A Comprehensive Investigation of Graph and Network Comparison: Techniques and Applications

Marissa Graham

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Emily Evans, Chair
Benjamin Webb
Christopher Grant

Department of Mathematics

Brigham Young University

Copyright © 2018 Marissa Graham

All Rights Reserved

ABSTRACT

A Comprehensive Investigation of Graph and Network Comparison: Techniques and Applications

Marissa Graham
Department of Mathematics, BYU
Master of Science

This is going to be the abstract.

Keywords:

Contents

List of Tables

List of Figures

Chapter 1. Background

We need to introduce all of the math that the different methods I want to talk about are going to use. Just quickly explain things and light context, so you don't have to google so much like when I was reading the fifty years paper. I'm going to cross reference things by what chapters use them so you know why it's important. We could move this stuff into the actual chapters, but I think it will be nice to have it all in one place so you know what you're getting into before reading.

This is also a good place to get everybody on the same page, terminology wise. How do we feel about a glossary? At minimum, a notation page would be great so I can just use G, A, etc. pretty freely throughout. It might not be too necessary, though.

1.1 Properties that all graphs have and also definitions

A **graph**[1] is formally defined as a finite, nonempty set V of **nodes** or **vertices**, combined with a set $E \subset V \times V$ of **edges** representing relationships between pairs of nodes. Throughout this work, we denote the number of vertices in a graph by n and the number of edges by m where not otherwise specified.

We normally deal with with **simple graphs**, which are those that do not have more than one edge between any pair of nodes (that is, a **multiedge**), and do not have any edges from a node to itself (a **self-edge** or **self-loop**). We should definitely still make sure that our math applies to generic graphs though and think about this, but most of the time it's simple. I think.

We also care about whether a graph is **directed** or **undirected**. In an undirected graph, we have an edge *between* two nodes, whereas in a directed graph we have edges *from* one node *to* another node. For simplicity, throughout this work we use the notation $v_i \leftrightarrow v_j$ for

[1]The term *network* is sometimes used interchangeably with the term *graph*. While they both refer to the same mathematical object, we attempt to follow the heuristic throughout of using the term *graph* to refer to a purely mathematical object and *network* to refer to a real-world system.

an undirected edge between nodes v_i and v_j, and $v_i \to v_j$ for a directed edge. We say two nodes are **adjacent** or **neighbors**when they are connected by an edge, and a node and an edge are **incident** when the edge is connected to the node. In the case of a directed graph, we call node v_i a **predecessor** of node v_j if there is an edge $i \to j$. In this case, v_j is a **successor** of v_i.

A graph can also be **weighted**, meaning each edge is assigned some real value w_{ij} representing the "strength" of the connection between nodes v_i and v_j. Generally these are positive.

In an undirected graph, the **degree** of a node is the number of nodes adjacent to it, or the sum of the weights of the incident edges in a weighted graph. In a directed graph, we distinguish between **in-degree** and **out-degree**, which are the number or total weight of a node's incoming and outgoing edges, respectively.

When studying real-world networks, we also make a distinction between **deterministic** and **random** networks, as discussed in (cite fifty years paper). This distinction is the same as that between a variable and a random variable A deterministic network is one in which the nodes and edges are "fixed"; that is, there is no other distinct graph which represents the same object and the nodes and edges do not change over time. In order to use graph theory to study real-world phenomena, we generally must use statistical inference methods and stuff and just treat it like a random variable instead. Some methods account for this and some don't and we should pay attention to this. Add "Statistical Analysis of Network Data" to your reference list because it discusses this.

1.2 Fancier properties and special graphs

These two should come up as you fill stuff out, so we're not going to do all of them quite yet.

Old list: Trees, Adjacency matrix, Laplacian, Maximum common subgraph and minimum common supergrpah, Isomorphisms and alignments (bring up exact and inexact here), Ker-

nels, Discussion of metrics with respect to graphs, Graphlets, Tanimoto index and Hamming distance between I'm still salty that the one survey paper didn't include

More: modularity, directed acyclic graphs, assortativity, connectivity, centrality notions and distributions, diameter, giant component

1.3 Basic algorithms and methods

Don't do psuedocode, this probably isn't even necessary, but you want to talk about the basic types of problems people want to solve using graphs aka the basic questions people want to answer about or using graphs, and the strategies for going about answering them. Go back and look at your initial notes from when dr. evans was explaining the six problems and find representative papers that talk about those different things.

Path finding and walks, Basic node similarity, Basic community detection

1.4 Other math background and definitions

Some of the stuff draws on math things that aren't network specific, and we should explain those too, because we're nice. What might be best is to be a bit colloquial, since you can't give a full explanation of EVERYTHING. Just like, use the terms, but also give an intuitive idea of why we're doing this and what the goal is and how it works.

Old: Spectral stuff in general, Edit distances?, Dynamical systems, Cost function (context of edit distances and alignment), Linear and quadratic programming, Weight matching, Kernels

Chapter 2. Introduction

2.1 This is why we care and the types of problems they're useful for

We're going to start off by talking about why graphs are useful. Then we're going to talk a bit about why graph similarity is useful, which should be easy after you've read all these papers. Just a general overview of the types of fields that naturally lend themselves to network analysis, with examples of the types of problems you'd see.

Really, networks are a useful way to model relationships between stuff for anything. It's obvious that relationships between things can be visualized with a network, and sometimes just looking at it is enough. Like a flowchart, or the atoms in a molecule. Something small where you want to see the structure of it. It's less obvious, I think, what else you can use that relationship for. That's less simple to answer, but a good place to start is by thinking about the kinds of questions you want to answer. You can come up with all kinds of algorithms and metrics, but it has to mean something

2.1.1 Infrastructure. For example, you can have networks where you want to measure and track the flow of traffic through the network, so you'd want to answer questions about how much stuff can flow through the network, how much redudancy you have, where it would make the most sense to add or remove things, and so on. The obvious use case is road networks, but air traffic control would also fall under this category, phone lines (really most public utilities, electricity, water, sewage, internet, all that), cell phone networks, all that. Is it planar? Almost planar? How do those restrictions affect it? Random walks? How do we model and calculate all this?

2.1.2 Social Networks and the World Wide Web. Then you've got social networks and the internet (specifically the world wide web, not the network of routers that make up

the internet itself), which kind of fall under the same category when it comes to questions you want to answer about it. Where are the communities? Who's important? What do we mean by important? How well is everybody connected to each other (useful question for the former type, as well as this)? Reciprocity, assortativity, small world.

2.1.3 Biology. THIS SECTION IS CURRENTLY WORD VOMIT BUT I'LL REDO IT AS I GO THROUGH AND READ PAPERS

Then there's biological networks. Wide variety of scales. Metabolic networks, protein-protein interaction networks, metabolic interaction networks, genetic regulatory, neural, ecological. What types of quesitons are we asking? For ecological networks, it's probably more similarity to the "flow of stuff" ones and the "social network" ones.

Metabolic networks probably go along with the "flow of stuff" category, just the difference is instead of having a "flow of stuff" network that you're trying to *build*, you have one that already exists and you're trying to figure out what all the little pieces do. Motifs are probably useful here, because the individual nodes by themselves aren't all that interesting to answer questions about. That's starting to get more into the network similarity thing. Similar with protein interactions. It's easy (or, well, *possible*) to figure out what stuff is dependent on other stuff but hard to say why.

Similarity between pieces of a metabolic network sounds very useful for evolutionary biology. Also looking at pieces across species should tell you what function the piece should serve? Also with both that (and protein-protein, and evolutionary), you're starting to get into the question of what's normal and what's abnormal and how would you fix it.

2.1.4 Computer Science. THIS SECTION IS CURRENTLY WORD VOMIT BUT I'LL REDO IT AS I GO THROUGH AND READ PAPERS

Also, computer science, besides just the internet. Queueing theory! Talk to dad about queueing theory! It's more graphs here than networks, because you're using a graph structure to represent the relationships between things, it's not TECHNICALLY "real world".

Computer vision and natural language processing really lend themselves well to graph-based study. Typically in this case you'd kind of pull the pieces out of an image and see how they relate to each other. If you wanna be stupid about it, just take the relation of all the pixels to each other. I'm not really sure how they construct these. But once you can figure out what the important pieces are you can compare it against other stuff. Like the way they do facial recognition visualization in every movie ever. Oh, that's also good for stuff like animation, I bet! And video games, or just anything where you're going to model it with a bunch of splines on a non-grid domain.

Language processing is good for this too. Relationships between words can tell you a lot about what they mean and how they're supposed to be used, especially if you have a large enough corpus. Like google's word2vec, so you can learn how to translate between languages. Here you're probably asking less "match this thing to something in a database", although that might actually be the case for translation, I don't know, but I feel like figuring out grammar and all that is definitely along the lines of asking "is this structure normal?". Again, this'll be better after I've read a bunch of the papers.

Put a footnote or side note or something here about the difference between "network" and "graph" and similarity vs isomorphism vs matching and such.

2.2 Dataset Creation

THERE ARE A LOT OF VERB TENSE ISSUES RIGHT NOW BUT I DON'T REALLY KNOW WHAT VOICE I'M SUPPOSED TO USE RIGHT NOW SO I'LL FIX IT LATER

Survey papers are obviously difficult in general. How do you know that you're getting everything? How do you know what's important? In particular, how do you know you haven't overlooked anything major, and that you know what's important, when you're not already an expert in the field? What about an interdisciplinary field? Would you have to be an expert in both?

A good place to start would seem to be with other people's survey papers. But what about

things that have come into existence since those survey papers? What if the survey papers assume more expertise than you have? How do you know you can trust their judgement on what's most important, and that they've got everything?

Obviously this is a major question for academic research in general. People writing survey papers care, but so does anybody who wants to get a sense of the field, to collaborate, or learn, or see if their work is original, and so on. This question is the whole reason things like Web of Science and Scopus and even Google Scholar itself exist, and a good example of why we care about network science in the first place. So we've got a really solid justification for looking at the citation network.

Why can we justify looking at the citation network of just network similarity papers, especially given how long this took to make? Well, first of all, YOU don't care about how long it took to make, so as long as it doesn't ruin the scope of my project, I don't need to justify THAT. More to the point, I'm not an expert in the field, and I'm not going to become one in a few months just by reading. The average reader doesn't have any good prior reason to trust my search process and my judgement about what's important. If I want my survey to be useful, I need to have a satisfying answer to that question. If I look at the citation network, I can guide my own reading, but I can also back up my assertion about what's important with an explanation of my process and standard centrality and community detection measures.

The main reason this needs justification at all is that citation network creation is not at all trivial. Some journals and databases can give you a citation network of the references in their own domain, but with such an interdisciplinary field, that only covers a small subset of the network needed. As a result, and since intellectual property restrictions mean you can't just scrape a whole citation network, I made it manually by making .txt file reference lists for relevant papers and constructing the network accordingly.

2.2.1 Approach. Relevant papers were determined by searching google scholar for "graph" or "network" + "alignment", "comparison", "similarity", "isomorphism", or "matching". I

went through the first five pages of results for each of the ten search terms and collected all topic-relevant papers. Then I set up an email alert for those same search terms, and have been continuing to add new relevant results to the database throughout the process. All graphics included are generated using the newest version of the database, which includes reference lists for 204 (UPDATE AS NECESSARY) papers and preprints up through June 5th, 2018 (UPDATE TO DATE OF DRAFT SUBMISSION). Any paper for which we have a reference list is referred to as a "parent" paper, and the references are referred to as the "child" papers.

In order to create the network, we need to be able to parse the freeform citations for each reference list in order to obtain metadata and recognize references as repeatedly cited. This is a difficult problem, as the parent papers in the database span over fifty years and represent a wide variety of citation styles, languages, and optical character recognition and unicode-related hiccups. Instead of attempting to parse a citation into component parts, we used the REST API to search for each record in the CrossRef database, which already has the metadata parsed for any reference it includes. References are marked as duplicate if their parsed metadata matches and both are verified as correct, or if both their metadata and original freeform citation match exactly.

The results of this search are considered verified if the title of the result can be found in the original freeform citation. This was the case for about 75% of the parent papers, and about half of their children. For unverified parents, I manually corrected or found title, year, author, DOI (if it existed), and URL information as well as reference and citation counts. I checked through the unverified child references and marked correct results. This occurred in about half of all cases, usually due to punctuation discrepancies, misspellings, or unicode issues. Results were counted as correct (but noted as "half-right") if CrossRef returned a review, purchase listing or similar for the correct item. I then went through the remaining incorrect references and manually parsed the author, title, and year from the citation, or looked them up if not included. I deleted any references which did not refer to a written

work of some kind; specifically, references simply citing a website, web service, database, software package/library, programming language, or "personal communication".

Then we write the citation network of the database relationships to a .gml file which can be loaded in Mathematica. By default, the nodes in this network include the title, year, reference and citation counts for each paper as properties. Including further metadata properties in the .gml file is not difficult, but additional string properties dramatically slow Mathematica's ability to load such a large network (5252 vertices and 6970 edges as of SUBMISSION DATE, takes about two minutes to load on a 2.6GHz 6th-gen quad core Intel Core i7 CPU with 16GB of RAM, Windows 10 Home, Mathematica 11.2 PUT THIS IN FOOTNOTE?), so they are not included by default.

The dataset itself and the code and source files used to generate it can be found on GITHUB REPOSITORY LINK, as well as documentation and instructions for using it to generate a similar dataset for any collection of properly-formatted reference list files.

2.3 Dataset Analysis

I've got several mathematica notebooks full of this. I want to leave it mostly there for now, though, because I don't want to have to create all the figures again every time I add a couple papers and re-run the analysis.

2.3.1 Overall Network Statistics.

- Very basic statistics about the network, e.g. size, degree distributions, how much it looks like a typical citation network
- Slightly more interesting statistics about the network as a whole: connectivity, diameter, assortativity, percentage in the giant component, clustering?
- Distributions for various centrality measures (closeness, betweenness, etc.)

- Repeat these three for the subnetwork of nodes with positive outdegree or indegree greater than 1
- Not including: reciprocity and actual SCC, obviously, since it's acyclic

2.3.2 Community Structure and Important Papers. Description of how I partitioned the network and the types of papers observed, types of important papers observed, what the community structure seems to be.

Note that a big problem with this is how comparatively rare it is for the papers to have positive outdegree, and most of the papers are only cited by one person. To do the partition, I took the weakly connected component and then the subgraph of all papers that have positive outdegree or indegree greater than 1.

Since I had to go through so many papers manually, I feel fairly confident in the assertion that the main two groups are computer science and bio, so I did a (modularity maximizing?) partition of the network into two groups. This made a lot more than two groups, though, so I looked at those to see which ones are contained in one or the other. TALK ABOUT RESULTS OF THIS.

Here is a table of the top papers for each community and overall by indegree, outdegree, betweenness, PageRank, and closeness centrality. These all go in the bibliography.

2.4 Structure of Paper

It's mainly CS and bio as far as applications go. This is mostly based on the fact that I went through pretty much all the titles. I could get some statistics on how many of them have bio-type words in the title. Look at the fringey guys with the modularity partition.

What are we doing with large networks? What problems are we trying to solve? What about with citation networks? Detect communities and determine which ones are important? Measure change over time?

What problems are we trying to solve when we're working with metabolic networks? Why

do we care about looking up proteins? How is DNA sequence alignment network similarity? Is it just that all these microRNA papers cite those kinds of papers a lot?

Mapping the brain to figure out what's normal and what's important and what does what? Is that really network similarity. Community detection requires a notion of similarity. Biology: figure out what stuff does? Why do we care about protein interactions?

Types of problems we're trying to use network similarity for:

- Searching for things in a large database of small graphs (fingerprint classification, protein search, facial recognition). Nearest neighbor type thing. Unique identifiers is really more of a node similarity thing, but oh well.
- Classify things as normal or anomalous (malware, cancer, trajectories, grammar,)

Chapter 3. CS

Look at what they call it. Graph matching seems to be more small graphs,

The main two kinds of applications seem to be computer vision and natural language processing. Both of those make sense. Unique identifiers, grading, image processing, large database search, e.g. handwriting and fingerprint/facial/iris recognition

Computer vision: fingerprint classification (lots of those, especially older ones), shape matching, pulling objects from an image,

Natural language processing: semantic relations, obviously

Niche things: malware classification.

Predictions and expectations for how it'll go: I think the CS stuff is pretty much exclusively graph matching. The standard techniques would be good to get from the thirty years paper, which is definitely on the list.

Chapter 4. Bio

I haven't read enough bio papers to really know for sure, but based on a quick glance through titles, bio applications seem to be protein folding (and molecular similarity? chemical st), metabolic interaction networks (duh) gene stuff, something something microRNA (same thing?), brain networks, SOCIAL networks, protein database search (should be similar to the graph matching ones where it's trying to find an image in a large database)

Main two: molecular structures and metabolic interactions

Predictions: social network and metabolic network relatively similar approach? Large networks of interactions. Molecular structure on a small scale should be like graph matching?

Chapter 5. Fringe

Citation network stuff would probably go here, I think. It's kind of in the middle.

Here's the ones that didn't really fit in either, or had their own little communities when we did modularity. I'm also curious about how the synthesis of it would go. And the papers that have parents from both communities.

Possibly fringe: air navigation route systems

You're not going to have the black or red books in the paper list but you still need to have them in your bibliography, don't forget.

Papers go in the bibliography if you read them to put them in the chapter. Any of the ones you've got printed out, so definitely fifty years. Not all the parents need to go in the bibliography, though. You need to cite/acknowledge CrossRef, for sure. What about python packages like gspread? Idk.

Bibliography

[1] Nobody Jr. My article please actually change when i do this, 2006.

[2] Alison Pease and Simon Colton. Computational Creativity Theory: Inspirations behind the FACE and the IDEA models. In *ICCC*, pages 72–77, 2011.

Index