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

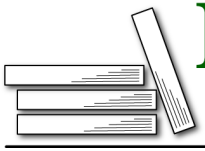
Complex Network Analysis in Python

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.

Thou wilt set forth at once because the journey is far and lasts for many hours; but the hours on the velvet spaces are the hours of the gods, and we may not say what time such an hour may be if reckoned in mortal years.

➤ *Lord Dunsany, Anglo-Irish writer and dramatist*

Preface

In science, technology, and mathematics, a network is a system of interconnected objects. Complex network analysis (CNA) is a discipline of exploring quantitative relationships in the networks with non-trivial, irregular structure. The actual nature of the networks (social, semantic, transportation, communication, economic, and the like) doesn't matter, as long as their organization doesn't reveal any specific patterns. This book was inspired by a decade of CNA practice and research.

Being a professor of mathematics and computer science at Suffolk University in Boston, I have experimented with complex networks of various sizes, purposes, and origins. I developed my first CNA software in an ad hoc manner in the C language—the language venerable yet ill-suited for CNA projects. The price of explicit memory management, cumbersome file input/output, and lack of advanced built-in data structures (such as maps and lists) was simply too high to justify a further commitment to C. At the moment I realized that there were affordable alternatives to C that did not require low-level programming (such as *Pajek [NMB11]* and Mathematica[1]), off I went.

Both systems that I mentioned had significant restrictions. Mathematica was proprietary (and, frankly, quite costly). My inner open source advocate demanded that I cease and desist using it, especially given that earlier versions of Mathematica didn't provide dedicated CNA support and failed to handle big networks. Pajek was proprietary, too, and not programmable. It took a joint effort of my inner open source advocate and inner programmer to push it to the periphery. (I still occasionally use Pajek, and I believe it's a great system for solving non-recurring problems.)

I felt delighted when, in search of open source, free, scalable, reliable, and programmable CNA software, I ran into `NetworkX`, a Python library still in its infancy. For the next several years, it became my tool of choice when it came to CNA simulation, analysis, or visualization.

1. www.wolfram.com/mathematica

About the Reader

This book is intended for graduate and undergraduate students, complex data analysis (CNA) or social network analysis (SNA) instructors, and CNA/SNA researchers and practitioners. The book assumes that you have some background in computer programming—namely, in Python programming. It expects from you no more than common sense knowledge of complex networks. The intention is to build up your CNA programming skills and at the same time educate you about the elements of CNA itself. If you're an experienced Python programmer, you can devote more attention to the CNA techniques. On the contrary, if you're a network analyst with less than an excellent background in Python programming, your plan should be to move slowly through the dark woods of data frames and list comprehensions and use your CNA intuition to grasp programming concepts.

About the Book

This book covers construction, exploration, analysis, and visualization of complex networks using NetworkX (a Python library), as well as several other Python modules, and Gephi, an interactive environment for network analysts. The book is not an introduction to Python. I assume that you already know the language, at least at the level of a freshman programming course.

The book consists of five parts, each covering specific aspects of complex networks. Each part comes with one or more detailed case studies.

Part I presents an overview of the main Python CNA modules: NetworkX, iGraph, graph-tool, and networkit. It then goes over the construction of very simple networks both programmatically (using NetworkX) and interactively (in Gephi), and it concludes by presenting a network of Wikipedia pages related to complex networks.

In Part II, you'll look into networks based on explicit relationships (such as social networks and communication networks). This part addresses advanced network construction and measurement techniques. The capstone case study —a network of "Panama papers"—illustrates possible money-laundering patterns in Central Asia.

Networks based on spatial and temporal co-occurrences—such as semantic and product networks—are the subject of Part III. The third part also explores macroscopic and mesoscopic complex network structure. It paves the way to network-based cultural domain analysis and a marketing study of Sephora cosmetic products.

If you cannot find any direct or indirect relationships between the items, but still would like to build a network of them, the contents of Part IV come to the rescue. You will learn how to find out if items are similar, and you will convert quantitative similarities into network edges. A network of psychological trauma types is one of the outcomes of the fourth part.

The book concludes with Part V: directed networks with plenty of examples, including a network of qualitative adjectives that you could use in computer games or fiction.

When you finish your journey, you'll be able to identify, sketch (both by hand, in Gephi, and programmatically), transform, analyze, and visualize several types of complex networks. You'll be able to interpret network measures and structure. The book doesn't aim to be a comprehensive CNA reference. Many discipline-specific aspects, such as triadic census, exponential random graph models (ERGMs), and network flows, as well as the whole story of network dynamics (evolution and contagion), have been intentionally left uncharted. The bibliography on page ? will take you to more destinations of your choice, whether they be economic networks, web scrapping, or classical social network analysis.

About the Software

This book uses Python 3.x and networkx 1.11. All Python examples in this book are known to work for the modules mentioned in the following table. All of these modules are included in the Anaconda distribution, with the exception of community,[2] toposort,[3] wikipedia,[4] and generalized,[5] which must be installed separately. Anaconda is provided by Continuum Analytics and is available for free.[6]

Package	Used version	Package	Used version
python	3.4.5	networkx	1.11
matplotlib	1.5.1	community	0.9
nltk	3.2.2	numpy	1.11.3
pandas	0.19.2	pygraphviz	1.3.1
wikipedia	1.4	scipy	0.18.1
toposort	1.5		

2. pypi.python.org/pypi/python-louvain
3. pypi.python.org/pypi/toposort
4. pypi.python.org/pypi/wikipedia
5. pragprog.com/titles/dzcnapy/source_code
6. www.continuum.io

The easiest way to install the missing modules is by running pip on your operating system shell command line.

```
⇛ pip install toposort
⇛ pip install wikipedia
⇛ pip install python-louvain
⇛ pip install pygraphviz
```

If you want to use module pygraphviz to layout networks, you first need to install Graphviz (including the developers add-on graphviz-dev).[7]

In September 2017, a new version of NetworkX was released, NetworkX 2.0. Appendix 2, *NetworkX 2.0*, on page ? provides useful information about converting your CNA scripts to the new version.

About the Notation

The following covers the specific notation used in this book.

Program Output

The book uses a left-pointed gray arrow in the left margin of a page to indicate program outputs. In the following scenario, print(1 + 2) is a Python statement, and 3 is the visual output of the statement.

```
print(1 + 2)
```

```
❮ 3
```

"This Chapter Uses X"

This chapter uses X

"This chapter/section uses X" informs you that the material in the chapter or section goes beyond the core Python and NetworkX. If you're unfamiliar with X, you'll probably understand the content but may experience difficulties with comprehending the included code snippets. You're advised to refresh your knowledge of the listed modules.

Directed Edges

NetworkX uses module Matplotlib for network visualization. You would expect directed edges to have an arrow at the head end, and Matplotlib fully supports arrows. However, NetworkX draws thick rectangular stubs instead. This is just something you'll have to get used to. If you need a publication-quality network image with arrows, consider using Gephi.

7. www.graphviz.org/

Online Resources

This book has its own web page[8] where you can find all the code for this book. There you'll also find the community forum, where you can ask questions, post comments, and submit errata.

Two other great community-operated resources for questions and answers are the Stack Overflow forum[9] and NetworkX Google discussion group.[10]

Now, let's get started!

Dmitry Zinoviev
dzinoviev@gmail.com
January 2018

8. pragprog.com/book/dzcnapy
9. stackoverflow.com/questions/tagged/networkx
10. groups.google.com/forum/#!forum/networkx-discuss