

Book proposal

To prepare your proposal, please:

1. Make a copy of this document.
2. Rename your copy as *Firstname Lastname — Working Title — Book Proposal*.
3. Craft your proposal, noting all prompts and guidance below.
4. Set document permissions to “anyone with the link can comment.”
5. Submit your proposal by emailing us the link: workwithus@oreilly.com.

Proposed book title

Hands-on Network Machine Learning with Scikit-Learn and Graspologic

Subtitle

Concepts, Tools, and Techniques to explore and exploit network data

Author(s)

Eric Bridgeford

Alex Loftus

Joshua Vogelstein

Section Contributors

Jesus Arroyo

Ali Saad-Eldin

Sambit Panda

Advisors

Carey E. Priebe

Christopher White

Weiwei Yang

Author's professional title(s) and affiliation(s):

Eric Bridgeford - PhD Student, Dept of Biostatistics, Johns Hopkins University

Alex Loftus - Master's Student, Dept of Biomedical Engineering, Johns Hopkins University

Joshua Vogelstein - Assistant Professor, Dept of Biomedical Engineering, Johns Hopkins University

Jesus Arroyo - Assistant Professor, Dept of Statistics, Texas A&M University

Ali Saad-Eldin - Software Engineer, Amazon

Sambit Panda - PhD Student, Dept of Biomedical Engineering, Johns Hopkins University

Carey E. Priebe - Professor of Applied Mathematics and Statistics, Johns Hopkins University

Christopher White - Managing Director, Microsoft Research

Weiwei Yang - Principal Development Manager, Microsoft Research

Mailing address(es):

Joshua T. Vogelstein, Dept of Biomedical Engineering, 3400 N. Charles St, Baltimore, MD 21218

Phone number(s) Preferred email address(es)

1. ebridge2@jhu.edu
2. aloftus2@jhu.edu
3. jovo@jhu.edu
4. jarroyo@tamu.edu
5. asaadel1@jhu.edu
6. spanda3@jhu.edu
7. cep@jhu.edu
8. chwh@microsoft.com
9. weiwya@microsoft.com

About the author (s)

Author biography (This bio will appear on Amazon. Please write in the third-person and be sure the biography answers the question "Why are you the best person to write this book?"):

Eric Bridgeford is a PhD student in the Department of Biostatistics at Johns Hopkins University. Eric's background includes Computer Science and Biomedical Engineering, and he is an avid contributor of packages to CRAN and PyPi for nonparametric hypothesis testing. Eric studies general approaches for statistical inference in network data, with applications to problems with network estimation in MRI connectomics data, including replicability and batch effects.

Alex Loftus is a master's student at Johns Hopkins University in the Department of Biomedical Engineering, with an undergraduate degree in neuroscience. He has worked on implementing network spectral embedding and clustering algorithms in Python, and helped develop an MRI pipeline to produce brain networks from diffusion MRI data.

Dr. Joshua T. Vogelstein is an Assistant Professor in the Department of Biomedical Engineering at Johns Hopkins University, with joint appointments in Applied Mathematics and Statistics, Computer Science, Electrical and Computer Engineering, Neuroscience, and Biostatistics. His research focuses on the statistics of networks in brain science (connectomes). His lab and collaborators have developed the leading computational algorithms and libraries to perform statistical analysis on networks.

Dr. Carey E. Priebe is Professor of Applied Mathematics and Statistics, and a founding member of the Center for Imaging Science (CIS) and the Mathematical Institute for Data Science (MINDS) at Johns Hopkins University. He is a leading researcher in theoretical, methodological, and applied statistics / machine learning; much of his recent work focuses on spectral network analysis and subsequent statistical inference. Professor Priebe is Senior Member of the IEEE, Elected Member of the International Statistical Institute, Fellow of the Institute of Mathematical Statistics, and Fellow of the American Statistical Association.

Chris White is Managing Director, Microsoft Research Special Projects. He leads mission-oriented research and software development teams focusing on high risk problems. Prior to joining Microsoft, he was a Fellow at Harvard for network statistics and machine learning. Chris's work has been featured in media outlets including Popular Science, CBS's 60 Minutes, CNN, the Wall Street Journal, Rolling Stone Magazine, TEDx, and Google's Solve for X. Chris was profiled in a cover feature for the Sept/Oct 2016 issue of Popular Science.

Weiwei Yang is a Principal Development Manager at Microsoft Research. Her interests are in resource efficient alt-SGD ML methods inspired by biological learning. The applied research group she leads aims to democratize AI by addressing issues of sustainability, robustness, scalability, and efficiency in ML. Her group has applied ML to address social issues such as countering human trafficking and to energy grid stabilizations.

LinkedIn profile

1. <https://www.linkedin.com/in/ericbridgeford>
2. <https://www.linkedin.com/in/jovo1/>
3. <https://www.linkedin.com/in/alex-loftus/>
4. <https://www.linkedin.com/in/ali-saad-eldin-09250317b>
5. <https://www.linkedin.com/in/sampan501>
6. <https://www.linkedin.com/in/cmileswhite>
7. <https://www.linkedin.com/in/weiwei-yang-32717528>

Twitter handle

1. @neuro_data

Author public speaking samples (YouTube, etc.)

1. [Unsupervised Learning with Network-Valued Data](#)
2. [Lifelong Learning Talk \(2/7/2020\)](#)
3. [Learning Salon Podcast- Website](#)
4. [Learning Salon YouTube channel](#)
5. [SBE Youtube Graph Theory Lecture](#)
6. [Slides for \(almost\) all talks given](#)

Marketing description

Please summarize the book in 1–2 paragraphs. How would you pitch it to a potential reader on the back cover? What makes your book unique in the marketplace?

This book covers the fundamentals of network machine learning, focusing on developing intuition on networks as statistical objects, doing so while paired with relevant Python tutorials. Concrete examples will develop both the fundamental intuition of working with networks, and the hands-on experience necessary to translate understanding into real-world applications.

We will show how to solve a network machine learning problem from start to finish (understanding the problem, visualizing the data, working with the data, and evaluating results). Readers with no prior knowledge of graph theory or network statistics will be able to utilize efficient and easy to use tools available for performing analyses on network data. Additionally, readers will be introduced to a range of techniques, such as representations, models, and algorithms for networks, in an accessible way.

About the topic

Briefly explain the topic and why it's important.

The use of network representations has experienced a boom in recent years with the rise of the internet and big data. Networks can be found everywhere, from social and spatial networks to biological networks of the brain. In general, networks represent a group of objects and the relationships between those objects. The diverse application of networks in many fields make them powerful statistical objects. The analysis of these networks has recently been made more accessible through the use of several available automated programming packages. We will use several of these packages to perform analysis, including networkX, graspologic, and scikit-learn.

Audience

In 1-2 paragraphs, explain who the primary audience is for your book. What professional positions do they hold, and what positions are they considering next in their careers? What knowledge do you assume they already have, and what skills do you assume they've mastered? What books do you assume they've read?

This book is intended to be a reference textbook for undergraduate and graduate level network machine learning classes. It can also be used by students and professionals (machine learning engineers, data scientists, etc.) interested in an introduction to the field of network machine learning. Readers should have a basic knowledge of mathematics, as well as some exposure to more advanced topics such as calculus, linear algebra, and statistics. Readers are also expected to have some background in programming, preferably in Python, as this book will contain Python tutorials. While many of our readers may have prior exposure to networks as an analytical object (e.g., using network summary statistics), we do not expect our readers to have any background using networks as a statistical object (e.g., dealing directly with network-valued random variables).

Estimate how many people will use the technology you're writing about. Please share any applicable statistics (e.g., Google Trends, analyst reports, blogs, leading companies adopting the topic of your book) indicating market use or market potential.

Machine learning and statistics represents an [estimated audience of nearly 2.720.000 professionals at present](#), with that number projected to grow at a constant-or-greater rate as machine learning continues to grow in popularity. Network statistics is a growing subfield of machine learning: NetworkX, for instance, a popular network analysis package, currently has [8.6k github stars](#) as well as over [30,000 average downloads per week](#) on combined OSX and Windows operating systems. Our book will apply readily to the majority of these individuals and users.

Please provide some scenarios that indicate how the audience will use your book. For example, will readers refer to it daily as a reference, or will they read it once to learn the concepts and then refer to it occasionally?

We expect this book to be the text of reference for network analysis in Python, analogous in the field of network machine learning to “Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow” in the area of Machine and Deep Learning (currently the #6 Amazon best-seller in “Computer Science”). The book will be useful as both a foundational introduction to the concepts of statistical inference for network-valued data and a day-to-day reference.

Further, new developments in computer science, such as graph neural networks, bridge the utility of network-valued data with the problem-solving applicability of neural networks. These techniques have been used to study complicated problems ranging from protein folding and drug discovery all the way to image segmentation. While this book does not plan to provide extensive and broad coverage of graph neural networks, we intend to provide lengthy intuition-focused discussions and series of code examples over the course of two sections which bridge the gap between traditional spectral embedding techniques for network-valued data and the intuition required to study graph neural networks. Existing resources on graph neural networks are extremely academic in nature, and we believe our book will provide value facilitating readers to become accustomed to the basic, high-level intuition they need to use these external tools.

What the reader will learn-and how to apply it

By the end of this book, readers will gain an intuitive understanding of the statistics and analysis of network data, and will have learned the tools and methodology needed to properly apply what they've

learned to the real data they may encounter, whether that be in an academic or professional setting. By introducing and detailing specific examples, the reader will understand how to frame and work through a network machine learning problem. Additionally, the reader will gain an understanding of the statistical properties of networks, as well as how to apply them with python packages through the use of algorithms, simulations, and models. The combination of mathematical description and hands-on application will build skills necessary to analyze, visualize, and understand complex networks.

Keywords

What are the trending terms for the topics covered in your book?

Machine learning, networks, graph embedding, graph clustering, representation learning

Other book features

Will there be a GitHub repository for code samples?

All of the code for the book will be included in a GitHub repository with Jupyter notebooks for each section. The notebooks will also be available to run via Binder or Google Colab, or will simply be rendered for viewing as static html. The in-progress rendering of the textbook, approximately 90% complete, can be found here:

<http://docs.neurodata.io/graph-stats-book/coverpage.html>

Software dependencies

What software updates or releases could potentially impact the release of your book?

Examples shown in the book will largely depend on numpy, scipy, scikit-learn, and graspologic. Numpy and scipy are Python packages for numerical programming, and scikit-learn is a package providing a high-level interface to common machine learning algorithms. Graspologic is an open-source Python package which provides utilities and algorithms for doing statistical analyses on network-valued data. The current stable release, v.0.1.0, will be sufficient to recreate all examples that will be included in the book. The authors of the book are major stakeholders of graspologic; thus it can be ensured that no updates or releases of it will negatively impact the release of the book.

Graspologic has the following direct required dependencies:

- numpy (v.1.8.1 or above)
- scipy (v.1.4.0 or above)
- scikit-learn (v.0.19.1 or above)
- networkx (v.2.1 or above)

- matplotlib (v.3.0.0 or above)
- seaborn (v.0.9.0 or above)

Some of the aforementioned packages might end up being used by our examples directly in addition to via graspologic. All the python packages are actively maintained, version controlled, open source, part of the common machine learning stack, and have a development philosophy of being backward compatible whenever possible. Hence, to the best of our knowledge, no software updates or releases will significantly impact the release of our book.

The only other package that the release of our book depends on is Jupyter Notebook, as well as its extension - Jupyter book - which we plan to utilize for typesetting purposes. The current release is sufficient for our purposes, and thus future releases should not impact the release of this book.

Competing titles

Please list three books from other publishers that will compete with your book. Include the title, author, ISBN, publisher name, and year of publication, and explain how your book will be different or better in timing, content, coverage, approach, or tone than each competing title.

In our exploration of competing titles, it is extremely important to highlight that there does not appear to be a single book which focuses on developing intuition for networks which have the following conditions:

1. Hands On: Includes coded, working examples
2. Network Machine Learning: Explores network machine learning approaches - specifically, this means that the book talks about quantities that it is estimating for the purposes of generalization to unseen edges, nodes, and networks, rather than merely computing properties of the network.
3. Python: Explains concepts using a Python and popular network analysis packages
4. Multiple Networks: Network analysis for single network and multiple-network problems

This combination will prove groundbreaking not only in developing the statistical intuition necessary for working with networks, but also will provide hands-on experience with regards to how to apply knowledge gained from the book to practical real-world examples. This uniquely positions our book as a potential landmark in network machine learning as data scientists become more familiar with the advantages afforded by working with network-valued data. Below, we explore competing titles, and where possible, indicate the desiderata that are lacking from the treatment described by the title in a table.

1. Al-Taie, Mohammed Zuhair and Seifedine Kadry. Python for Graph and Network Analysis. Springer International Publishing, 2017, doi:10.1007/978-3-319-53004-8.

2. Deo, Narsingh. "Graph Theory with Applications to Engineering and Computer Science." Dover Publications, 1974.

3. Brandes, Ulrik and Thomas Erlebach. Network Analysis. Springer, 2005, doi:10.1007/b106453.

4. Luke, Douglas. A User's Guide to Network Analysis in R. Springer, 2015, doi:10.1007/978-3-319-23883-8.

5. Kolaczyk, Eric D. Statistical Analysis of Network Data. Springer-Verlag, 2009, doi:10.1007/978-0-387-88146-1.

Book Title	Hands-on	Network Machine learning	Python	Multiple networks
Proposed Book	✓	✓	✓	✓
Python for Graph and Network Analysis	✗	✓	✓	✗
Graph Theory with Applications to Engineering and Computer Science	✗	✗	✗	✗
Network Analysis	✗	✗	✗	✗
A User's Guide to Network Analysis in R	✓	✓	✗	✗
Statistical Analysis of Network Data	✗	✓	✗	✗

Related O'Reilly titles

Which O'Reilly book(s) cover similar topics or related technology? Please list title and author(s).

1. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. O'Reilly Media, 2019, www.oreilly.com/library/view/hands-on-machine-learning/9781492032632.

2. Needham, Mark and Amy E. Hodler. Graph Algorithms. O'Reilly Media, 2019, www.oreilly.com/library/view/graph-algorithms/9781492047674.

3. Lanum, Corey L. Visualizing Graph Data. Manning Publications, 2016, www.oreilly.com/library/view/visualizing-graph-data/9781617293078.

Book outline

Include a detailed outline for the project here, following the example format shown. (If proposing a second edition, please highlight changes from the previous edition.)

Section 1: Foundations

- Chapter 1 - The Network Machine Learning Landscape
 - What Is A Network?
 - Definition of a Network
 - Adjacency Matrix/network equivalence
 - Why Study Networks?
 - Examples Of Applications
 - Types Of Networks
 - Directed / Undirected
 - Weighted / binary
 - Hollow / self-loops
 - Single-network vs. Multi-network
 - Matched vs. unmatched
 - Types of Network Learning Problems
 - Summarize network data
 - Identify edge/node/community/network of interest
 - Find correspondence between nodes/communities/networks
 - Predict missing/errorful edges
 - Main Challenges Of Network Learning
 - High Dimensionality
 - Non-Euclidean-ness
 - Various representations
 - Network isomorphism / $n!$ Permutations
 - Probability Notational Overview
 - Exercises
- Chapter 2 - End-to-end Biology Network Machine Learning Project
 - Look at the big picture
 - Frame the Problem
 - Select a Performance Measure
 - Check the Assumptions
 - Notation
 - Get the Data
 - Create the Workspace
 - Download the Data
 - Take a Quick Look at the Data Structure
 - 1. netspec
 - Discover and Visualize the Data to Gain Insights
 - Visualizing Network Data
 - Looking at Simple network statistics.
 - 1. Number of nodes
 - 2. Number of edges, sum of edge weights, etc.

- 3. In and out degree distributions
- Prepare the Data for Network Algorithms
 - Data Cleaning
 - 1. Getting rid of isolated nodes, pendants, pizza nodes
 - 2. Cutting edges/nodes by weight/degree
 - Handling Text and Categorical Attributes
 - 1. Node and edge metadata
 - Transformation techniques
 - 1. Diagonal augmentation
 - 2. Pass-to-ranks
 - 3. Graph cuts
 - 4. Edge clipping
 - Transformation Pipelines
- Select and Train a Model
 - Why we don't have training and evaluation set
 - Evaluating estimates without cross-validation
- Fine-Tune Your Model
 - Grid Search
 - Randomized Search
 - Analyze the Best Models
- Chapter 3 - End-to-end Business Network Machine Learning Project
 - Look at the big picture
 - Frame the Problem
 - Select a Performance Measure
 - Check the Assumptions
 - Get the Data
 - Create the Workspace
 - Download the Data
 - Take a Quick Look at the Data Structure
 - Discover and Visualize the Data to Gain Insights
 - Visualizing Network Data
 - Looking at Simple network/network statistics
 - Prepare the Data for Network Algorithms
 - Data Cleaning
 - Handling Text and Categorical Attributes
 - Custom Transformers

Section 2: Representations

- Chapter 4 - Properties of Networks As Statistical Objects
 - Matrix Representations
 - Adjacency Matrix
 - 1. Augmenting the Adjacency Matrix
 - Laplacian Matrix
 - 1. Variations on the Laplacian
 - Representations of Networks (including ERGM, spectral, and GNN)
 - Network-wise representations

- Community-wise representations
 - Node-wise representations
 - Edge-wise representations
- Properties of Networks
- The Nonidentifiability Problem
- Regularization
 - Degree trimming
 - De-sparsifying
- Chapter 5 - Why Use Statistical Models?
 - Why use statistical models?
 - Network Models
 - Erdos-Renyi model
 - Stochastic Block Model
 - Structured Independent Edge Model
 - Random Dot Product Graph
 - Generalized Random Dot Product Graph
 - Degree corrected models
 - Inhomogeneous Erdos Renyi
 - Multi-Network models
 - Joint random dot product graphs
 - Common Subspace Independent-Edge Model
 - Correlated Heterogeneous Model
- Chapter 6 - Learning Network Representations
 - Why embed networks?
 - Low-dimensional representations of networks
 - ML methods for Euclidean data
 - The Nonidentifiability Problem
 - Spectral Embedding Methods
 - Adjacency spectral embedding
 - Laplacian spectral embedding
 - Estimating Parameters in Network Models
 - Random walk/diffusion based methods
 - Node2vec
 - Graph Neural Networks
 - GCN
 - Multiple-Network representation learning
 - Omnibus embedding
 - Multiple adjacency spectral embedding
 - Joint Representation Learning
 - CASC
 - Omni (build a $n \times n$ feature matrix, and jointly embed)
- Chapter 7 - Theoretical Results
 - Theory for Single Network Models
 - Theory for Multiple-network Models
 - RDPG/COSIE
 - Timeseries

- Theory for Graph Matching

Section 3: Applications

- Chapter 8 - Leveraging Representations for Single Network Applications
 - Community Detection
 - “Two Truths” phenomena
 - Testing for difference between communities
 - Edge-wise communities
 - Vertex-wise communities
 - Model Selection
 - Vertex Nomination
 - Anomaly detection
 - Out-of-sample Embedding
- Chapter 9: Leveraging Representations for Multiple Network Applications
 - Two sample hypothesis testing
 - Description
 - Case study
 - Graph Matching
 - What is graph matching?
 - Graph matching algorithm
 1. GM
 2. Padded/Matched GM
 3. Seeded GM
 - Vertex Nomination via Seeded Graph Matching
- Chapter 10: Algorithms for more than 2 Networks
 - Anomaly detection
 - What is anomaly detection?
 - Anomaly detection algorithm
 - Case study
 - Testing for Significant Edges
 - Testing for Significant Vertices

Specs and schedule

How many pages do you expect the book to be? Will you be using illustrations or screenshots?

We expect this book to be roughly 450-650 pages. We will rely heavily on figures and illustrations to explain concepts. Most figures will be generated in Jupyter Book directly from Python code. A corresponding github repository will include all data necessary for generating all figures and data analyses in the book.

Do any special considerations apply to your plans for the book, such as using an unusual format, color, or hard-to-get illustrations or anything else calling for atypical resources?

We will be writing in Jupyter Book. We anticipate that a draft in jupyter book with fixed software dependencies will enhance the reproducibility of the book and serve as a future guide for later projects as to develop data analysis textbooks.

Please outline your anticipated delivery schedule (MM/DD/YYYY). Two initial draft chapters delivered by:

The book first draft is near-completed. See [Chapter 5](#) and [Chapter 8](#).

Half-draft manuscript delivered by:

Well over half of the manuscript is completed. See the links above for links to the remaining chapters which are completed.

Full-draft manuscript delivered by:

We anticipate a full manuscript to be delivered by June 30, 2022.

Final manuscript ready for production by:

We anticipate that the full final manuscript will be prepared for production (on our end) by August 01, 2022.