Extracted from:

Complex Network Analysis in Python

Recognize → Construct → Visualize → Analyze → Interpret

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Complex Network Analysis in Python

Recognize → Construct → Visualize → Analyze → Interpret

Dmitry Zinoviev
edited by Adaobi Obi Tulton

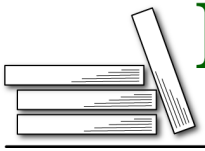
Complex Network Analysis in Python

Recognize → Construct → Visualize → Analyze → Interpret

Dmitry Zinoviev

The Pragmatic Bookshelf

Raleigh, North Carolina



Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Managing Editor: Brian MacDonald
Supervising Editor: Jacquelyn Carter
Development Editor: Adaobi Obi Tulton
Indexing: Potomac Indexing, LLC
Copy Editor: Nicole Abramowitz
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

Printed in the United States of America.
ISBN-13: 978-1-68050-269-5
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—January 2018

To my beautiful and most intelligent wife,
Anna, and to our children: graceful ballerina,
Eugenia, and romantic gamer, Roman.

Know Thy Networks

In general, a network is yet another—relational—form of organization and representation of discrete data. (The other one being tabular, with the data organized in rows and columns.) Two important network concepts are entities and the relationships between them. Depending on a researcher's background, entities are known as nodes (the term we'll use in this book), actors, or vertices. Relationships are known as edges (preferred in this book), links, arcs, or connections. We will casually refer to networks as "graphs" (in the graph-theoretical meaning of the word), even though graphs are not the only way to describe networks.

Graphs and Graphs

When it comes to mathematics, the word "graph" has at least two different meanings. In algebra and calculus, a graph of a function is a continuous line chart or surface plot. In graph theory, a graph is a set of discrete objects (vertices, depicted diagrammatically as dots), possibly joined by edges (depicted as lines or arcs). We will always use the latter definition unless explicitly stated.

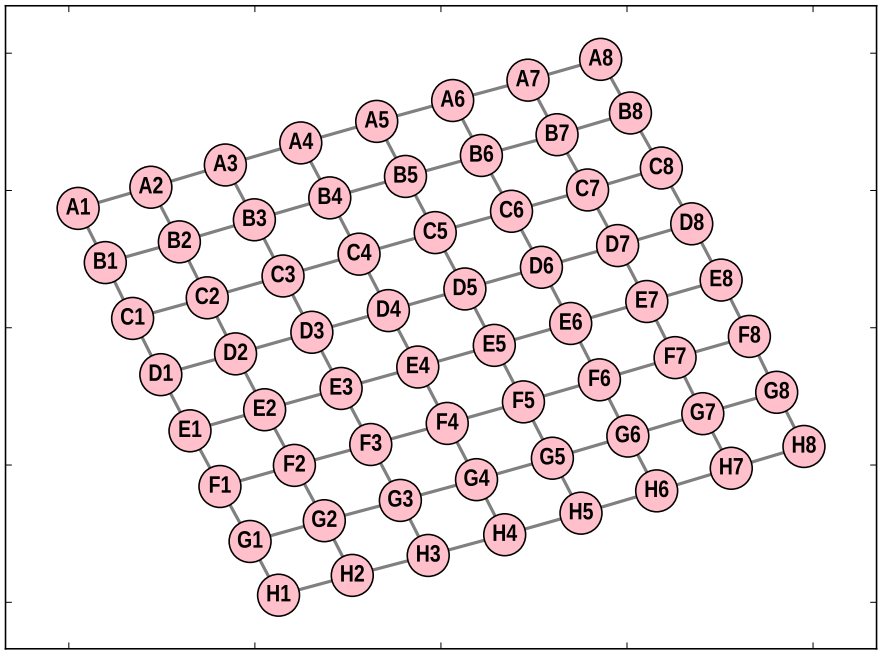
Network nodes and edges are high-level abstractions. For many types of network analysis, their true nature is not essential. (When it is, we decorate nodes and edges by adding properties, also known as attributes.) What matters is the discreteness of the entities and the binarity of the relationships. A discrete entity must be separable from all other entities—otherwise, it is not clear how to represent it as a node. A relationship typically involves two discrete entities; in other words, any two entities either are in a relationship or not. (An entity can be in a relationship with itself. Such a relationship is called *reflexive*.) It is not directly possible to use networks to model relationships that involve more than two entities, but if such modeling is really necessary, then you can use hypergraphs, which are beyond the scope of this book.

Once all of the above conditions are met, you can graphically represent and visualize a node as a point or circle and an edge as a line or arc segment. You can further express node and edge attributes by adding line thickness, color, different shapes and sizes, and the like.

Let's have a look at some really basic—so-called "classic"—networks.

In a checkerboard, each field is an entity (node) with three attributes: "color" ("black" or white"), "column" ("A" through "H"), and "row" (1 through 8). "Being next to" is the relationship between two entities. There is an edge connecting two nodes if the nodes "are next to" each other. As a matter

of fact, "being next to" is one of the foundational relationships that leads to spatial networks. You can see a "checkerboard" network, also known as a mesh or grid, in the following figure.

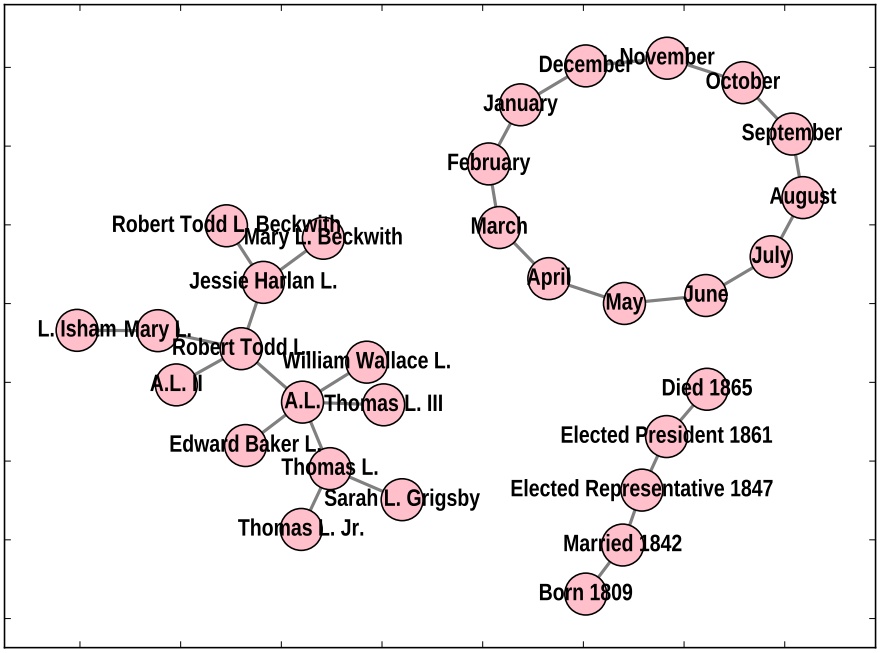


In a timeline of our life, each life event (such as "birth," "high school graduation," "marriage," and eventually "death") is an entity with at least one attribute: "time." "Happening immediately after" is the relationship: an edge connects two events if one event occurs immediately after the other, leading to a network of events. Unlike "being next to," "happening immediately after" is not symmetric: if A happened immediately after B (there is an edge from A to B), then B did not happen after A (there is no reverse edge).

In a family tree, each person in the tree is an entity, and the relationship could be either being "a descendant of" or "an ancestor of" (asymmetric). A family tree network is neither spatial nor strictly temporal: the nodes are not intrinsically arranged in space or time.

In a hierarchical system that consists of parts, sub-parts, and sub-sub-parts (such as this book), a part at any level of the hierarchy is an entity. The relationship between the entities is "a part of": a paragraph is "a part of" a subsection, which is "a part of" a section, which is "a part of" a chapter, which is "a part of" a book.

All the networks listed previously are simple because they have a regular or almost regular structure. A checkerboard is a rectangular grid. A timeline is a linear network. A family tree is a tree, and such is a network of a hierarchical system (a special case of a tree with just one level of branches is called a star). The following figure shows more simple networks: a linear timeline of Abraham Lincoln (A.L.), his family tree, and a ring of months in a year. (A ring is another simple network, which is essentially a linear network, wrapped around.)



Make no mistake: a simple network is simple not because it is small, but because it is regular. For example, any ring node always has two neighbors; any tree node (except for the root) has exactly one antecedent; any inner grid node has exactly four neighbors, two of which are in the same row and the other two in the same column. The complete world timeline has billions of events. The humankind "family tree" has billions of individuals. We still consider these networks simple.

What is a complex network, then?

A complex network has a non-trivial structure. It is not a grid, not a tree, not a ring—but it is not entirely random, either. Complex networks emerge in nature and the man-made world as a result of decentralized processes with no global control. One of the most common mechanisms is the preferential attachment (*Emergence of Scaling in Random Networks [BA99]*), whereby nodes with more

edges get even more edges, forming gigantic hubs in the core, surrounded by the poorly connected periphery. Another evolutionary mechanism is transitive closure, which connects two nodes together if they are already connected to a common neighbor, leading to densely interconnected network neighborhoods.

Let's glance at some complex networks. The following table shows the major classes of complex networks and some representatives from each class.

Technological networks	Communication systems; transportation; the Internet; electric grid; water mains
Biological/ecological networks	Food webs; gene/protein interactions; neural system; disease epidemics
Economic networks	Financial transactions; corporate partnerships; international trade; market basket analysis
Social networks	Families and friends; email/SMS exchanges; professional groups
Cultural networks	Language families; semantic networks; literature, art, history, religion networks (emerging fields)

The networks in the table pertain to diverse physical, social, and informational aspects of human life. They consist of various nodes and edges, some material and some purely abstract. However, all of them have common properties and behaviors that can be found in complex networks and only in complex networks, such as community structure, evolution by preferential attachment, and power law degree distribution.

Enter Complex Network Analysis

Complex network analysis (CNA), which is the study of complex networks—their structure, properties, and dynamics—is a relatively new discipline, but with a rich history.

You can think of CNA as a generalization of social network analysis (SNA) to include non-social networks.

Social networks—descriptors of social structures through interactions—have been known as "social groups" since the late 1890s. Their systematic exploration began in the 1930s. In 1934, J.L. Moreno (*Who Shall Survive? [Mor34]*) developed sociograms—graph drawings of social networks. Eventually, sociograms became the de facto standard of complex network visualization.

John Barnes coined the term "SNA" in 1954 (*Class and Committees in a Norwegian Island Parish [Bar54]*). Around the same time, rapid penetration

of mathematical methods into social sciences began, leading to the emergence of SNA as one of the leading paradigms in contemporary sociology.

Social network analysis addresses social networks at three levels: microscopic, mesoscopic, and macroscopic. At the microscopic level, we view a network as an assembly of individual nodes, dyads (pairs of connected nodes; essentially, edges), triads (triples of nodes, connected in a triangular way), and subsets (tightly knit groups of nodes). A mesoscopic view focuses on exponential random graph models (ERGMs), scale-free and small-world networks, and network evolution. Finally, at the macroscopic level, the more general complex network analysis fully absorbs SNA, abstracting from the social origins of social networks and concentrating on the properties of very large real-world graphs, such as degree distribution, assortativity, and hierarchical structure (*Exploring Complex Networks [Str01]*). You will see the definitions and explanations of some of these properties and the Python ways of calculating them later in the book.

But first, let's get your hands dirty (possibly physically dirty) and sketch a real complex network on a sheet of paper.