A Librarian's Guide to Graphs, Data and the Semantic Web

James Powell, Matthew Hopkins
Los Alamos National Laboratory

Introduction

Single file, rarely out of step with one another, a large contingent of ants marches almost as a single pulsing organism. In intimate proximity to one another, they make their way toward the remnants of a careless human's lunch. From above it is hard to see the stream of tiny ants emerging from a crater-like mound of sand.

As individuals, they are not gifted with keen sight. And despite such close quarters, they don't feel crowded, rushed or claustrophobic. They effortlessly climb obstacles that would at our scale seem to be insurmountable boulders. They build nests, fights off invaders and lifts massive items many times its own weight all in support of the collective.

Yet they have no leader. The colony is self-organizing. Like a road crew repainting the stripes along a stretch of highway, they constantly refresh the trail that guides their parade until it is no longer needed. Their highway is a forage line, marked by a pheromone trail. Some random ant discovered the food and excitedly established the trail for others to follow. They make quick work of the breadcrumbs and then attend to other matters. Some ants are soldiers, some are harvesters, some are diggers, and others just clean, switching roles as needed. You'll find no multitasking here.

Their behavior may be simple, but the aggregate results are complex. Ants can quickly adapt to the challenges of the local environment, achieving things as a community that would be impossible for a single ant. They do so by following simple rules, playing simple roles, and occasionally submitting to the whims of chance. They are a perfect example of what scientists call a complex system.

Ants endure and thrive in a dynamic world full of unknowns even though they have no hope of comprehending that world in its entirety. Simple rules guide individuals to perform whatever task is most pressing for their collective survival in the moment. An essential ingredient of those rules is randomness. Food foraging starts out as a random, directionless activity. When ants find a food source, they make pheromone trails to the source. The trails dissipate unless they are reinforced. When the food is gone, the pheromone trail fades away, and the ants usually retreat to their nest or move on to another source of food. Randomness takes over again. We used to believe that the universe was deterministic, that if you knew all the initial conditions and all the rules that govern it, then you could know everything that would ever happen. But ants know better. They don't plan, they react.

DRAFT

In our universe, the best models we have come up with to anticipate what can or might happen are based on probabilities. Inherent in probabilities is an element of randomness. Ants do perfectly fine by individually focusing on discrete, simple tasks. They react rather than acting under centralized direction, without anticipating anything. The colony as a whole knows how to survive. But separate some ants from the colony and the results can be disastrous.

The ant mill phenomena is illustrative of how important it is for individual ants to remain connected to their colony. If a few ants lose track of their forage trail, they begin to follow one another. Pretty soon they form a circle and they will march around and around until they all die. As a collective, ants manifest complex behaviors that ensure their survival. This behavior is more sophisticated than one would assume possible given what is known about an individual component. Emergence is the term for this phenomena, and it is characteristic of complex systems.

The study of complex systems is relatively new, and there are many details around the edges that are a bit hazy. Even the precise definition of a complex system is a subject of debate among those who study them. Complex systems exist all around us and they inhabit the space between organized simplicity (simple, recurring patterns) and chaos. Complex doesn't necessarily imply complicated in the sense that the word complicated is often used. An ant isn't a complicated animal, but an ant colony is a complex system. Complicated systems are not necessarily complex. A car is quite complicated under the hood, but the components are highly individualized and play distinct roles for which they were specifically designed. There seems to be a tipping point somewhere between very simple and completely chaotic where complexity is manifest. In other words, an ant and its relationship to other ants, is only as complex as it needs to be in order for an ant colony to survive and adapt to its environment.

Why the sudden interest in complex systems? Well, in the 20th century, science hit a wall. The era when a single individual laboring tirelessly for a lifetime was able to understand and make significant contributions in a field was drawing to a close. For centuries, progress had come from the process of reductionism. Reductionism supposes that a complex phenomenon can ultimately be understood by comprehending its constituent components. But reductionism was reaching its limits. New knowledge increasingly depended on a considerable foundation of existing knowledge within and across disciplines. Some phenomena mysteriously manifested capabilities that were not suggested by an understanding of their parts. A great deal had been achieved without the benefit of modern computing capabilities. But it was inevitable that we'd reach a threshold where it was beyond a single person's ability to move a field forward. Reductionism was yielding lower returns. Scientific advancement began to depend more and more on large data sets, simulations, modeling, statistical analysis, machine learning, and these advancements suggested that there may be some unknown, poorly understood self-organizing principles that played a

DRAFT

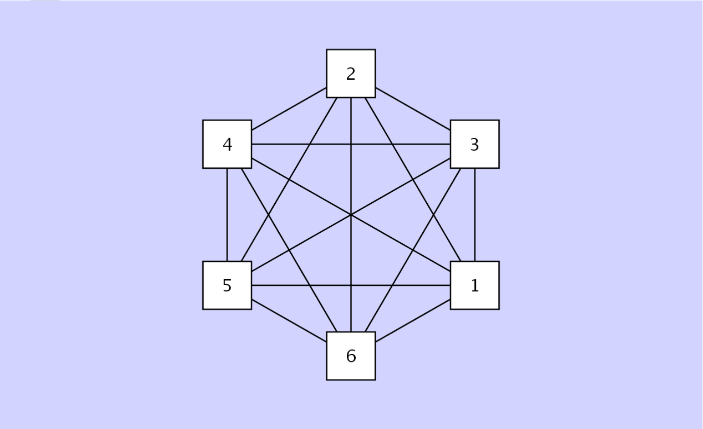
crucial role in some natural systems. Our ability to study complex systems is due in no small part to the advances in computing over the last few decades. And one of the techniques we use to understand complex systems is to model them as graphs.

In this book we talk a lot about models and modeling. A graph is a type of model. A model is a representation of some other thing. A paper airplane is a model that bears some resemblance to an actual airplane, and can even fly. The advantage of a paper airplane is that you can make and fly your own without killing anyone. Models are good for exploring aspects of a thing, and for testing ideas about it without doing any harm. Mathematicians often represent their models as formulae and they breath life into a model by assigning values to its variables.

Graphs are not merely models; they are mathematical models. This type of graph is not the same as a bar or pie chart you're likely familiar with. It is a model of things and their relationships with one another. This model can simultaneously capture complexity and simplicity. A graph model of a complex system can reveal how it has achieved balance within itself without overcompensating, and with its environment, without eventually rendering that environment unlivable.

Visually, a graph is most often represented as a collection of dots with lines between them. The dots are an abstraction for the things that exist in a system, process, or knowledge space, and the lines are relationships among them. Graph theory is the field of mathematics that is concerned with analyzing and exploring graphs. Here's a visualization of a simple graph, together with the formula that mathematicians use to describe a graph:

G= (V,E)
All this says is, a graph G is a set of vertices (1,2,3,4,5,6) and the edges that connect them.



Graphs have been used to gain a better understanding of many phenomena. For example, there was a long standing mystery in biology regarding the energy requirements of an organism as its size increased. Intuitively, one would guess that if you doubled the size of an animal, it would require twice as much food. But it turns out that's not the case: a 75% increase is all that's required. How can that

DRAFT

be? The solution to this mystery lies in the answer to the question: what do a city, a leaf, and a lung have in common? Geoffrey West of the Santa Fe Institute explains:

"The key lies in the generic mathematical properties of networks. Highly complex self-sustaining systems ... require close integration of many constituent units that require an efficient supply of nutrients and the disposal of waste products. ...this servicing – via, for instance, circulatory systems in organisms, or perhaps transport systems in cities – is accomplished through optimized, space-filling, fractal-like branching networks whose dynamical and geometric constraints are independent of specific evolved organismic design."

Blood vessels, the vascular system in leaves, and transportation routes to and within a city are not just like networks, they are networks. Nature settled on power law networks to solve many problems. We're not far behind. The infrastructure that supports our cities, our regional and global transportation systems, and our power grids are networks that have similar properties to evolved network systems. So it should come as no surprise that graph models are useful for understanding many aspects of the natural world.

A graph is a great way to model contacts involved in the spread of disease. The graph starts with patient 0 – the first person infected by a disease. From previous research, it may be known that this particular disease has a reproduction number of 2. That means that every infected person infects on average two more people. The incubation period may be on average four days before symptoms appear. Patient 0 was infected 8 weeks ago, and at least 125 people have shown up in area hospitals with the disease over the last seven weeks. Health care workers begin interviewing patients to determine who had contact with whom. If the contact preceded infection, and only one infected individual is identified, then there's a clear relationship. Pretty soon a tree like graph emerges that confirms that the reproduction rate has been right around 2. The good news is, as time passes, the graph has more and more nodes with no new connections at all. Quarantine and early treatment are working. The disease seems to be burning itself out. The graph model provides tangible evidence of this. As you might guess from this example, epidemiologists love graphs.

In fact, many disciplines love graphs. Sociologists study communities and communication with graphs. Chemists can study graph representations of compounds before they ever attempt to produce them in the lab. Physicists can study phase transitions and myriad other aspects of matter and energy. Astronomers can study galactic clusters with graphs. Environmental scientists can represent food chains with graphs. Cell biologists can model metabolic processes in a cell and even get some sense for how such a process evolved over time. Economists can study purchasing or investing patterns among consumers. City planners can study traffic patterns to figure out where mass transit might relieve congested roadways. When that transit system is in place,

DRAFT

the subway map daily commuters will use is itself a graph. In a connected world, being able to analyze connections is a powerful tool. A portion of this book is devoted to the use of graphs in various fields, and we use these reviews to illustrate various important concepts in graph theory, often building on concepts introduced in earlier chapters.

Tim Berners Lee looked to graph theory for inspiration as he developed a concept for a global network of information. His vision included a democratic model where anyone could encode and share any facts. These facts would be comprehensible by both man and machine. He sometimes even referred to this as the Global Giant Graph. His goal was to push the granularity of information representation down to the level of individual facts. These facts would be shared and interconnected. And like the World Wide Web before it, there would be an ecosystem of standards and technologies upon which this system would be built.

We refer to this Web of interconnected knowledge by various names, including linked open data, the linked data web, or simply the Semantic Web. At the core is a simple information model which defines the structure of a fundamental unit of knowledge called a triple. A triple is a graph segment made up of two nodes and a directed edge between them. These segments reside in a larger knowledge graph. This graph coexists with the World Wide Web. Nodes are represented by unique network identifiers that give a thing a unique name and provide a pointer to it. Identifiers make it possible to add more graph segments that further describe these things. This simple information representation model results in a complex web of knowledge. Sound familiar?

This book is intended to provide an overview of a number of graph-related topics. We have attempted whenever possible to write it so that each chapter can be consulted independently of any other chapter, but there is at times an unavoidable progression of background knowledge that is prerequisite for understanding later chapters.

It is a book of lists. Lists encapsulate and summarize in a way that most everyone finds helpful on some level. Of course when we do start off with a list, we follow up with a narrative that expands the list elements into the things you really need to know about a given topic.

It is a book of examples. To the extent possible, we don't just talk about a given technology, but try to show you examples and explain what it's good for. Standards documents tend to be broad and deep, as they should. We tend to feature vertical slices of a given technology which we hope will give you a good idea of some of the uses for it and an incentive to learn more.

It is a book for people who love books and reading. We try to bring the topics to life with anecdotes from current research papers, other more technical sources and even historical texts and novels. These are intended to engage, enlighten,

DRAFT

and occasionally even entertain.

It is a book that covers more topics at an introductory level, rather than a few topics at a deeper level. You may never need to delve deeply into some topics we cover, but we aim to make sure that if you ever hear about or encounter the topic again, it will be familiar, not foreign.

It is, on occasion, a book of code. Some chapters discuss program code, markup and information representation languages. If you never plan to write software, you may not need to delve deeply into the handful of chapters that discuss programming and APIs, but we do our best to provide a non-technical overview before we show you any code. If the code is more detail than you need, skip forward and share the code with a programming friend or colleague. If they are tasked with solving a problem using that particular technology and are new to it, they may thank you profusely, as we strive to introduce concepts from a beginner's perspective.

It is a book to help you get things done. These are big topics. There are more rabbit holes here than in the proverbial briar patch. We steer you clear of the rabbit holes by highlighting solid, widely adopted technologies, not standards and technologies still under development that may or may not come to fruition. We introduce you to things that you can use now. We provide concrete examples throughout, and we conclude the book with in-depth case studies of two real-world applications. We believe learning is good, doing is good, but learning while doing is better.

With this book, we will introduce you to some of the finer points of graph theory and the Semantic Web. We hope to provide you with a solid conceptual framework of graph theory, and a comprehensive overview of semantic web technologies, which we think makes this book unique in this field. For librarians, it will help you understand how patrons in a variety of fields might be modeling aspects of their field of research, and how you might apply graph theory to some information discovery and retrieval challenges in your library. For information technologists such as software developers, it will give you the background you need to understand how graph theory and the semantic web can be leveraged to represent knowledge. There are many open source tools, software libraries, and standards for graph data and for the semantic web. We will introduce some specific tools but also give you the knowledge you need to find and evaluate other similar tools.

Although this book is not specifically about complexity science, it is a guidebook to some of the technologies used to model and elucidate aspects of complex systems. There are many excellent books about complexity and complex systems, including Melanie Mitchell's “Complexity: a Guided Tour”, “Deep Simplicity” by John Gribbin, and “Simply Complexity” by Neil Johnson, just to name a few. Any or all of these resources may help you understand the field and

DRAFT

its myriad applications.

Libraries are constantly striving to provide new and better ways to find information. We've been doing this for hundreds of years. Graph theory and semantic web technologies offer a way for libraries to re-invent themselves and their services, based on relationships. Finding things in libraries used to depend on physical access to those things. Then we introduced layers of abstraction: classification schemes, call number systems, card catalogs, online public access catalogs, and finding aides, to name a few. Yet relationships may be the most natural way for users to find things. Graphs and the semantic Web are great at modeling and exploring relationships. Libraries can model the relationships among content, topics, creators and consumers. Rather than mapping all our previous solutions onto the semantic Web, perhaps it is time to develop a "relationship engine" that can leverage these new technologies to totally reinvent the library experience.

Humans are obviously not ants. But humans can model ant behavior using graphs. We can model the inner processes of an ant's physiology using graphs. We can model an ant colonies' relationship to its environment using graphs. We can model what we know about ants using graphs. Graphs are tools that help us understand complex systems. In a non-deterministic universe, we face myriad unknown and complex challenges, some will be random events, and some will be of our own making. It is contingent upon us to use every means at our disposal to augment our ability to comprehend complexity. Turning our back on complexity is no longer an option. Graph models of complex systems, and the practice of embedding knowledge into graphs, are powerful tools for comprehending complexity—and they enable us to use that understanding to our advantage.

DRAFT

Chapter 1
Graphs in Theory

Abstract
Graph theory has a humble beginning, as a solution to a puzzle. Residents of the Prussian city of Kongisberg, which was bisected by a river, had long pondered this puzzle. It took Leonard Euler, a mathematician to not only solve the problem, but to do so in such a novel way that his solution launched a new field of mathematics. This chapter reviews the early history of graph theory, starting with the famous story of the seven bridges of Konigsberg. It will also present other early and historically significant uses of graph theory such as Stanley Milgram's landmark “small world problem” study. Some fundamental topics are introduced, such as vertexes and edges. We close out the chapter with an exploration of the four color problem, which was explored and solved using a special type of graph called a planar graph. The four color theorem inspired a mathematician who today lends his name to a special number much coveted by mathematicians, the Erdos number.

Keywords *Seven Bridges of Konigsberg, Leonard Euler, Eulerian path, Stanley Milgram, six degrees of separation, planar graph, four color theorem, Erdos number*

Chapter 2
Graphs and How to Make Them

Abstract
We encounter graphs all around us both in the physical world and in representations of that world. It turns out that simple graphs are intuitively easy for people to understand, and so they are often used to represent information. Our families are collections of entities (people) who have relationships to one another (marriage, or by birth). Science has found a great many uses for graphs as they explore systems which have a great many actors and numerous relationships among them. They have devised numerous methods over the years to explore these graphs and quantify their characteristics. This chapter introduces a few graphs we encounter in everyday life, and then explores some common characteristics of graphs. It discusses some of the challenges related to modeling systems as graphs. Finally it provides a brief overview of major categories of graph analytic techniques.

Keywords *graphs, networks, graph modeling, graph analytics, graph visualization, centrality measures, distributions, cluster analysis*

Chapter 3
Graphs and the Semantic Web

Abstract
The Semantic Web is a collection of technologies and standards that utilize graphs to model knowledge. It is called the Semantic Web because the technologies coexist with and leverage Web technologies that allow us to access and create links between Web pages. It uses links to formalize representations of knowledge that convey their meaning in a context that is not unlike how humans classify things as they process information. The vision behind something like the Semantic Web has existed for decades, but only with the advent of computer networks and the World Wide Web, has it been possible to make it a reality. The formalization is an abstract model called RDF, which relies on a fundamental knowledge unit referred to as a triple. This triple is a graph segment with two nodes and a directed edge between them. This chapter introduces some of the

DRAFT

fundamental technologies and concepts behind the Semantic Web. It introduces the process of modeling information using RDF. It explains the basics of linking within the Semantic Web, and illustrates how graph analytics can be applied to RDF graphs.
Keywords *Semantic Web, RDF, triple, statement, truth value, IRI, URI, URL, linked data*

Chapter 4
RDF and Its Serializations

Abstract

Graphs and Semantic Web graphs represent a continuum between simply representing data and indicating that there is some relationship among entities in the data set, and a formalized way of representing those entities and specifying those relationships. This chapter delves deeper into RDF and modeling information using RDF. It then introduces the notion of serializations. RDF is an abstract model, a general specification for modeling data for the Semantic Web. Serializations are the actual manifestation of RDF in a particular format, stored in a text file. We look at several serializations including Turtle, N-triples or Notation3 (N3), RDF/XML, the new Javascript object data format JSON-LD, and the microformat RDFa, which can be embedded into Web pages.
Keywords *RDF modeling, serializations, turtle, N3, RDF/XML, JSON-LD, RDFa*

Chapter 5
Ontologies

Abstract

The graph segments that make up triples in the Semantic Web have another dimension of connections that link subjects, predicates, and some object values to structured vocabularies called ontologies. This is how the RDF graph model encourages the use of shared terminology for instance data. Ontologies describe the kinds of things that exist in a particular knowledge domain, and their relationships to one another. These are called classes. RDF nodes are then identified as being of a particular class that is defined within a given ontology. In addition, the edges in triples are give the same treatment. They are defined in ontologies as properties. This chapter discusses the process of modeling and using ontologies. It introduces RDF schema and OWL, which are languages for formally defining ontologies. It provides examples of ontology entries and their serializations and use. Finally it touches on advanced concepts such as OWL profiles and reasoning.
Keywords *ontologies, logic, reasoning, RDF schema, OWL, taxonomies, classes, object properties, data properties, profiles*

Chapter 6
SPARQL

Abstract

Although RDF data serializations can be published directly on the Web, most Semantic Web projects generate enough data that the use of a specialized data storage system is warranted. Data storage systems that are specifically designed for Semantic Web data are most commonly referred to as triplestores. Although they may use any number of techniques for storing and indexing triples, they provide a common set of capabilities including importing RDF data serialized in various formats such as RDF/XML, data update functionality, managing aggregations of triples as distinct repositories or graphs, indexing triples for faster searching, inferencing, and a standard way of searching RDF

DRAFT

triples. SPARQL is the standard query language for RDF. This chapter provides an overview of triplestore functionality and services. It introduces the SPARQL query language and presents examples that illustrate SPARQL's capabilities. Finally it briefly discusses the concept of a SPARQL query endpoint, which is an outward facing discovery mechanism provides by a triplestore to help others find your data and link to it.
Keywords *SPARQL, where clause, filters, bindings, RDF patterns, select clause, variables, SPARQL query endpoint, Linked Data Fragments, triplestore, CRUD*

Chapter 7
Inferencing, Reasoning & Rules

Abstract

By virtue of the fact that a graph defines things and their relationships to one another, it is possible to follow paths in a graph or compare entities and their connections, to deduce, or infer, knowledge that isn't explicitly represented. Humans do this all the time but are barely aware of it most of the time. The Semantic Web embraces and facilitates computational reasoning over triples because ontologies are defined based on the formal branch of mathematics called first order logic. Every graph segment represents a statement of fact, which is called an axiom. Logic provides a formalized language for specifying the implications of statements, and for evaluating a series of statements to determine new knowledge or to validate the truth of a condition represented by those statements. This chapter discusses the notion of logic and the formal version used by the Semantic Web called first order logic, and some of the challenges in computational logic. It introduces the operators and structure of logic statements. Finally it shows how ontologies, triples and rules relate to logic, and shows how rules can be used by Semantic Web applications to use semantic data in ways that mimic human understanding of information.
Keywords *axiom, inferencing, reasoning, logic, first order logic, the frame problem, closed world, open world, rules, SWRL, N3 rules*

Chapter 8
Understanding Linked Data

Abstract

As a graph representation of information, the Semantic Web is subject to “the network effect” which is the result of users publishing and linking their triples to other triples. The Semantic Web defines several principles that enable linked data: the RDF model, which is a graph segment, the IRI which uniquely defines a thing, the use of shared ontologies, and the ability to link out to other RDF triples. Linked Open Data refers to the world wide collection of published and interconnected RDF graphs. This chapter explains the idea of linked data and illustrates the mechanisms that enable it. It looks at a few technologies that facilitate the discovery of linked data sets and the entities they describe. It provides a more in-depth overview of a few linked data collections, how they are constructed and published.
Keywords *Linked data, Linked Open Data, IRI, rdfs:seeAlso, owl:sameAs, DBPedia, Geonames, Swoogle, Spotlight, Europeana, Linked Open Vocabularies, Linked Data Platform, Marmotta*

DRAFT

DRAFT

Chapter 9
Library Networks – Co-authorship, Citation, and Usage Graphs

Abstract

The next six chapters step back from semantic Web graphs to look at graph models of systems in various domains, starting with libraries. Traditional use of library resources often involves what is generically referred to as "research." What is meant in this particular context is the process of locating materials on a particular topic or produced by a particular author, determining co-authorship and topical relationships, reviewing the citation relationships, and then exploring all of these various relationships to identify other items of potential interest. These organic, explicit networks grow over time and exhibit some of the characteristics found in other types of networks. This chapter looks at how this data can be modeled as graphs, and the process of analyzing and traversing these graphs. It illustrates the use of some common graph theory techniques in the context of library data and systems, including co-authorship analysis and citation graphs between papers. It also briefly explores how usage data can be modeled using graphs to explore research trends and generate recommendations for library patrons. It introduces first mover advantage and scale free networks.

Keywords *co-authorship graphs, citation networks, subject-author graphs, preferential attachment, first mover advantage, scale free networks*

Chapter 10
Networks in Life Sciences

Abstract

Graphs and networks have important applications in the life sciences because they can model the characteristics of complex systems and complex patterns of interaction. These models can be used to predict and affect change in such systems. For example, the spread of an infectious disease within a population can be modeled as a network. Social or physical interactions among entities can predict the likely path a disease may take. With a reasonably accurate model, one can then conduct experiments that might otherwise be considered unethical, such as how effective a vaccine might be in halting the spread of a disease if introduced at specific points in the graph. This chapter looks two areas in life sciences, the spread of disease and the modeling of food webs, and what network models of these systems may tell us. Since these models are directed graphs, it introduces a couple of common path patterns found in directed graphs. It also looks at small, recurring patterns in graphs called motifs.

Keywords epidemiology, food webs, paths, cycles, Hamiltonian cycle, Eulerian cycle, motifs

Chapter 11
Biological Networks

Abstract

Biological organisms have evolved processes and information propagation models over billions of years. These processes are quite complex but lend themselves to graph modeling. These models can reveal interdependent processes, and characteristics of the network that make it robust to environmental challenges. Our own attempts to classify living organisms, such as taxonomic classification schemes, also tend toward graph models. This chapter explores how network science has been used in biology and bioinformatics, especially genomic networks and cellular metabolic networks, as well as new applications in post-genomic biological systems analysis. It explores the analysis of large datasets, through such concepts as graph alignment and weighted networks. It will also explore the fact that these networks within our bodies have been guided and refined by natural selection over millions of generations

Keywords genomic networks, cellular metabolic networks, graph alignment, trees, weighted networks, biological taxonomic classification, phylogenetic networks

Chapter 12
Networks in Economics and Business

Abstract

This chapter looks at graph representations of business organizations, the economy they comprise, and even the cities they inhabit. Some topics include network representations of management and collaboration within organizations, supply chains, and economic models. Many of these models predate the emerging science of networks, but nevertheless reflect an understanding that businesses and economic interactions run on connections, attracting investors with capital, bringing together creative people to implement ideas, managing supply lines for raw materials to create products, tackling challenges related to moving products from place to place, and stores to connect consumers to those products.

Keywords Matthew principle, herding, supply chain, pyramidal network, economies of scale, social mixing

Chapter 13
Networks in Chemistry and Physics

Abstract

Like other physical sciences, the depths of physics and chemistry can be plumbed using networks. Some of the earliest approaches to modeling molecular structures were graph based. The physical arrangement of atoms in matter can also be modeled using graphs. Dynamic graphs can be used to explore phenomena such as phase transitions and percolation. This chapter looks at how graphs are used to model and understand processes in physics. Interestingly enough, the patterns that emerge when modeling phase transitions have shown to occur in graph models of other systems. This chapter also looks at chemical graph models and how they enable certain types of chemistry in silica. Graph concepts introduced include percolation, phase transitions, and graph rewriting.

Chapter 14
Social Networks

Abstract

This chapter introduced the concept of social network analysis, which, although not invented by, was certainly made famous by Stanley Milgram and his “six degrees of separation” findings. It described how a social network can be represented as a graph, with people as nodes and the relationships among people serving as edges where edges can also reflect things like the whether the relationship is strong or weak, and whether is has directionality. It introduced some terms specific to the social network analysis community such as homophily and multiplexity, and it described the graph analytic

DRAFT

techniques that can be used to explore social networks. Finally we explored two upper level semantic web ontologies for representing facts about a social network, the Friend of a Friend ontology for expressing facts about people, and the Organization ontology, for describing organizations, and a person's relationship to and role in an organization.
Keywords *Social Network Analysis, six degrees of separation, small world network, resiliency, homophily, multiplexity, reciprocity, Friend of a Friend*

Chapter 15
Upper Ontologies

Abstract

The process of defining an ontology is typically an effort to document what is known in a particular domain, what things it contains, their properties and their relationships to one another. But there are fundamental types of knowledge that are not the purview of any particular knowledge domain. Everyone from a biologist to a sociologist to a librarian may be interested in referencing a geographic location. Physical objects have dimensions, structure, and displace space. A chemical reaction, a rocket launch, a genetic mutation, and an opera all have a starting point in time. This chapter introduces some ontologies that have broad applications. These include the ORG ontology for describing organization, the Event ontology which is concerned with temporal information, the Provenance ontology which is used to ascribe attribution related to the creation of an intellectual product, two location ontologies, an ontology for defining thesauri, and a pair of ontologies that can be used to describe some aspects of various kinds of scientific datasets.
Keywords *FOAF, Organization ontology, Event ontology, PROV-O, Data Catalog, SKOS, OAI-ORE, Geonames, WGS84*

Chapter 16
Library Metadata Ontologies

Abstract

The Semantic Web is not a library. It is not even a catalog, per se. It is a way of representing, sharing and linking with basic statements of fact about things. The granularity of the semantic Web is not at the level at which libraries have been traditionally concerned with information. Libraries collected and described intellectual products that consist of some form of narrative, a book, a journal, a research paper, a movie, an album, or a song. More recently libraries have begun to assist scientists in cataloging their datasets. But the cataloging of individual facts was left to Webster's, encyclopedias, survey papers and the like. And yet it makes a great deal of sense to take the fined grained descriptive data that libraries have meticulously crafted about these various objects, and generate semantic Web data from it, because this represents knowledge. This chapter takes a look at the issues and challenges of publishing library metadata as semantic Web data. It briefly reviews several applicable ontologies that have been used by various projects to create triples about bibliographic metadata and subject and author authority data. It also looks at a few Linked Open Data efforts that have been undertaken by libraries.
Keywords *MARC, VIAF, Dublin Core, BIBO, RDA, Schema.org, BIBFRAME, DAMS, MODS, MADS*

DRAFT

Chapter 17
Time

Abstract
Time requires special consideration in computing. Business systems that use relational databases must stay synchronized and use consistent ways of representing a point in time, an interval of time that has a start and endpoint, and for indicating that something holds true to the present time, or that a start or endpoint is undefined. Comparing points in time is relatively straight forward, but intervals are a different matter entirely. Allen's Temporal Intervals defines the various ways that time intervals can be related to one another, such as meets, overlaps with, contains, after, etc. This model enabled reasoning about time intervals. Graphs are often used to model complex systems, and complex systems are usually dynamic systems that change over time. There are various ways of handling this issue, but modeling time in graphs remains an active area of research. Likewise, time is treated in a variety of ways in the semantic Web. Sometimes it is just an attribute of a thing, sometimes it is the thing of interest and everything else is a characteristic of the event. This chapter looks at the challenges of representing and reasoning about time. It describes some ways that graph models have incorporated time data. It also provides a brief survey of various temporal ontologies used in the semantic Web.

Keywords *ISO 8601, RFC3339, Network Time Protocol, Allen's Temporal Intervals, timepoints, burstiness, reification, Observation ontology, Geologic Timescale ontology, Time ontology*

Chapter 18
Drawing and Serializing Graphs

Abstract
Graphs tend to be inherently complex and correspondingly challenging to represent visually. Sometimes the process of analyzing a graph involves an ad-hoc pipeline of tools and techniques. This necessitates that there be a way to represent the data that consitutes a graph in a standard format suitable for interchange between tools. As with RDF, there are a number of serialization formats for graph data. In this chapter, we will explore two topics. The first the general task of representing the data that describes nodes and edges without regard to any specific layout. Serializations may differ in syntax, but in essence they are just lists of nodes and some means to indicate whether, and in what direction, an edge exists between them. XML and JSON figure prominently among graph serialization formats. The remainder of this chapter is concerned with graph drawing and graph visualization tools. Graph serializations encode the model and its data. By itself, this data is incomprehensible to a human. But a human can make sense of a visualization of a graph. A graph can be drawn in many ways: a lattice, hierarchically, a random tangle, spread out according to rules that mimic gravity or magnetic attraction, fanned out in a circle, or represented in a 3-d model that you could spin and fly around in. Interactivity, graph analytics, and the ability to filter nodes and edges play an important role in comprehending a visualization of a graph.

Keywords *adjacency matrix, graph serialization, GraphML, XGMML, GEXF, GDF, GraphSON, graph drawing, layout algorithms, Cytoscape, Gephi, GUESS, D3*

DRAFT

Chapter 19
Graph Analytics Techniques

Abstract

There are many ways to analyze a graph. There are graph wide metrics to quantify attributes of a graph such as diameter, density, etc. You can compare your instance graph with a comparable regular, small world or random graph. The shape and size of a graph may tell you some interesting things about that graph. Finer grained characteristics that are also useful consider connectivity of nodes in the graph, characteristics of paths, and the presence and size of clusters within the graph. This chapter provides a conceptual overview of analytics that fall into those three categories. It describes several node-based metrics such as degree and betweenness centrality, which can be used to learn more about a given node or about the distribution of degrees of connectivity within a graph. Path metrics can be used to calculate the shortest path between nodes, the average path length, or to look for special kinds of paths such as cycles in directed graphs. Clusters are groupings of nodes that have a special relationship to one another as compared to the graph as a whole, revealed by the relatively interconnectedness of nodes that fall within the cluster as compared to those which do not.

Keywords *node metrics, degree, degree centrality, betweenness centrality, path analysis, path traversal, cycle, cluster, partition, clique, motif, diameter*

Chapter 20
Graph Analytics Software Libraries

Abstract

Graph analytics is a burgeoning field that leverages the combined efforts of researchers who model and study graph models of a variety of different systems and processes. Activities include characterizing a graph's overall structure, determining how information flows through a network, extracting subgraphs, finding clusters and patterns of connectivity, quantifying graph wide metrics, determining how similar a graph model is to a reference graph type such as a small world or scale free graph. These are things that tend not to readily lend themselves to visual inspection of a graph. In this chapter we will revie some of the more common analytic techniques that are broadly applicable to many different types of graphs. Then we will look at several opensource graph analytic software libraries written in Java and Python. We close the chapter with a brief overview of graph edit distance and its applications.

Keywords *consciousness, connections, paths, distributions, clustering, Jung, JGraphT, NetworkX, Graph Edit Distance*

Chapter 21
Semantic Repositories and How to Use Them

Abstract

Triples can accumulate rapidly regardless of the data source, because the process of reducing knowledge into collections of interrelated triples results in a lot of data. You can publish your triples directly to the Web in a file containing RDF/XML or some other serialization of your data. But the you are left with various challenges such as how to track of all your item URIs, all of the triples triples you've generated, much less perform any sort of transactions or searches on this content. So as with any large collection of structured data, it makes sense to use some kind of database. In this chapter we will look at a special type of database designed for RDF, called a triplestore. We'll look at some of

DRAFT

the core functionality provided by all triplestores, including repository management, searching, and transactions. We also explore several software APIs that can be used to write semantic Web applications that can query and add data to triplestores.
Keywords VIVO, s*emantic repository, triplestore, API, OpenRDF Sesame, Mulgara, Jena*

Chapter 22
Graph Databases and How to Use Them

Abstract

Graph models tend to be quite large when they are populated with node and edge data describing a real system, community or process. This data can be stored on a local filesystem and loaded as needed by analysis tools, but eventually the file becomes so large that loading and processing time required makes this impractical. Fortunately there are databases that natively store data as graphs. These are general purpose graph databases which are able to accommodate arbitrary graph data, unlike specialized graph databases like triplestores. In this chapter we will look at at several graph databases along with some of their core features, their query and path traversal languages, and their administrative and query interfaces. Finally we will look at some graph database APIs, which enable programmers to write graph-based software applications that can query, traverse, analyze and manipulate graph data.
Keywords *graph database, graph computing, Gremlin, Titan, Blueprints, Cypher, Neo4J, Helios, TinkerPop*

Chapter 23
Case Studies

DRAFT