GRAPH STRUCTURE & MODELLING

CASE STUDY 1

Author :	*Paula Dwan*
Email :	*paula.dwan@gmail.com*
Student ID :	*13208660*
Course :	*MSc Advanced Software Engineering*
Module :	*COMP-47270 Computational Network Analysis and Modelling*
Lecturer :	*Dr. Neil Hurley*
Email :	*neil.hurley@ucd.ie*
Due Date :	*20 April 2015*

Table of Contents

1 Case Study Requirements

Determine the qualitative nature of the networks you are studying and write up a report.

Do some of the following:

1. For the directed networks, sketch the **Broder et al** picture of the network.

   Number of nodes in *(SCC)* strongest connected components and *In-* and *Out-* and other sections of the network.

2. Fit a line to a log-log plot of the degree distribution.

   Compute the slope of this line to determine the parameter β of the power-law degree distribution model.

3. Simulate the Price model to generate a network of n nodes for a particular power-law parameter α. [1]

   Compute all of the above parameters for the **Price model.**

   Which (real-life) network/s does the Price Model best represent?

[1] *Note that* $\alpha = \beta - 1$

2 Broder et Al

2.1 *Overview of Broder et Al*

In 2000, Broder and various colleagues conducted a large-scale analyis of the web. They concluded that as a directed graph there was a recongnisable structure in place that of a bow-tie (see following figure.) In short, if a page x is connected to page y via a directed edge then a hyperlink connects page x to page y.

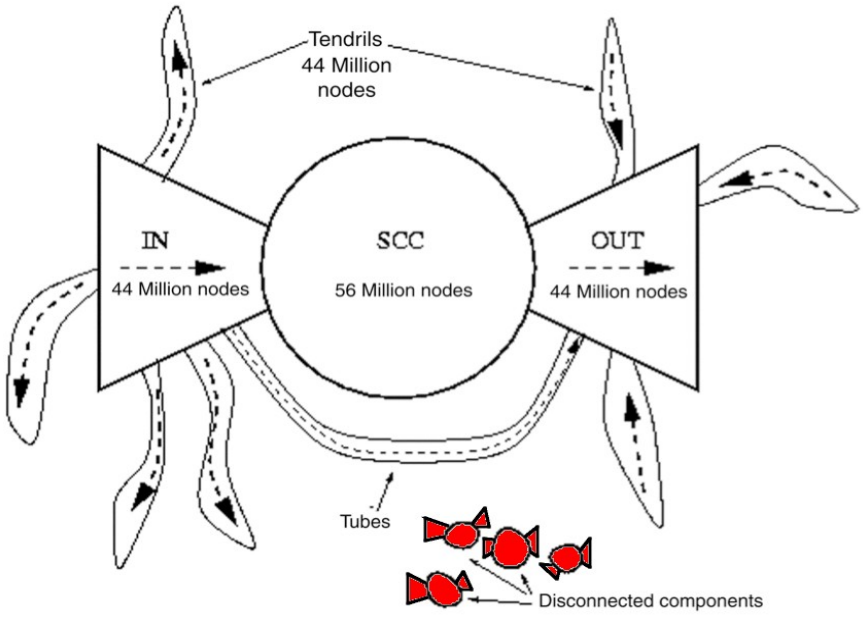


Figure 1 : Bow-tie model of the web graph (Broder et Al, 2000).

2.1.1 Components : Bow-Tie Model

Component	Explanation
IN	
SCC	
OUT	

Component	Explanation
Tubes	
Disconnected component	
Tendrils	

2.1.2 How Calculated Using Python

There is no one function in NetworkX or in python to evaluate a directed graph and confirm what node is part of what component, therefore node must be examined to see what connections exist and in what direction.

Component	Explanation
IN	
SCC	
OUT	
Tubes	
Disconnected component	
Tendrils	

2.1.3 Functions used

2.1.3.1 Function : strongly_connected_components (G)

Information	Generate nodes in strongly connected components of graph
Parameters	**G** : NetworkX Graph An directed graph.
Returns	**comp** : generator of lists A list of nodes for each strongly connected component of G.
Raises	NetworkXNotImplemented: If G is undirected.
Notes	Uses Tarjan's algorithm with Nuutila's modifications. Nonrecursive version of algorithm.
References	Depth-first search and linear graph algorithms, R. Tarjan SIAM Journal of Computing 1(2):146-160, (1972). On finding the strongly connected components in a directed graph. E. Nuutila and E. Soisalon-Soinen Information Processing Letters 49(1): 9-14, (1994)..
Source	https://networkx.github.io/documentation/latest/reference/algorithms.component.html https://networkx.github.io/documentation/latest/reference/generated/networkx.algorithms.components.strongly_connected.strongly_connected_components.html

2.1.3.2 Function : number_strongly_connected_components (G)

Information	Return number of strongly connected components in graph.
Parameters	G : NetworkX graph A directed graph.
Returns	n : integer Number of strongly connected components
Source	https://networkx.github.io/documentation/latest/reference/generated/networkx.algorithms.components.strongly_connected.number_strongly_connected_components.html

2.1.4 Function : IN–degree distributions

Information	
Parameters	
Returns	
Source	

2.1.5 Function : OUT–degree distributions

Information	
Parameters	
Returns	
Source	

3 Plot the Degree Distribution

3.1 Overview

3.1.1 Components : Log-Log Plot

Component	Explanation

3.1.2 How Calculated Using Python

Component	Explanation

3.1.3 Functions Used

3.1.3.1 Function : degree_histogram (G)

Information	Return a list of the frequency of each degree value.
Parameters	G : NetworkX graph A graph.
Returns	**hist** : list A list of frequencies of degrees. The degree values are the index in the list.
Notes	The bins are width one, hence len(list) can be large (Order(number_of_edges))
Source	https://networkx.github.io/documentation/latest/reference/generated/networkx.classes.function.degree_histogram.html

3.2 Calculate Parameter b for Power-Law Degree Distribution Model

4 Price Model

4.1 Overview

Price's model (named after the physicist Derek J. de Solla Price) is a mathematical model for the growth of social networks. It was the first model which generalized the Simon model[1] to be used for networks, especially for growing networks. Price's model belongs to the broader class of network growing models (together with the highly influential Barabási–Albert model) whose primary target is to explain the origination of networks with strongly skewed degree distributions. The model picked up the ideas of the Simon model reflecting the concept of rich get richer, also known as the Matthew effect. Price took the example of a network of citations between scientific papers and expressed its properties. His idea was that the way how an old vertex (existing paper) gets new edges (new citations) should be proportional to the number of existing edges (existing citations) the vertex already has. This was referred to as cumulative advantage, now also known as preferential attachment. Price's work is also significant in providing the first known example of a scale-free network (although it was named later). His ideas were used to describe many real-world networks such as the Web.

Usage :

Although many real-world networks are thought to be scale-free, the evidence often remains inconclusive, primarily due to the developing awareness of more rigorous data analysis techniques.[3] As such, the scale-free nature of many networks is still being debated by the scientific community. A few examples of networks claimed to be scale-free include:

- Social networks, including collaboration networks. Two examples that have been studied extensively are the collaboration of movie actors in films and the co-authorship by mathematicians of papers.
- Many kinds of computer networks, including the internet and the webgraph of the World Wide Web.
- Some financial networks such as interbank payment networks [12][13]
- Protein-protein interaction networks.
- Semantic networks.[14]
- Airline networks.

Scale free topology has been also found in high temperature superconductors.[15] The qualities of a high-temperature superconductor — a compound in which electrons obey the laws of quantum physics, and flow in perfect synchrony, without friction — appear linked to the fractal arrangements of seemingly random oxygen atoms and lattice distortion.[16]

4.1.1 Components : Log-Log Plot

Component	Explanation

4.1.2 How Calculated Using Python

Component	Explanation

4.1.3 Functions Used

5 Conclusions

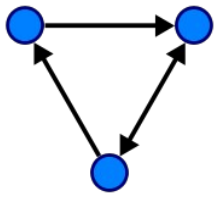
6 REFERENCES

6.1 OVERVIEW

Datasets : http://snap.stanford.edu/data/index.html

6.1.1 NETWORK TYPES

1. **Directed** : directed network (http://en.wikipedia.org/wiki/Directed_graph)



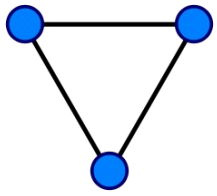
In mathematics, and more specifically in graph theory, a directed graph (or digraph) is a graph, or set of nodes connected by edges, where the edges have a direction associated with them. In formal terms, a digraph is a pair G=(V,A) (sometimes G=(V,E)) of:[1]

- a set V, whose elements are called vertices or nodes,
- a set A of ordered pairs of vertices, called arcs, directed edges, or arrows (and sometimes simply edges with the corresponding set named E instead of A).

It differs from an ordinary or undirected graph, in that the latter is defined in terms of unordered pairs of vertices, which are usually called edges.

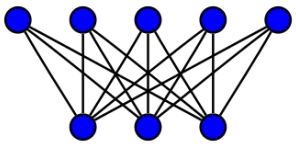
A digraph is called "simple" if it has no loops, and no multiple arcs (arcs with same starting and ending nodes). A directed multigraph, in which the arcs constitute a multiset, rather than a set, of ordered pairs of vertices may have loops (that is, "self-loops" with same starting and ending node) and multiple arcs. Some, but not all, texts allow a digraph, without the qualification simple, to have self loops, multiple arcs, or both.

2. **Undirected** : undirected network (http://en.wikipedia.org/wiki/Graph_(mathematics)#Undirected_graph)



An undirected graph is one in which edges have no orientation. The edge (a, b) is identical to the edge (b, a), i.e., they are not ordered pairs, but sets {u, v} (or 2-multisets) of vertices. The maximum number of edges in an undirected graph without a self-loop is n(n - 1)/2.

3. **Bipartite** : bipartite network (http://en.wikipedia.org/wiki/Bipartite_graph)



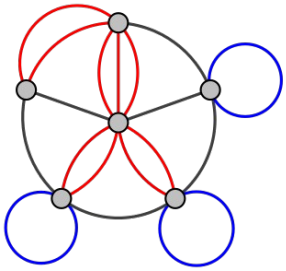
A complete bipartite graph with m = 5 and n = 3

In the mathematical field of graph theory, a bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V (that is, U and V are each independent sets) such that every edge connects a vertex in U to one in V. Vertex set U and V are often denoted as partite sets. Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles.[1][2]

The two sets U and V may be thought of as a coloring of the graph with two colors: if one colors all nodes in U blue, and all nodes in V green, each edge has endpoints of differing colors, as is required in the graph coloring problem.[3][4] In contrast, such a coloring is impossible in the case of a non-bipartite graph, such as a triangle: after one node is colored blue and another green, the third vertex of the triangle is connected to vertices of both colors, preventing it from being assigned either color.

One often writes G=(U,V,E) to denote a bipartite graph whose partition has the parts U and V, with E denoting the edges of the graph. If a bipartite graph is not connected, it may have more than one bipartition;[5] in this case, the (U,V,E) notation is helpful in specifying one particular bipartition that may be of importance in an application. If $|U|=|V|$, that is, if the two subsets have equal cardinality, then G is called a balanced bipartite graph.[3] If all vertices on the same side of the bipartition have the same degree, then G is called biregular.

4. **Multigraph** : network has multiple edges between a pair of nodes (http://en.wikipedia.org/wiki/Multigraph)



A multigraph with multiple edges (red) and several loops (blue).

Not all authors allow multigraphs to have loops.

In mathematics, and more specifically in graph theory, a multigraph is a graph which is permitted to have multiple edges (also called parallel edges[1]), that is, edges that have the same end nodes. Thus two vertices may be connected by more than one edge.

There are two distinct notions of multiple edges:

- Edges without own identity: The identity of an edge is defined solely by the two nodes it connects. In this case, the term "multiple edges" means that the same edge can occur several times between these two nodes.
- Edges with own identity: Edges are primitive entities just like nodes. When multiple edges connect two nodes, these are different edges.

A multigraph is different from a hypergraph, which is a graph in which an edge can connect any number of nodes, not just two.

For some authors, the terms pseudograph and multigraph are synonymous. For others, a pseudograph is a multigraph with loops.

- **Temporal** : for each node/edge we know the time when it appeared in the network

- **Labeled** : network contains labels (weights, attributes) on nodes and/or edges

6.1.2 NETWORK STATISTICS

Dataset statistics	
Nodes	Number of nodes in the network
Edges	Number of edges in the network
Nodes in largest WCC	Number of nodes in the largest weakly connected component

Edges in largest WCC	Number of edges in the largest weakly connected component
Nodes in largest SCC	Number of nodes in the largest strongly connected component
Edges in largest SCC	Number of edges in the largest strongly connected component
Average clustering coefficient	Average clustering coefficient In graph theory, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterised by a relatively high density of ties; this likelihood tends to be greater than the average probability of a tie randomly established between two nodes (Holland and Leinhardt, 1971;[1] Watts and Strogatz, 1998[2]). Two versions of this measure exist: the global and the local. The global version was designed to give an overall indication of the clustering in the network, whereas the local gives an indication of the embeddedness of single nodes.
Number of triangles	Number of triples of connected nodes (considering the network as undirected)
Fraction of closed triangles	Number of connected triples of nodes / number of (undirected) length 2 paths
Diameter (longest shortest path)	Maximum undirected shortest path length (sampled over 1,000 random nodes)
90-percentile effective diameter	90th percentile of undirected shortest path length distribution (sampled over 1,000 random nodes)

6.2 *Social circles: Google+ (Social / Directed)*

6.2.1 Dataset information

http://snap.stanford.edu/data/egonets-Gplus.html

This dataset consists of 'circles' from Google+. Google+ data was collected from users who had manually shared their circles using the 'share circle' feature. The dataset includes node features (profiles), circles, and ego networks. Data is also available from Facebook and Twitter.

Dataset statistics	
Nodes	**107614**
Edges	13673453
Nodes in largest WCC	107614 (1.000)
Edges in largest WCC	13673453 (1.000)
Nodes in largest SCC	69501 (0.646)
Edges in largest SCC	9168660 (0.671)
Average clustering coefficient	0.4901
Number of triangles	1073677742
Fraction of closed triangles	0.6552
Diameter (longest shortest path)	6
90-percentile effective diameter	3

6.2.2 Source (citation)

J. McAuley and J. Leskovec. Learning to Discover Social Circles in Ego Networks. NIPS, 2012.

6.2.3 FILES

File	Description
gplus.tar.gz	Google+ (132 networks)
gplus_combined.txt.gz	Edges from all egonets combined
readme-Ego.txt	Description of files

6.3 SOCIAL CIRCLES: TWITTER (SOCIAL / DIRECTED)

6.3.1 DATASET INFORMATION

http://snap.stanford.edu/data/egonets-Twitter.html

This dataset consists of 'circles' (or 'lists') from Twitter. Twitter data was crawled from public sources. The dataset includes node features (profiles), circles, and ego networks. Data is also available from Facebook and Google+.

Dataset statistics	
Nodes	81306
Edges	1768149
Nodes in largest WCC	81306 (1.000)
Edges in largest WCC	1768149 (1.000)
Nodes in largest SCC	68413 (0.841)
Edges in largest SCC	1685163 (0.953)
Average clustering coefficient	0.5653
Number of triangles	13082506
Fraction of closed triangles	0.06415
Diameter (longest shortest path)	7
90-percentile effective diameter	4.5

6.3.2 SOURCE (CITATION)

J. McAuley and J. Leskovec. Learning to Discover Social Circles in Ego Networks. NIPS, 2012.

6.3.3 FILES

File	Description
twitter.tar.gz	Twitter data (973 networks)
twitter_combined.txt.gz	Edges from all egonets combined
readme-Ego.txt	Description of files

6.4 CITING SNAP

We encourage you to cite our datasets if you have used them in your work.

You can use the following BibTeX citation:

```
@misc{snapnets,
  author       = {Jure Leskovec and Andrej Krevl},
  title        = {{SNAP Datasets}: {Stanford} Large Network Dataset Collection},
  howpublished = {\url{http://snap.stanford.edu/data}},
  month        = jun,
  year         = 2014
}
```