RS 445/660 – Network Science – Fall 2015

Handout 7.1 – Algorithm for Generating Directed Power Law Graphs

Ref: Text by Easley & Kleinberg (Chapter 18).

Note: The following algorithm creates a directed graph where indegrees of nodes have a power law distribution. (Each node has an outdegree of 1.)

Input: Integer \( n \) (number of nodes) and a probability value \( p \). (It is assumed that the nodes of the graph are numbered 1, 2, \ldots, n and that there are no edges in the graph initially.)

Output: A directed graph where indegrees of nodes have a power law distribution.

Steps of the algorithm:

1. **Initialize:** Add a directed self-loop around Node 1. (See Item 1 in “Additional Notes” below.)

2. for \( j = 2 \) to \( n \) do
   
   (i) Choose a node \( x \) uniformly at random from \{1, \ldots, j − 1\}.
   
   (ii) Choose Step (a) below with probability \( p \) and Step (b) below with probability \( 1 − p \).
       
       (Each iteration executes exactly one of the steps (a) and (b) below.)
       
       (a) Add the directed edge \((j, x)\) to \( G \).

       (b) Let \( y \) be the node to which \( x \) has a directed edge. Add the directed edge \((j, y)\) to \( G \).

3. Output the resulting directed graph \( G(V, E) \).

Additional Notes:

1. Adding the self loop in Step 1 ensures that Node 1 also has an outdegree of 1.

2. Step 2(ii)(b) above is called the copy step and this is the key to obtaining the power law behavior.

3. An analysis is presented in the Easley/Kleinberg text to argue that (for large enough \( n \)), the fraction of nodes with indegree \( k \) follows an approximate power law \( k^{-c} \), where \( c = 1 + 1/(1 − p) \).

4. As mentioned above, each node in the resulting directed graph has an outdegree of 1. The above algorithm can be generalized so that \( t \geq 2 \) of outgoing edges from node \( j \) get added in each iteration instead of a single edge.