortest paths for a single node (single source shortest paths), and to calculate the "all pairs shortest paths"[16] (which means to find the shortest path between any two nodes in the graph). Since we need information only on paths up to a given length, we have to modify the algorithms to stop on finding all paths of a given length. In this section, we present two algorithms for calculating the fractal dimension.

The classic shortest path algorithms start with some initial upper bounds on the node distances, and then apply one of the following two types of relaxation:

- *Edge Relaxation:* Test whether travelling along a new edge reduces the currently known path lengths to the node.
- *Path Relaxation:* Test whether travelling through a given vertex reduces the currently known path lengths between any pair of nodes.

The nodes are assumed to have information only about their nearest neighbors. The initial bounds on the distances could be, for example, a large sentinel value for all nodes except for nearest neighbor nodes, for which the distance is set to one. A widely-known algorithm for finding the single source shortest paths is Dijkstra's algorithm. This algorithm builds the tree of shortest paths by starting with the source node, and adding one edge at a time, always choosing an edge which gives the shortest path from the source to a node not already on the tree. For "all pairs shortest paths" in dense graphs, Floyd's algorithm is useful. If the graphs are sparse, the Bellman–Ford algorithms are efficient for the "all pairs shortest paths".

We choose the Bellman-Ford class of graph algorithms as the basis for our first algorithm. This is because these algorithms are simple, and are efficient for the problem at hand. In this class of algorithms each node maintains a one-dimensional vector of its distances from all other nodes. The algorithm considers the network's edges in any order, and relaxes along each edge, and makes N such passes. Initially the distance vectors contain upper bounds, say infinity for all nodes except for nearest neighbor nodes, for which the distance is set to one.

In the implementation that we choose, each node iteratively exchanges its distance vector with its nearest neighbors. Thus, if node 0 is connected to node 1 and node 1 is connected to node 2, after the first iteration node 0 can update its distance vector with the information that node 2 is at a distance of 2. After k iterations, all nodes at a distance of k are identified. This property makes the implementation of the algorithm suitable for calculating the graph surface function and the complex network zeta function. This implementation can be called the distance vector algorithm.

The distance vector algorithm is simple and converges quickly. It is used on the internet in the Routing Information Protocol, a widely-used protocol in IP

networks. For these applications, some additional steps are necessary, since the links are dynamic and can go down at any time. For dynamic links, one has to guard against the algorithm not converging. Also, in these applications one needs to know not only the distances to other nodes, but also routing information, i.e., the next hop that needs to be taken to reach any node. Thus, a set of protocols based on Dijkstra's algorithm is also widely-used in internet routing. For the static graph application that we consider here, the additional sophistication is unnecessary. The simplest implementation of the distance vector algorithm is all that is needed to calculate the complex network surface and volume functions, and from them the complex network zeta function.

For each round of iterations in the distance vector algorithm, one needs to do $O(N^2\langle k\rangle)$ additions and comparisions, where $\langle k\rangle$ is the average vertex degree for the graph and N is the number of nodes in the graph. Thus, if one is interested in finding nodes at a distance of r, $O(N^2\langle k\rangle r)$ operations will be required. The operations involved are very simple, essentially addition of one to the distance estimates, comparision, and assignments.

The other algorithm is based on the adjacency matrix for the complex network. The adjacency matrix A for a complex network is

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge joining vertices } i, j, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

(For undirected networks, A is symmetric.) If we square $(I+A)$ (where I is the unit matrix, with ones along the diagonal and zeros in the off-diagonal entries), then the off-diagonal entries of $(I+A)^2$ are non-zero only if there is a path of length 2 or 1 between the pair of nodes. For higher powers $(I+A)^k$ the off-diagonal entries are non-zero only if there is a path of length k or less between the node pair. Thus, by repeatedly squaring $(I+A)$, we can build up the information to calculate the graph volume for distances which are powers of two. One need not explicitly multiply the integers, one can use a boolean representation, in which $1+1=1$, $1+0=0+1=1$ and $0+0=0$ and $1*1=1$, $1*0=0*1=0$ and $0*0=0$. This may be a useful algorithm for dense graphs. For the sparse graph that arises in the model studied in Sec. 4, the distance vector algorithm is a better choice.

4. Application to Model

In this section, we review a class of models first presented in Ref. 2. The models have fractal dimension, and transition from a one-dimensional regular lattice to a two-dimensional regular lattice. The starting network is a one-dimensional lattice (a ring) of N vertices with periodic boundary conditions. Each vertex is joined to its neighbors on either side, which results in a system with N edges. The network is extended by taking each node in turn and, with probability p, adding an edge to a new location m nodes distant.

When the rewiring probability $p = 0$, we have a one-dimensional regular lattice of size N. When $p = 1$, every node is connected to a new location and the graph

Table 1. Dimension versus probability, for different seed values.

p	Model A	Model B	Model C	Mean	σ
0.0	1.00	1.00	1.00	1.00	0.00
0.005	1.31	1.30	1.29	1.30	0.006
0.01	1.51	1.48	1.53	1.51	0.02
0.015	1.62	1.63	1.69	1.65	0.03
0.02	1.73	1.70	1.81	1.75	0.05
0.025	1.76	1.80	1.88	1.81	0.05
0.03	1.85	1.86	1.93	1.88	0.03

is essentially a two-dimensional lattice with m and N/m nodes in each direction. For p between 0 and 1, we have a graph which interpolates between the one- and two-dimensional regular lattices. The graphs are parametrized by:

$$size = N, \tag{13}$$

$$shortcut\ distance = m,\ and \tag{14}$$

$$rewiring\ probability = p. \tag{15}$$

We calculated the graph dimension for $N = 16384$ and $m = 128$, and varied p between 0.0 and 0.03. We studied three models, with different starting seed values for the random number generator. We numerically evaluated the dimension of the models by doing a linear regression of the volume on distance on a log–log plot. Table 1 shows the dimension calculation for three different seed values for the random number generator, and the mean value and standard deviation. Figure 1 shows the dependence of the dimension on the probability of shortcuts. The transition from a dimension of 1 to 2 occurs sharply, as p varies from 0 to 0.03. Also, the slope of the graph is large at $p = 0$ and falls off monotonically. Thus, it seems likely that when the system size N approaches ∞ the transition from one- to two-dimensional behavior will occur as an infinitely sharp jump at $p = 0$. The author would like to thank the anonymous referee for pointing out that this possibility needs to be mentioned. This question is being studied further. What is interesting about this behavior is that we have a large range of probability values where the dimension is integer, but where the graph is not a discrete regular lattice. As we see from the results in Sec. 5, not all properties of the graph exhibit such a sharp transition.

5. Vertex Degree Distribution

Table 2 shows the mean vertex degree and the standard deviation of the vertex degree for different probability values. The mean vertex degree $\langle k \rangle$ varies as $2 + 2p$, where p is the probability of shortcuts. The standard deviation σ varies as $\sqrt{2p(1-p)}$. The maximum variation in the vertex degree distribution occurs when the model is farthest from being a regular lattice, i.e., when $p = 0.5$. The variation of the average node vertex degree and the standard deviation with the

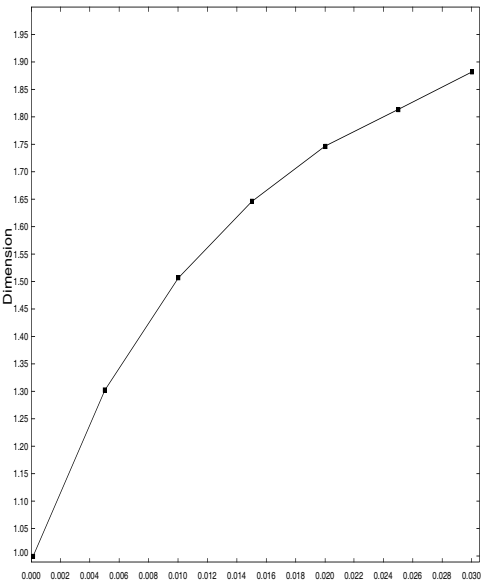

Fig. 1. Dependence of the dimension on the probability of shortcuts.

Table 2. Vertex distribution versus probability.

Probability p	Mean vertex degree $\langle k \rangle$	Standard deviation σ
0.0	2.00	0.00
0.1	2.19	0.42
0.2	2.39	0.56
0.3	2.59	0.64
0.4	2.79	0.69
0.5	2.99	0.71
0.6	3.18	0.70
0.7	3.39	0.66
0.8	3.59	0.57
0.9	3.79	0.44
1.0	4.00	0.00

probability p follows from the fact that there are $(1-p)^2$ nodes of degree 2, $2p(1-p)$ nodes of degree 3, and p^2 nodes of degree 4.

There is an interesting relation between the vertex degree distribution and the eignevalues of the adjacency matrix. It can be shown that the largest eigenvalue of the adjacency matrix has a lower bound given by the average vertex degree of the graph. For the model studied here, the lower bound becomes the largest eigenvalue at the end points of the model, $p = 0$, when the largest eigenvalue is two, and $p = 1$, when the largest eigenvalue is four.

6. Conclusion

Algorithms to calculate fractal dimension were presented, and applied to study the fractal dimension of a class of models. The transition of the dimension in the models was seen to occur rather sharply from 1 at $p = 0$ to essentially 2 at the small value $p = 0.03$. The slope of the graph of dimension versus probability p is large at $p = 0$ and falls off monotonically. It appears that the transition will likely become infinitely sharp in the large system limit ($N \to \infty$) and occur at $p = 0$. The vertex degree distribution was studied in the models. It does not exhibit the sharp transition that the dimension exhibits. The vertex degree distribution shows the smallest variability when the models are close to regular discrete lattices.

References

1. O. Shanker, *Mod. Phys. Lett. B* **21** (2007) 639–644.
2. O. Shanker, *Mod. Phys. Lett. B* **21** (2007) 321–326.
3. O. Shanker and G. Motta, Use of word relationship graphs to analyze texts in different languages, Technical Report (2007), http://www.geocities.com/oshanker/TextAnalysis.pdf.
4. O. Shanker, Complex network dimension and path counts, Technical Report (2007), http://www.geocities.com/oshanker/complexNetworkZetaFunction.pdf.
5. J. J. Dai, J. I. Lathrop, J. H. Lutz and E. Mayordomo, *Theor. Comput. Sci.* **310** (2004) 1–33.
6. D. Doty, X. Gu, J. H. Lutz, E. Mayordomo and P. Moser, Zeta-dimension, in *Proc. 30th Int. Symp. on Mathematical Foundations of Computer Science* (Springer-Verlag, 2005), pp. 283–294.
7. J. H. Lutz, *SIAM Journal on Computing* **32** (2003) 1236–1259.
8. J. H. Lutz, *Information and Computation* **187** (2003) 49–79.
9. J. H. Lutz, *Mathematical Logic Quarterly* **51** (2005) 62–72.
10. L. Staiger, *Information and Computation* **103** (1993) 159–194.
11. L. Staiger, *Theory of Computing Systems* **31** (1998) 215–229.
12. T. M. Apostol, Modular functions and Dirichlet series in number theory, in *Graduate Texts in Mathematics*, 2nd edn. (Springer, 1990), p. 41.
13. G. H. Hardy and M. Riesz, *The General Theory of Dirichlet's Series* (Cambridge University Press, Cambridge, 1915).
14. E. Cahen, *Annales de l'École Normale supérieure* **3** (1894) 75–164.
15. H. M. Edwards, *Riemann's Zeta Function* (Academic Press, 1974).
16. R. Sedgewick, *Algorithms in C*, Part 5 (Addision Wesley, San Francisco, 2002).