Methods and Applications for Detecting Structure in Complex Networks

by

Elizabeth A. Leicht

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Physics)
in the University of Michigan
2008

Doctoral Committee:

Professor Mark E. Newman, Chair
Professor Dante E. Amidei
Professor Leonard M. Sander
Assistant Professor Lada A. Adamic
Assistant Professor Gavin S. Clarkson

To my parents

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor Mark Newman, for his insights and guidance over the past several years. His pursuit of research excellence is truly inspirational. Mark has always challenged me to do my best work and I know I have learned a great deal from him. Also, I would like to thank Professor Gavin Clarkson for his unique perspective on my research and willingness to collaborate.

I would like to thank Beth Percha, Gourab Ghoshal, and Brian Karrer for many useful discussions as well as for being excellent officemates. Many thanks go to Kevin McGrath; he has always been willing to share his vast knowledge of computers and programming with me. Additionally, I would like to thank Professor Jean Krisch for all of her wisdom.

I thank Michael Busha for his energy, his love, and his optimism which inspire me and keep me going. Finally, I want to thank my parents whose years of love and support have allowed me to reach this point.

This work uses data from Add Health, a program project designed by J. Richard Udry, Peter S. Bearman, and Kathleen Mullan Harris, and funded by a grant P01-HD31921 from the National Institute of Child Health and Human Development, with cooperative funding from 17 other agencies. Special acknowledgment is due Ronald R. Rindfuss and Barbara Entwisle for assistance in the original design. Persons interested in obtaining data files from Add Health should contact Add Health, Carolina Population Center, 123 W. Franklin Street, Chapel Hill, NC 27516–2524 (addhealth@unc.edu).

CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER 1

Introduction

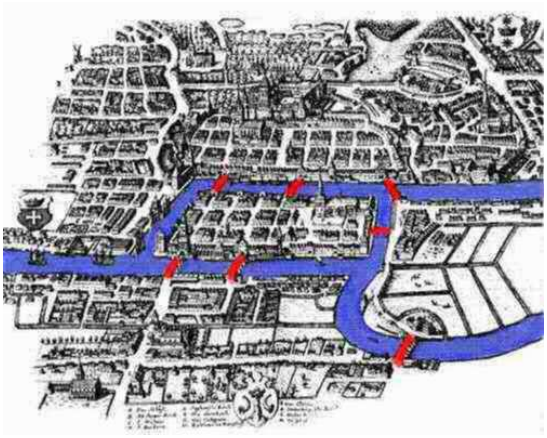
1.1 Introduction

Networks have recently attracted considerable attention from physicists, and a significant body of research has established networks as a basis for the mathematical representation of a wide range of complex systems. Systems of human social interaction, as well as a variety of other informational, biological, and technological systems, have all been studied as networks. Due to their simple and flexible nature, networks are capable of serving as a basis for models of an extremely large range of seemingly unrelated systems.

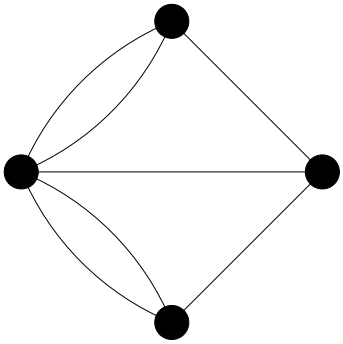
1.1.1 Origins

In its most basic form a **network** is a set of objects we term **vertices**, also known as "nodes" (in math and computer science) or "actors" (in the social sciences). Pairs of vertices are connected via **edges**, also known as "links" or "ties," which represent real relationships between the vertex pairs.

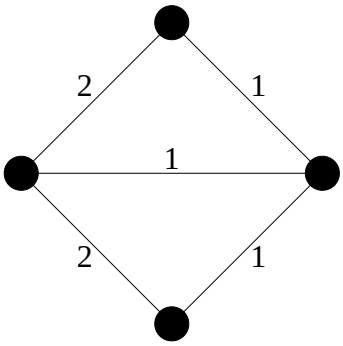
The study of networks is rooted in the field of mathematical graph theory, a fundamental area of study in discrete mathematics. In some networks literature "graph" is used as a synonym of "network." The first proof in the field of graph theory is believed by some people to be the solution to the Königsberg Bridge Problem, written by Leonhard Euler in 1735. The city of Königsberg, Prussia (now Kaliningrad, Russia) is divided by a river which splits into two branches around two islands.



(a)



(b)



(c)

Figure 1.1. The Königsberg Bridge Problem is considered by some to be the first proof written in the field of graph theory. A visualization of the problem (a) as a standard map (the river is highlighted in blue and the bridges are highlighted in pink), (b) as a network where the vertices represent the land masses and the edges represent the bridges, and (c) as a network where the vertices represent the land masses and the edges are weighted to represent the number of bridges connecting the land masses.

Figure 1.1(a) is a map of the city which spans the four land masses with the river highlighted in blue and the bridges that connect the four land masses highlighted in pink. The question arose amongst the residents of the city as to whether it was possible to walk a route through the city crossing each bridge once, and only once. We will not provide the details of Euler's solution to the bridge-crossing problem in this dissertation, but suffice it to say, he was able to mathematically prove that it was impossible to find a path that crossed every bridge once and no bridge more than once.

In Fig. 1.1(b) we represent the Königsberg Bridge Problem as a network. In the figure, each land mass is represented by a vertex and each bridge, a physical connection between two land masses, as an edge. This example, in fact, also introduces one type of embellishment for an edge, the **multiedge**. In a network, edges need not all represent the same strength of relation between vertices. With the network of bridges in Königsberg, there are two instances when a pair of land masses are connected by more than one bridge. Figure 1.1(b) shows the network with these multiedges present, but another way to visualize the network is to give all edges weights. **Weighted edges** replace multiedges in Fig. 1.1(c). Note, unlike multiedges, weighted edges are not required to have integer values.

While the connection between mathematical graph theory and networks is strong, the two fields are distinct. Network theory can be viewed as having a different "flavor" from graph theory. Graph theory has more emphasis on artificial or random graphs (see Section 1.3.1) while networks research is directed more towards the study of real networks found in the world. In addition to mathematical graph theory, networks research also counts among its influences the longstanding work of a number of sociologists on social networks.

Many credit Jacob Moreno with the first use of networks with points (vertices) and lines (edges) to model systems of social interaction. An article published in the

New York Times on April 3, 1933 under the headline "Colored Lines Show Likes and Dislikes of Individual and of Groups" reported on a presentation given by Moreno on social interactions among the residents at the New York State Training School for Girls. This article is often cited as Moreno's introduction of what he termed a "sociogram," but what we clearly recognize today as a visualization of a social network.

Moreno believed social configurations had observable structure and that by drawing pictures of the social interactions of individuals in a group, a researcher could see the structure and understand the impact of said structure on the whole collection of individuals [83]. His book, *Who Shall Survive?: A New Approach to the Problems of Human Interrelations*, published in 1934, contains many of his so-called sociograms, including several illustrating the patterns of friendship among school children. We highlight one of these networks, a network of friendships between 14 boys and 18 girls in a seventh-grade class. To gather the data, the children were asked which other children they would most want seated near them [61]. This method of gathering social network data allows for another variation on the basic concept of network edges, **directed edges**.

In the Königsberg Bridge example the edges are **undirected** because we assume the bridges can be crossed in either direction. We call a network with only undirected edges an **undirected network**. In contrast, in Moreno's social network, one child may pick another child to be seated near them, but the second child may not reciprocate. This scenario allows edges to have direction. A directed edge results from a relationship between two vertices which is somehow unequal. In a network diagram we represent directed edges as arrows instead of lines using a convention to determine the direction of the arrow. In this example, the tail of the arrow is at the choosing child while the head of the arrow is at the chosen child. The directed edges of this social network are shown in Fig. 1.2.

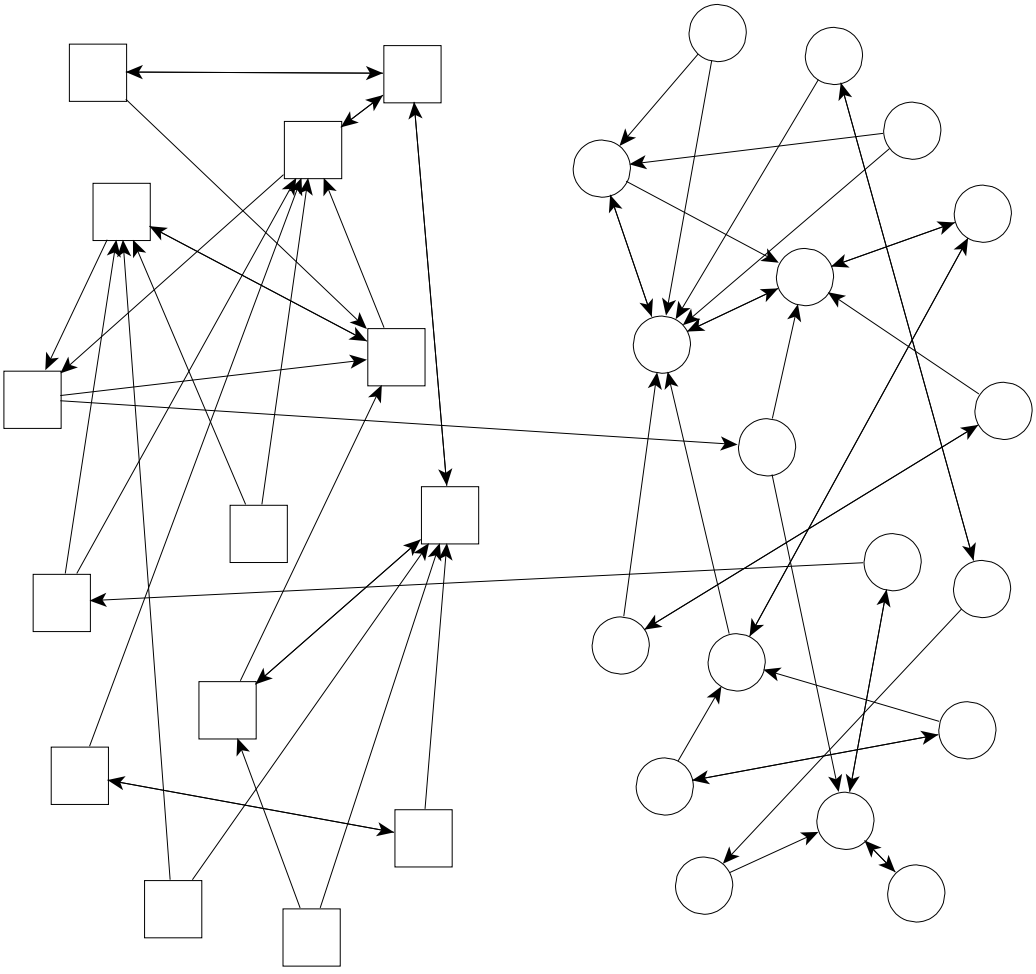


Figure 1.2. In this illustration of the social network formed by 14 boys (squares) and 18 girls (circles) in a seventh-grade class the edges are directed. A directed edge runs from one student (tail) to a second student (head) if the first student chose the second student as someone whom they would want seated near them. This figure is adapted from the work published by Moreno [61].

While a network is basically a collection of vertices and edges, variations and embellishments such as weighted or directed edges are possible. Another edge variation not depicted in either network example is the concept of **self-edges** or **self-loops**. A self-edge occurs when the two endpoints of an edge are at the same vertex. Self-edges do not exist in all types of networks, but can be notable when they occur in real networks.

This dissertation will present many networks representing a variety of systems. While the networks will represent very disparate systems, they are all built from the same fundamental building blocks presented in this section.

1.1.2 Interest to physicists

Many other researchers in the social sciences have followed the work of Moreno in their studies of social networks. Work on networks has also been pursued in other disciplines, especially in the information sciences where there has been a great deal of work on networks of academic papers. Such networks are called **citation networks**; the vertices are the academic papers themselves and directed edges represent the citation of one paper by another. We will discuss citation networks at greater length later in this chapter, as well as in Chapter 5.

However, many suggest that a true paradigm shift regarding research on networks occurred during the 1990's. The rise of the Internet, World Wide Web, and online databases made large network data sets of thousands or millions of vertices easily available to researchers for the first time. An example of such a large network is shown in Fig. 1.3, which depicts a map of the Internet on November 22, 2003 created by the OPTE project. The vertices represent servers and the edges represent physical optical fiber cable connections between pairs of servers. The colors in the figure were used to provide a sense the physical locations of servers based on IP address. For example, the areas colored in light blue are identified as being in Asia/the Pacific

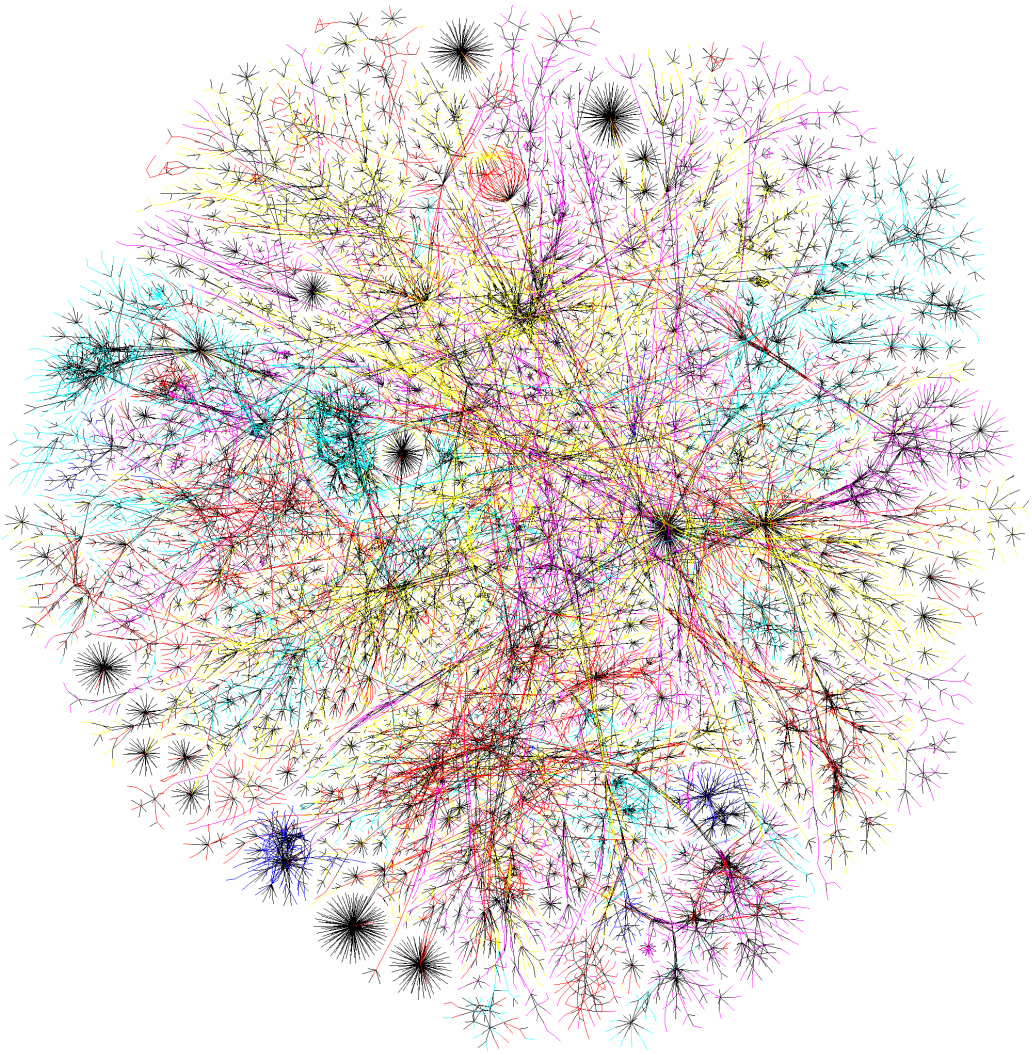


Figure 1.3. A visualization of the the Internet on November 22, 2003. The vertices represent servers and the edges represent physical optical fiber cable connections between servers. Some network edges were not included in this visualization for aesthetic reasons. This image was created by the OPTE project and was distributed under the Creative Commons License.

while North America is represented by the sections in yellow and Europe by the sections in pink.

We immediately notice the comparative difficulty posed by a large network such as the Internet in Fig. 1.3 as compared with the small social network of Moreno in Fig. 1.2. In the small network, drawing the vertices and edges on a sheet of paper was an excellent tool for understanding the structure of the network. In Moreno's network it is easy to see the separation of boys and girls or to observe the vertices representing very popular children. When considering small networks, our eyes are very well adapted to identifying patterns of connection among vertices that form the overarching network structure. However, our capacity to detect structure in a network breaks down when we consider large systems. Yes, we can see some semblance of structure in the map of the Internet, but not in the way we can for Moreno's small social network. Instead, we require mathematical and statistical methods to find structure.

This shift towards studying large-scale properties of networks was partially responsible for the increased interest of physicists in the study of networks. When we discuss social, biological, ecological, or informational networks, one might be curious as to why physicists participate in such research. An article by Philip Ball insists that the interest of physicists in networks is actually the completion of a circle begun in the 19th century when the work on social statistics influenced the development of statistical physics. The statistical physicists of the time were able to "abandon a strict Newtonian determinism and instead to trust a 'law of large numbers' " as they confronted systems with innumerable particles, each of which possessed individual behavior that was beyond their understanding [4]. Thus, physicists working with networks have an interest not in the behavior of individual vertices, but in the system: the general structure of networks and the laws which govern that structure. The level of abstraction can be seen to go even further with physicists who are not

limiting themselves to the study of only one variety of networks, such as social networks or information networks. Instead, these physicists are interested in analyzing the general properties and laws of all networks, dealing in specifics when necessary, but staying as abstract as possible.

1.1.3 Overview of the chapter

We began this chapter by introducing some basic concepts regarding networks and the origins of networks research. In the next two sections of the chapter we will provide a review of existing research concerning network structure. In Section 1.2 we will outline key measures of network structure that have been developed and that serve as building blocks for our own research. In Section 1.3 we present the concept of random graphs, the standard null models used to understand the significance of the structure detected in real networks. Finally, we conclude the chapter with Section 1.4, where we will outline the content of the remaining chapters of this dissertation.

1.2 Standard measures of network structure

In the previous section, we introduced the most basic concepts regarding networks and motivated the need for mathematically-based analysis procedures to replace the human eye in detecting the hidden structure of large networks. We now present some characteristics of this structure that have been identified in many types of complex networks. In addition, we introduce a series of methods derived by various authors to measure these aspects of structure.

1.2.1 Adjacency matrix

We have already established vertices and edges with their various embellishments as the basic building blocks of networks. Now we can begin to put them in a mathematical framework. We label the number of vertices in a network as n and represent a network mathematically as an n by n matrix, the **adjacency matrix**. The entries

in the adjacency matrix $\mathbf{A}$ are,

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge from vertex } j \text{ to vertex } i \\ 0, & \text{otherwise} \end{cases} \tag{1.1}$$

assuming a network with unweighted edges. To represent a weighted network we let the element A_{ij} equal the weight of the edge from vertex j to vertex i. Obviously in an undirected network the adjacency matrix is symmetric: if there is an edge from j to i there is also an edge from i to j. In a directed network, the adjacency matrix is usually not symmetric.

1.2.2 Paths in a network

While edges are one of the basic building block of networks, we are sometimes interested in more than just direct connections between vertices. A **path of length** l is an alternating sequence of vertices and edges starting and ending with vertices and traversing a portion of a network. For example, in a social network where an edge represents friendship between two individuals, we find the friends of friends of an individual i by looking for all vertices connected to i via a path of length $l = 2$.

In terms of the adjacency matrix we find the number of paths of length $l = 2$ from a vertex j to a vertex i through a vertex k by calculating the value of $A_{ik}A_{kj}$. To count the number of paths from j to i through any vertex, we sum the previous expression over all k,

$$\text{number of paths of length two from } j \text{ to } i = \sum_{k=1}^{n} A_{ik}A_{kj}. \tag{1.2}$$

We recognize this as matrix multiplication and rewrite the equation as

$$\text{number of paths of length two from } j \text{ to } i = \left[\mathbf{A}^2\right]_{ij}. \tag{1.3}$$

That is, the number of paths from j to i is the ij-th element of the square of the adjacency matrix.

We could easily write a proof by induction to prove that the total number of paths of any length l between two vertices is calculated in a similar manner. Here, we omit the proof, but give the resultant equation,

$$\text{number of paths of length } l \text{ from } j \text{ to } i = \left[\mathbf{A}^l\right]_{ij}. \tag{1.4}$$

We must note that the paths described above differ from the graph theoretical interpretation of paths. In particular, we can only easily calculate self-intersecting paths using the adjacency matrix multiplication method. Calculating non-self-intersecting paths is a much more complicated undertaking and one we do not tackle in this dissertation. However, we will return to the concept of self-intersecting paths of various lengths in Chapter 2.

A variation on calculating paths between two vertices in a network is calculating **loops**, that is paths where the starting and ending vertices are identical. Loops of length $l = 2$ are really only interesting in directed networks. In that case, the loop indicates the directed edges between vertices are **reciprocated edges**. Loops of length $l = 3$ are also known as **triangles** since they form triangular structures in networks. They play a role in the calculation of network transitivity, a measure we will address in Section 1.2.4.

1.2.3 Degree, average degree, and degree distribution

In an undirected network, the **degree** of a vertex is equal to the number of edges incident upon that vertex. We represent the degree of a vertex i as k_i and calculate it in a network of n vertices as

$$k_i = \sum_{j=1}^{n} A_{ij}. \tag{1.5}$$

A list of the degree of each vertex in a network is known as the **degree sequence** of the network.

In a directed network, both the **in-degree** and the **out-degree** of a vertex are measured. The in-degree of vertex i, k_i^{in}, is equal to the number of directed edges

pointing from other vertices to vertex i (the sum of directed edge heads at vertex i),

$$k_i^{\text{in}} = \sum_{j=1}^{n} A_{ij}. \tag{1.6}$$

The out-degree of vertex i, k_i^{out}, equals the number of directed edges from vertex i to other vertices (the sum of all directed edge tails at vertex i),

$$k_i^{\text{out}} = \sum_{j=1}^{n} A_{ji}. \tag{1.7}$$

Of course, both an **in-degree sequence** and an **out-degree sequence** exist for a directed network.

The **average degree** of an undirected network is the sum of the degree of each vertex in the network, $\sum_{i=1}^{n} k_i$, divided by the number of vertices in the network. We calculate $\sum_{i=1}^{n} k_i$ by remembering that each edge has two endpoints, and that the sum over vertex degree equals the total number of edge endpoints. Thus, the sum over vertex degree equals twice the total number of edges in the network, m, making the average degree,

$$\langle k \rangle = \frac{\sum_{i=1}^{n} k_i}{n} = \frac{2m}{n}. \tag{1.8}$$

In a directed network, the total number of edge heads, $\sum_{i=1}^{n} k_i^{\text{in}}$, equals the total number of tails, $\sum_{i=1}^{n} k_i^{\text{out}}$, and also equals the total number of edges, m. Therefore, in a directed network **average in-degree** equals **average out-degree**, or

$$\langle k^{\text{in}} \rangle = \frac{\sum_{i=1}^{n} k_i^{\text{in}}}{n} = \frac{m}{n} = \frac{\sum_{i=1}^{n} k_i^{\text{out}}}{n} = \langle k^{\text{out}} \rangle. \tag{1.9}$$

Since measuring the degree of each vertex is straightforward, it is unsurprising that one basic way to characterize network topology is by the **degree distribution**, $\{p_k\}$. We define p_k to be the fraction of vertices with degree k in a particular network. That is, if n_k equals the number of vertices with degree k in a network of size n, $p_k = \frac{n_k}{n}$. In directed networks, the **in-degree distribution** and the **out-degree distribution** are considered separately.

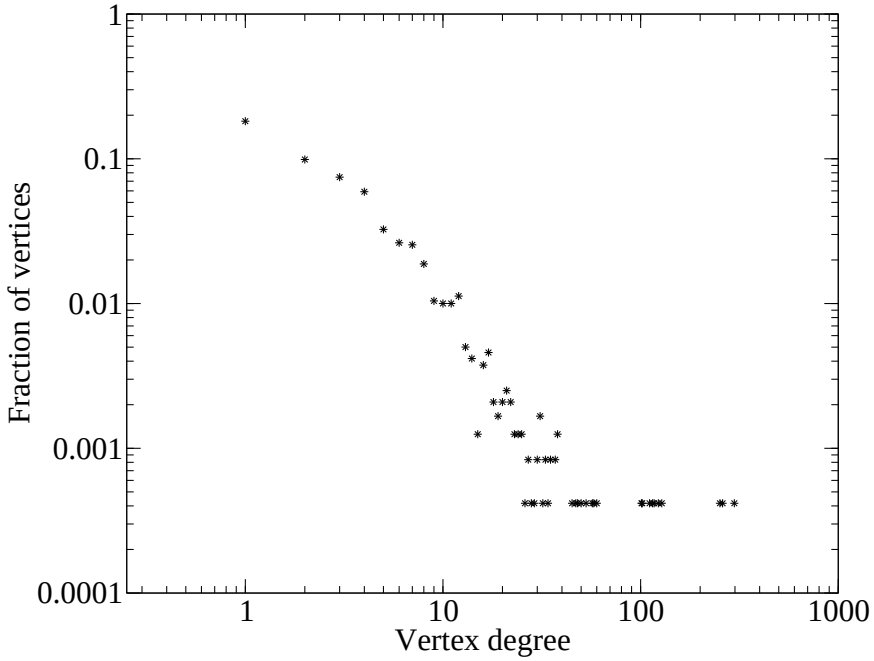
A significant body of research has been devoted to measuring the degree distributions of real networks. In the 1960's and 1970's the physicist-turned-science-historian Derek de Solla Price may have been the first to present one key type of degree distribution, now widely known as the hallmark of a scale-free network [72, 73]. Price studied scientific citation networks in which the vertices represent papers published in scholarly scientific journals and the edges represent the citation of one paper in the text of another paper.

We can illustrate Price's work by recreating one of his key plots with our own citation data. Our citation data comes from papers published to the arXiv in the high-energy theory (hep-th) section. The arXiv is an electronic archive maintained by Cornell University Library as a repository for papers in physics, mathematics, computer science, and others. Papers published to the arXiv may be pre-prints submitted before an article is published in a scholarly journal or versions of papers appearing in scholarly journals. The data was made available by the KDD Cup and we examine the in-degree distribution for all papers published in the hep-th section of the arXiv in 1994 in Fig. 1.4(a). Just as Price found, the in-degree distribution is not random or regular, but is highly right skewed with a long tail towards large in-degree. Price identified the highly skewed degree distribution in his data as a power-law distribution, where

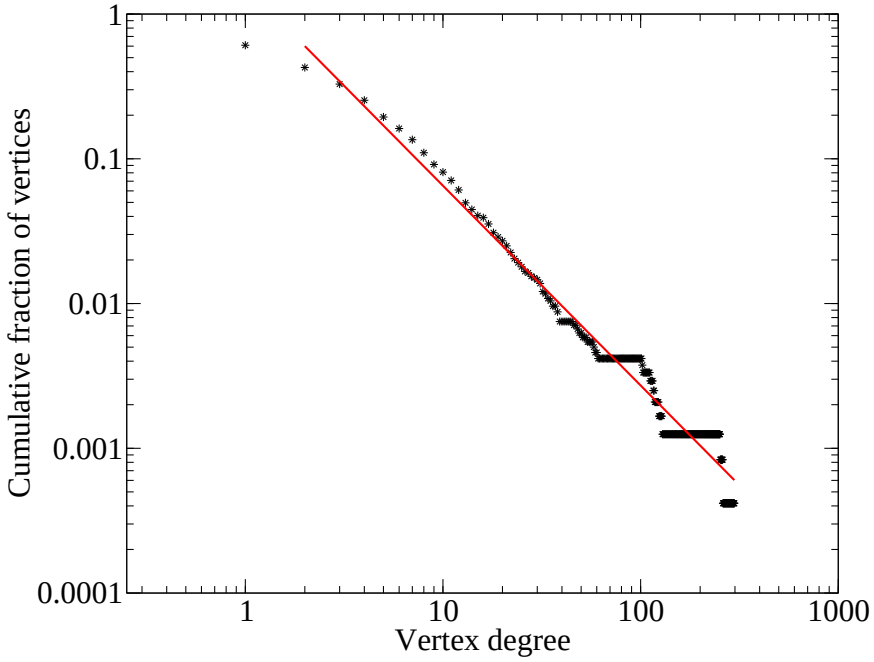
$$p_k \propto k^{-\alpha}. \tag{1.10}$$

Power-law degree distributions are now synonymous with scale-free networks, since power-laws have the property of possessing the same functional form at all scales. However, the term is somewhat misleading because the scale-free behavior refers only to the degree distribution, and does not necessarily hold for any other aspects of the structure of a network.

Measuring this tail of the degree distribution is tricky. In Fig. 1.4(a) the tail is very noisy, owing to the fact that there are not enough data for reasonable statistics—



(a)



(b)

Figure 1.4. (a) The in-degree distribution and (b) the cumulative in-degree distribution for papers published in 1994 on the arXiv in the hep-th category. The in-degree for each paper was calculated using only citations it received within two years of publication.

a common problem. One method used to circumvent this problem is to plot the cumulative distribution function,

$$P_k = \sum_{k'=k'}^{\infty} p_{k'}, \tag{1.11}$$

where P_k is the probability that a vertex has a degree greater than or equal to k. We then rewrite the power-law distribution as

$$P_k \propto k^{-\gamma}, \tag{1.12}$$

where $\gamma = \alpha - 1$ [11]. A plot of the cumulative degree distribution for the citation data, with a power-law fit is provided in Fig. 1.4(b).

In his original work, Price estimated the value of α to be 2.5 or 3.0 for citation networks [72]. The data plotted in Fig. 1.4 clearly do not fit power-law behavior very well for small k, but do reasonably illustrate a power-law distribution with $\alpha = 2.38$ for large k.

The preferential attachment model originally proposed by Price in 1976 [73] and later revisited by Barabási and Albert [5] offers a solution for how power-law degree distributions emerge in networks. The model simulates network growth by beginning with an initial configuration of vertices and then introduces a new vertex and a new edge (or edges) into the network at each time, t_i, in a series of time steps. If the new edge (or edges) connects the new vertex to an existing vertex, where the existing vertex is chosen with a probability proportional to its degree, the degree distribution is ultimately a power law with an exponent in line with those observed in real networks.

While power-law or scale-free networks have received considerable attention, other types of degree distributions are found in real networks. Other distributions observed in real networks include exponential distributions and power-laws with exponential cutoffs. In Section 1.3.1 we will discuss the degree distribution of random networks and how they compare to real degree distributions.

1.2.4 Transitivity or Clustering

In Section 1.2.2 we introduced the concept of paths that begin and end on the same vertex, loops, and made special mention of loops with $l = 3$, which we termed triangles. Triangles are frequently observed in certain kinds of networks, especially in social networks. The idea of **transitivity** in networks makes use of this observation by suggesting that if there is an edge between two vertices i and j and there is also an edge between vertices j and k, then the probability that there is an edge between i and k is greater than if the path $i - j - k$ did not exist. In other words, the friend of your friend is also likely to be your friend [88]. Transitivity is also known in the literature as clustering, but clustering has also come to have an additional meaning in the context of network structure. We quantify network clustering with the **(global) clustering coefficient**,

$$C = \frac{3 \times \text{number of triangles in the network}}{\frac{1}{2} \times \text{number of paths of length two}}. \tag{1.13}$$

The clustering coefficient measures the probability that a randomly chosen path of length two in a network is actually part of a triangle. The factor of three in the numerator accounts for the contribution each triangle makes to the three different paths of length two and the factor of one-half in the denominator takes care of the double counting of paths of length two. Counting all paths of length two in a network counts both $i - j - k$ and $k - j - i$ as distinct paths.

An alternative to this global measure of clustering is a measure of local clustering put forward by Watts and Strogatz [89]. They proposed a **local clustering coefficient** for every network vertex,

$$C_i = \frac{\text{number of triangles including vertex } i}{\text{number of paths of length two centered on vertex } i}. \tag{1.14}$$

In the language of social networks C_i is the probability that a pair of friends of vertex i are themselves friends. Vertices with degree zero or one cannot be the central vertex

for a path of length two, making C_i undefined. In those cases we take C_i to be zero. This local measure is turned into a global measure by averaging C_i over all vertices in the network,

$$C_{\mathrm{WS}} = \frac{1}{n}\sum_{i=1}^{n} C_i, \tag{1.15}$$

where we denote the clustering coefficient with the subscript WS to indicate it is different from the clustering coefficient in Eq. (1.13).

Both measures of transitivity, or clustering, are excellent examples of how direct observation of real networks led to the creation of a specialized metric. Many social networks were observed to have an overabundance of triangles, and the metric simply defined this observation in a quantifiable way.

1.2.5 Measures of similarity

Interest in the local structure surrounding individual vertices is the basis for measures of **structural equivalence**. Structural equivalence is a general concept, but we can summarize it by saying that two vertices are structurally equivalent if they share many of the same network neighbors. A number of measures of structural equivalence originated in other fields and were subsequently applied to networks. All of the definitions concern vertex network neighbors, so we define Γ_i to be the neighborhood of vertex i in a network, i.e., the set of vertices directly connected to i via an edge.

The measure **cosine similarity** was proposed by Salton in 1983 [81]. Originally a method for measuring the similarity between any two vector quantities, it was reshaped as a network measure and for undirected networks is written as

$$\sigma_{\mathrm{cosine}} = \frac{|\Gamma_i \cap \Gamma_j|}{\sqrt{|\Gamma_i|\,|\Gamma_j|}} = \frac{\sum_{k=1}^{n} A_{ik}A_{jk}}{\sqrt{\sum_{k=1}^{n} A_{ik}^2}\sqrt{\sum_{j=1}^{n} A_{jk}^2}} = \frac{[\mathbf{A}^2]_{ij}}{k_i k_j}. \tag{1.16}$$

A related measure predating cosine similarity by several decades is the **Jaccard index** also known as the **Jaccard similarity coefficient**. It was originally proposed by Jaccard as a statistic for comparing the similarity and diversity of sample sets [38].

It is defined as the intersection of two sample sets divided by their union,

$$\sigma_{\text{Jaccard}} = \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i \cup \Gamma_j|} \tag{1.17}$$

where we've omitted the expression in terms of the adjacency matrix, as that representation is not as compact as cosine similarity.

Euclidean distance is a measure of structural equivalence developed by Burt [15, 88]. For an undirected network, the Euclidean distance between two vertices is written as

$$x_{ij} = \sqrt{|\Gamma_i| + |\Gamma_j| - 2(\Gamma_i \cup \Gamma_j)} = \sqrt{\sum_{k=1,k\neq i,j}^{n} (x_{ik} - x_{jk})^2}. \tag{1.18}$$

Euclidean distance is actually a measure of dissimilarity, as $x_{ij} = 0$ for vertices that have a perfectly similar local structure and increases as the local structure becomes less similar.

For directed networks two specific structural measures, **bibliographic coupling** and **co-citation**, arose from the study of citation networks. Bibliographic coupling was introduced by M. M. Kessler, who claimed that two papers are bibliographically coupled if they both cite the same paper [22, 44]. The strength of the coupling is based on how many papers they cite in common,

$$B_{ij} = \sum_{k=1}^{n} A_{ki} A_{kj} \tag{1.19}$$

or

$$\mathbf{B} = \mathbf{A}^T \mathbf{A}. \tag{1.20}$$

Co-citation is the companion to bibliographic coupling and was independently proposed in 1973 by two information scientists, Irina Marshakova and Henry Small. If two documents are both cited by a third document, the two documents are said to be co-cited [22]. As with bibliographic coupling, the strength of the co-citation of two papers is based on how many total papers cite the two papers in common,

$$C_{ij} = \sum_{k=1}^{n} A_{ik} A_{jk} \tag{1.21}$$

or

$$\mathbf{C} = \mathbf{A}\mathbf{A}^T. \tag{1.22}$$

While the methods in this section may seem very basic, they are network measures with a long history of use in the literature.

1.2.6 Centrality

Historically there has been great interest in identifying the most influential or **central** vertices in a network. Many measures have been proposed to calculate vertex centrality and here we give a short review of some of the more well-known of these measures.

The first and most simple measure is **degree centrality** where we count the number of edges incident on a vertex, the degree of the vertex [75]. In an undirected network, vertices with high degree are considered more central because they are connected to many other vertices. In directed networks such as the World Wide Web or a citation network, both the in-degree and out-degree are measures of centrality.

A more sophisticated measure is **eigenvector centrality** [12]. Here the centrality of a vertex i, x_i, is defined to be proportional to the sum of the centrality of each connected vertex,

$$x_i \propto \sum_{j=1}^{n} A_{ij} x_j. \tag{1.23}$$

This equation is easily rewritten in matrix form and transformed into an eigenvalue problem,

$$\lambda \mathbf{x} = \mathbf{A}\mathbf{x}. \tag{1.24}$$

The Perron–Frobenius theorem tells us that there is only one eigenvector with only non-negative entries, which is the unique eigenvector corresponding to the largest eigenvalue. Thus, vertex centrality is proportional to the corresponding element of the leading eigenvector of the adjacency matrix. Unfortunately, this method fails for directed acyclic networks such as citation networks.

An alternative measure of centrality actually predating eigenvalue centrality was proposed by Katz, a statistician working with social networks. In 1953 Katz proposed a measure of vertex centrality which he used to represent the status of individuals in a social network [42]. In his method, every vertex has some centrality or status score x_i, and the status of an individual is based upon the the status of his friends (not unlike eigenvalue centrality) as well as some inherent individual status,

$$x_i = \sum_{j=1}^{n} \alpha A_{ij} x_j + y_i, \tag{1.25}$$

where $0 < \alpha \leq 1$ is a parameter used to control the amount of status transmitted via a link of friendship between pairs of individuals. The parameter y_i is the inherent status of the individual i. We can think of this as we would an infection process, where people are infected by the status of their friends with probability α. We rewrite Eq. (1.25) in matrix form and solve for $\mathbf{x}$,

$$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{y} \tag{1.26}$$

where $\mathbf{I}$ is the identity matrix. We can expand Eq. (1.26) in a power series in terms of $\mathbf{A}$,

$$(\mathbf{I} - \alpha\mathbf{A})^{-1} = \sum_{r=0}^{\infty} (\alpha A)^r = \mathbf{I} + \alpha\mathbf{A} + \alpha^2\mathbf{A}^2 + \cdots . \tag{1.27}$$

In order to make this series converge, it can be shown that we must pick α such that $\alpha < \lambda_1^{-1}$, where λ_1 is the largest eigenvalue of the adjacency matrix, $\mathbf{A}$.

Brin and Page [14] also developed an algorithm for measuring vertex centrality, known as "PageRank." This measure does not greatly differ from those measures already presented: centrality is again a quantity x_i and the centrality of a vertex is proportional to the sum of the centrality of each connected vertex. However, the measure includes a normalization factor for the out-degree of each connected vertex,

$$x_i = \sum_{j=1}^{n} \alpha \frac{A_{ij}}{k_j^{\text{out}}} x_j + y_i, \tag{1.28}$$

where y_i is the inherent centrality of the vertex and is usually defined to be $\frac{(1-\alpha)}{n}$. This normalization by the out-degree of the connected vertices allows less weight in the centrality score to come from those vertices which point to many other vertices.

An interesting and more sophisticated variant of centrality was proposed by Kleinberg [45] and works very well for directed networks, especially citation networks. The method recognizes that vertices with high in-degree are not the only type of vertex of interest in a network and that vertices with high out-degree are also interesting. Consequently, in this variant, each vertex has two values of centrality known as its **authority score** and its **hub score**, the first derived from the incoming edges and the second from the outgoing edges.

In this view, a **hub** is a vertex that points to many important authorities, for instance a review paper in a citation network, while an **authority** is a vertex pointed to by many important hubs, such as an important or authoritative research article. In the most simple version of the method, the authority score x_i of vertex i is proportional to the sum of the hub scores y_j of the vertices citing it,

$$x_i = \frac{1}{\lambda} \sum_j A_{ij} y_j, \tag{1.29}$$

for some constant λ, while the hub score is proportional to the sum of the authority scores of the vertices it cites,

$$y_i = \frac{1}{\mu} \sum_j A_{ji} x_j. \tag{1.30}$$

In matrix form, these equations are

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{x}, \qquad \mathbf{A}^T \mathbf{x} = \mu \mathbf{y}. \tag{1.31}$$

Or, eliminating either $\mathbf{x}$ or $\mathbf{y}$,

$$\mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda\mu \mathbf{x}, \tag{1.32}$$

$$\mathbf{A}^T \mathbf{A}\mathbf{y} = \lambda\mu \mathbf{y}. \tag{1.33}$$

Thus $\mathbf{x}$ and $\mathbf{y}$ are eigenvectors of the symmetric matrices $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ (the co-citation and bibliographic coupling matrices respectively seen in Section 1.2.5). In Kleinberg's formulation of the problem, one focuses on the leading eigenvector of each of the matrices, although in principle there could be useful information to be gleaned from other eigenvectors.

1.2.7 Community structure

Until this point we have discussed aspects of network structure that mainly concern the network as a whole (degree distributions, global clustering, centrality, etc.) or the very local structure around individual vertices (local clustering, similarity, etc.). Another active area of networks research proposes that many networks display **community structure**, a property where sub-groups of vertices form communities with their own distinct patterns of edge placement.

As it is natural to think of community structure in terms of social networks, we can easily form an illustration of this concept by thinking of a group of people in a social network who have most of their friends within the group and only a few friends outside the group. In fact, the concept of communities is well established in the literature of social analysis [83, 88]. However, the idea of having tightly connected communities is useful beyond the realm of social networks. For example, in information networks such at the World Wide Web or citation networks of academic publications, communities of documents (web pages or academic papers) could all pertain to the same topic, unifying the community in terms of content, and making it an attractive target for identification.

Spectral methods

The task of community detection, or partitioning, is not unique to the study of networks. Computer scientists have long been concerned with the problem of graph partitioning. The need to partition graphs arises in many contexts in computer

science. One example is found in the development of computer algorithms for parallel computing where a number of tasks, n, must be divided across two processors. Many of these n tasks need to communicate with one another, and while communication within a processor is reasonably fast, communication between processors is relatively slow. Thus, it is desirable to minimize the number of communications that must occur between processors.

One well known solution to this problem is **spectral bisection** which divides a network into two communities using the network **Laplacian**. The network Laplacian, $\mathbf{L}$, for a network with n vertices, is an $n \times n$ matrix with elements

$$L_{ij} = \delta_{ij} k_i - A_{ij}, \tag{1.34}$$

where δ_{ij} is the Kronecker delta while k_i and A_{ij} are, of course, a vertex degree and an adjacency list element respectively. From the definition of matrix elements given in Eq. (1.34) it is clear that all rows/columns sum to zero, making the vector $\mathbf{v} = (1, 1, \ldots 1)$ an eigenvector of $\mathbf{L}$ with corresponding eigenvalue $\lambda = 0$. Also, it can be shown that $\mathbf{L}$ is always a positive semi-definite, meaning that $\lambda = 0$ is the smallest eigenvalue of $\mathbf{L}$.

If a network can be divided into c communities, where all edges fall only within communities and no edges connect communities, then we can imagine writing the Laplacian in block diagonal form,

$$\mathbf{L} = \begin{pmatrix} \square & 0 & & 0 \\ 0 & \square & & 0 \\ & & \ddots & \\ 0 & 0 & & \square \end{pmatrix} \tag{1.35}$$

where the $\square$s represent the sub-Laplacian for one of the c communities and the 0s represent matrices of zeros. There would then be c eigenvalues equal to zero.

However, when a network is not perfectly divisible into c communities with edges falling only within communities and not between communities, there is only the single

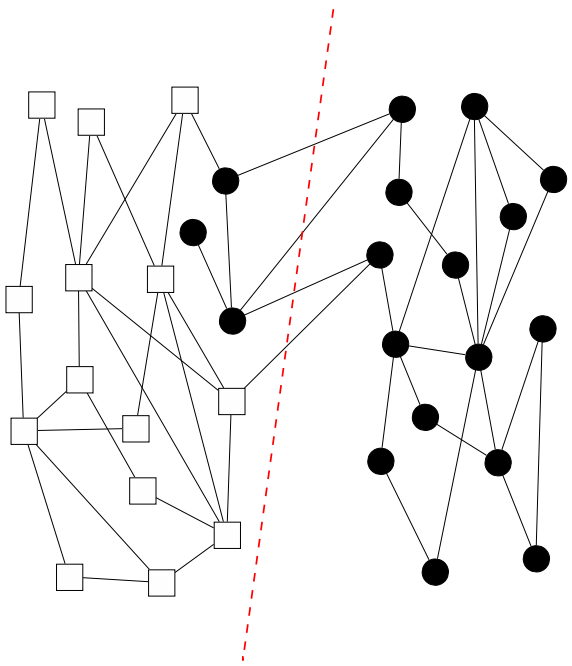


Figure 1.5. The symmetrized and undirected version of Moreno's network of school children divided into two communities via spectral bisection. Again the girls are represented by circles, this time shaded black, while the boys are represented by squares, in white. The dashed line separates the two groups of children.

eigenvalue equal to zero ($\lambda_1 = 0$) and $c - 1$ eigenvalues slightly larger than zero. Thus, for real networks the number of communities can be approximated by finding the number of eigenvalues close to zero [63].

In the special case of a network having only two communities, the work of Fiedler [28, 29] which was later revisited by Pothen *et al.* [71] provides a solution for the actual identification of the vertices in each community. Fielder proposed that the signs of the elements of the eigenvector $\mathbf{v}_2$, corresponding to that second smallest eigenvalue, λ_2, which he called the **algebraic connectivity**, could be used to approximate the splitting of a network into two communities. Here we present an application of this method of spectral bisection to Moreno's social network of school children, which we introduced earlier in this chapter. Note we used the symmetrized version of the network, which ignores edge direction, as the spectral bisection requires an undirected

network. We will revisit the idea of turning directed networks into undirected networks in Chapter 3. In Fig. 1.5 we see a reasonable split of the students into two communities. The split into two groups does not exactly fall along gender lines, as perhaps would would assume after viewing the first illustration of this network in Fig. 1.2. However, we see only three girls included with the fourteen boys in one group and the remaining fifteen girls in the second group.

For division of a network into more than two groups, repeated bisection can be used. However, the accuracy of the partitions determined by repeated bisection is usually unknown. In addition, the point at which to stop sub-dividing the network can be difficult to determine unless the number of communities in the network is known ahead of time [11, 63].

Hierarchical clustering

Sociologists also have a long standing interesting in the process of community detection in networks. A significant part of their research has been directed towards finding community structure in real networks. A widely used technique, known as **hierarchical clustering**, groups vertices into subsets or communities such that the vertices within a community are similar to one another in some sense [11, 63, 83, 88].

There are many methods for calculating the similarity of two vertices and several of these methods were presented in Section 1.2.5. The general technique of hierarchical clustering proceeds in the same way regardless of the particular similarity measure chosen. The process starts with calculating the similarity, x_{ij}, of all pairs of vertices in the network. The method then looks to an empty network with the same number of vertices n as the original real network. Then, starting with the pair of vertices with the greatest similarity in the network, the vertices are joined via an edge in the new network. After that the similarity threshold, α, is gradually decreased and more vertex pairs are joined together. The method can continue until all vertices in the network are connected in one component. However, network communities can

be extracted for any intermediate value of similarity α between which the network is neither empty nor connected as a single component.

More than one system exists for the process of connecting vertices as the new network is built-up. One common method, known as **single linkage**, allows any two vertices i and j to be connected as soon as $\alpha = x_{ij}$, and a third vertex k to be connected in a community with i and j as soon as $\alpha = x_{ik}$ or $\alpha = x_{jk}$. Thus, for a given α it is guaranteed that any pair of vertices in the network with $x_{ij} \geq \alpha$ will be in the same community. However, we are not guaranteed that $x_{ij} \geq \alpha$ for all pairs in a given community.

An alternative means of conducting hierarchical clustering is the method of **complete linkage**. This method also begins with an empty network where each vertex is its own component, and vertices are connected together in order of decreasing similarity. However, the requirement for the formation of communities is far more stringent than in the single linkage method. In this case, multiple connected vertices are in the same component at a given value of α only if all pairs of vertices i and j in the component have similarity $x_{ij} \geq \alpha$. There may exist a vertex external to the community with similarity to a vertex in the community greater than or equal to α, but the vertex is not included in the community if its similarity with any other vertex in the community is less than α.

The entire process is frequently represented as a **dendrogram**, a visualization of the vertices coalescing into communities. We give an example of this process and the resulting dendrograms in Figs. 1.6 and 1.7, again we use the data from Moreno's social network of school children. As a first example we used the hierarchical clustering capabilities in the network analysis program Pajek. The method in fact uses a dissimilarity measure instead of a similarity measure. Two perfectly similar vertices have a dissimilarity value equal to zero. To create the dendrogram, edges are added from vertex pairs with the smallest dissimilarity to the largest dissimilarity

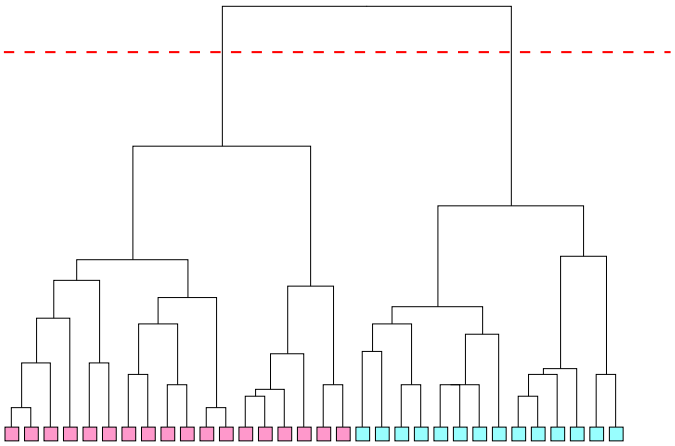


Figure 1.6. The symmetrized and undirected version of Moreno's network of school children divided into communities via hierarchical clustering using Pajek. The vertices are depicted at the bottom of the dendrograms with the boys represented by blue squares and the girls by pink squares. The dashed lines represent the communities as they exist in the network given a certain similarity threshold.

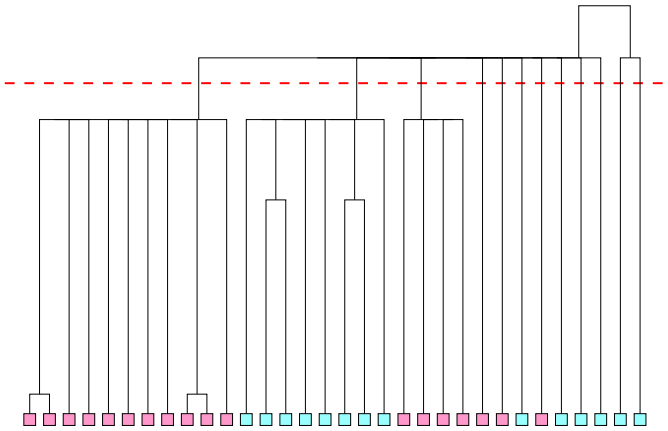
using the single linkage technique.

The dendrogram shown in Fig. 1.6 demonstrates how the network analysis program Pajek performs hierarchical clustering on this social network. With hierarchical clustering there is no built-in way to find the correct number of communities, but we placed a horizontal line in the plot at one given division to show how this hierarchical clustering technique can find the two gender segregated communities.

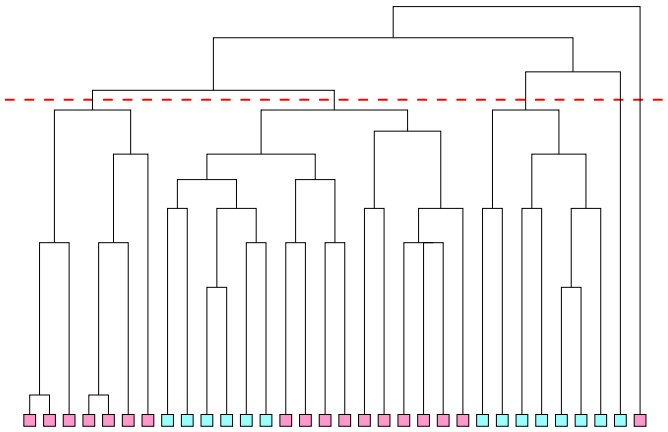
However, with hierarchical clustering methods any measure of vertex similarity or dissimilarity may be used. As a second illustration we use Euclidean distance, which we presented in Section 1.2.5. The two dendrograms shown in Fig 1.7 show hierarchical clustering via single linkage and complete linkage. In this case the method does not perform very well. We place a horizontal line in both of these plots to see how communities can be identified. We notice here that identified communities are more mixed in gender then was seen in method from Fig. 1.6.

Interest from physicists

Physicists have, in the past few years, entered into the quest to find a method to divide networks into communities. At this time numerous methods have been proposed,



(a)



(b)

Figure 1.7. The symmetrized and undirected version of Moreno's network of school children divided into two communities via hierarchical clustering with (a) single linkage and (b) complete linkage. The vertices are depicted at the bottom of the dendrograms with the boys represented by blue squares and the girls by pink squares. The dashed lines represent the communities as they exist in the network given a certain similarity threshold.

including some of our own work which will be presented in subsequent chapters. Many view a notable method proposed by Girvan and Newman as the start of current interest from physicists on the topic.

Girvan and Newman proposed that communities are naturally composed of vertices that are highly connected to each other and less connected to other vertices in the network. They suggested that the communities could be detected by identifying the edges running between the highly connected groups of vertices. The idea can be likened to a problem of information flow on a network. If information were to flow through the network, the structure of the network would force a great deal of the information to flow along edges connecting the communities. The implication is that if the edges with high flow were removed, the network would break up into communities. The means by which they proposed to identify the edges connecting communities was by measuring **edge betweenness**. In short, edge betweenness measures the number of shortest paths that run along a given edge. A shortest path between two vertices is exactly what it seems to be, the path that runs from a vertex j to a vertex i with the fewest number of steps.

The algorithm, which is now known as the Girvan-Newman (GN) algorithm, is composed of the following steps for a network with n vertices:

1. Calculate the edge betweenness of all edges in the network.

2. Remove the edge with the highest betweenness score.

3. Repeat until all the edges in the network are removed, that is the network has n components.

The requirement of recalculating the edge betweenness of all edges in the network after the removal of one edge has been shown to be an important aspect of the method [67].

The division of the network can be visualized with the aid of a dendrogram similar to the one used in the hierarchical clustering method. However, the process for creating the dendrogram is a reversal of hierarchical clustering. In this case the dendrogram is read from the top, the connected network of n vertices, to the bottom, the n components of one vertex each.

As with hierarchical clustering, the method provides no way to determine the correct number of communities. However, Girvan and Newman additionally proposed a measure to quantify the the quality of a particular network division. The **modularity** for the division of a network into c communities is measured by constructing a $c \times c$ symmetric matrix, $\mathbf{e}$, where the element e_{ij} is the fraction of all edges in the original network that connect vertices in group i to vertices in group j. Thus, the fraction of all edges which connect two vertices in the same community is the trace of the matrix $\mathbf{e}$. The fraction of all edges connected to a vertex in group i is $\sum_j e_{ij}$, which can be labeled as a_i. If edges were to fall between vertices in a network independent of the communities to which those vertices belong, then it would be the case that $e_{ij} = a_i a_j$ [67]. Thus, modularity can be measured as,

$$Q = \sum_i (e_{ii} - a_i^2) = \text{Tr } \mathbf{e} - ||\mathbf{e}^2|| \tag{1.36}$$

where $||\mathbf{x}||$ is the sum of the elements of the matrix $\mathbf{x}$. Modularity is a measure of the real number of edges between vertices in the same community minus the expected number of such edges were the network randomly connected without regard to community structure. Consequently, values of Q range between zero and one with values of Q close to zero for networks with little to no community structure and networks with Q close to one having strong community structure. The spectrum of Q values can be calculated as the edges are removed using the method of Girvan and Newman, with local peaks in the value of Q indicating good divisions of the network into communities.

We give an example of the GN algorithm applied to the social network of Moreno

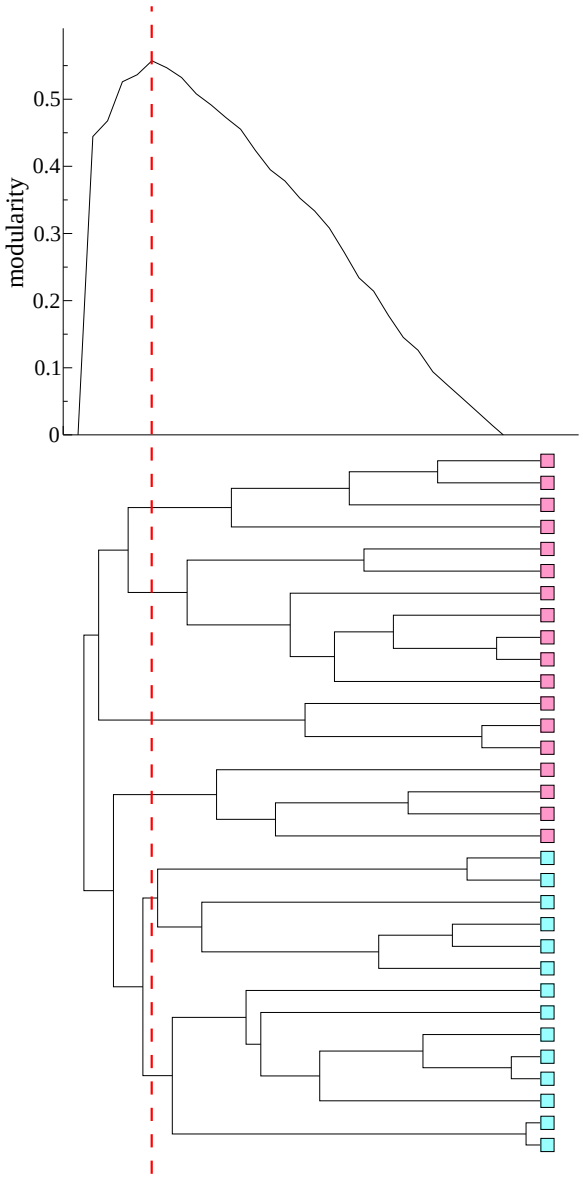


Figure 1.8. The symmetrized and undirected version of Moreno's network of school children divided into two communities via the GN algorithm. The dashed line shows the division of the network that maximizes modularity. The vertices are depicted at the bottom of the dendrogram with the boys represented by blue squares and the girls by pink squares.

in Fig. 1.8. At the top of the figure we plot the modularity of the network as edges are removed and at the bottom we show the dendrogram of communities. The vertical dashed line cuts across the dendrogram at the division achieving maximum modularity. The network is split into five communities, three with only girls and two with only boys.

The work of Girvan and Newman has been followed by many other proposed methods for community detection. Radicchi *et al.*proposed a method also posited on the removal of edges based on betweenness score, but used a different measure of betweenness than the one used by Girvan and Newman [74]. The review article by Danon *et al.* [20] gives an overview of many of the methods.

1.3 Random graphs

The previous section focused on standard methods for detecting structure in real networks. An important topic we did not address is the issue of how we know that the detected structure is significant beyond random chance. In this section we present two types of random graphs: Poisson random graphs and generalized random graphs. These random graphs or networks are used in two contexts, to create artificial or simulated networks and as null models for comparison with real networks. We will return to these models throughout this dissertation.

1.3.1 Poisson random graph

The **Poisson random graph** is a classic network model first proposed by Solomonoff and Rapoport in 1951 [85]. The same model was later and independently proposed by Erdős and Rényi [23, 24, 25]. This method for constructing a network starts with n vertices and connects all possible pairs of vertices with a probability p. This model, known as an **Erdős Rényi (ER) random graph**, disallows multiedges and self-edges and is denoted G_{np}. An alternative way to specify the random graph is to

again start with n vertices, but preselect the number of edges m in the network. The method again starts with n unconnected vertices. Pairs of vertices are then considered in random order and are connected via an edge. Vertex pairs are connected until the network has a total of m edges. This method for constructing a random network again disallows multi-edges and self-edges [11]

We note that a graph created using either of these methods is only one realization of the many statistically possible graphs in the ensemble of all possible graphs created using either the G_{nm} or G_{np} framework. In fact, usually the notation G_{nm} or G_{np} is not used in reference to a single graph, but to the ensemble of all possible graphs using the model.

The degree distribution for an ER random graph is a binomial distribution,

$$p_k = \begin{pmatrix} n-1 \\ k \end{pmatrix} p^k (1-p)^{n-1-k} \tag{1.37}$$

where p_k is the probability that network vertex has degree k. In the equation p^k is the probability of the existence of k edges while $(1-p)^{n-1-k}$ is the probability of the absence of the remaining $n-1-k$ possible edges. The number of possible edges connected to one vertex is $n-1$ because self-edges are disallowed. The leading term, $\begin{pmatrix} n-1 \\ k \end{pmatrix}$, is a combinatorial factor that accounts for the number of ways there are to choose the k endpoints. Now, we also note that in the limit of large system size (large n) if the mean degree is held fixed ($\langle k \rangle = p(n-1) = \text{constant}$) the degree distribution is nicely approximated by a Poisson degree distribution,

$$p_k = \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!}, \tag{1.38}$$

which has led to these networks also being called Poisson random networks [11]. Figure 1.9 is a histogram of the degree distribution for a single realization of a G_{np} model where $n = 1000$ and $p = 0.1$. We also plot the corresponding Poisson distribution and observe the excellent correspondence.

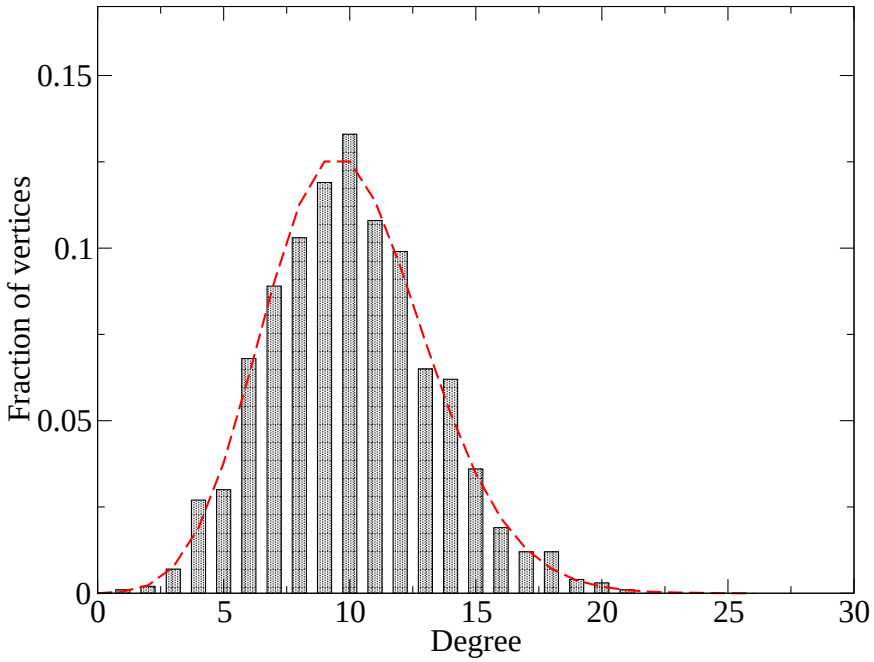


Figure 1.9. The degree distribution for an ER random network where $n = 1000$ and $p = 0.1$, with the real distribution plotted as a bar graph and the Poisson approximation plotted as the dashed line.

Clearly, the degree distribution of such a random network does not reflect the previously described degree distributions detected in real networks, such as power-law degree distributions. We will describe a method of creating random networks with a given degree distribution in the next section. The ER network model does function as a null model for comparison with real networks, and the model can be altered to include some types of networks structure, such as communities (see Chapters 3 and 4).

1.3.2 Generalized random graphs

We noted in the previous section that while ER model can be used to create random networks, the networks produced lack a realistic degree distribution. In Section 1.2.3 we discussed the degree distributions of real networks. Clearly, an improvement to the ER model is to allow for a more realistic degree distribution. The following type of random network has been studied by many authors since the 1970's [9, 58, 59].

However, the description we give stems from the work of Molloy and Reed [58].

The **configuration model** is a method where we sample a random network with a specific degree sequence (see Section 1.2.3) constructed with a prescribed degree distribution, $\{p_k\}$. The prescribed degree distribution is used to construct a degree sequence for the network where the degrees of the vertices, the k_i's (where $i = 1 \dots n$) are independent, identically-distributed random integers drawn from the degree distribution p_k. This is equivalent to giving each vertex k_i stubs, half-edges, or edge endpoints. We then choose two stubs at random and connect them via an edge. We repeat this process until all the stubs have been used to form edges. Again, the configuration model is really defined as the ensemble of all possible graphs produced from a given degree distribution.

1.4 Outline of dissertation

In this chapter we have introduced many concepts regarding network structure. We have motivated the idea that for small networks we can use our eye to peruse a visualization of the network for interesting structural features, but we cannot do so for the largest networks of interest today. Instead we must use specialized methods to measure network structure, some of which have been reviewed in this chapter.

However, we note that not all network examples found in this paper are networks of thousands or millions of vertices. On the contrary, we frequently rely on smaller networks of tens or hundreds of vertices to support our claims about the mathematical techniques we use to detect network structure. With these smaller networks the reader can compare our mathematically-rooted results with the conclusions of the eye.

That said, we set forth in this dissertation new research regarding the detection of structure in real networks. The remainder of this dissertation is divided into two parts which approach the detection of network structure through different means.

The first part, Chapters 2 and 3, follows the pattern set-forth by many of the

methods described in this introductory chapter. That is, we propose to detect a given type of network structure and we present an implementation of a method to complete the task. In Chapter 2 we derive a measure of vertex similarity based upon network structure. The method builds on existing ideas concerning the calculation of vertex similarity, but generalizes and extends the scope to large networks. We then address, in Chapter 3, the detection of communities or modules in the specific class of networks, known as directed networks. The method extends an existing technique for the directed network subset. These two chapters are based on the author's publications [50] and [51] respectively.

In the second part, Chapters 4 and 5, we move away from the specialized methods that have dominated the field of networks research. Instead we propose two methods for network structure detection based on probabilistic techniques. In Chapter 4 we propose a method for detecting network structure that does not require *a priori* knowledge of the type of structure for which we are searching. We base this method on the well known statistical method of the expectation-maximization algorithm. This chapter is based on the author's publication [68]. The work presented in Chapter 5 also uses the framework of the expectation-maximization algorithm, but focuses on detecting changes in networks evolving with time and is based on the author's publication [49].

CHAPTER 2

Vertex similarity in networks

2.1 Introduction

There are many situations concerning real networks in which it would be useful to be able to answer questions such as "How similar are these two vertices?" or "Which other vertices are most similar to this vertex?" Of course, there are many senses in which two vertices can be similar. In the network of the World Wide Web, for instance, in which vertices represent Web pages, two pages might be considered similar if the text appearing on them contains many of the same words. In a social network representing friendships between individuals, two people might be considered similar if they have similar professions, interests, or backgrounds. This chapter considers ways of determining vertex similarity based solely on the structure of a network. Given only the pattern of edge placement in a network, we ask, can we define useful measures that tell us when two vertices are similar? Similarity of this type is sometimes called **structural similarity**, to distinguish it from social similarity, textual similarity, or other similarity types. It is a basic premise of networks research that the structure of a network reflects real information about the involved vertices. Thus, it seems reasonable that meaningful structural similarity measures might exist.

The problem of quantifying the similarity of two vertices in a network is not new. The most common approach taken in previous work has been to focus on so-called structural equivalence (see Section 1.2.5). To briefly review the idea, two vertices are considered structurally equivalent if they share many of the same network

neighbors. For instance, it may be reasonable to conclude that two individuals in a social network have something in common if they share many of the same friends. Let Γ_i be the neighborhood of vertex i in a network, i.e., the set of vertices that are directly connected to i via an edge. We can then represent the number of friends i and j have in common as

$$\sigma_{\text{unnorm}} = |\Gamma_i \cap \Gamma_j|, \tag{2.1}$$

where $|x|$ indicates the cardinality (i.e., number of elements in) of the set x. This count of common friends can represent an unnormalized measure of similarity between two vertices. Section 1.2.5 introduced some ways to normalize this rudimentary measure of similarity. Cosine similarity is an example of such a measure,

$$\sigma_{\text{cosine}} = \frac{|\Gamma_i \cap \Gamma_j|}{\sqrt{|\Gamma_i|\,|\Gamma_j|}}. \tag{2.2}$$

There are, however, many cases in which vertices occupy similar structural positions in networks without having common neighbors. For instance, two store clerks in different towns occupy similar social positions by virtue of their numerous professional interactions with customers. However, it is quite unlikely that they have any customers in common. Considerations of this kind lead us to an extended definition of network similarity known as **regular equivalence**. In this case, vertices are said to be similar if they are connected to other vertices *that are themselves similar*. It is upon this idea that the measures we will develop in this chapter are based.

Regular equivalence is clearly a self-referential concept: one needs to know the similarity of the neighbors of two vertices before one can compute the similarity of the two vertices themselves. It comes as no surprise to learn, therefore, that traditional algorithms for computing regular equivalence have an iterative or recursive nature. Two of the best known such algorithms, REGE and CATREGE [13], proceed by searching for optimal matching between the neighbors of the two vertices. In addition, other authors have also formulated the calculation of regular equivalence in networks

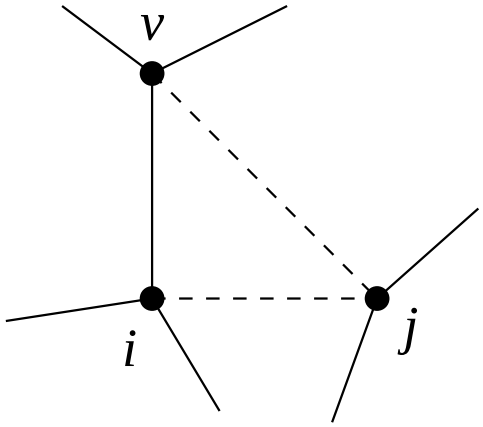


Figure 2.1. A vertex j is similar to vertex i (dashed line) if i has a network neighbor v (solid line) that is itself similar to j.

as a optimization problem [6].

In this chapter we propose a method implemented using different tactics. Our measure of similarity uses a linear algebra framework. The fundamental statement of our approach is that vertices i and j are similar if either of them has a neighbor v that is similar to the other—see Fig. 2.1. Coupled with the additional assumption that vertices are trivially similar to themselves, this gives, as we will see, a sensible and straightforward formulation of the concept of regular equivalence for undirected networks. The method has substantial advantages over other similarity measures: it is global, unlike cosine similarity and related measures, it depends on the whole graph and allows vertices to be similar without sharing neighbors; it has a transparent theoretical rationale, which more complex methods like REGE and CATREGE lack [13]; it avoids the convergence problems that have plagued optimization methods; and it is comparatively fast, since its implementation can take advantage of standard, hardware optimized, linear algebra software.

Some previous authors have also considered similarity measures based on matrix methods [40, 10]. We discuss the differences between our measure and other measures in Section 2.2.3.

The content of this chapter is organized in several sections. We derive our struc-

tural similarity measure in Section 2.2. In Section 2.3, we test the measure on simulated and real-world networks. Finally, in Section 2.4 we give a discussion of our results.

2.2 A measure of similarity

The starting point for our measure of similarity is the assumption that the edges in a network are themselves indicators of similarity between the vertices they connect. Thus, for instance, we assume that two people in a social network are more likely to be connected if they are similar, in some social sense, than if they are dissimilar. The edges of the network provide the raw data from which we will deduce more subtle similarity values, including values for pairs of vertices that are not directly connected.

It is worth noting that it is not always the case that the edges in a network fall between similar vertices. Some networks are said to be **disassortative** [62], meaning that edges preferentially connect vertices that are different in some way. Although the measures derived in this chapter may convey useful information even in those cases, we will for the purposes of argument assume that the networks at which we are looking are not disassortative; rather they are **assortative** and edges tend to connect vertices that are fundamentally similar.

This then leads us immediately to the idea of regular equivalence: a pair of vertices i, j are similar to one another if any pair u, v of their neighbors are similar. In fact, an even simpler one-step expression of the principle is possible: vertex i is similar to j if i has any network neighbor v that is itself similar to j. This idea, illustrated in Fig. 2.1, forms the basis for the measure of similarity developed here. At first glance this definition might appear less satisfactory than the two-step version, having an asymmetry between i and j that the two-step definition lacks. As we will see, however, it makes no difference to the results if we swap vertices i and j: the mathematical expression for the similarity turns out to be the same and hence the definition is in

fact symmetric.

This definition of similarity is clearly recursive and hence we need to provide some starting point for the recursion in order to make the results converge to a useful limit. The starting point we choose is to make each vertex similar to itself, which is natural in most situations. Our definition of similarity will thus have two components: the neighbor term of the previous paragraph and the self-similarity.

Thus, our first guess at the form of the similarity (we will improve it later) is to write the similarity S_{ij} of vertex i to vertex j as

$$S_{ij} = \phi \sum_v A_{iv} S_{vj} + \psi \delta_{ij}, \tag{2.3}$$

where δ_{ij} is the Kronecker delta function and A_{iv} is an element of the adjacency matrix of a symmetric network as defined in Section 1.2.1. Additionally, ϕ and ψ are free parameters whose values control the balance between the two components of the similarity.

Considering S_{ij} to be the ij element of a similarity matrix $\mathbf{S}$, we can write Eq. (2.3) in matrix form as

$$\mathbf{S} = \phi \mathbf{A}\mathbf{S} + \psi \mathbf{I}, \tag{2.4}$$

where $\mathbf{I}$ is the identity matrix. If rearranged, this equation can also be written as $\mathbf{S} = \psi[\mathbf{I} - \phi\mathbf{A}]^{-1}$. As we see, the parameter ψ merely contributes an overall multiplicative factor to our similarity. Since in essentially all cases we will be concerned not with the absolute magnitude of the similarity, but only with the relative similarity of different pairs of vertices, we can safely set $\psi = 1$, eliminating one of our free parameters, and giving

$$\mathbf{S} = [\mathbf{I} - \phi\mathbf{A}]^{-1}. \tag{2.5}$$

This expression for similarity bears a close relation to the matrix-based centrality measure of Katz [42] (see Section 1.2.6). In fact, the Katz centrality of a vertex is equal to the sum of that vertex's similarities to every other vertex. This is a natural

concept: a vertex is prominent in a network if it is closely allied with many other vertices.

We can also consider the similarity of i and j when j has a neighbor v that is similar to i. In that case,

$$S_{ij} = \phi \sum_v S_{iv} A_{vj} + \psi \delta_{ij}. \tag{2.6}$$

It is trivial to show that this leads to precisely the same expression for similarity as in Eq. (2.5) and we can set $\psi = 1$ as before. Thus, as we claimed above, our definition provides only one similarity value for any pair of vertices, given by the symmetric matrix $\mathbf{S}$ of Eq. (2.5).

The remaining parameter ϕ in Eq. (2.5) is still free. To shed light on the appropriate value for this parameter, let us expand the similarity as a power series,

$$\mathbf{S} = \mathbf{I} + \phi \mathbf{A} + \phi^2 \mathbf{A^2} + \ldots. \tag{2.7}$$

Noting that the element $\left[\mathbf{A}^l\right]_{ij}$ is equal to the number of (possibly self-intersecting) network paths of length l from i to j, this equation gives us an alternative, term-by-term interpretation of our similarity measure. The first term says that a vertex is identically similar to itself. The second term says that vertices that are immediate neighbors of one another have similarity ϕ. The third term says that vertices that are distance two apart on the network have similarity ϕ^2 and so forth.

However, also notice that vertex pairs having many paths of a given length are considered more similar than those pairs that have few. The similarity of vertices i and j acquires a contribution ϕ^2 for *every* path of length two from i to j. We note, however, that some pairs of vertices are *expected* to have one or even many such paths between them: vertices with very high degree, for instance, will almost certainly have one or several paths of length two connecting them, even if connections between vertices are made at random. Thus, simple counts of number of paths are

not enough to establish similarity. We need to know when a pair of vertices has more paths of a given length between them than we would expect by chance.

This suggests a strategy for choosing ϕ. We will normalize each term in our series by dividing the number of paths of length l (given by the power of the adjacency matrix) by the *expected* number of such paths, were vertices in the network connected at random. Then each term will be greater or less than unity by a factor representing the extent to which the corresponding vertices have more or fewer paths of the appropriate length than would be expected by chance. In fact, there is no single choice of the parameter ϕ that will simultaneously achieve this normalization for every term in the series. Yet, if we allow a slight modification of Eq. (2.5), then there is a choice for ϕ that approximately achieves normalization for every term and achieves it exactly in the asymptotic limit of high terms in the series.

2.2.1 Expected number of paths

Let us generalize the series, Eq. (2.7), to allow an independent coefficient for each term and for each vertex pair i, j:

$$S_{ij} = \sum_{l=0}^{\infty} C_l^{ij} \left[\mathbf{A}^l\right]_{ij}. \tag{2.8}$$

Let us also choose (for the moment) each coefficient to equal one over the expected number of paths of the corresponding length between the same pair of vertices on a network with the same degree sequence as the network under consideration, but in which the vertices are otherwise randomly connected. Such a network is called a configuration model and was previously introduced in Section 1.3.2.

The zeroth-order coefficient C_0^{ij} is trivial: there are no paths of length zero between vertices i and j unless $i = j$. So $C_0^{ij} = \delta_{ij}$. The first-order coefficient C_1^{ij} is more interesting. If vertices i and j have degrees k_i and k_j respectively, then we can calculate the expected number of paths of length one between them as follows. For any of the k_i edges emerging from vertex i, there are $2m$ places where it could terminate.

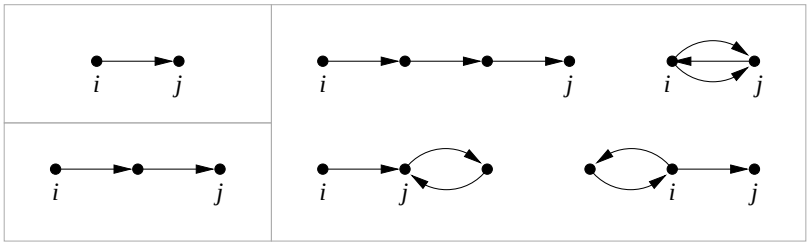


Figure 2.2. There is only one possible topology for paths of length one between distinct vertices, and only one for paths of length two, but there are four possible topologies for paths of length three.

(The total number of edges in the network is m.) Of these, k_j end at vertex j and result in a direct path of length one from i to j. Thus, for each edge emerging from i there is a probability $k_j/2m$ of a length-one path to j. Overall, the expected number of such paths is $k_ik_j/2m$ and the first-order coefficient is

$$C_1^{ij} = \frac{2m}{k_ik_j}. \tag{2.9}$$

Now consider the second-order term in the series. A path of length two between i and j must go through a single intermediate vertex v, whose degree we denote k_v. Using the argument of the preceding paragraph, the expected number of paths of length one from i to v is $k_ik_v/2m$. This step uses up one of the edges emerging from v, leaving $k_v - 1$ remaining edges. The expected number of paths of length one from v to j, given that there is already a path from i to v, is then $(k_v - 1)k_j/2m$. The expected number of paths of length two from i to j via v is just the product of the two terms, $k_ik_v(k_v - 1)k_j/(2m)^2$. Summing over all v, the total expected number of paths of length two is

$$\frac{k_ik_j}{(2m)^2}\sum_v k_v(k_v - 1) = \frac{k_ik_j}{2m}\left(\frac{\langle k^2\rangle - \langle k\rangle}{\langle k\rangle}\right), \tag{2.10}$$

where $\langle k\rangle$ and $\langle k^2\rangle$ are the mean degree and mean-square degree of the network respectively. Note, we have made use of the result $2m = n\langle k\rangle$, where n is the total number of vertices in the network. C_2^{ij} is then the reciprocal of this quantity.

For paths of length three and greater, the calculations become more complicated. Since paths can be self-intersecting, we have to consider topologies for those paths

that include loops or that traverse the same edge more than once. There exists only one topology for paths of length one or two between a specified pair of vertices. However, there are four distinct topologies for paths of length three (Fig. 2.2). To find the expected number of all paths of a given length between two vertices we must sum over all possible path topologies. The end result for the number of paths of length three between two vertices is

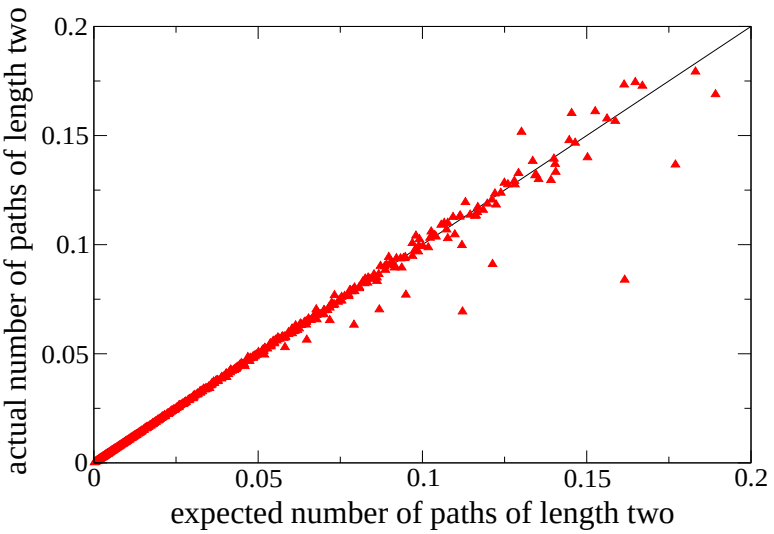
$$\frac{k_i k_j}{2m}\left[\left(\frac{\langle k^2\rangle - \langle k\rangle}{\langle k\rangle}\right)^2 + k_i + k_j - 1\right]. \tag{2.11}$$

As a check on our calculations, in Fig. 2.3, we compare our analytic expressions for the numbers of paths of length two and three to actual path counts for randomly generated networks. We find the actual path counts by calculating the average number of paths (of length two and three respectively) between two vertices of given degree over multiple realizations of a network using the configuration model. In the figures, there is increased scatter in the numerical data at longer path lengths due to finite size effects in the network, but overall the agreement between analytic and numerical calculations is good.

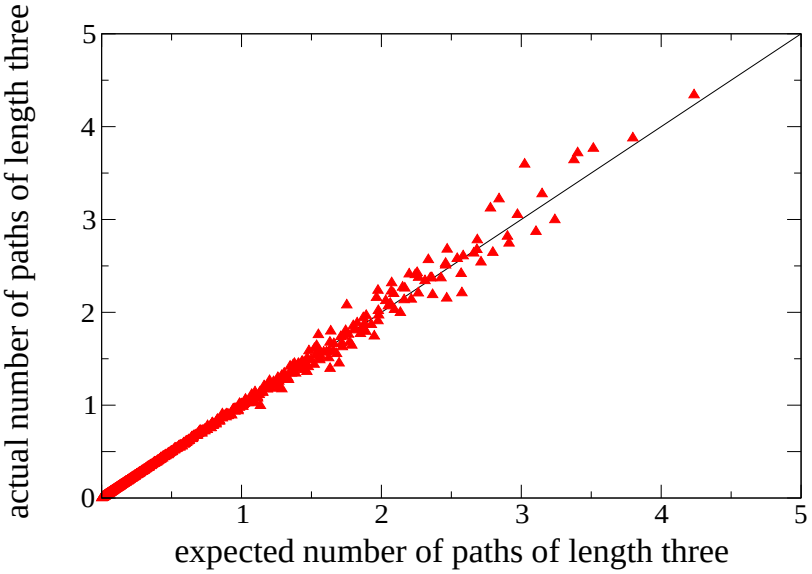
The expected number of paths of length l from i to j can be written as the jth element of the vector $\mathbf{p}_l$ given by

$$\mathbf{p}_l = \mathbf{A}^l \mathbf{v}, \tag{2.12}$$

where the vector $\mathbf{v}$ has all elements zero except for $v_i = 1$. In the limit of large l, the vector $\mathbf{p}_l$ tends toward (a multiple of) the leading eigenvector of the adjacency matrix. Consequently, in this limit we have $\mathbf{p}_{l+1} = \lambda_1 \mathbf{p}_l$, where λ_1 is the largest eigenvalue of $\mathbf{A}$. Thus, the number of paths from i to j increases by a factor of λ_1 each time we add one extra step to the path length. The first step of the path violates this rule: we know the number of paths increases by exactly a factor of k_i on the first step. Furthermore, since our paths are constrained to end at vertex j, the last step must end at one of the k_j edges emanating from j, out of a total of $2m$ possible places



(a)



(b)

Figure 2.3. The actual number of paths of length two (a) and three (b) between vertex pairs in a configuration model versus the expected number of paths given by Eq. (2.10) for (a) and Eq. (2.11) for (b).

that it could end. This introduces a factor of $k_j/2m$ into the expected number of paths. Hence, to within a multiplicative constant, the number of paths of length l from i to j, for large l, should be $(k_i k_j/2m)\lambda_1^{l-1}$.

This expression is not in general correct for small l. It is, however, correct for the particular case $l = 1$ (paths of length one see Eq. (2.9)) and we expect it to be approximately correct for other intermediate values of $l > 1$. Guided by these results, we therefore choose the constants C_l^{ij} appearing in Eq. (2.8) to take the values:

$$C_l^{ij} = \frac{2m}{k_i k_j}\,\lambda_1^{-l+1}, \tag{2.13}$$

for $l \geq 1$, with $C_0^{ij} = \delta_{ij}$. These values approximate the desired values based on expected numbers of paths and are asymptotically correct in the limit of large l.

2.2.2 Derivation of the similarity

There is one more issue we need to deal with with before we arrive at a final expression for our similarity. If we simply substitute C_l^{ij} from Eq. (2.13) into Eq. (2.8) we produce a series that unfortunately does not converge. To see this, note that in the limit of very long path lengths most networks will have roughly the same number of paths between vertices as would be expected by chance. (The local structure around the vertices in question is unimportant in this case, because the long length of the path means that correlation with its start and end points is weak.) This means that the terms in the series (2.8) will tend to unity for large l and, since there are an infinite number of them, the sum will diverge. On the other hand, if, rather than being constant, consecutive terms were to decrease by even the tiniest factor at each order, the sum would converge, as do all series with exponentially decreasing terms. Thus, we can ensure convergence by introducing an extra numerical factor α, giving the

series

$$\begin{aligned} S_{ij} &= \delta_{ij} + \frac{2m}{k_i k_j} \sum_{l=1}^{\infty} \alpha^l \lambda_1^{-l+1} \left[\mathbf{A}^l\right]_{ij} \\ &= \left[1 - \frac{2m\lambda_1}{k_i k_j}\right] \delta_{ij} + \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1}\mathbf{A}\right)^{-1}\right]_{ij}. \end{aligned} \tag{2.14}$$

In physical terms, the effect of the parameter α is to reduce the contribution of long paths relative to short ones. That is, for $0 < \alpha < 1$, our similarity measure considers vertices to be more similar if they have a greater than expected number of short paths between them, rather than if they have a greater than expected number of long ones. While this is a natural route to take, it does mean we have introduced a new free parameter into our calculations. This seems a fair exchange: we have traded the infinite number of free parameters in the expansion of Eq. (2.8) for a single parameter. We discuss the appropriate choice of value for α in Section 2.3.2.

The first term in Eq. (2.14) is diagonal and only impacts the similarity of vertices to themselves. Generally we are not interested in this similarity, so we will henceforth drop the term. Thus, our final expression for the similarity is

$$S_{ij} = \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1}\mathbf{A}\right)^{-1}\right]_{ij}. \tag{2.15}$$

Equivalently, we could write this in matrix form as

$$\mathbf{S} = 2m\lambda_1 \mathbf{D}^{-1} \left(\mathbf{I} - \frac{\alpha}{\lambda_1}\mathbf{A}\right)^{-1} \mathbf{D}^{-1}, \tag{2.16}$$

where $\mathbf{D}$ is the diagonal matrix having the degrees of the vertices as its diagonal elements: $D_{ij} = k_i \delta_{ij}$.

This similarity measure takes exactly the form we postulated in Eq. (2.5) with $\phi = \alpha/\lambda_1$, except for an overall multiplier, which is trivial, and the leading factor of $1/k_i k_j$, which is not. This factor compensates for the fact that we expect there to be more paths between pairs of vertices with high degree simply because there are more ways of entering and leaving such vertices. Its presence is crucial if we wish to compare the similarities of vertex pairs having very different degrees.

In practical terms, the calculation of the similarity matrix is most simply achieved by direct multiplication. Dropping the constant factor $2m\lambda_1$ for convenience, we can rewrite Eq. (2.16) in the form of Eq. (2.3) thus:

$$\mathbf{DSD} = \frac{\alpha}{\lambda_1}\mathbf{A}(\mathbf{DSD}) + \mathbf{I}. \tag{2.17}$$

Making any guess we like for an initial value of **DSD**, such as $\mathbf{DSD} = 0$, we iterate this equation repeatedly until it converges. In practice, for the networks studied here, we have found good convergence after 100 iterations or less.

2.2.3 Comparison with previous similarity measures

Several other authors have proposed vertex similarity measures based on matrix methods similar to ours [40, 10]. Jeh and Widom [40] have proposed a method that they call "SimRank," predicated, as ours is, on the idea that vertices are similar if their neighbors are similar. In our notation, their measure is

$$S_{ij} = \frac{C}{k_i k_j} \sum_{u,v} A_{iu} A_{vj} S_{uv}, \tag{2.18}$$

where C is a constant. While this expression bears some similarity to ours, Eq. (2.3), it also has an important difference. Starting from an initial guess for S_{ij}, one can iterate to converge on a complete expression for the similarity. The final expression contains terms representing path counts between vertex pairs, as in our case. However, since the adjacency matrix appears twice on the right-hand side of Eq. (2.18), the expression includes *only paths of even length*. This can make a substantial difference to the resulting figures for similarity. An extreme example would be a tree or a square lattice, in which vertices are separated either only by paths of even length or only by paths of odd length. In such cases, those vertices that are separated only by paths of odd length will have similarity zero. Even vertices that are directly connected to one another by an edge will have similarity zero. Most people would consider this result counterintuitive, and our measure, which counts paths of all lengths, seems clearly preferable.

Blondel *et al.* [10] considered similarity measures for directed networks, i.e., based on asymmetric adjacency matrices, which is a more complex situation than the one we consider. However, for the special case of a symmetric matrix, the measure of Blondel *et al.*can be written as

$$S_{ij} = C \sum_{u,v} A_{iu} A_{vj} S_{uv}, \tag{2.19}$$

where C is again a constant. This is very similar to the measure of Jeh and Widom, differing only in the omission of the factor $1/k_i k_j$. Like the measure of Jeh and Widom, it can be written in terms of paths between vertices, but counts only paths of even length, so that again vertices separated only by paths of odd length have similarity zero.

Work related to vertex similarity has been termed **vertex proximity** by Koren *et al.* [47]. Their idea of vertex proximity is focused towards measuring the potential of information exchange or the speed of information exchange between two vertices that are not directly connected in a network. It is easy to see the relation of this topic to our work on vertex similarity. Specifically, we propose that more similar vertices possess a greater number of paths between them than less similar vertices, and the number of paths between two vertices is also a factor in determining flow of information between those vertices. The work on vertex proximity begins with calculating the "weight of a path" between two vertices, an idea that is related to our idea of calculating the expected number of paths between vertices. Koren *et al.* recognized, as we did, that the paths of interest between vertex pairs in real networks are really just the non-self-intersecting paths, but that counts of those paths cannot easily be calculated. At this point their method greatly diverges from ours a they proposed a novel way of estimating the number of non-self-intersecting paths, which they term **simple paths**, based on calculating only the most probable edges.

2.2.4 A measure of structural equivalence

An interesting corollary of the theory developed in the previous sections is an alternative measure of structural equivalence. The structural equivalence measures derived from Eq. (2.1) can be viewed as similarity measures that count only the paths of length two between vertex pairs; the number of common neighbors of a pair of vertices is exactly equal to the number of paths of length two. Thus, structural equivalence can be thought of as just one term—the second-order term—in the infinite series that defines our measure of regular equivalence.

The specific measures (cosine similarity, the Jaccard index, etc.) differ from one another in their normalization. The developments outlined in this chapter suggest another possible normalization, one in which we divide the number of paths of length two by its expected value in the configuration model. We derived an exact expression in Eq. (2.10). The normalized regular equivalence measure is then

$$\sigma = \frac{2m}{k_i k_j} \left(\frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \right) |\Gamma_i \cap \Gamma_j|. \tag{2.20}$$

If we are concerned only with the comparative similarities of different pairs of vertices within a given graph, then we can neglect multiplicative constants and write

$$\sigma = \frac{|\Gamma_i \cap \Gamma_j|}{k_i k_j} = \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i|\,|\Gamma_j|}. \tag{2.21}$$

This is, we feel, the appropriate way to normalize Eq. (2.1). It gives high similarity to vertex pairs that have many common neighbors as compared not to the maximum number possible but to the *expected* number of such neighbors. The normalization, therefore, highlights vertices that have a statistically improbable coincidence of neighborhoods. Of course, one could define similar measures for paths of length one or three or any other length. One could also combine all such lengths, which is precisely what our overall similarity measure does.

2.3 Tests of the method

In this section we test our method on a number of different networks. Our first example is a set of computer-generated networks designed to have known similarities between vertices. In the following sections we also test the method against some real-world examples.

2.3.1 Stratified model network

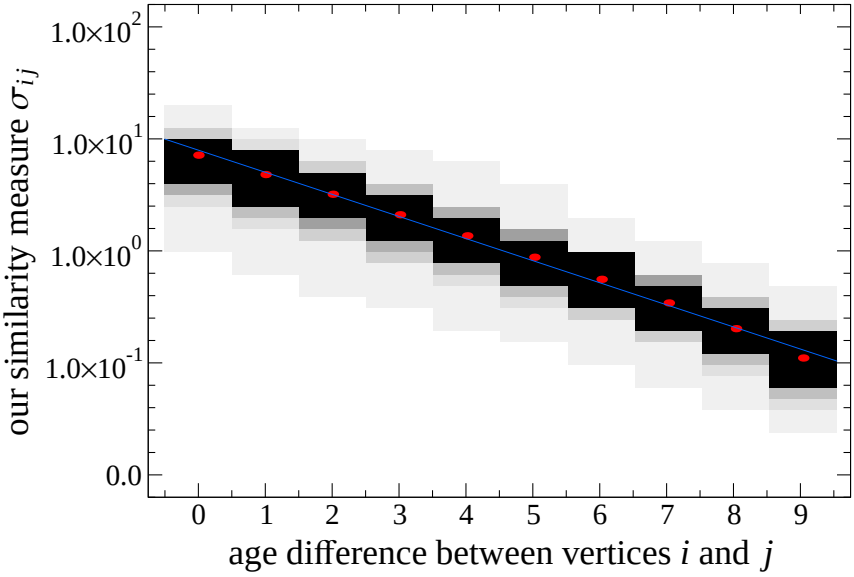
In many social networks, individuals make connections with others preferentially according to some perceived similarity, such as age or income. Such networks are said to be **stratified**, and stratified networks present a perfect opportunity to test our similarity measure: ideally we would like to see that given only the network structure our measure can correctly identify vertices that are similar in age (or whatever the corresponding variable is) even when the vertices are not directly connected to one another.

As a first test of our measure, we have created artificial stratified networks on a computer. Such networks offer a controlled structure for which we believe we know the "correct" answers for vertex similarity. In our model networks, each of $n = 1000$ vertices was given one of ten integer "ages." Then edges were created between vertices with probability

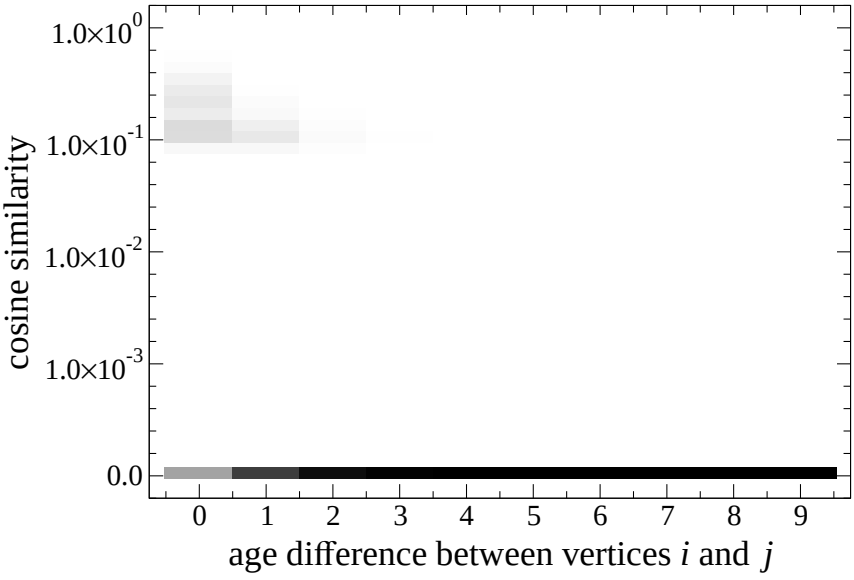
$$P(\Delta t) = p_0 e^{-a\Delta t}, \tag{2.22}$$

where Δt is the difference in ages of the vertices and p_0 and a are constants, whose values in our calculations were chosen to be $p_0 = 0.12$ and $a = 2.0$. Thus, the probability of "acquaintance" between two individuals drops by a factor of e^2 for every additional year separating their ages.

In order to calculate our similarity measure for this or any network we need first to choose a value for the parameter α appearing in Eq. (2.15). In the present calculations we used a value of $\alpha = 0.97$, which, as we will see, is fairly typical. Since α must



(a)



(b)

Figure 2.4. Density plots of vertex similarity in our stratified network model using (a) the method of this chapter and (b) cosine similarity. The points in plot (a) give the average similarity as a function of age difference and the line is a least-squares fit to a straight line.

be strictly less than one if Eq. (2.15) is to converge, $\alpha = 0.97$ is quite close to the maximum possible value. We discuss in the following section why values close to the maximum are usually desirable.

Figure 2.4(a) shows a density plot of the similarity values for all vertex pairs in the model network not directly connected by an edge, on a semi-log scale as a function of the age difference between the vertices. The average similarity as a function of age difference is also plotted, along with a fit to the data. We exclude directly connected pairs in the figure because it is trivial that such pairs will have high similarity and most of the interest in our method is in its ability to detect similarity in nontrivial cases.

For comparison, we also show in Fig. 2.4(b) a density plot of the cosine similarity, Eq. (2.2), for the same network. As the plots show, cosine similarity is in this case a much less revealing measure of similarity. It is only possible for cosine similarity to be nonzero for a pair of vertices if there exists a path of length two between them. Vertices with an age difference of three or more rarely have such a path in this network and, as Fig. 2.4(b) shows, such vertices therefore nearly all have a cosine similarity of zero. Thus cosine similarity finds only highly similar vertices in this case and entirely fails to distinguish between vertices with age differences between 3 and 9. Our similarity measure by contrast distinguishes these cases comfortably.

2.3.2 Choice of α

Our similarity measure, Eq. (2.15), contains one free parameter α, which controls the relative weight placed on short and long paths. This parameter lies strictly in the range $0 < \alpha < 1$, with low values placing most weight on short paths between vertices and high values placing weight more equally both on short and long paths. (Values $\alpha > 1$ would place more weight on long paths than on short, but for such values the series defining our similarity does not converge.)

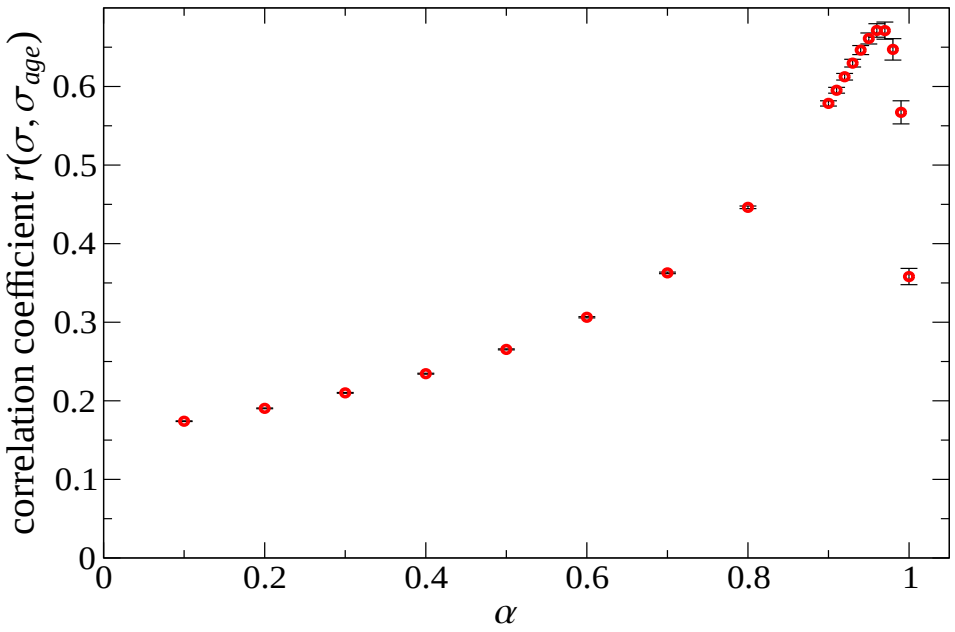


Figure 2.5. The correlation coefficient $r(\sigma, \sigma_{\text{age}})$ for correlation between our similarity measure and the probability of connection, Eq. (2.22), in our stratified model, for a range of values of α. The values given are averaged over an ensemble of graphs generated from the model. The maximum value is found to occur for $\alpha \simeq 0.97$.

There is, in general, no single value of α that works perfectly for every network, but experience suggests some reliable rules of thumb. Our stratified network model, for instance, provides a good guide. Consider Fig. 2.5, here we have calculated the correlation coefficient of the similarity values for vertex pairs determined using our method, against the probabilities, Eq. (2.22), of connections between the vertices, which, following the ideas outlined at the beginning of Section 2.2, we consider to be a fundamental measure of vertices' *a priori* similarity. As the figure shows, the correlation is quite low for small values of α, but becomes strong as α approaches unity. Only as α gets very close to unity does the correlation fall off again. This appears to imply that a value of $\alpha = 0.9$ or greater should give the best results in this case. Furthermore, it appears that, for values of α in this range, the precise value does not matter greatly, all values around the maximum in the correlation coefficient giving roughly comparable performance.

This we have found to be a good general rule: values of α close to the maximum value of 1 perform the best, with values in the range 0.90 to 0.99 being typical. Within this range the results are not highly sensitive to the exact value. We give another example to reinforce this conclusion below.

The large typical values of α mean that paths of different lengths are weighted almost equally in our similarity measure. In other words, it appears that our measure works best when long paths are accorded almost as much consideration as short ones. This contrasts strongly with structural equivalence measures like the Jaccard index and the cosine similarity, which are based exclusively on short paths—those of length two. Indeed, these measures can be considered analogous to measures such as ours in the limit of small α, where all the weight is placed on the shortest paths, which effectively means paths of length two when we are considering vertex pairs that are not directly connected. Thus, in a sense, our measure, with its near-maximal value of α, can be considered at the farthest possible extreme from the traditional structural equivalence measures.

2.3.3 Thesaurus network

We now consider two applications of our method to real-world networks. The first is to a network of words extracted from a supplemented version of the 1911 U.S. edition of *Roget's Thesaurus* [54]. The thesaurus consists of a five-level hierarchical categorization of English words. For example, the word "paradise" (level five) is cataloged under "heaven" (level four), "superhuman beings and regions" (level three), "religious affections" (level two), and "words relating to the sentient and moral powers" (level one). Here we study the network composed of the 1000 level-four words, in which two such words are linked if one or more of the level-five words cataloged below them are common to both. For instance, the level-four words "book" and "knowledge" are connected because the entries for both in the thesaurus contain the level-five terms

word	our measure		cosine similarity	
alarm	warning	32.014	omen	0.51640
	danger	25.769	threat	0.47141
	omen	18.806	prediction	0.34816
hell	heaven	63.382	pleasure	0.40825
	pain	28.927	discontent	0.28868
	discontent	7.034	weariness	0.26726
mean	compromise	20.027	gravity	0.23570
	generality	19.811	inferiority	0.22222
	middle	17.084	littleness	0.20101
water	plunge	33.593	dryness	0.44721
	air	25.267	wind	0.31623
	moisture	25.267	ocean	0.31623

Table 2.1. The words most similar to "alarm," "heaven," "mean," and "water," in the word network of the 1911 edition of *Roget's Thesaurus*, as quantified by our similarity measure and by the more rudimentary cosine similarity of Eq. (2.2). We used a value of 0.98 for the parameter α.

"book learning" and "encyclopedia."

In Table 2.1 we show the words most similar to the words "alarm," "hell," "mean," and "water," as ranked first by our similarity measure and second by cosine similarity. We used a value of $\alpha = 0.98$ in this case, on the grounds that this value gave the best performance in other test cases (see below).

Since cosine similarity can be regarded as a measure of the number of paths of length two between vertices, it tends in this example to give high similarity scores for words at distance two in the thesaurus—synonyms of synonyms, antonyms of synonyms, and so forth. For example, cosine similarity ranks "pleasure" as the word most similar to "hell," probably because it is closely associated with hell's antonym "heaven." By contrast, our measure ranks "heaven" itself first, which appears to be a more sensible association. Similarly, cosine similarity links "water" with "dryness", whereas our measure links "water" with "plunge."

2.3.4 Friendship network of high school students

As a second real-world test of our similarity measure, we apply it to a set of networks of friendships between school children. The network data were collected as part of

the National Longitudinal Study of Adolescent Health (AddHealth) [8], and describe 90 118 students at 168 schools, including their school grade (i.e., year), race, and gender, as well as their recent patterns of friendship. It is well known that people with similar social traits tend to associate with one another [56], so we expect there to be a correlation between similarity in terms of personal traits and similarity based on network position. This gives us another method for checking the efficacy of our similarity measure.

The AddHealth data were gathered through questionnaires handed out to students at 84 pairs of American schools, a school pair typically consisting of one junior high school (grades 7 and 8, ages 12–14) and one high school (grades 9–12, ages 14–18). Here we consider the data as a set of networks in which each network contains students from one pair of schools, with students in all six grades.

The questionnaires asked respondents, among other things, to "List your closest (male/female) friends. List your best (male/female) friend first, then your next best friend, and so on. (Girls/Boys) may include (boys/girls) who are friends and (boy/girl) friends." For each of the friends listed, the student was asked to state in which of five particular activities they had participated recently with that friend, such as "you spent time with (him/her) last weekend." From these answers a weight $w(i,j)$ was assigned to every ordered pair of students (i,j) such that $w(i,j)$ is 0 if i has not listed j as a friend, or 1 plus the number of activities conducted otherwise. (The additional 1 is necessary because some students list another as a friend but have not participated in any of the listed activities with them recently.) From these weights we construct an unweighted, undirected friendship network by adding a link between vertices i and j if $w(i,j)$ and $w(j,i)$ are both greater than or equal to a specified threshold value W. As it turns out, our conclusions are not very sensitive to the choice of W; the results described here use $W = 2$.

The networks derived in this way are not necessarily connected; they may, and

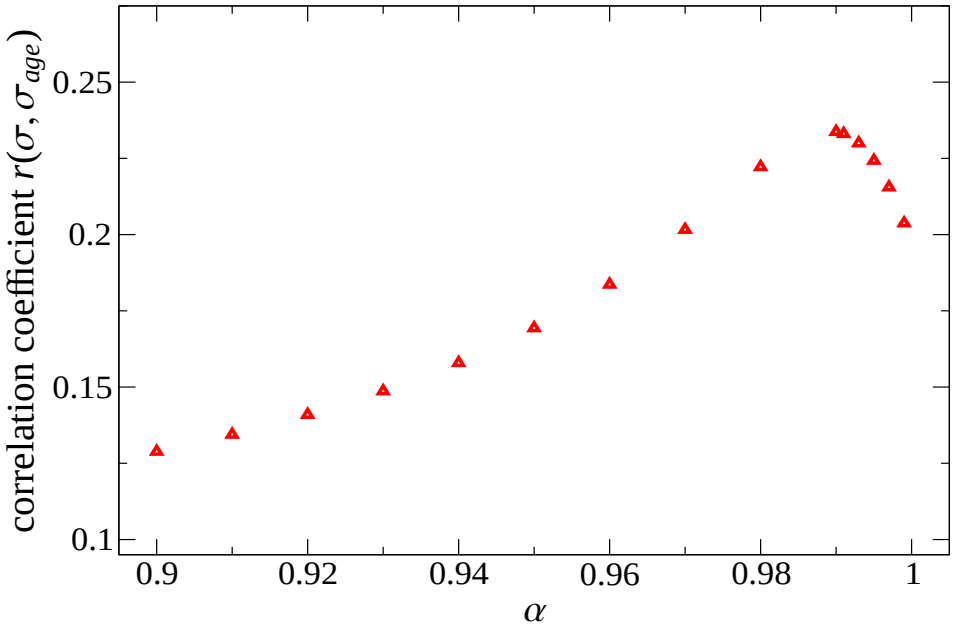


Figure 2.6. The correlation coefficient for correlation between our similarity measure and the age difference of all vertex pairs in a single network, as a function of α. This plot is typical for the school networks studied.

often do, consist of more than one component for each school studied. To simplify matters we consider only the largest component of each network. The largest component in some of the networks is quite small, however, so to avoid finite size effects we have focused on networks whose largest component contains more than 1000 students.

We first test our similarity measure using the method we used for the stratified network of Section 2.3.1: we determine the linear correlation coefficient between age difference (measured as difference in grade) and our network similarity measure, for all vertex pairs in a network. We have calculated this correlation coefficient for a range of values of α, the free parameter in our measure, and for a selection of different networks. The results for one particular network are shown in Fig. 2.6. In this case the correlation coefficient is maximized for $\alpha \simeq 0.99$, which is again close to the maximum possible value of 1. For other networks we find maxima in the range from 0.96 to 0.99, which is in accord with the results of Section 2.3.2. (We also calculated correlation coefficients for the similarity measures of Jeh and Widom [40]

		similarity ratios			
school	n	SG:DG	SG:DG*	SR:DR	SR:DR*
A	1090	8.0	6.1	1.1	1.1
B	1302	6.2	4.4	2.6	2.6
C	1996	2.2	1.9	5.0	5.0
D	1530	3.3	2.6	4.0	3.6

Table 2.2. Network size n and ratios of average similarity values for school networks in the AddHealth data set. The column labeled SG:DG gives the ratio of average similarity for students in the same grade (SG) to average similarity for students in different grades (DG). The column labeled SR:DR gives the ratio of average similarity for students of the same race (SR) to average similarity for students of different races (DR). Columns marked with asterisks (*) give values of the same ratios but omitting vertex pairs connected directly by an edge.

and Blondel *et al.* [10] discussed in Section 2.2.3. These values were routinely lower than ones found with our measure.)

These correlations between age difference and network similarity appear to indicate that our similarity measure is able to detect some aspects of the social structure of these networks. To investigate this further, we have also calculated the average similarity of vertex pairs that have a known common characteristic, either grade or race, comparing that average with the average similarity for vertex pairs that differ with respect to the same characteristic. The values of α used were those corresponding to the peak in the correlation, as above. The results are given in Table 2.2.

For school A, for example, the average similarity for pairs of students in the same grade is a factor of eight greater than that for pairs in different grades. It is possible, however, that this impressive difference could result purely from the contribution to the similarity from vertex pairs that are directly connected by an edge. It would come as no surprise that such pairs tend to be in the same grade. To guard against this, we give in the fourth column of Table 2.2 results for calculations in which all directly connected vertex pairs did not contribute to the calculation of average similarity. Even with these pairs removed we see that same-grade vertex pairs are on average significantly more similar than pairs from different grades.

We have also made similar calculations with respect to the race of students. Stu-

dents in school A did not appear to have any significant division along racial lines (columns five and six of Table 2.2), but this school was almost entirely composed of students of a single race anyway, so this result is not very surprising; it seems likely that the numbers were just too small to show a significant effect. School B was similar. Schools C and D, however, show a marked contrast. In school C, the average similarity for students of the same race is a factor of five greater than the average similarity for students of different races. School C had a population split 2:1 between two racial groups, in marked contrast with schools A and B. School D similarly appears to be divided by race, although a little less strongly. In this case there is a three-way split within the population between different racial groups. Possibly this more even split with no majority group was a factor in allowing more friendships between students from different racial groups.

These results indicate that our measure of similarity is able to identify real social similarity between vertices in these networks. That is, using only the structure of the network, our similarity measure identifies students of the same race and in the same grade to be more similar to each other than students of different grades or different races. For comparison, we performed similar calculations using the similarity measures proposed by Jeh and Widom [40] and Blondel *et al.* [10], finding again that the average similarity of pairs of vertices sharing characteristics was higher than for pairs of vertices that differed by the same characteristic. However, the factors by which these methods differentiated between vertices with similar characteristics and vertices with different characteristics was consistently less than with our measure.

2.4 Discussion

In this chapter we have proposed a measure of structural similarity for pairs of vertices in networks. The method is fundamentally iterative, with the similarity of a vertex pair being given in terms of the similarity of the vertices' neighbors. Alternatively,

our measure can be viewed as a weighted count of the number of paths of all lengths between the vertices in question. We expect the measure to be applicable to any network where the vertices do not have a tendency to attach to dissimilar vertices. The weights appearing in this count are asymptotically equal to the expected numbers of network paths between the vertices, which we express in terms of the leading eigenvalue of the adjacency matrix of the network and the degrees of the vertices of interest. The resulting expression for our similarity measure is given in Eq. (2.16).

We have tested our measure against computer-generated and real-world networks, with promising results. In tests on computer-generated networks the measure is particularly good at discerning similarity between vertices connected by relatively long paths, an area in which more traditional similarity measures such as cosine similarity perform poorly. In tests on real-world networks the method was able to extract sensible synonyms to words from a network representing the structure of Roget's Thesaurus, and showed strong correlations with similarity of age and race in a number of networks of friendship among school children. Taken together, these results seem to indicate that the measure is capable of extracting useful information about vertex similarity based on network topology.

The strength of similarity measures such as ours is their generality—in any network where the function or role of a vertex is related in some way to its structural surroundings, structural similarity measures can be used to find vertices with similar functions. For instance, similarity measures can be used to divide vertices into functional categories [53, 76, 92] or for functional prediction in cases where the functionality of vertices is partly known ahead of time [36]. Additionally, our method may be applied to "link prediction" [52]. We believe that the application of similarity measures to problems such as these will prove a fruitful topic for future work.

CHAPTER 3

Community structure in directed networks

3.1 Introduction

Many networks are found to display community structure dividing naturally into communities or modules with dense connections within communities, but sparser connections between them. Communities are of interest both in their own right as functional building blocks within networks and for the insights they offer into the dynamics or modes of formation of networks. A large volume of research has been devoted to the development of algorithmic methods for discovering communities. An introduction to the topic is found in Section 1.2.7.

Nearly all of these methods, however, have one thing in common: they are intended for the analysis of undirected network data. Many of the networks that we would like to study are directed, including the World Wide Web, food webs, many biological networks, and even some social networks. A common approach to detecting communities in directed networks has been simply to ignore the direction of edges and apply algorithms designed for undirected networks. This works reasonably well in some cases, although in others it does not, as we will illustrate in this chapter. Even in the cases where it works, however, it is clear that in discarding the direction of edges we are throwing away a good deal of information about the structure of our network, information that, at least in principle, could allow us to make a more accurate determination of the communities.

Several previous studies, including our own, have touched on this problem [3, 34,

68, 80], but they have typically not tackled the community structure question directly. In this chapter, we propose a method for finding communities in directed networks that makes explicit use of the information contained in edge direction. The method is an extension of the well established modularity optimization approach for undirected networks [65], an approach that has been shown to be both computationally efficient and highly effective in practical applications [20]. We outline our technique for detecting communities in directed networks in Section 3.2. In Section 3.3, we apply our method to both real and simulated networks. Finally, we conclude with a brief discussion of the results in Section 3.4.

3.2 The method

Recall from Section 1.2.7 that the premise of the modularity optimization method is that a good division of a network into communities will give high values of the benefit function Q,

$$\begin{aligned} Q = &\text{(fraction of edges within communities)} \\ &- \text{(expected fraction of such edges)}. \end{aligned} \tag{3.1}$$

Large positive values of the modularity indicate when a statistically surprising fraction of the edges in a network fall within the chosen communities; it tells us when there are more edges within communities than we would expect on the basis of chance. Given a division of a network into communities, it is easy to calculate the first term in Eq. (3.1), but the second term requires some additional information about the expected number of edges within a community. In effect, we are requiring ourselves to compare the real network data to some null model of network data where edges are placed a random.

In recent work on complexity reduction in networks, Arenas *et al.* [3] have proposed a generalization of the modularity to directed networks, which can be under-

stood in the following way. The expected positions of edges in a directed network depend on their direction. Consider two vertices, A and B. Vertex A has high out-degree but low in-degree while vertex B has the reverse situation. This means that a given edge is more likely to run from A to B than *vice versa*, simply because there are more ways it can fall in the first direction than in the second. Hence if we *observe* in our real network that there is an edge from B to A, it should be considered a bigger surprise than an edge from A to B and thus should make a bigger contribution to the modularity, since modularity should be high for statistically surprising configurations.

In practice, exact optimization of the modularity is computationally hard, so practical methods based on modularity optimization make use of approximate optimization schemes such as greedy algorithms, simulated annealing, spectral methods, and others [33, 57, 64, 66, 79].

We put these insights to work adapting a method for community detection, previously only known for undirected networks [64], for directed networks. Like the method for undirected networks, our starting constraint for the null model is to fix the total number of vertices, n, to be the same as in the real network of interest. Apart from that basic constraint, for the moment, we leave the null model flexible. We represent the real number of edges from a vertex j to a vertex i as the ijth element of the adjacency matrix, A_{ij} (see Section 1.2.1), but we need a way to write the expected number of such edges.

Let us propose another matrix, $\mathbf{P}$, where the element P_{ij} is the expected number of edges from vertex j to vertex i. We can now write the real number of edges minus the expected number of edges from vertex j to vertex i as $A_{ij} - P_{ij}$. This expression can be used to rewrite Eq. (3.1) with more mathematical formalism. Modularity, Q, equals the real minus expected number of edges between all vertex pairs when both vertices are in the same community. Note, we must count pairs j to i and i to j separately since the direction of the edges is important. If we define the community

of a vertex i to be c_i, we can write modularity as,

$$Q = \frac{1}{m} \sum_{ij} \Big[A_{ij} - P_{ij} \Big] \delta_{c_i,c_j}, \tag{3.2}$$

where δ_{c_i,c_j} is the Kronecker delta function and is equal to 1 if i and j are in the same group $c_i = c_j$ and 0 otherwise. The factor of $\frac{1}{m}$ in Eq. (3.2) is merely a convention which makes this equation compatible with previous definitions of modularity. The term does not impact the maximization of Q as it is an overall multiplying factor. We note that this equation is similar to the ones given independently by Newman [65] and White and Smyth [91] regarding the calculation of modularity for undirected networks.

Let us now impose an additional constraint on Q. Let $Q = 0$ if all network vertices are placed in the same group. This constraint follows directly from Eq. (3.1); if all vertices are in one community then all edges fall within the community and all expected edges fall within the community making the two quantities equal and forcing $Q = 0$. We can rewrite Eq. (3.2) assuming $c_i = c_j$ for all i and j,

$$Q = \sum_{ij} \Big[A_{ij} - P_{ij} \Big] = 0 \tag{3.3}$$

or

$$\sum_{ij} P_{ij} = \sum_{ij} A_{ij} = m. \tag{3.4}$$

Thus, imposing $Q = 0$ when all network vertices fall into one community forces us to choose a null model that not only has the same number of vertices, n, as our original network data, but also has the same number of edges, m. However, we still do not know how those edges are distributed among the n^2 pairs of vertices in the network. In order to determine edge placement in the null model, we take inspiration from the method for undirected networks. In that case, the degree of each vertex in the null model is required to be equal to the degree of a corresponding vertex in the real network. For directed networks, we must keep both the in-degree and out-degree of

each vertex fixed between the real network and the null model. Consequently, we impose the following constraints on our null model,

$$\sum_i P_{ij} = k_j^{\text{out}}, \tag{3.5}$$

and

$$\sum_j P_{ij} = k_i^{\text{in}}. \tag{3.6}$$

By making these choices we automatically satisfy Eq. (3.4) since $\sum_i k_i^{\text{in}} = \sum_i k_i^{\text{out}} = m$.

One simple null model satisfying both Eqs. (3.5) and (3.6) is related to the configuration model, which was introduced in Section 1.3.2. In this model edges are placed randomly between network vertices, subject only to the allowed in- and out-degree of each vertex. Thus, if we consider a directed edge from vertex j to vertex i the probability that the tail of a directed edge falls at vertex j depends only on the out-degree of j, k_j^{out}. Likewise, the probability that the head of a directed edge falls at vertex i depends only on the in-degree of i, k_i^{in}. These two probabilities are independent; consequently, we can write the expected number of edges from j to i as

$$P_{ij} = f(k_i^{\text{in}})g(k_j^{\text{out}}) \tag{3.7}$$

where f and g are both functions. We know from Eqs. (3.5) and (3.6) that

$$\sum_i P_{ij} = g(k_j^{\text{out}}) \sum_i f(k_i^{\text{in}}) = k_j^{\text{out}} \tag{3.8}$$

and

$$\sum_j P_{ij} = f(k_i^{\text{in}}) \sum_j g(k_j^{\text{out}}) = k_i^{\text{in}}. \tag{3.9}$$

Thus, $f(k_i^{\text{in}}) = Fk_i^{\text{in}}$ and $G = g(k_j^{\text{out}})$ where F and G are constants. We also know from Eq. (3.4) that

$$m = \sum_{ij} P_{ij} = FG \sum_{ij} k_i^{\text{in}} k_j^{\text{out}} = FGm^2. \tag{3.10}$$

Therefore, $FG = m^{-1}$ and we must have,

$$P_{ij} = \frac{k_i^{\text{in}} k_j^{\text{out}}}{m}. \tag{3.11}$$

We can now return to Eq. (3.2), and substitute Eq. (3.11) in for P_{ij},

$$Q = \frac{1}{m} \sum_{ij} \left[A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right] \delta_{c_i,c_j}, \tag{3.12}$$

which is a special case of the formula given in [3]. Note that edges $j \to i$ do indeed make larger contributions to this expression if k_i^{in} and/or k_j^{out} is small.

As in the undirected case we can make use of the modularity to find network communities by searching for the division of the network that maximizes Q. One can in principle make use of any of the methods previously applied to modularity maximization, such as simulated annealing or greedy algorithms. Here we derive the appropriate generalization of the spectral optimization method of Newman [66], which is both computationally efficient and appears to give excellent results in practice.

We consider first the simplified problem of dividing a directed network into just two communities. We define s_i to be $+1$ if vertex i is assigned to community 1 and -1 if it is assigned to community 2. Note that this implies that $\sum_i s_i^2 = n$. Then, $\delta_{c_i,c_j} = \frac{1}{2}(s_i s_j + 1)$ and

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right] (s_i s_j + 1) = \frac{1}{2m} \mathbf{s}^T \mathbf{B} \mathbf{s}, \tag{3.13}$$

where $\mathbf{s}$ is the vector whose elements are the s_i, $\mathbf{B}$ is the so-called modularity matrix with elements

$$B_{ij} = A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m}. \tag{3.14}$$

Our goal is now to find the $\mathbf{s}$ that maximizes Q for a given $\mathbf{B}$.

In the undirected case the modularity matrix is symmetric, but in general is not in the directed case. The lack of symmetry in the modularity matrix will cause technical problems if we blindly attempt to duplicate the eigenvector-based machinery

presented for undirected networks in [66]. We can, however, restore symmetry to the problem by the following trick. Noting that Q is a scalar and therefore equal to its own transpose, we take the transpose of Eq. (3.13) to give $Q = (2m)^{-1}\mathbf{s}^T\mathbf{B}^T\mathbf{s}$ and then take the average of this expression and Eq. (3.13) to give

$$Q = \frac{1}{4m}\mathbf{s}^T(\mathbf{B} + \mathbf{B}^T)\mathbf{s}. \tag{3.15}$$

The matrix $\mathbf{B} + \mathbf{B}^T$ is manifestly symmetric and it is on this matrix that we focus forthwith. Notice that $\mathbf{B} + \mathbf{B}^T$ is not the same as the modularity matrix for a symmetrized version of the network in which edge direction is ignored. Consequently, we expect methods based on the true directed modularity to give different results, in general, from methods based on the undirected version.

The leading constant $1/4m$ in Eq. (3.15) is conventional, but makes no difference to the position of the maximum of Q, so for the sake of clarity we neglect it in defining our optimization procedure.

Following [66], we now write $\mathbf{s}$ as a linear combination of the eigenvectors $\mathbf{v}_i$ of $\mathbf{B} + \mathbf{B}^T$ thus: $\mathbf{s} = \sum_i a_i\mathbf{v}_i$ with $a_i = \mathbf{v}_i^T \cdot \mathbf{s}$. As a result, we can write

$$Q = \sum_i a_i\mathbf{v}_i^T(\mathbf{B} + \mathbf{B}^T)\sum_j a_j\mathbf{v}_j = \sum_i \beta_i(\mathbf{v}_i^T \cdot \mathbf{s})^2, \tag{3.16}$$

where β_i is the eigenvalue of $\mathbf{B} + \mathbf{B}^T$ corresponding to eigenvector $\mathbf{v}_i$. Let us assume the eigenvalues to be labeled in decreasing order $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_n$. Under the normalization constraint $\mathbf{s}^T \cdot \mathbf{s} = n$ the maximum of Q is achieved when $\mathbf{s}$ is chosen parallel to the leading eigenvector $\mathbf{v}_1$, but normally this solution is forbidden by the additional condition that $s_i = \pm 1$. We do the best we can, however, and make $\mathbf{s}$ as close as possible to parallel with $\mathbf{v}_1$, meaning we choose the value of $\mathbf{s}$ that maximizes $|\mathbf{v}_1^T \cdot \mathbf{s}|$. Note,

$$|\mathbf{v}_1^T \cdot \mathbf{s}| = \left|\sum_i v_i^{(1)} s_i\right| \leq \sum_i |v_i^{(1)}| \tag{3.17}$$

where $v_i^{(1)}$ is the ith element of the eigenvector $\mathbf{v}_1$. The triangle inequality allows us to put an upper bound on the maximum possible value of $|\mathbf{v}_1^T \cdot \mathbf{s}|$. The inequality

becomes an equality when all of the elements s_i are chosen to force the terms in the sum on the left-hand side to be all positive (or all negative). We then find the maximum of $|\mathbf{v}_1^T \cdot \mathbf{s}|$ when $v_i^{(1)} s_i \geq 0$, which occurs if s_i has the same sign as $v_i^{(1)}$ for all i. Consequently, modularity is maximized when we choose,

$$s_i = \begin{cases} +1, & \text{if } v_i^{(1)} > 0 \\ -1, & \text{if } v_i^{(1)} < 0 \end{cases} . \tag{3.18}$$

If $v_i^{(1)} = 0$ then $s_i = \pm 1$ are equally good solutions to the maximization problem. Thus, we arrive at a simple algorithm for splitting a network into two communities: we calculate the eigenvector corresponding to the largest positive eigenvalue of the symmetric matrix $\mathbf{B} + \mathbf{B}^T$ and then assign vertices to communities based on the signs of the elements of this eigenvector.

As in the undirected case, the spectral method typically provides an excellent guide to the broad outlines of the optimal partition, but may err in the case of individual vertices, a situation that can be remedied by adding a "fine-tuning" stage to the algorithm in which vertices are moved back and forth between communities in an effort to increase the modularity, until no further improvements can be made [66]. We have incorporated such a fine-tuning in all the calculations presented here.

We can illustrate the splitting of a directed network into two communities by looking at Moreno's network of seventh-grade students. We applied several community structure detection methods to the symmetrized version of this network in Section 1.2.7. With the algorithm introduced in this chapter we need not ignore edge direction. In Fig. 3.1, we see that the network does split into communities exactly along gender lines.

So far, we have discussed the division of a network into two communities. There are a number of ways of generalizing the approach to more than two communities, but the simplest, which we adopt here, is repeated bisection. That is, we first divide the network into two groups, then subdivide those groups and so on. The process

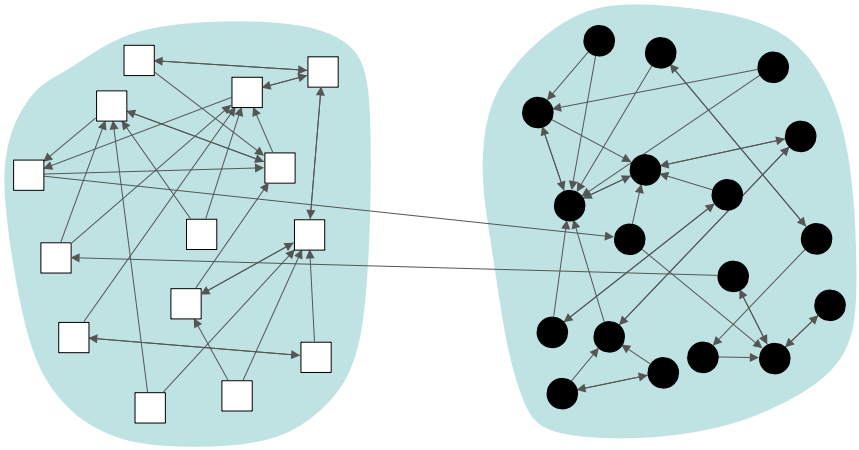


Figure 3.1. The original directed version of Moreno's network of school children divided into two communities by the algorithm of this chapter. Again the girls are represented by black circles and the boys are represented by white squares. The two shaded regions represent the two communities detected by our method.

stops when we reach a point at which further division does not increase the total modularity of the network.

The subdivision of a community contained within a larger network requires a slight generalization of the method above. Consider the change in modularity ΔQ of an entire network when a community g within it is subdivided. Defining s_i as before for vertices in g, we find

$$\begin{aligned}\Delta Q &= \frac{1}{2m}\left[\sum_{i,j\in g}(B_{ij}+B_{ji})\frac{s_i s_j+1}{2} - \sum_{i,j\in g}(B_{ij}+B_{ji})\right] \\ &= \frac{1}{4m}\sum_{i,j\in g}\left[(B_{ij}+B_{ji}) - \delta_{ij}\sum_{k\in g}(B_{ik}+B_{ki})\right]s_i s_j \\ &= \frac{1}{4m}\mathbf{s}^T\Big(\mathbf{B}^{(g)}+\mathbf{B}^{(g)T}\Big)\mathbf{s},\end{aligned} \tag{3.19}$$

where we have made use of $s_i^2 = 1$ for all i and

$$B_{ij}^{(g)} = B_{ij} - \tfrac{1}{2}\delta_{ij}\sum_{k\in g}(B_{ik}+B_{ki}). \tag{3.20}$$

In other words, $\mathbf{B}^{(g)}$ is the sub-matrix of $\mathbf{B}$ for the subgraph g with the average of the appropriate row and column sums subtracted from each diagonal element. Although $\mathbf{B}^{(g)}$, like $\mathbf{B}$, is in general asymmetric, the sum $\mathbf{B}^{(g)}+\mathbf{B}^{(g)T}$ is symmetric and hence

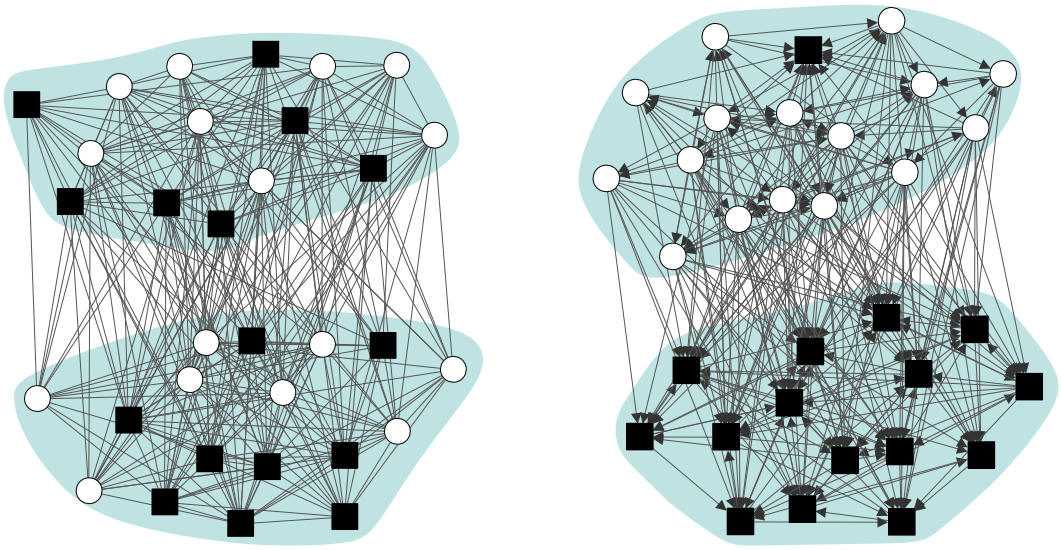
Eq. (3.19) has the same functional form as Eq. (3.15) and we can apply the same method to maximize ΔQ.

Our complete algorithm for discovering communities or groups in a directed network is thus as follows. We construct the modularity matrix, Eq. (3.14), for the network and find the most positive eigenvalue of the symmetric matrix $\mathbf{B} + \mathbf{B}^T$ and the corresponding eigenvector. Each vertex is assigned to one of two groups depending on the sign of the appropriate element of the eigenvector and then we fine-tune the assignments as described above to maximize the modularity. We then further subdivide the communities using the same method, but with the generalized modularity matrix, Eq. (3.20), fine tuning after each division. If the algorithm finds no division giving a positive value of ΔQ for a particular community, then we can increase the modularity no further by subdividing this community and we leave it alone. When all communities reach this state the algorithm ends.

3.3 Applications

We now give a number of examples of the application of the method. For illustrative purposes, we first consider an artificial computer-generated network, designed specifically to test the performance of the algorithm. In this network of 32 vertices, vertex pairs are connected by edges independently and uniformly at random with some probability p. The edges are initially undirected. The network is then divided into two groups of 16 vertices each and edges that fall within groups are assigned directions at random, but edges between groups are biased so that they are more likely to point from group 1 to group 2 than *vice versa*.

By construction, there is no community structure to be found in this network if we ignore edge directions—the positions of the edges are entirely random—and this is confirmed in Fig. 3.2(a), which shows the results of the application of the undirected modularity maximization algorithm. If we take the directions into account, however,

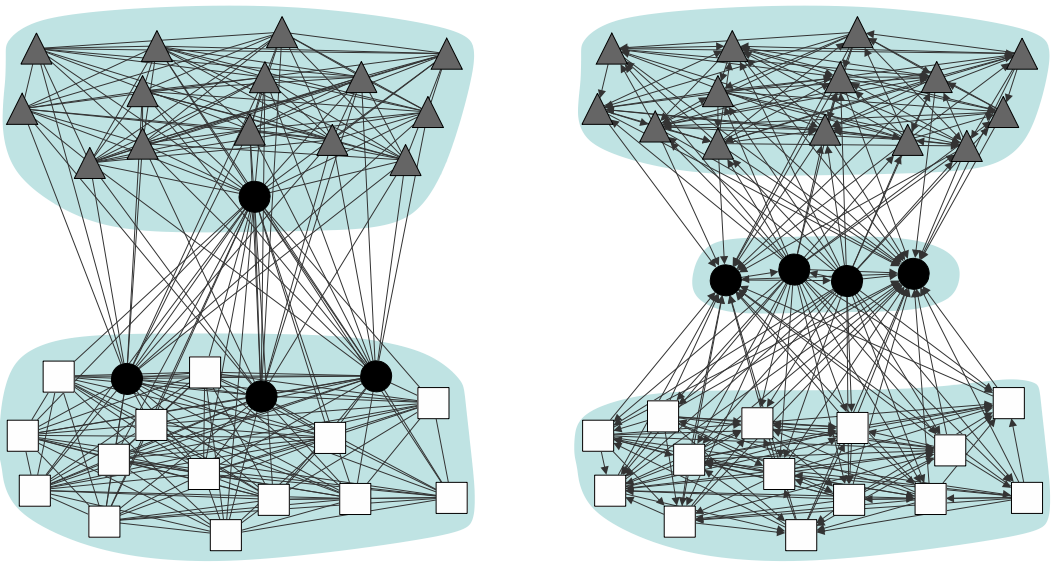


(a) (b)

Figure 3.2. Community assignments for the two-community random network described in the text using (a) a standard modularity maximization that ignores edge direction and (b) the algorithm of this chapter. The shaded regions represent the communities discovered by the algorithms; the true community assignments are denoted by vertex shape and color.

using the algorithm presented in this chapter, the two communities are detected almost perfectly: just one vertex out of 32 is misclassified—see Fig. 3.2(b).

Even in networks where there is clear community structure contained in the positions of the edges, it is still possible for the directions to contribute additional useful information. As an example of this type of behavior, consider the network shown in Fig. 3.3, which has 32 vertices and three communities. For two of the communities, containing 14 vertices each, there is a high probability of connection between pairs of vertices that fall in the same community, but a lower probability if the vertices are in different communities. Structure of this kind, in which edge direction does not play a role, can in principle be found by algorithms designed for undirected networks. The third community, however, is different. It has four vertices, each of which has a high probability of connection to every other vertex. The only feature that distinguishes this third community as separate is the direction of its edges—two of the four ver-



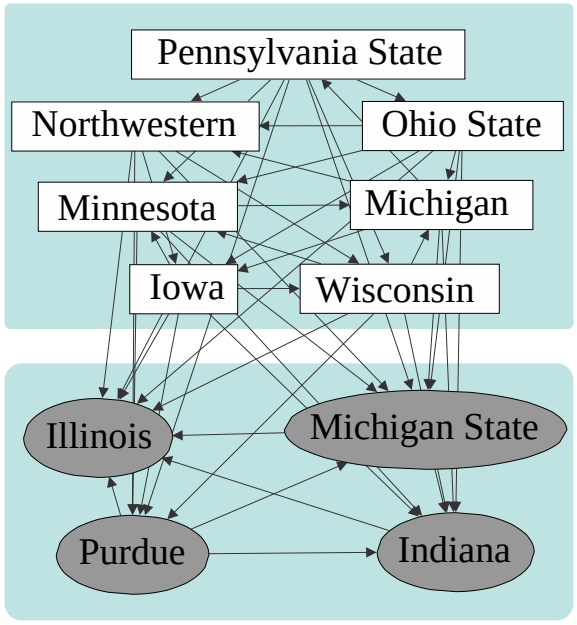
(a) (b)

Figure 3.3. Community assignments for the three-community random network described in the text as generated by (a) standard undirected modularity maximization and (b) the algorithm of this chapter.

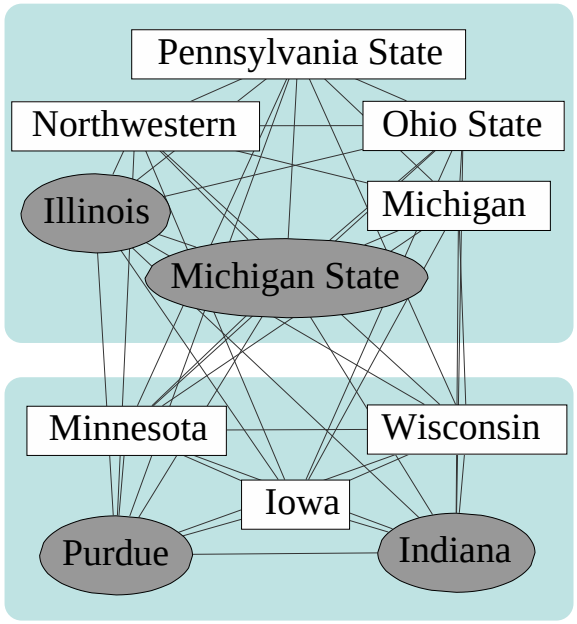
tices have high probability of in-going edges, the other two have high probability of outgoing edges, and there are also a small number of additional edges running from the former to the latter.

Applied to this network, the standard undirected community detection algorithm finds the two large communities with ease, but the remaining community is not found and its vertices are dispersed by the algorithm among the other communities (Fig. 3.3(a)). Our directed algorithm, on the other hand, finds all three communities without difficulty (Fig. 3.3(b)). Again the algorithm has made use of information contained in the edge directions to identify structures not accessible to other methods.

Turning now to real-world networks, consider Fig. 3.4, which shows a network representation of a sporting competition. Networks of this kind have received some attention in the recent literature for their clear, but nontrivial community structure. The vertices in the network represent the teams in one of the regional competitions or



(a)



(b)

Figure 3.4. Community assignments for the network of American football teams competing in the "Big Ten" conference in 2005 as generated by (a) the algorithm of this chapter and (b) a standard undirected modularity maximization. The shaded regions represent the communities discovered by the algorithms while vertex shapes and colors indicate whether teams won or lost a majority of their games during the season.

"conferences" of U.S. universities in the game of American football. Edges join pairs of teams that played one another during the 2005 football season. Most previous studies have represented such networks as undirected, but useful information can be extracted from a directed version in which the edges point from the winner to the loser of each game [70].

Figure 3.4(a) shows the two communities found in this network when edge direction is taken into account. The teams are shaded according to whether they won or lost a majority of their (within-conference) games and, as the figure shows, the two communities correspond precisely to these two groups in this case—the algorithm has divided the more successful and less successful teams into different communities using the information contained in the edge directions of the network. (Similar results are seen in the networks for other years.) If, however, we ignore the edge directions of the network and apply the undirected modularity algorithm, the method entirely fails to identify the two groups, as shown in Fig. 3.4(b), indicating that in this case a crucial part of the community information is contained in the edge directions.

3.4 Discussion

In summary, we have presented a method for detecting community structure in directed networks that makes explicit use of information contained in edge directions, information that most other algorithms discard. Our method is an extension of the established modularity maximization method widely used to determine community structure in undirected networks. We have applied the method to a variety of networks, both real and simulated, showing that it is able to recover known community structure and extract additional and revealing information not available to algorithms that ignore edge direction. The computational efficiency of the algorithm is essentially identical to that of the corresponding algorithm for undirected networks and hence we see no reason to use the undirected algorithm on directed graphs; we rec-

ommend the use of the full directed algorithm in all cases where researchers wish to analyze both edge placement and edge direction.

CHAPTER 4

Mixture models and exploratory analysis in networks

4.1 Introduction

In Chapter 1 we outlined the history of networks research and indicated that recent studies of networks are fundamentally different from earlier work due to the sheer scale of networks being analyzed. We showed that these large networks cannot be easily visualized in a way that allows for "analysis to be conducted by eye." Instead, we have been obliged to turn to topological measures, computer algorithms, and statistics to understand the structure of modern networks. In fact, much of the current research on networks is, in effect, aimed at answering the question "How can we tell what a network looks like, when we can't actually look at it?"

The typical approach to this problem involves defining measures or statistics to quantify network features of interest: centrality indices [88, 83], degree distributions [2, 27, 46], clustering coefficients [89], and community structure measurements [32, 20] are all invaluable tools for shedding light on the topology of networks. The first two chapters of this dissertation followed the traditional approach of identifying a structural feature of interest and then constructing a specialized method to detect this structure. Our reliance on measures like these, however, has a downside: they require us to know what we are looking for in advance, before we can decide what to measure. People measure clustering coefficients, for instance, because (presumably) they think there may be interesting clustering in a network; they measure

degree distributions because they believe the degree distribution may show interesting features. This approach has certainly worked well—many illuminating discoveries have been made this way. However, it raises an uncomfortable question: could there be interesting and relevant structural features of networks that we have failed to find simply because we haven't thought to measure the right thing?

To some extent this is an issue with the whole of scientific endeavor. In any field, thinking of the right question can demand as much insight as thinking of the answer. However, there are also things we can do to help ourselves. In this chapter we describe a technique that allows us to detect structure in network data while making only rather general assumptions about what that structure is. Methods of this type are referred to by statisticians as "exploratory" data analysis techniques, and we will make use of a number of ideas from the statistical literature in the developments that follow.

We focus on the problem of classifying or clustering the vertices of a network into groups such that the members of each group are similar in some sense. This already narrows the types of structure we consider substantially, but leaves a large and useful selection of types still in play. Some of these types of structure have been considered in the past, but the range of possibilities considered here is far larger than that of previous work. For instance, many researchers have examined community structure in networks—also called "homophily" or "assortative mixing"—in which vertices divide into groups such that the members of each are mostly connected to other members of the same group [32, 20]. "Disassortative mixing," in which vertices have most of their connections outside their group, has also been discussed to a lesser extent [37, 65, 26]. Effective techniques have been developed that can detect structure of both of these types. But, what should we do if we do not know in advance which type to expect, or if our network has some other type of structure entirely whose existence we are not even aware of? One can imagine an arbitrary number of other types of division

among the vertices of a network, most of which have probably never been considered explicitly in the past. One possibility, for instance, is a network in which, although there is no conventional assortative mixing, there are certain "keystone" vertices and group membership is defined by which particular keystone or set of keystones a vertex is connected to. Another possibility is a network in which there is both assortative and disassortative mixing between members of the same groups, the groups themselves being defined by the fact that their vertices have the same pattern of preferences and aversions, rather than by any overall assortative or disassortative behavior at the group level. And there are certainly many other possibilities. Such complex structures cannot be detected by the standard methods available to us at present, and moreover it seems unlikely in many cases that appropriate specialized detection methods will be developed because of the chicken-and-egg nature of the problem: we would have to know the form of the structure in question to develop such a method, but without a detection method, we cannot discover that form in the first place.

This chapter proposes a new approach to the analysis of structure network data that employs a broad and flexible definition of vertex classes, parameterized by an extensive number of variables and hence encompassing an essentially infinite variety of structural types in the limit of large network size. Certainly this definition includes the standard assortative and disassortative structures and, as we will see, the method we propose will detect those structures when they are present. However, it is also able to detect a wide variety of other structural types, including those described above as well as many others. Furthermore, it does so without requiring that we specify which particular structure we are looking for: the algorithm simultaneously finds the appropriate assignment of vertices to groups and the parameters defining the meaning of those groups, so that upon completion the calculation tells us not only the best way of grouping the vertices, but also the definitions of the groups themselves. Our method, which is based on the established numerical technique known as

the expectation-maximization algorithm, is also fast and simple to implement. We demonstrate the algorithm with applications to a selection of real-world networks and computer-generated test networks.

4.2 The method

The method we describe is based on a mixture model, a standard construct in statistics, although one that has not yet found wide use in studies of networks. The method works well for both directed and undirected networks, but is somewhat simpler in the directed case, so let us start there.

Suppose we have a network of n vertices connected by directed edges, such as a Web graph or a food web. We represent the network using our standard mathematical formulation in terms of an adjacency matrix with elements $A_{ij} = 1$ if there is an edge from j to i and 0 otherwise.

Suppose also that the vertices fall into some number c of classes or groups and let us denote by g_i the group to which vertex i belongs. We will assume that these group memberships are unknown to us and that we cannot measure them directly. In the language of statistical inference they are **hidden** or **missing** data. Our goal is to infer them from the observed network structure. (The number of groups c can also be inferred from the data using standard methods [1, 82], but for the moment we will treat it as given.) To infer the group memberships we adopt a standard approach for such problems: we propose a flexible (mixture) model for the groups and their properties, then vary the parameters of the model in order to find the best fit to the observed network.

The model we use is a stochastic one that parameterizes the probability of each possible configuration of group assignments and edges. We define θ_{ri} to be the probability that a (directed) link from a particular vertex in group r connects to vertex i. In the World Wide Web, for instance, θ_{ri} would represent the probability that a hy-

perlink from a Web page in group r links to Web page i. In effect θ_{ri} represents the "preferences" of vertices in group r about which other vertices they link to. In our approach, it is these preferences that define the groups: a "group" is a set of vertices that all have similar patterns of connection to others. [1] The idea is similar in philosophy to the block models proposed by White and others for the analysis of social networks [90], although the realization and the mathematical techniques employed are different.

We also define π_r be the (currently unknown) fraction of vertices in group or class r, or equivalently the probability that a randomly chosen vertex falls in r. The parameters π_r, θ_{ri} satisfy the normalization conditions

$$\sum_{r=1}^{c} \pi_r = 1, \qquad \sum_{i=1}^{n} \theta_{ri} = 1. \tag{4.1}$$

These quantities specify a network model flexible enough to describe many different types of structure. For instance, if θ_{ri} is larger than average for vertices i that are themselves members of group r, the model displays assortative mixing, with vertices being connected primarily within their own groups. Conversely, if θ_{ri} is large for vertices not in r we have disassortative or k-partite structure. And many other more complex types of structure are possible for other parameter choices.

The quantities in our theory thus fall into three classes: measured data $\{A_{ij}\}$, missing data $\{g_i\}$, and model parameters $\{\pi_r, \theta_{ri}\}$. To simplify the notation we will henceforth denote by A the entire set $\{A_{ij}\}$ and similarly for $\{g_i\}$, $\{\pi_r\}$, and $\{\theta_{ri}\}$.

The standard framework for fitting models like the one above to a given data set is likelihood maximization, in which one maximizes with respect to the model parameters the probability that the data were generated by the given model. Maximum likelihood methods have occasionally been employed in network calculations in the

[1] We could alternatively base our calculation on the patterns of ingoing rather than outgoing links and for some networks this may be a useful approach. The mathematical developments are entirely analogous to the case presented here.

past [41, 17, 35], as well as in many other problems in the study of complex systems. In the present case, our fitting problem requires us to maximize the likelihood $\Pr(A, g|\pi, \theta)$ with respect to π and θ, which can be done by writing

$$\Pr(A, g|\pi, \theta) = \Pr(A|g, \pi, \theta) \Pr(g|\pi, \theta), \tag{4.2}$$

where

$$\Pr(A|g, \pi, \theta) = \prod_{ij} \theta_{g_j,i}^{A_{ij}}, \quad \Pr(g|\pi, \theta) = \prod_j \pi_{g_j}, \tag{4.3}$$

so that the likelihood is

$$\Pr(A, g|\pi, \theta) = \prod_j \biggl[\pi_{g_j} \prod_i \theta_{g_j,i}^{A_{ij}} \biggr]. \tag{4.4}$$

In fact, one commonly works not with the likelihood itself but with its logarithm:

$$\mathcal{L} = \ln \Pr(A, g|\pi, \theta) = \sum_j \biggl[\ln \pi_{g_j} + \sum_i A_{ij} \ln \theta_{g_j,i} \biggr]. \tag{4.5}$$

The maximum of the two functions is in the same place, since the logarithm is a monotonically increasing function.

Unfortunately, g is unknown in our case, which means the value of the log-likelihood is also unknown. We can, however, usually make a good guess at the value of g given the network structure A and the model parameters π, θ. More specifically, we can, as shown below, calculate the probability distribution $\Pr(g|A, \pi, \theta)$ and from it calculate an expected value $\overline{\mathcal{L}}$ for the log-likelihood by averaging over g, thus:

$$\begin{aligned} \overline{\mathcal{L}} &= \sum_{g_1=1}^{c} \ldots \sum_{g_n=1}^{c} \Pr(g|A, \pi, \theta) \sum_j \biggl[\ln \pi_{g_j} + \sum_i A_{ij} \ln \theta_{g_j,i} \biggr] \\ &= \sum_{jr} \Pr(g_j = r|A, \pi, \theta) \biggl[\ln \pi_r + \sum_i A_{ij} \ln \theta_{ri} \biggr] \\ &= \sum_{jr} q_{jr} \biggl[\ln \pi_r + \sum_i A_{ij} \ln \theta_{ri} \biggr], \end{aligned} \tag{4.6}$$

where to simplify the notation we have defined $q_{jr} = \Pr(g_j = r|A, \pi, \theta)$, which is the probability that vertex j is a member of group r. (In fact, it is precisely these probabilities that will be the principal output of our calculation.)

This expected log-likelihood represents our best estimate of the value of $\mathcal{L}$ and the position of its maximum represents our best estimate of the most likely values of the model parameters. Finding this maximum still presents a problem, however, since the calculation of q requires the values of π and θ, while the calculation of π and θ requires q. The solution is to adopt an iterative, self-consistent approach that evaluates both simultaneously. This type of approach, known as an expectation-maximization or EM algorithm, is common in the literature on missing data problems. In its modern form it is usually attributed to Dempster *et al.* [21], who built on theoretical foundations laid previously by a number of other authors [55].

Following the conventional development of the method, we calculate the expected probabilities q of the group memberships given π, θ, and A thus:

$$q_{ir} = \Pr(g_i = r|A, \pi, \theta) = \frac{\Pr(A, g_i = r|\pi, \theta)}{\Pr(A|\pi, \theta)}. \tag{4.7}$$

The factors on the right are given by summing over the possible values of g in Eq. (4.4),

$$\begin{aligned} \Pr(A, g_j = r|\pi, \theta) &= \sum_{g_1=1}^{c} \cdots \sum_{g_n=1}^{c} \delta_{g_j,r} \Pr(A, g|\pi, \theta) \\ &= \sum_{g_1=1}^{c} \cdots \sum_{g_n=1}^{c} \delta_{g_j,r} \prod_k \left[\pi_{g_k} \prod_i \theta_{g_k,i}^{A_{ik}} \right] \\ &= \left[\pi_r \prod_i \theta_{ri}^{A_{ij}} \right] \left[\prod_{k \neq j} \sum_{s=1}^{c} \pi_s \prod_i \theta_{si}^{A_{ik}} \right], \end{aligned} \tag{4.8}$$

and

$$\begin{aligned} \Pr(A|\pi, \theta) &= \sum_{g_1=1}^{c} \cdots \sum_{g_n=1}^{c} \Pr(A, g|\pi, \theta) \\ &= \prod_k \sum_{s=1}^{c} \pi_s \prod_i \theta_{si}^{A_{ik}}, \end{aligned} \tag{4.9}$$

where δ_{ij} is the Kronecker δ symbol. Substituting into Eq. (4.7), we then find

$$q_{jr} = \frac{\pi_r \prod_i \theta_{ri}^{A_{ij}}}{\sum_s \pi_s \prod_i \theta_{si}^{A_{ij}}}. \tag{4.10}$$

Note that q_{jr} correctly satisfies the normalization condition $\sum_r q_{jr} = 1$.

Once we have the values of the q_{jr}, we can use them to evaluate the expected log-likelihood, Eq. (4.6), and to find the values of π, θ that maximize it. One advantage of the current approach now becomes clear: because the q_{jr} are known, fixed quantities, the maximization can be carried out purely analytically, obviating the need for numerical techniques such as Markov chain Monte Carlo. Introducing Lagrange multipliers to enforce the normalization conditions, Eq. (4.1), and differentiating, we find that the maximum of the likelihood occurs when

$$\pi_r = \frac{1}{n} \sum_j q_{jr}, \qquad \theta_{ri} = \frac{\sum_j A_{ij} q_{jr}}{\sum_j k_j q_{jr}}, \tag{4.11}$$

where $k_j = \sum_i A_{ij}$ is the out-degree of vertex j and we have explicitly evaluated the Lagrange multipliers using the normalization conditions. The application of Lagrange multipliers is covered more fully in Section 5.2 of the next chapter.

Equations (4.10) and (4.11) define our expectation-maximization algorithm. Implementation of the algorithm consists merely of iterating these equations to convergence and the output is the probability q_{ir} for each vertex to belong to each group, plus the probabilities θ_{ri} of links from vertices in each group to every other vertex, the latter effectively giving the definitions of the groups. The calculation converges rapidly in practice: typical runtimes for the networks studied were fractions of a second. (Some theoretical results are known for convergence of algorithms in this class, see Dempster *et al.* [21] and Wu [93].)

The obvious choice of starting values for the iteration is the symmetric choice $\pi_r = 1/c$, $\theta_{ri} = 1/n$, but unfortunately these values are a trivial (unstable) fixed point of Eqs. (4.10) and (4.11). In our calculations we have instead used starting conditions that are perturbed randomly a small distance from this fixed point. A random starting condition also gives us an opportunity to assess the robustness of our results. Except in special cases (such as the trivial fixed point above), EM algorithms are known to converge to local maxima of the likelihood [55] but not always to global maxima,

and hence it is possible to get different solutions from different starting points. The method works well in cases where it frequently converges to the global maximum or where it converges to local maxima that are close to the global maximum, giving good if not perfect solutions on most runs. In practice, we find for some networks that the method almost always converges to the same solution or a very similar one, whereas for others it is necessary to perform several runs with different initial conditions to find a good maximum of the likelihood. In the calculations presented in this chapter, we have in each case taken the division of the network giving the highest likelihood over the runs performed.

The developments so far apply to the case of a directed network. Most of the networks studied in the recent literature, however, are undirected. The model used above is inappropriate for the undirected case because its edges represent an inherently asymmetric, directed relationship between vertices in which one vertex chooses unilaterally to link to another, the receiving vertex having no say in the matter. The edges in an undirected network, by contrast, usually represent symmetric relationships. In a social networks of friendships, for instance, the edges would typically be drawn as undirected because two people can become friends only if both choose to be friendly towards the other. To extend our method to undirected networks we need to incorporate this symmetry into our model, which we do as follows. Once again, we define θ_{ri} to be the probability that a vertex in group r "chooses" to link to vertex i, but we now specify that a link will be formed only if two vertices both choose each other. Thus, the probability that an edge falls between vertices i and j, given that i is in group s and j is in group r, is $\theta_{ri}\theta_{sj}$, which is now symmetric. This probability satisfies the normalization condition $\sum_{ij}\theta_{ri}\theta_{sj} = 1$ for all r, s and setting $r = s$ we find

$$\sum_{ij}\theta_{ri}\theta_{rj} = \Bigl[\sum_i \theta_{ri}\Bigr]^2 = 1, \tag{4.12}$$

and hence $\sum_i \theta_{ri} = 1$ as before.

Now the probability $\Pr(A|g,\pi,\theta)$ in Eq. (4.4) is given by

$$\Pr(A|g,\pi,\theta) = \prod_{i>j}\Big[\theta_{g_i,j}\theta_{g_j,i}\Big]^{A_{ij}} = \prod_{ij}\theta_{g_j,i}^{A_{ij}}, \tag{4.13}$$

exactly as in the directed case, where we have made use of the fact that $A_{ji} = A_{ij}$ for an undirected network. (We have also assumed there are no self-edges in the network, edges that connect a vertex to itself, so that $A_{ii} = 0$ for all i.)

The remainder of the derivation now follows as before and results in precisely the same Eqs. (4.10) and (4.11), for the final algorithm.

4.3 Example applications

We now apply our method to a series of example networks. We use both real network data and simulated network data to highlight the strengths of our algorithm.

4.3.1 Karate club network

The first example is the much-discussed "karate club" network of friendships between 34 members of a karate club at a U.S. university, assembled by Zachary [94] by direct observation of the club's members. This network is of particular interest because the club split in two during the study as a result of an internal dispute, and Zachary recorded the membership of the two factions after the split.

Figure 4.1 shows the best division of this network into two groups found using the EM method with $c = 2$. The shades of the vertices in the figure represent the values of the variables q_{i1} for each vertex on the scale shown (or equivalently the values of q_{i2}, since $q_{i1} + q_{i2} = 1$ for all i). As we can see, the algorithm assigns most of the vertices strongly to one group or the other; in fact, all but 13 vertices are assigned 100% to one of the groups. Thus, the algorithm finds a strong split into two clusters in this case, and indeed if one simply divides the vertices according to the cluster to which each is most strongly assigned, the result corresponds perfectly to the division observed in real life (denoted by the shaded regions in the figure).

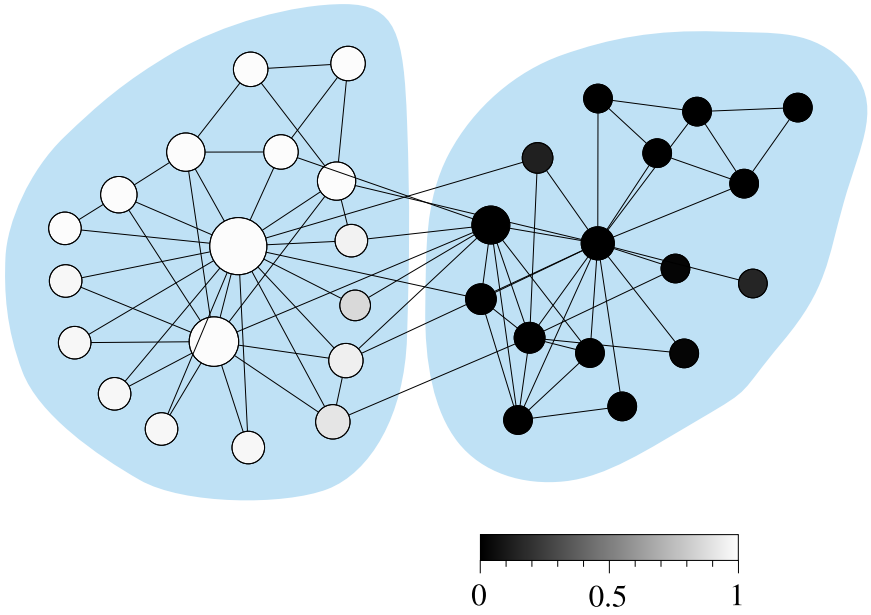


Figure 4.1. Application of the method described here to the "karate club" network of Ref. [94]. The two shaded regions indicate the division of the network in the real world, while the shades of the individual vertices indicate the decomposition chosen by the algorithm. The sizes of the vertices indicate the probabilities θ_{1i}, for each vertex to be connected to vertices in the left group, with the probabilities ranging from 0 for the smallest vertices to 0.19 for the largest.

However, the algorithm reveals much more about the network than this. First, where appropriate, it can return probabilities for assignment to the two groups that are not 0 or 1 but lie somewhere between these limits, and for 13 of the vertices in this network it does so. For some of these 13 vertices the values of q_{ir} are still very close to 0 or 1, but for some they are not. Inspection of the figure reveals in particular a small number of vertices with intermediate shades of gray along the border between the groups. There has been some discussion in the recent literature of methods for divining "fuzzy" or overlapping groups in networks; rather than dividing a network sharply into groups, it is sometimes desirable to assign vertices to more than one group and a number of authors have proposed possible ways of doing this [78, 69, 7, 65]. The present algorithm offers an alternative method that is particularly attractive because of the clear definition of the overlap: the values of the q_{ir} give the precise probability that a vertex belongs to a specified group, given the observed network structure.

The algorithm also returns the distributions or preferences θ_{ri} for connections

from vertices in group r to each other vertex i. For instance, in Fig. 4.1 we indicate by the sizes of vertices the distribution θ_{1i} of connections from vertices in group 1, which is the left-hand group in the figure, to each other vertex. As we can see, two vertices central to the group have high connection probabilities, while some of the more peripheral vertices have smaller probabilities. Thus, the values of θ_{ri} behave as a kind of centrality measure, indicating how important a particular vertex is to a particular group. This could form the basis for a practical measure of within-group influence or attraction in social or other networks. Note that, in this case this measure is not high for vertices that are central to the other group, group 2; the measure is sensitive to the particular preferences of the vertices in just a single group.

4.3.2 Network of English words

In Fig. 4.2 we show the results of its application to an adjacency network of English words taken from Ref. [65]. In this network the vertices represent 112 commonly occurring adjectives and nouns in a particular body of text (the novel *David Copperfield* by Charles Dickens), with edges connecting any pair of words that appear adjacent to each other at any point in the text. Because adjectives typically occur next to nouns in English, most edges connect an adjective to a noun and the network is thus approximately bipartite or disassortative. This can be seen clearly in Fig. 4.2, where the two shaded groups represent the adjectives and nouns and most edges are observed to run between groups.

Analyzing this network using our algorithm we find the classification shown by the shades of the vertices. Once again, most vertices are assigned 100% to one class or the other, although there are a few ambiguous cases, visible as the intermediate shades of gray. As Fig. 4.2 makes clear, the algorithm's classification corresponds closely to the adjective/noun division of the words—almost all the black vertices are in one group and the white ones in the other. In fact, 89% of the vertices are correctly

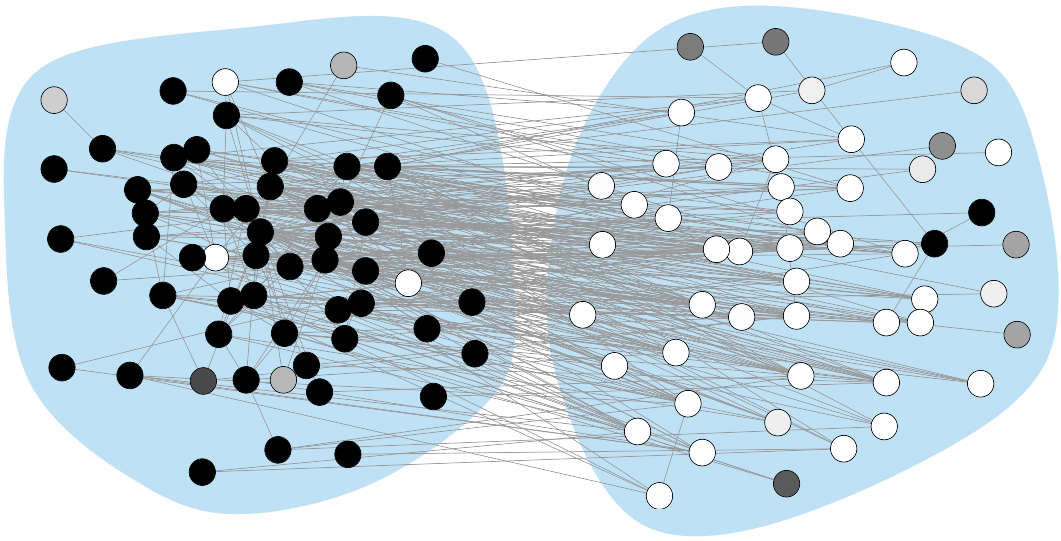


Figure 4.2. The network taken from [65] in which the vertices represent 112 commonly occurring adjectives and nouns in the novel *David Copperfield* by Charles Dickens. The edges connect any pair of words that appear adjacent to each other at any point in the text. Since adjectives typically occur next to nouns in English, most edges connect an adjective to a noun and thus run between groups making the network approximately bipartite. The two shaded regions represent the real groups of adjectives and nouns respectively and the shades of the individual vertices represent the classification found by the algorithm.

classified by our algorithm in this case.

The crucial point to notice, however, is that the algorithm is not merely able to detect the bipartite structure in this network, but it is able to do so without being told that it is to look for bipartite structure. The exact same algorithm, unmodified, finds both the assortative structure of Fig. 4.1 and the disassortative structure of Fig. 4.2. This is the strength of the present method: it is able to detect a wide range of structure types without knowing in advance what type is expected. Other methods are able to detect particular kinds of structure, and in many cases do a good job, but they tend to be narrowly tailored to that job. Typically a new method or algorithm has to be devised for each new structural type.

4.3.3 Simulated assortative and disassortative networks

We emphasize the ability of the algorithm to detect both assortative and disassortative community structure with Fig. 4.3, in which we show the results of the application

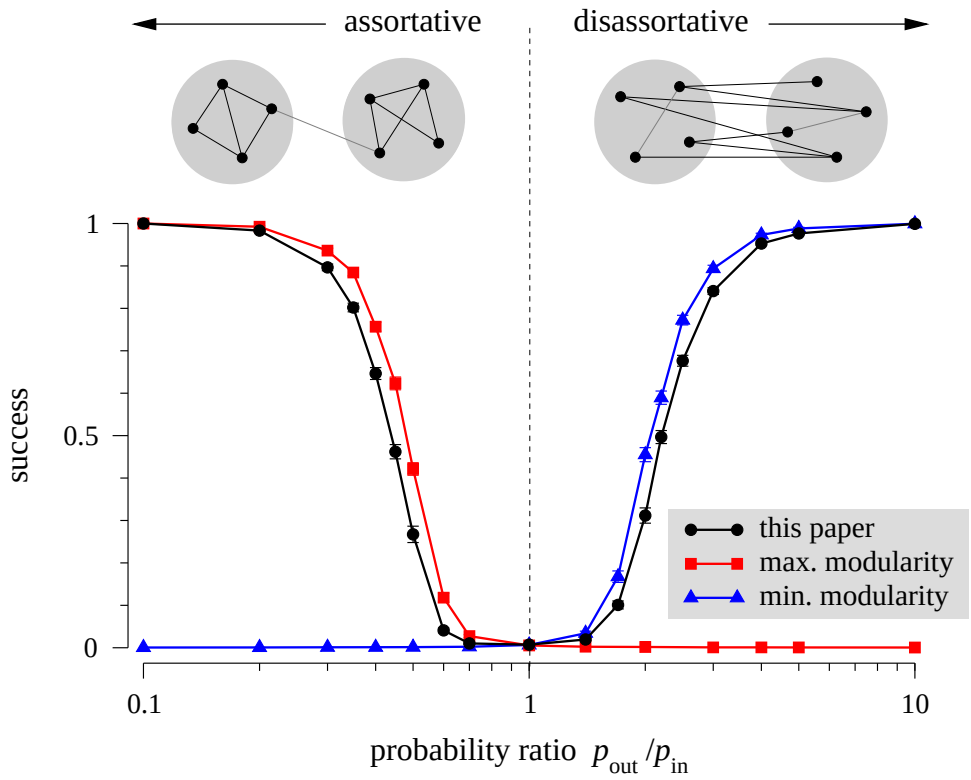


Figure 4.3. Results of the application of three algorithms to a set of computer generated networks with two groups each. The horizontal axis varies the structure of the networks from assortative to disassortative, while the vertical axis indicates the success of the algorithms at detecting the groups, as measured by the mutual information index of Danon *et al.* [20]. Each point is averaged over 100 network realizations.

of our method to a set of computer generated networks. In this test, we generated networks of fixed size $n = 128$, divided into two groups of 64 vertices each. Edges were placed between pairs of vertices in the same group with probability p_{in} and between pairs in different groups with probability p_{out}. We then varied the ratio $p_{\text{out}}/p_{\text{in}}$ of the two probabilities, while keeping the mean degree of all vertices fixed, in this case at 16. When $p_{\text{out}}/p_{\text{in}}$ takes values below 1, we thus produce a network with assortative mixing, while for values above 1 the network is disassortative.

Figure 4.3 shows how successful (or unsuccessful) our algorithm is in detecting the known groups in these networks, as quantified using the mutual information index of Danon *et al.* [20], which is 1 when the groups are identified perfectly and 0 when there is no correlation between the true groups and those found by the algorithm.

The circles (•) in the figure show the results for our algorithm and as we can see the algorithm successfully detects the known groups for values of $p_{\text{out}}/p_{\text{in}}$ both above and below 1, i.e., for both assortative and disassortative cases. When the ratio is close to 1, meaning that edges are placed without regard for the group structure, then, unsurprisingly, the algorithm is unable to detect the groups, since the network contains no signature of their presence.

The two other curves in the figure show the performance of the spectral modularity maximization (squares ■) and minimization (triangles ▲) algorithms of Ref. [65], which are designed specifically to detect assortative and disassortative structure respectively. Two interesting features deserve comment. (1) The specialized spectral algorithms slightly out-perform our maximum likelihood method on the tasks for which they were designed—they are able to detect structure for values of $p_{\text{out}}/p_{\text{in}}$ closer to 1. This is not surprising: the spectral algorithms are, in a sense, given more information to start with, since we tell them what type of structure to look for. The EM algorithm, on the other hand, is told very little about what to look for and has to work out more for itself. (2) The modularity-based algorithms, however, fail to detect any structure outside their domains of validity. The modularity maximization method is incapable of detecting the disassortative structure present for $p_{\text{out}}/p_{\text{in}} > 1$, and the minimization method is similarly incapable of detecting assortative structure. This illustrates the advantages of the present method as a flexible technique that detects whatever type of structure is present, rather than being focused on answering one specific question.

4.3.4 A directed social network

As we have seen, our method is applicable to both directed and undirected networks. In Fig. 4.4, we show an example application to a directed network, a social network of high school students taken from from the U.S. National Longitudinal Study of

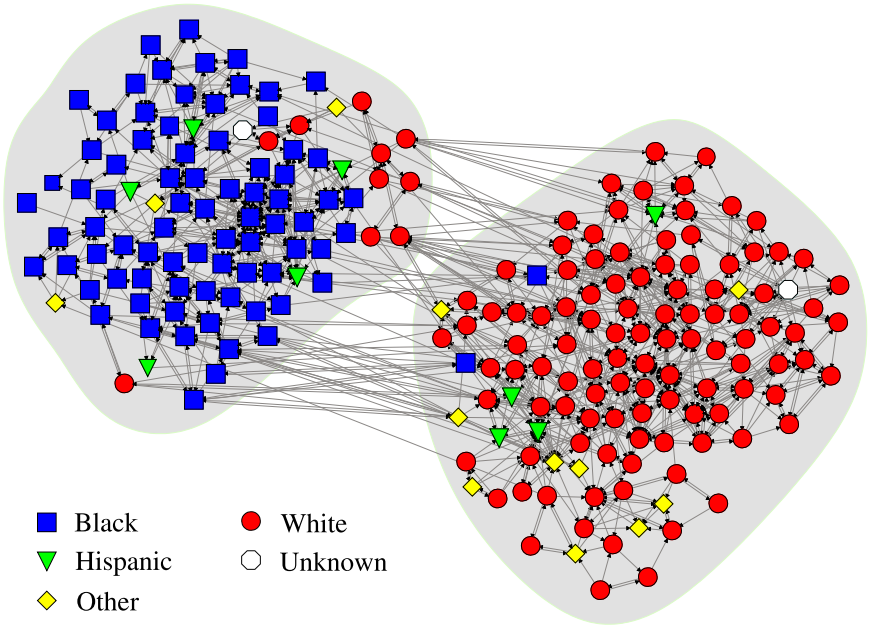


Figure 4.4. A directed social network of U.S. high school students and the division into two groups found by the directed version of our method. Vertex shapes show the (self-identified) ethnicity of the students.

Adolescent Health (the "AddHealth" study). Students were asked to identify their friends within the school and a response in which student A identifies B as a friend is represented as a directed edge from A to B. In contrast to the common view, discussed earlier, of friendship as a symmetric relationship running in both directions between the individuals it connects, a remarkable number of the friendships identified in this study, more than half, are found to run in only one direction, so that a directed representation of the network is indispensable for capturing the structure of the data.

Applying the directed version of our method to this network with $c = 2$ produces the division shown in Fig. 4.4. This example is striking because, like many of the networks in the AddHealth data set, the groupings are found to correlate strongly with student ethnicity as shown by the shapes of the vertices [60]. In this case, one of the two groups contains most of the black students in the school and the other most of the white students, with the few members of other ethnic groups distributed more evenly.

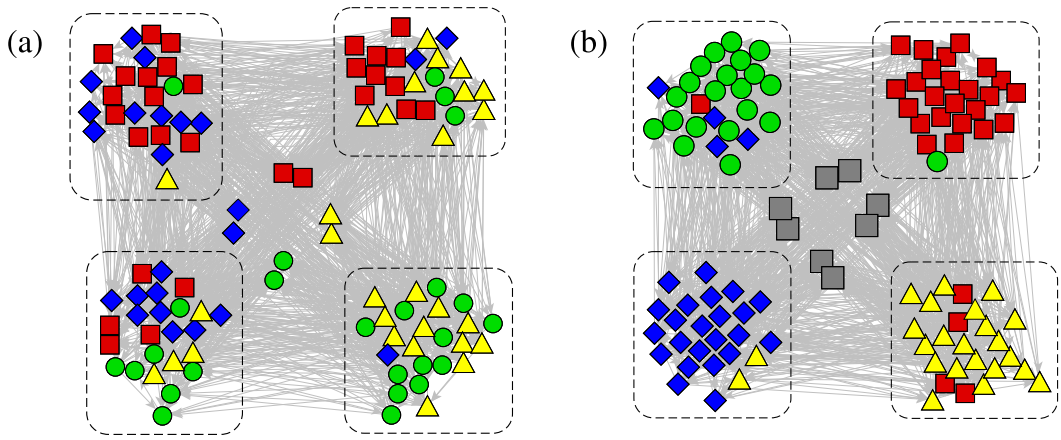


Figure 4.5. The four-group network described in the text, in which connections between vertices are entirely random, except for connections to the eight keystone vertices in the center. Each of the four groups (dashed boxes) is distinguished solely by the unique pattern of its connections to the keystone vertices. Vertex shapes represent the groups to which vertices are assigned by our analyses using (a) standard modularity maximization and (b) the maximum likelihood method of this chapter.

4.3.5 "Keystone" network

Finally, lest we give the impression that our method is suitable for detecting only assortative and disassortative mixing in networks, let us give one more example of a completely different kind. In Fig. 4.5 we show a four-group computer-generated network of a form mentioned in the introduction: there are a small number of "keystone" vertices in the network, and group membership affects only the propensity to link to these vertices. All other connections are purely random.

The network is again a directed one, with a total of 108 vertices. Of the 108 vertices, 100 are divided into four groups of 25 each, and directed edges are placed uniformly at random between them such that the mean degree (both in and out) is 10. The remaining 8 vertices are denoted keystone vertices and the other vertices link to them depending on their group membership. Specifically, the vertices in groups A, B, C, and D link to keystone vertices $\{1, 2, 3, 4\}$, $\{3, 4, 5, 6\}$, $\{5, 6, 7, 8\}$, and $\{7, 8, 1, 2\}$ respectively. Thus, no keystone vertex is uniquely identified with any group, but each group has a unique signature set of keystones. It is only the pattern of the keystone links that distinguishes the groups and nothing else. The network is not assortative

by the standard definition: the randomly placed edges fall within or between groups purely by chance, and the links to the keystones, although not random, are equally likely to fall within or between groups.

Figure 4.5(a) shows what happens when we analyze this network using a standard modularity maximization technique. The dashed boxes in the figure outline the four groups of vertices and the shapes show the group assignments found by the analysis. While the modularity maximization does find four groups, the groups found do not correspond to the known division of the network—each box contains a substantial number of vertices of at least two shapes and in some cases more. The maximum likelihood analysis, by contrast, has no difficulty in discerning the structure of the network. Figure 4.5(b) shows the results of applying our expectation-maximization algorithm with $c = 4$, and as we can see, the algorithm has, without any prior information on the type of structure contained in the network, discovered the structure and correctly assigned almost all of the vertices to their four groups. The 8 keystone vertices, which are shown in the center of Fig. 4.5(b), are not assigned to any group by the algorithm, but are instead divided (almost) equally between all four (meaning that q_{ir} is close to 0.25 for all r). Thus, the algorithm has, in effect, accurately deduced the five classes of vertices present in the network. Moreover, an examination of the final values of the model parameters θ will tell us exactly what type of structure the algorithm has discovered. In principle, considerably more complex structures than this can be detected as well.

4.4 Discussion

In this chapter, we have described a method for exploratory analysis of network data in which vertices are classified or clustered into groups based on the observed patterns of connections between them. The method is more general than previous clustering methods, making use of maximum likelihood techniques to classify vertices

and simultaneously determine the definitive properties of each class. The result is a simple algorithm that is capable of detecting a broad range of structural signatures in networks, including conventional community structure, bipartite or k-partite structure, fuzzy, or overlapping classifications, and many mixed or hybrid structural forms that have not been considered explicitly in the past. We have demonstrated the method with applications to a variety of examples, including real-world networks and computer-generated networks. The method's strength is its flexibility, which will allow researchers to probe observed networks for general types of structure without having to specify in advance what type they expect to find.

CHAPTER 5

Large-scale structure of time evolving citation networks

5.1 Introduction

Citation networks, the principal focus of this chapter, have been studied quantitatively almost from the moment citation databases first became available. The physicist-turned-science-historian Derek de Solla Price authored two celebrated papers in the 1960s and 1970s highlighting the power-law degree distributions in networks of scientific papers and developing models to explain their origin [72, 73]. A discussion of Price's work can be found in Section 1.2.3.

A citation network is an information network in which a vertex represents a document of some kind and an edge between two vertices represents the citation of one document by another. Citation networks differ from other networks in a number of important ways. First, they are directed: citations go from one document to another and hence constitute an inherently asymmetric relationship between the vertices involved. Mathematically, the network can be represented by an adjacency matrix of the kind described in Section 1.2.1. In a directed network the adjacency matrix is, in general, asymmetric.

A second feature of citation networks is that they evolve over time as new documents are created. The evolution of the network takes a special form, in that vertices and edges are added to the network at a specific time and cannot be removed later. This permanence of vertices and edges means that the structure of the network is

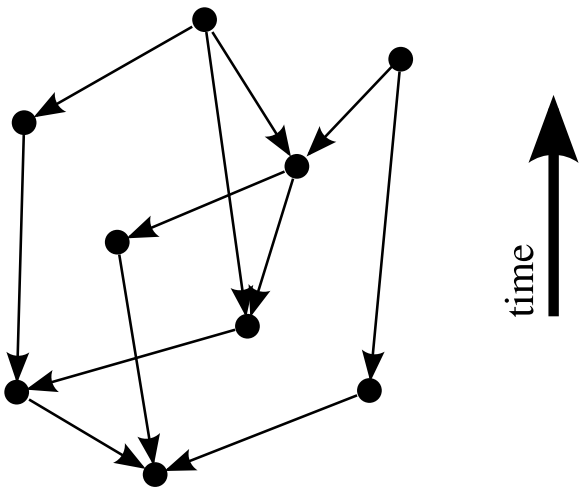


Figure 5.1. Citations run from vertices created at later times to those created at earlier times—in the opposite direction to the arrow of time.

mostly static: it changes only at the **leading edge** of the network, the time at which new documents are added. Citation networks differ in this respect from other information networks such as the World Wide Web, in which vertices and edges can be removed as well as added, and edges can be repositioned after they are added. This limited form of time evolution found in citation networks makes them, in some ways, a simpler and cleaner laboratory for the study of network growth than the Web.

The combination of the two features of citation networks described above leads to a third: citation networks are acyclic, meaning there are no closed loops of citations of the form A cites B cites C cites A, or longer. When a new vertex is added to a citation network it can cite any of the previously existing vertices, but it cannot cite vertices that have not yet been created. This gives the network a clear **arrow of time**, with all edges pointing backwards in time as shown in Fig. 5.1. As a result it is typically possible, starting from a given vertex, to find a path of citations that takes us back in time through the network, but it is not possible to find one that takes us forward again, so no closed loops exist. (Real citation networks are often not perfectly acyclic. For example, a scientific paper can sometimes cite work that is forthcoming but not yet published, resulting in a closed loop in the network. However, such loops

are rare and necessarily short, being limited by the narrow span of time over which it is possible to predict future publications. In practice, therefore, it is usually a good approximation to assume the network to be acyclic.)

Citation networks arise in a variety of different areas. We have mentioned networks of scientific citations, which have been studied by many authors since the classic work of Price mentioned above. (See, for instance, the book by Egghe and Rousseau [22] or any volume of the journal *Scientometrics*, which is entirely devoted to the quantitative analysis of scholarly authorship and citation patterns.) Citation networks of patents have, to a lesser extent, also been studied. Patents cite other patents for a variety of reasons, but most often to establish their originality and distinction from previous work. Extensive data on patent citations have become available in recent years, allowing the construction of very large citation networks [16, 39]. For example, the time evolution of the United States patent citation network has been studied recently by Csárdi *et al.* [19], although their approach to the analysis of temporal patterns, while both interesting and useful, is quite different from the one adopted in this chapter. Very recently, there has also been some interest in legal citation networks, networks of legal opinions written by judges and others, which cite one another to establish precedent [18, 30, 31]. We make extensive use of one particular legal citation network, the network of opinions of the United States Supreme Court, as an example in this chapter, although the techniques we will be considering are certainly applicable to other networks as well.

Given the wide interest in and unique structure of citation networks, it is instructive to investigate what can be learned from an analysis of the statistical patterns present in these networks. A variety of previous studies focused on relatively standard network measures such as degree distributions [72, 77, 84]. To investigate the time-dependent structure that is the special property of citation networks, however, other methods are needed. In this chapter we present several techniques that, as we

will show, are—both individually and collectively—capable of revealing interesting new structure in these networks.

5.2 A mixture model of citation patterns

The first analysis we describe makes use of a stochastic mixture model of the citation process, which is fitted to the observed network data using the likelihood optimization technique known as the expectation-maximization algorithm.

A crucial property affecting the structure of citation networks is the temporal pattern of citation to documents following their publication. It is interesting, for instance, to ask if there are typical patterns that documents follow. Are there more citations immediately after publication than later, or do they grow in frequency over time? Are documents more likely to cite recent precedents or older, better-established ones? Do documents tend to cite others published during a particular time period? There could also be more than one common pattern, with different documents following different patterns. If so, how can we determine those patterns, and how can we tell which pattern particular documents follow, given that citation data are inherently noisy?

As an example, we consider the network of legal citations between cases handed down by the Supreme Court of the United States, from its inception in 1789 until the present day. We will use this example throughout this chapter; it is well documented, shows clear and interesting structural signatures, and has been studied much less than other types of citation networks in the past. While we are using the network primarily as an illustrative device, the results we derive are, in many cases, of interest in their own right and not just as a demonstration of our methods.

Consider Table 5.1, which gives the dates of the citations received so far by a single example opinion handed down by the Supreme Court in the year 1900. We will take citation profiles such as this as the basic inputs in our analysis.

year	cites	year	cites	year	cites
1900	1	1907	2	1925	1
1901	4	1910	1	1936	1
1902	3	1912	2	1947	1
1904	1	1920	1		

Table 5.1. The number of citations per year received by a single opinion handed down by the Supreme Court in the year 1900.

One interesting question is whether there are distinct eras of citation in the history of this (or any) citation network. Are there, for instance, eras in which a certain set of documents are well cited, followed perhaps by another era or eras in which that set falls out of favor to be replaced by a different one? Many readers can probably think of anecdotal cases of behavior like this in scientific citation networks. Here we place these observations on a firm analytic foundation.

We will attempt to divide the vertices in a citation network into groups by identifying similarities in their citation profiles. Other methods have been proposed for identifying times of significant change in the structure of networks. Sun *et al.* [86] proposed a method for identifying **change-points** in time evolving networks (not only in citation networks). Their method utilizes information theoretic principles such as lossless compression schemes to identify change-points or times at which significant change in the structure of a network occur.

However, our method takes a very different approach. We define a set of citation profiles and then self-consistently assign each case to the profile it best fits while at the same time adjusting the shape of the profiles to best fit the cases assigned to them. The means by which we accomplish this task is the expectation–maximization (EM) algorithm [21, 55].

In Chapter 4 we described the application of this method to the classification of vertices in static networks, both directed and undirected. Here we describe a different application to the analysis of the temporal profiles of citations.

Suppose we have a network of n vertices representing our documents and we believe that they can be divided into c groups, each of which is characterized by a particular probability distribution of citations over time. This is a different method from the one presented in Chapter 4. Here we we are interested in the "time" at which a vertex is cited, and not which other vertex cited it. In Chapter 4 our interest was in which vertex pairs were connected by edges. Our approach to finding the groups will be to fit the network to a model consisting of two parts: (1) a set of **time profiles** $\{\theta_r(t)\}$, one for each group, such that $\theta_r(t)$ is the probability that a particular citation received by a document in group r is made during year t; (2) a set of **probabilities** π_r, such that π_r is the probability that a randomly chosen document belongs to group r. (Here the π_r's represent the same thing as they did in Chapter 4). Just as in Chapter 4 we fit this model to the observed data. However, in this case the observed data are the citations over time.

Suppose that document i belongs to group g_i and let $z_i(t)$ be the number of citations that the document receives in year t. Then the probability that document i received the particular citations it did and is in group g_i, given the model parameters, is

$$\Pr(z_i, g_i|\pi, \theta) = \Pr(z_i|g_i, \pi, \theta) \Pr(g_i|\pi, \theta). \tag{5.1}$$

Again, we use π, θ to denote the entire set $\{\pi_r, \theta_r\}$. Assuming random and uncorrelated citations drawn from the time profile $\theta_{g_i}(t)$, the terms on the right-hand side are given by

$$\Pr(z_i|g_i, \pi, \theta) = k_i! \prod_{t=t_1}^{t_2} \frac{[\theta_{g_i}(t)]^{z_i(t)}}{z_i(t)!}, \tag{5.2}$$

$$\Pr(g_i|\pi, \theta) = \pi_{g_i}, \tag{5.3}$$

where $k_i = \sum_t z_i(t)$ is the in-degree of document i, i.e., the total number of citations it receives, and t_1 and t_2 are the first and last years of data in our dataset.

Now taking the product over all vertices, the likelihood of the entire data set is

$L = \prod_{i=1}^{n} \Pr(z_i, g_i|\pi, \theta)$. Of course, we will again work with the logarithm $\mathcal{L}$ of the likelihood,

$$\mathcal{L} = \ln L = \sum_{i=1}^{n} \Big[\ln \Pr(g_i|\pi, \theta) + \ln \Pr(z_i|g_i, \pi, \theta) \Big]. \tag{5.4}$$

Just as in the previous EM based method, if we write the probability of a particular assignment of vertices to groups as $\Pr(\{g_i\}|z, \pi, \theta)$, we can then calculate the expected value of the log-likelihood as the average of Eq. (5.4) over all possible assignments thus:

$$\begin{aligned}
\overline{\mathcal{L}} &= \sum_{g_1=1}^{c} \cdots \sum_{g_n=1}^{c} \Pr(\{g_i\}|z, \pi, \theta)\, \mathcal{L} \\
&= \sum_{g_1=1}^{c} \cdots \sum_{g_n=1}^{c} \Pr(\{g_i\}|z, \pi, \theta) \\
&\qquad \times \sum_{i=1}^{n} \Big[\ln \Pr(g_i|\pi, \theta) + \ln \Pr(z_i|g_i, \pi, \theta) \Big] \\
&= \sum_{i=1}^{n} \sum_{r=1}^{c} \Pr(g_i = r|z_i, \pi, \theta) \\
&\qquad \times \Big[\ln \Pr(g_i = r|\pi, \theta) + \ln \Pr(z_i|g_i = r, \pi, \theta) \Big] \\
&= \sum_{i=1}^{n} \sum_{r=1}^{c} q_{ir} \Big\{ \ln \pi_r + \ln k_i! + \\
&\qquad\qquad \sum_{t=t_1}^{t_2} \Big[z_i(t) \ln \theta_r(t) - \ln z_i(t)! \Big] \Big\},
\end{aligned} \tag{5.5}$$

where we have introduced the shorthand notation

$$q_{ir} = \Pr(g_i = r|z_i, \pi, \theta) \tag{5.6}$$

for the probability that vertex i belongs to group r, given the model and the observed citation pattern.

Again, this expected log-likelihood represents our best estimate of the value of the log-likelihood given what we know about the system. By maximizing it, we can now calculate a best estimate of the most likely values of the model parameters a process that involves two steps almost identical to those in Chapter 4: first, we estimate

the group membership probabilities q_{ir}; second, we use those probabilities in the maximization of $\overline{\mathcal{L}}$. We take these steps in turn.

To calculate the q_{ir} we observe that

$$q_{ir} = \Pr(g_i = r|z_i, \pi, \theta) = \frac{\Pr(z_i, g_i = r|\pi, \theta)}{\Pr(z_i|\pi, \theta)}. \tag{5.7}$$

The two factors on the right can be determined by summing Eq. (5.1) over the appropriate sets of variables and making use of Eqs. (5.2) and (5.3) to give

$$q_{ir} = \frac{\pi_r \prod_t [\theta_r(t)]^{z_i(t)}}{\sum_k \pi_k \prod_t [\theta_k(t)]^{z_i(t)}}. \tag{5.8}$$

Once we have this expression, we can use it to evaluate the log-likelihood, Eq. (5.5), and hence to find the values of the model parameters that maximize the likelihood. The maximization is again helped by the fact that π_r and θ_r enter Eq. (5.5) in independent terms. Considering π_r first and noting that it must satisfy the normalization condition $\sum_r \pi_r = 1$, we introduce a Lagrange multiplier α and then differentiate, holding q_{ir} constant, to get

$$\begin{aligned} 0 &= \frac{\partial}{\partial \pi_r}\left\{\sum_{ir} q_{ir} \ln \pi_r + \alpha\left[1 - \sum_r \pi_r\right]\right\} \\ &= \frac{1}{\pi_r}\sum_{i=1}^{n} q_{ir} - \alpha. \end{aligned} \tag{5.9}$$

Rearranging this expression gives

$$\pi_r = \frac{1}{\alpha}\sum_{i=1}^{n} q_{ir}. \tag{5.10}$$

The Lagrange multiplier α is then fixed by the condition $\sum_r \pi_r = 1$, and so

$$\sum_{r=1}^{c} \pi_r = 1 = \frac{1}{\alpha}\sum_{ir} q_{ir} = \frac{n}{\alpha}, \tag{5.11}$$

where we have made use of $\sum_r q_{ir} = 1$. Thus π_r is given by

$$\pi_r = \frac{1}{n}\sum_i q_{ir}. \tag{5.12}$$

In other words, the prior probability of a vertex belonging to group r is just the average over all vertices of the conditional probability of belonging to group r. This equation for π_r is identical to Eq. (4.11).

Similarly, the θ_r satisfy the normalization condition $\sum_t \theta_r(t) = 1$ for all r, so we introduce a set of c Lagrange multipliers $\{\beta_r\}$ and write

$$\frac{\partial}{\partial\theta_r(t)}\biggl\{\sum_{ir} q_{ir} \sum_{t=t_1}^{t_2} z_i(t) \ln\theta_r(t) + \sum_r \beta_r\biggl[1 - \sum_t \theta_r(t)\biggr]\biggr\} = 0. \tag{5.13}$$

Again holding q_{ir} constant and employing Eq. (5.2), we find

$$\sum_i q_{ir} \frac{z_i(t)}{\theta_r(t)} - \beta_r = 0, \tag{5.14}$$

or

$$\theta_r(t) = \frac{\sum_i q_{ir} z_i(t)}{\sum_i q_{ir} k_i}, \tag{5.15}$$

where we have evaluated β_r using the normalization condition and the fact that $\sum_t z_i(t) = k_i$ by definition.

To calculate the optimal values of the model parameters, as well as the group membership variables q_{ir} we again turn to numerical iteration. Starting from an initial guess about the values of $\{\pi_r, \theta_r(t)\}$, we evaluate Eq. (5.8) and then use the results to make an improved estimate of the model parameters from Eqs. (5.12) and (5.15). Under reasonable conditions this process is known to converge upon iteration to a self-consistent solution.

5.2.1 Example

As a demonstration of this EM method, we have applied it to the citation network of Supreme Court cases described in Section 5.2. Applied to this network, the algorithm will divide the network into any requested number c of groups, such that each group is characterized by a distinctive pattern of citations to cases in that group. We have

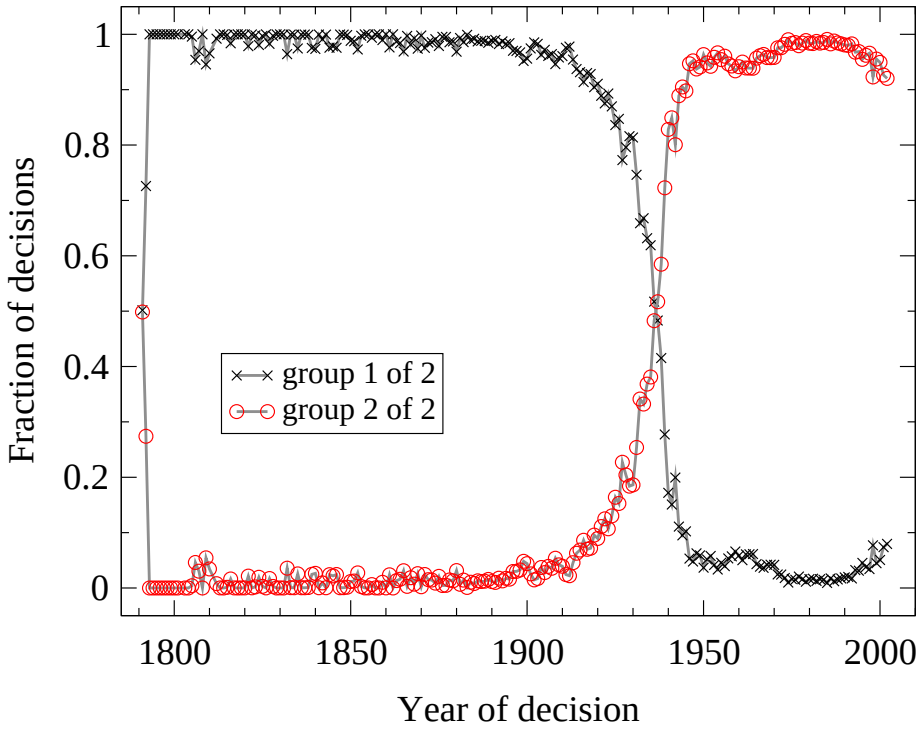


Figure 5.2. Results of the application of the EM analysis with $c = 2$ to the network of citations between Supreme Court opinions. The two curves show the fraction of cases assigned to each of the two groups found, as a function of time.

performed the analysis for a variety of different values of c. We begin with the simplest case, $c = 2$, of division into two groups. Starting with random initial values for $\{\pi_r, \theta_r\}$ and applying the EM iteration, Eqs. (5.8), (5.12), and (5.15), the parameters rapidly converge to a clear split of the network into two groups. Figure 5.2 shows the fraction of cases assigned by the algorithm to each of the groups as a function of time. Cases are assigned in proportion to their probability of membership in each of the groups so that, for instance, a case belonging to group 1 with probability 0.7 and to group 2 with probability 0.3 contributes 0.7 of a case to the first group and 0.3 of a case to the second.

Figure 5.2 reveals a dramatic split between the two groups: the best fit, in the maximum likelihood sense, of the mixture model with two groups to these data produces one group containing practically all cases before 1937 and another containing practically all cases after. This breakpoint coincides with a significant constitutional

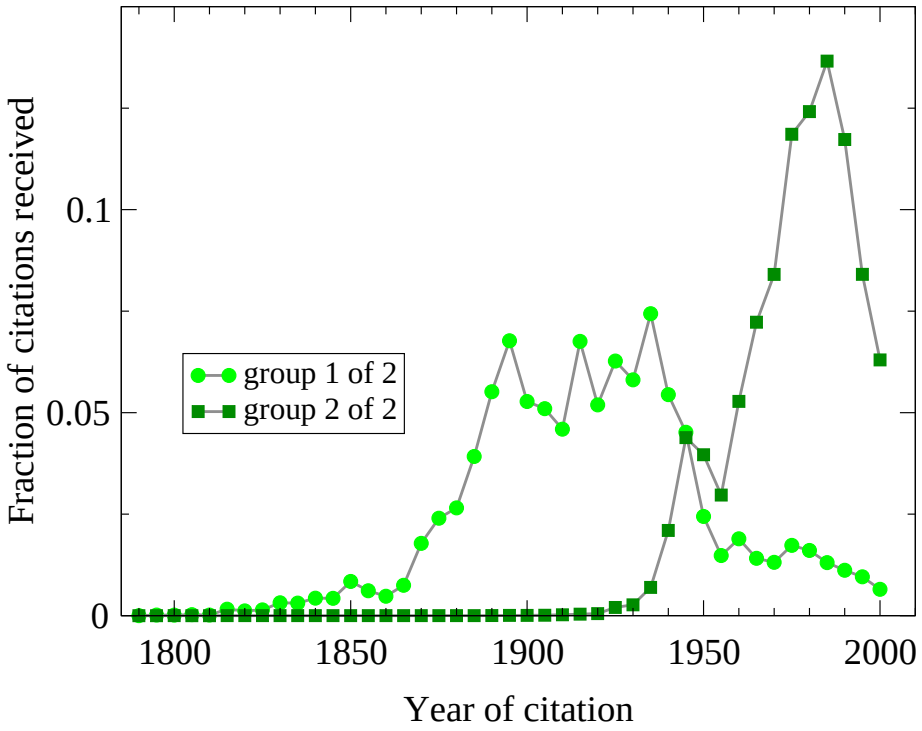


Figure 5.3. The citation profiles $\theta_r(t)$ generated by the EM algorithm with $c = 2$ for the Supreme Court citation network.

crisis for the Supreme Court. For the interested reader, we provide some further analysis in Section 5.5.

The EM algorithm tells us, in this case, that the Supreme Court's rulings split quite cleanly into groups with distinct citation profiles. That is, the opinions of the court can be distinguished sharply by the cases that later cited them. The citation profiles themselves, meaning the temporal citation patterns represented by the parameters $\{\theta_r\}$ in the model, are shown in Fig. 5.3. As we can see, each profile covers a distinct time period. The time period covered by of each profile also correspond closely to time spanned by the groups depicted in Fig. 5.2. This implies that the opinions that cite cases in each of our groups were handed down during roughly the same eras as the cited cases. This is not surprising if one assumes that the group divisions reflect different legal ideologies, but it is important to bear in mind that our analysis does not require it: it would be perfectly possible to detect groups that

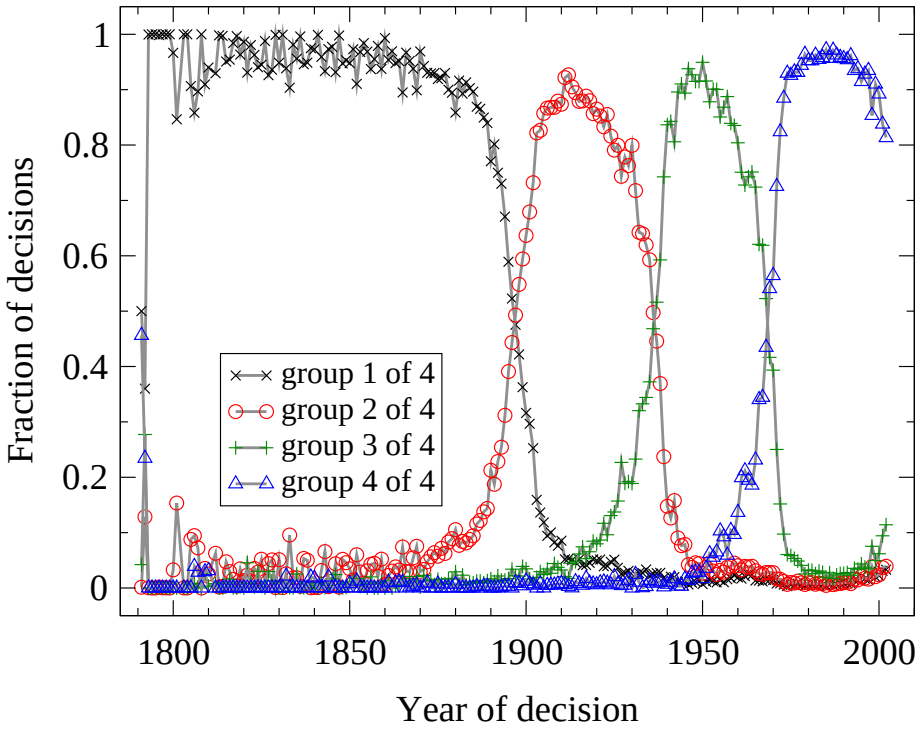


Figure 5.4. Results of the application of the EM analysis with $c = 4$ to the network of citations between Supreme Court opinions.

were distinguished by citations received during some entirely different era of the court arbitrarily later in its history, or even in no era at all but scattered widely over time.

We can also ask about best fits to the model for numbers of groups c greater than two. It is always the case that larger values of c will give better fits to the data, since larger values give us more parameters to fit with, but we must be wary of overfitting. In practice, we have been able to extract useful information about networks by comparing the results for a variety of small values of c. Rigorous methods for deciding optimal values of c, such as minimum description length, methods based on approximations to the marginal likelihood, or information theoretic measures, have been developed for other applications of the EM algorithm [1, 82]. For the moment we simply describe the results for various values of c.

Figure 5.4 shows results for the Supreme Court network with $c = 4$. The method again finds clear groups of cases, and as in the $c = 2$ case they are strongly delineated

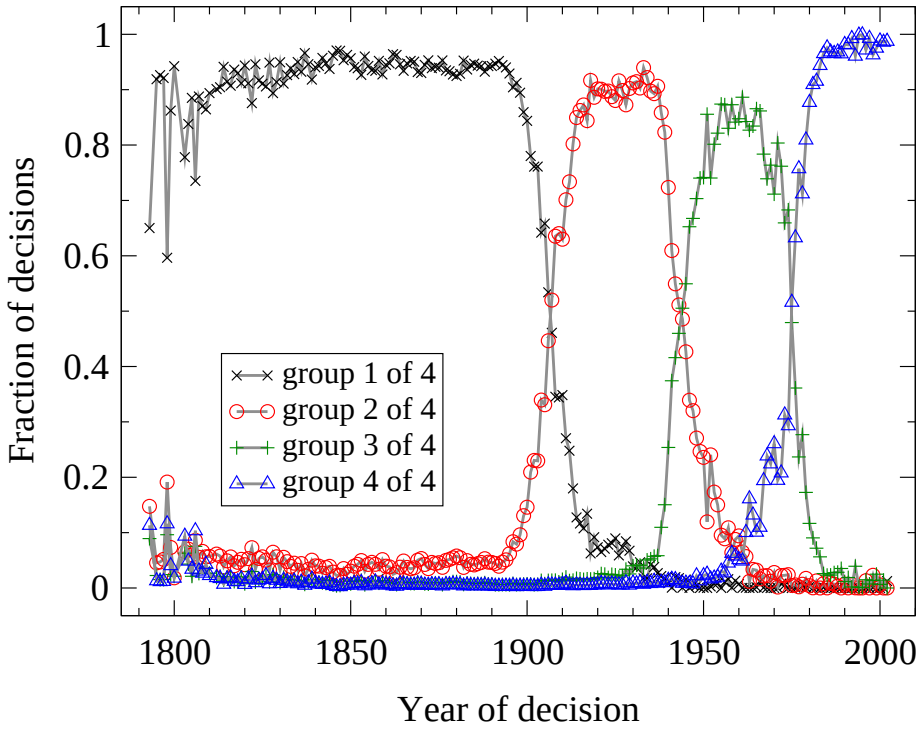


Figure 5.5. Results of the application of the EM algorithm with $c = 4$ to data for citations *made* (rather than received) by opinions in our Supreme Court dataset. The groups found are quite similar to those for the analysis based on citations received.

according to the dates of the opinions and thus appear to offer evidence for the presence of distinct eras in the Court's history. In particular, the analysis finds a clear grouping of cases between 1897 and 1937, corresponding approximately to the so-called *Lochner* era of Supreme Court jurisprudence, the significance of which is described in Section 5.5.

In these analyses we have characterized our documents by the pattern of citations they receive. However, one can just as easily look at the pattern of citations that documents *make* and this also, at least in some cases, can be a useful cue for detecting patterns in the network. The EM algorithm can be applied to this analysis as well. The developments are identical and the same computer code can be used. Figure 5.5, for example, shows the results of the application of this method to citations made by the opinions in our Supreme Court dataset, with $c = 4$. As the figure shows, the results are remarkably similar to those for citations received: it appears that, in this

case at least, there is a high degree of agreement about how cases should be classified into eras. This could indicate agreement between the opinions' writers and those that came after them, about the position staked out by individual opinions within the larger body of literature represented in our data set.

5.3 Clustering in citation networks

The general problem of the division of networks into groups of related vertices has been studied extensively in the past. The classic problem of clustering or community detection, which we introduced in Section 1.2.7, is to find groups of vertices within networks that have a higher than average density of internal edges and relatively few connections to these rest of the network. The second analysis technique we investigate for citation networks is a clustering method of this kind. As we will see, it is instructive to compare the results with those of our EM analysis in the previous section. The two methods do not do the same thing: the EM analysis groups together vertices that have similar time profiles to their citations, while the community analysis groups together vertices that are specifically linked to one another by edges. Nonetheless, as we will show, the two approaches can produce similar outcomes. One instance of this is the example of the Supreme Court data set.

Considerable effort has been devoted to the development of methods to find community structure within networks. Here we make use a method proposed by Newman [66]. This is, in fact, the same framework for community structure detection as we presented in Chapter 3, but for undirected networks. We choose the method for undirected networks as it seems reasonable to consider edges in a citation network to be a sign of connection between documents, and that connection exists regardless of the direction the edge runs in. If we did not make this choice, we would have to modify the clustering method to account for the time evolution of the network. So we simply ignore the directions in our analysis and apply the eigenvector calculation

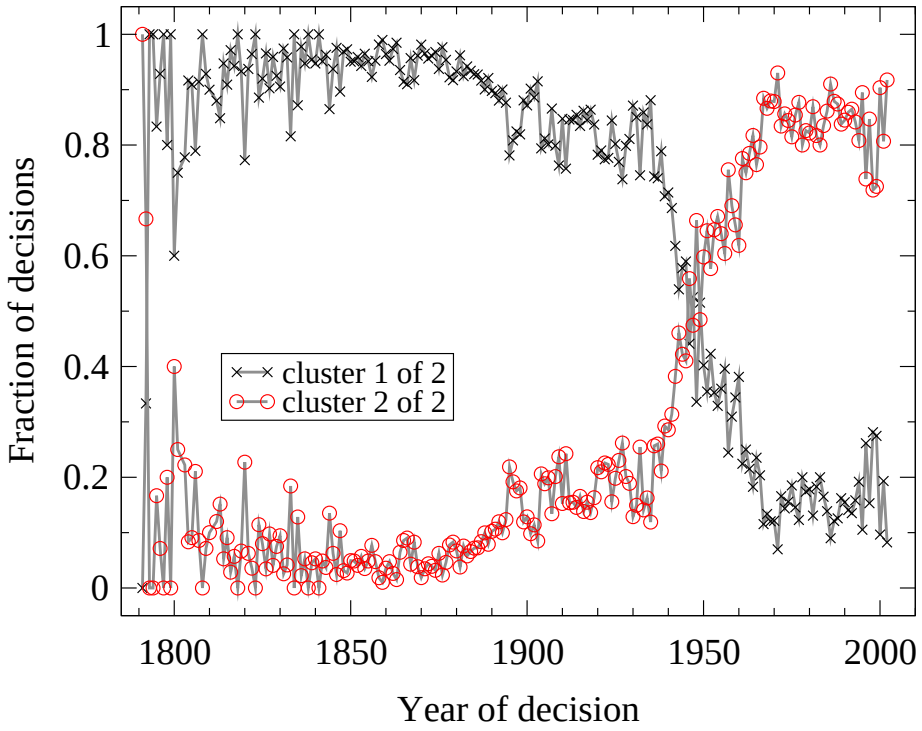


Figure 5.6. A histogram of the number of decisions versus the year of the decision for cases assigned to each group in the two-way split produced by the modularity maximization algorithm.

to the undirected network. This approach has been taken before by other authors and appears to work well—see, for example, Ref. [48].

We should recall that this method involved the repeatedly subdivision of the network into smaller and smaller groups. This aspect of the method is particularly attractive for the purposes of our present analysis, because it allows us to observe the major divisions in the network first, followed by more minor ones, and to stop the process at any point to compare with our other analyses.

We can visualize the results of our clustering analysis in a manner similar to our visualizations of the output of the EM algorithm, as a histogram over time. The results for the leading split of the Supreme Court network into two clusters are depicted in this way in Fig. 5.6. The results are similar to those for the EM algorithm, with a significant break around 1937. This appears to bolster the conclusions of our EM analysis: there have been separate periods in the Court's history that left

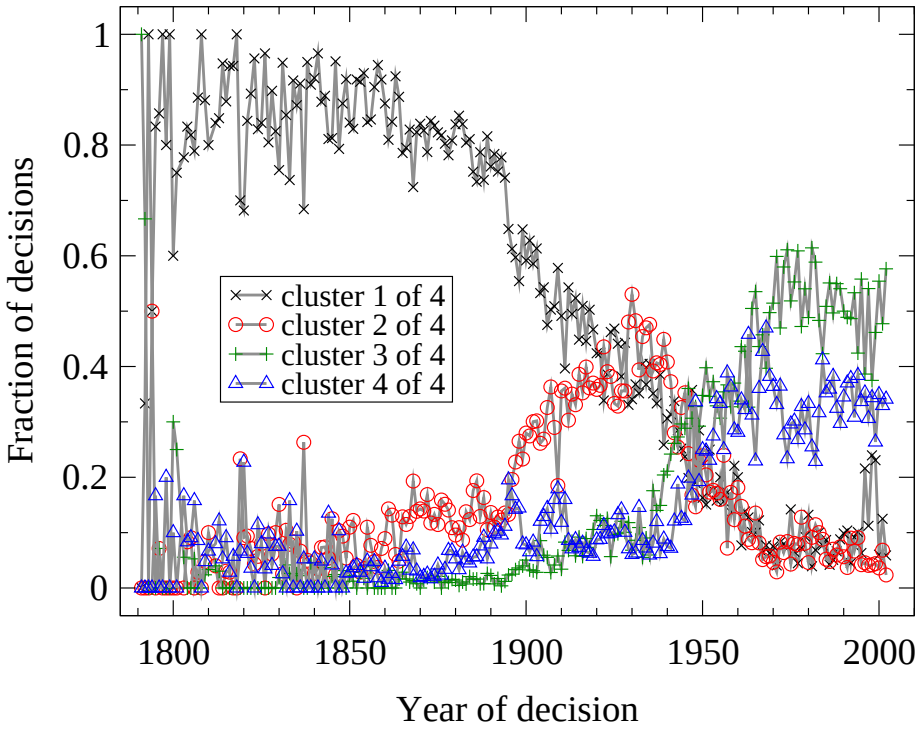


Figure 5.7. A histogram of the number of decisions versus the year of the decision for cases assigned to each group in the four-way split produced by the modularity maximization algorithm.

identifiable signatures in the citation record. There are some differences between the two sets of results, particularly the early "tail" to the second group in the clustering analysis and an overall difference in the number of cases assigned to each group. A possible explanation for these differences is that the EM analysis makes use only of citations received by cases, whereas the clustering analysis, which ignores edge direction, takes into account both citations received and citations made. This allows the classification into groups of some vertices that were unclassifiable with the EM algorithm by virtue of their never receiving any citations. (About 10% of cases were never cited.) It could also be responsible for the tail in the second group because citations made, which are necessarily to cases in the past, connect vertices to earlier times, perhaps pulling them from the second group into the first in the clustering analysis.

As with the EM analysis, we can go further and look at splits into larger numbers

of groups. For instance, Fig. 5.7 shows the best split into four groups according to the modularity-based approach. Again, the split is similar in overall form to the split found by the EM algorithm with $c = 4$, although the results are not as clean as those for the EM algorithm. As before, a new split point appears around 1900, which could be associated with the start of the *Lochner* era.

5.4 Vertex authority score and time evolution

For our third analysis, we turn away from studies of groups or clusters and focus on another class of network measures: centrality scores, which quantify the importance or influence of individual vertices in a network. As we will see, the pattern of centrality scores as a function of time in our evolving citation networks can reveal interesting patterns.

We provided an introduction to the various measures of centrality for network vertices in Section 1.2.6. For our analysis of the network of United States Supreme Court decisions we use the method of Kleinberg [45], which works well for acyclic networks. In this variant each vertex has two centralities, known as the authority score and the hub score, the first derived from the incoming links and the second from the outgoing links. In this view a "hub" is a vertex that points to many important authorities—a review paper in a citation network, for instance—while an authority is a vertex pointed to by many important hubs—such as an important or authoritative research article on a particular subject. A more complete discussion of this technique is found in Section 1.2.6.

Again, taking the Supreme Court network as an example we note that hub and authority scores have been previously applied to Supreme Court cases by Fowler and Jeon [30], who showed that it can be revealing to calculate scores not only for a complete citation network, but also for subsets of the network containing only opinions handed down on or before a certain date, effectively recreating the citation network

as it existed at that date. We have adopted a similar approach in our calculations, calculating authority scores for the network as it existed at some time t. We focus primarily on the most central cases: those with the highest scores.

Figure 5.8 shows one particularly informative statistic, the average age of the ten highest-ranked cases in our data set as a function of the year at which the network is cut off. As the plot shows, there is a marked trend for the average age to increase with the passage of time. This is precisely the behavior one would expect if the top authorities in the network are remaining the same as time goes by. Every once in a while, however, the plot shows a sudden and precipitous drop in the average age, indicating that a much younger set of vertices have, in a short space of time, taken over as the new leaders in the authority score rankings. Thus the plot indicates a repeated pattern in the evolution of the network in which a certain set of vertices—certain cases considered by the Supreme Court—remain the top authorities for substantial periods of time before being swiftly replaced by a different set. One example of such a turnover can be seen in Fig. 5.8 around 1900 and a smaller one around 1940, dates that, as we have seen, correspond roughly to the beginning and end of the *Lochner* era. Another very large dip in the curve occurs around 1970. (Our four-group EM analysis also found a group division at approximately the same point—see Fig. 5.4.) The large size of this dip may be due in part to the much larger number of cases decided per year by the Supreme Court in more recent decades than in its earlier history, which makes it easier for newly appearing cases to quickly become top authorities. The results of the centrality analysis are thus compatible with but different from those of previous sections. Such variations are one reason why a variety of different analytic techniques are useful in studies of network structure.

The behavior described is clearest in the age of the top ten vertices, but persists if a different number is used. Figure 5.8 shows the results of the same calculation for the top 50, 100, and 500 authorities, and in each case a similar pattern of maturation

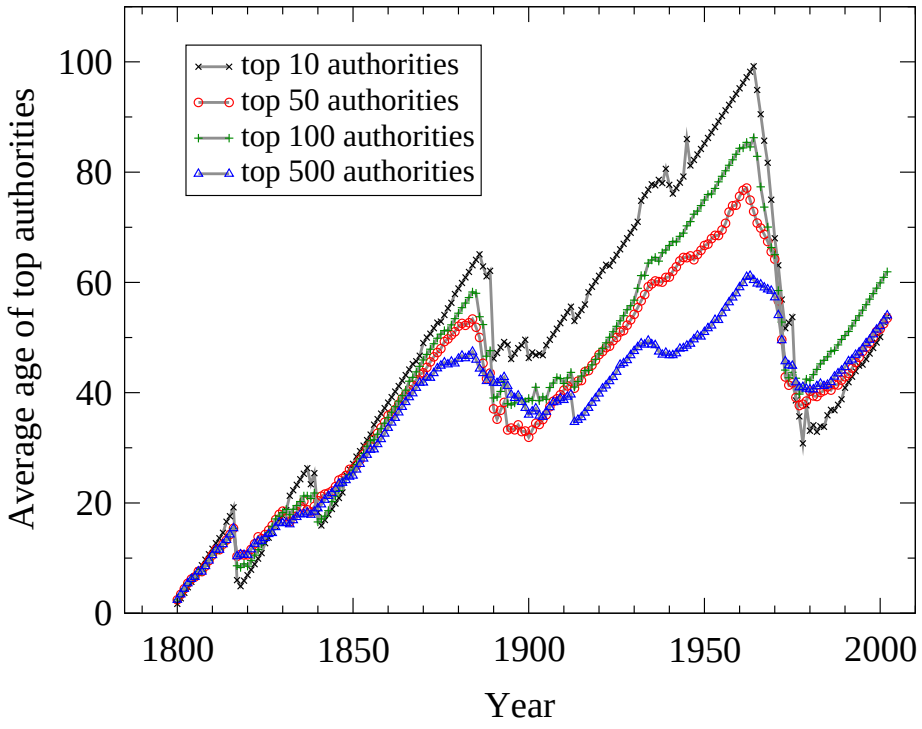


Figure 5.8. The average age of the highest-authority cases in the Supreme Court citation network as a function of time.

followed by swift renewal is visible.

5.5 Implications for legal scholarship

Although the purpose of this chapter is primarily to highlight new methods for the analysis of network data, the ultimate goal of these methods is, of course, to give researchers insight into the structure and meaning of their data. Thus, it is interesting to ask whether the analyses described here do indeed shed light on the system studied—in this case, the network of citations among Supreme Court cases. In fact the results do appear to offer an interesting perspective on the workings of the Supreme Court, as we argue briefly in this section.

The United States underwent a transition from an agricultural economy to an industrial economy in the latter part of the nineteenth century. Federal and state legislators adapted to the new economic environment by passing laws that regulated emerging industries. These regulations, however, were not without opposition from

those who preferred a *laissez-faire*, or "hands-off," approach. Among those outspoken in opposition were several members of the Supreme Court and, beginning in 1897, the court began invalidating a number of cases that imposed regulations on industry and business, starting with *Allgeyer v. Louisiana*. The legal doctrines of **substantive due process** and **freedom of contract** were merged together into a significant limitation on the police power of the state. After *Allgeyer*, any statute, ordinance, or administrative act that imposed any kind of limitation upon the right of private property or freedom of contract became suspect, even if the regulation was intended to promote safety and general welfare [43].

The most famous (or infamous) of the cases to use substantive due process to invalidate state regulation was *Lochner v. New York* in 1905, a case that became so notorious that this entire era of jurisprudence, between 1897 and 1937, came to be known as the *Lochner* era. During the *Lochner* era the Supreme Court struck down nearly 200 regulations [87]. The *Lochner* era is clearly visible, for example, in our EM analysis with $c = 4$ (Fig. 5.4)—the analysis picks out one group of cases with start and end dates that correspond closely to the accepted dates of the era.

Ultimately, the Supreme Court's hostility to state and federal regulation began to interfere with the "New Deal" programs instituted by US President Franklin Roosevelt to combat the Great Depression. Between 1934 and 1936, the court invalidated more federal statutes than during any other two-year period in its history and by 1936 nearly all of the statutes passed as part of the New Deal had been struck down. In response, Roosevelt launched, in the early months of 1937, a counteroffensive against the Supreme Court in which he proposed to appoint to the court up to six additional justices more receptive to the New Deal. This "court packing" plan was, to say the least, highly controversial, but Roosevelt had the support of significant majorities in both houses of Congress, and the nation as a whole, still in the throes of the Great Depression, was eager for something new.

Following Roosevelt's proposal, the court abruptly reversed course and, beginning in March of 1937, validated a series of state and federal measures. Contemporary commentators have humorously dubbed this change the "switch in time that saved nine," but whether the switch was substantive or illusory has been the subject of much debate. Some scholars believe that the court responded to political pressure while others have suggested that the court already contained a majority of justices who would have been inclined to sustain the New Deal if legislation had been drafted better or if certain unanswered questions had been appropriately posed to the court.

Our EM analysis shows a clear break around 1937, corresponding closely to the end of the *Lochner* era. It is important to appreciate that this analysis takes into account only citations received by cases. Thus, the opinions of the Supreme Court appear to have taken a substantial change of direction, not merely in their conclusions, but also in their arguments. Later cases cited the new opinions rather than those coming before them because, presumably, their arguments better supported the decisions of the post-1937 court. Consequently, our analysis appears to indicate not merely a change in case outcomes that was a natural, if novel, result of positions long held by the sitting justices, but also a more fundamental change in legal thinking itself—or at least its expression in the written opinions of the court and the later citation of those opinions.

5.6 Discussion

In this chapter we have described several methods for the analysis of citation networks, which are acyclic directed graphs of citations between pairs of documents. Using the network of citations between opinions handed down by the United States Supreme Court as an example, we have described and demonstrated three analysis techniques. The first makes use of a probabilistic mixture model fitted to the observed network structure using an expectation–maximization algorithm. The second is a network

clustering method making use of the recently introduced method of modularity maximization. The third is an analysis of the patterns of time variation in eigenvector centrality scores, particularly the "authority" score introduced by Kleinberg [45].

When applied to the Supreme Court network, each of these analyses reveals interesting structure, particularly highlighting qualitative changes in citation patterns that may be associated with specific eras of legal thought in the Supreme Court. However, it is in combination that the methods become most effective. Features that appear clearly in analyses performed using several different techniques possess correspondingly greater persuasive force. In the case of the Supreme Court, there emerges quite a clear picture of the eras of the court as marked by shifts in citation patterns, particularly around the time of the so-called *Lochner* era in the early 20th century.

CHAPTER 6

Conclusions

In this dissertation, we developed several new methods for detecting structure in complex networks. We also presented the results obtained from applying these new methods to a variety real and simulated networks.

The visualization of networked systems as two-dimensional drawings dates back, at least, to Moreno's work in the 1930's and much further if we include the field of graph theory. For a small network, a good deal of information is gleaned from a simple two-dimensional sketch of the network; indeed, we may easily identify highly connected vertices, densely connected communities of vertices, or other distinct types of network structure. However, when we visualize large networks our eyes tend to lose the ability to identify specific structural aspects. Instead, we see just a jumble of vertices and edges.

Numerous methods tackle the problem of discovering what a networks looks like when direct visualization yields poor results. The methods all aim to detect different types of network structure, but they share a common aspect of their methodology. Such methods usually begin with an a priori assumption of the form the network structure will take; then, a means of detecting and measuring the particular type of network structure is derived and implemented on networks. In the first part of this dissertation, Chapters 2 and 3, we followed this path and added our own methods for detecting distinct aspects of network structure.

Specifically, in Chapter 2 we identified the structural similarity of network vertices.

Using techniques from linear algebra, we derived a new method for measuring the similarity of any pair of vertices in a network. The application of our method to both real and simulated networks illustrated its strengths, especially when compared with existing procedures for detecting vertex similarity.

In Chapter 3, we pursued the idea of community detection in directed networks. In the past, community detection in directed networks involved transforming a directed network into an undirected network (by throwing away edge direction), and implementing a method for community detection designed for undirected networks. However, we showed that one method for detecting community structure in undirected networks could be re-derived for directed ones, while making explicit use of the information contained in the edge directions. We also highlighted some specific examples of directed networks where community detection is significantly improved when edge direction is retained.

In the second part of this dissertation, Chapters 4 and 5, we changed our approach to the problem of detecting network structure. In these two chapters, we used the machinery of probabilistic mixture models and the expectation-maximization (EM) algorithm to probe network structure.

In Chapter 4, we applied the EM algorithm to networks as an exploratory data analysis technique. Our technique detected general patterns of connection among vertices and assigned them to groups based on that structure. The patterns of connection being sought among vertices were never pre-specified—a departure from previous techniques. Nevertheless, our application of the EM algorithm proved able to detect a wide range of structural signatures in networks.

The EM algorithm was again discussed in Chapter 5, but this time as a tool for detecting structure in time evolving networks. We proposed that distinct time periods in the history of an evolving network may be identified using the EM algorithm. As a particular example, we examined the citation network of United States Supreme

Court Decisions. Our method did detect changes in the structure of the network over time. Furthermore, the times at which we identified changes in the network structure tended to be aligned with eras in the history of the Court that have been the subject of debate amongst legal scholars.

The content of this dissertation was divided between two different types of methodology. The first two chapters used linear algebra based methods to detect specific types of network structure. The final two chapters turned to probabilistic techniques for general exploration of network structure. These probabilistic techniques should not be been seen as replacements for the more traditional liner algebra techniques. In Chapter 4, we highlighted the fact that our technique based on the EM algorithm identified a variety of types of network structure without being told ahead of time the type of structure to seek. However, specialized methods outperformed our measure when we limited the scope of our search to one specific type of structure. Consequently, probabilistic based techniques are not a replacement, but an additional tool. We should not abandon measure of specific types of network structure. Instead, probabilistic techniques are for exploratory data analysis, while specific linear algebra techniques probe deeper into very specific types of network structure.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] H. Akaike, *A new look at the statistical identification model*, IEEE Trans. Auto. Control, 19 (1974), pp. 716–723.

[2] R. Albert, H. Jeong, and A.-L. Barabási, *Diameter of the world-wide web*, Nature, 401 (1999), pp. 130–131.

[3] A. Arenas, J. Duch, A. Fernández, and S. Gómez, *Size reduction of complex networks preserving modularity*, New J. Phys., 9 (2007), p. 176.

[4] P. Ball, *The physical modelling of society: a historical perspective*, Physica A, 314 (2002), pp. 1–14.

[5] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks*, Science, 286 (1999), pp. 509–512.

[6] V. Batagelj, P. Doreian, and A. Ferligoj, *An optimizational approach to regular equivalence*, Social Networks, 14 (1992), pp. 121–135.

[7] J. Baumes, M. Goldberg, and M. Magdon-Ismail, *Efficient identification of overlapping communities*, in Proceedings of the IEEE International Conference on Intelligence and Security Informatics, New York, 2005, Institute of Electrical and Electronics Engineers.

[8] P. S. Bearman, J. Moody, and K. Stovel, *Chains of affection: The structure of adolescent romantic and sexual networks*, Am. J. Sociol., 110 (2004), pp. 44–91.

[9] E. A. Bender and E. R. Canfield, *The asymptotic number of labeled graphs with given degree sequences*, Journal of Combinatorial Theory A, 24 (1978), pp. 296–307.

[10] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. V. Dooren, *A measure of similarity between graph vertices: Applications to synonym extraction and web searching*, SIAM Review, 46 (2004), pp. 647–666.

[11] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, *Complex networks: Structure and dynamics*, Physics Reports, 424 (2006), pp. 175–308.

[12] P. F. Bonacich, *Power and centrality: A family of measures*, Am. J. Sociol., 92 (1987), pp. 1170–1182.

[13] S. P. Borgatti and M. G. Everett, *Two algorithms for computing regular equivalence*, Social Networks, 15 (1993), pp. 361–376.

[14] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks, 30 (1998), pp. 107–117.

[15] R. S. Burt, *Positions in networks*, Social Forces, 55 (1976), pp. 93–122.

[16] G. Clarkson and D. DeKorte, *The problem of patent thickets in convergent technologies*, in Progress in Convergence: Technologies for Human Wellbeing, W. S. Bainbridge and M. C. Roco, eds., vol. 1093 of Annals of the New York Academy of Science, New York Academy of Sciences, 2006, pp. 180–200.

[17] A. Clauset, M. E. J. Newman, and C. Moore, *Structural inference of hierarchies in networks*, in Proceedings of the 23rd International Conference on Machine Learning, New York, 2006, Association of Computing Machinery.

[18] F. B. Cross, T. A. Smith, and A. Tomarchio, *Determinants of cohesion in the supreme court's network of precedents*, SSRN eLibrary, (2006).

[19] G. Csárdi, K. J. Strandburg, L. Zalanyi, J. Tobochnik, and P. Érdi, *Modeling innovation by a kinetic description of the patent citation system*, Physica A, 374 (2007), pp. 783–793.

[20] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, *Comparing community structure identification*, J. Stat. Mech., (2005), p. P09008.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, J. R. Statist. Soc. B, 39 (1977), pp. 185–197.

[22] L. Egghe and R. Rousseau, *Introduction to Informetrics*, Elsevier, 1990.

[23] P. Erdős and A. Rényi, *On random graphs*, Publicationes Mathematicae, 6 (1959), pp. 290–297.

[24] ——, *On the evolution of random graphs*, Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 5 (1960), pp. 17–61.

[25] ——, *On the strength of connectedness of a random graph*, Acta Mathematica Scientia Hungary, 12 (1961), pp. 261–267.

[26] E. Estrada and J. A. Rodríguez-Velázquez, *Spectral measures of bipartivity in complex networks*, Phys. Rev. E, 72 (2005), p. 046105.

[27] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On power-law relationships of the internet topology*, Computer Communications Review, 29 (1999), pp. 251–262.

[28] M. Fiedler, *Algebraic connectivity of graphs*, Czech. Math. J., 23 (1973).

[29] ——, *A property of eigenvectors of non-negative symmetric matrices and its application to graph theory*, Czech. Math. J., 25 (1975), pp. 619–633.

[30] J. H. Fowler and S. Jeon, *The authority of Supreme Court precedent*, Social Networks, 30 (2008), pp. 16–30.

[31] J. H. Fowler, T. R. Johnson, J. F. Spriggs II, S. Jeon, and P. J. Wahlbeck, *Network analysis and the law: Measuring the legal importance of Supreme Court precedents*, Political Analysis, 15 (2007), pp. 324–346.

[32] M. Girvan and M. E. J. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 7821–7826.

[33] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, *Modularity from fluctuations in random graphs and complex networks*, Phys. Rev. E, 70 (2004), p. 025101.

[34] ——, *Module identification in bipartite and directed networks*, Phys. Rev. E, 76 (2007), p. 036102.

[35] M. B. Hastings, *Community detection as an inference problem*, PRE, 74 (2006), p. 035102(R).

[36] P. Holme and M. Huss, *Role-similarity based functional prediction in networked systems: Application to the yeast proteome*, J. R. Soc. Interface, 2 (2005), pp. 327–333.

[37] P. Holme, F. Liljeros, C. R. Edling, and B. J. Kim, *Network bipartivity*, Phys. Rev. E, 68 (2003), p. 056107.

[38] P. Jaccard, *Etude comparative de la distribution florale dans une portion des alpes et du jura*, Bulletin de la Société Vaudoise des Sciences Naturelles, 37 (1901), pp. 547–579.

[39] A. B. Jaffe and M. Trajtenberg, *Patents, Citations and Innovations: A Window on the Knowledge Economy*, MIT Press, Cambridge, MA, 2002.

[40] G. Jeh and J. Widom, *SimRank: A measure of structural-context similarity*, in KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2002, Association of Computing Machinery, pp. 538–543.

[41] J. H. Jones and M. S. Handcock, *An assessment of preferential attachment as a mechanism for human sexual network formation*, Proc. R. Soc. London B, 270 (2003), pp. 1123–1128.

[42] L. Katz, *A new status index derived from sociometric analysis*, Psychometrika, 18 (1953), pp. 39–43.

[43] A. H. Kelly, W. A. Harbison, and H. Belz, *The American Constitution: Its origins and development*, Norton, New York, 7th ed., 1991.

[44] M. M. Kessler, *Bibliographic coupling between scientific papers*, American Documentation, 14 (1963), pp. 10–25.

[45] J. M. Kleinberg, *Authoritative sources in a hyperlinked environment*, J. ACM, 46 (1999), pp. 604–632.

[46] J. M. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, *The Web as a graph: Measurements, models and methods*, in Proceedings of the 5th Annual International Conference on Combinatorics and Computing, T. Asano, H. Imai, D. T. Lee, S.-I. Nakano, and T. Tokuyama, eds., no. 1627 in Lecture Notes in Computer Science, Berlin, 1999, Springer, pp. 1–18.

[47] Y. Koren, S. C. North, and C. Volinsky, *Measuring and extracting proximity in networks*, in KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2006, ACM, pp. 245–255.

[48] A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor, *Compartments revealed in food-web structure*, Nature, 426 (2003), pp. 282–285.

[49] E. A. Leicht, G. Clarkson, K. Shedden, and M. E. J. Newman, *Large-scale structure of time evolving citation networks*, Eur. Phys. J. B, 59 (2007), pp. 75–83.

[50] E. A. Leicht, P. Holme, and M. E. J. Newman, *Vertex similarity in networks*, Phys. Rev. E, 73 (2006), p. 026120.

[51] E. A. Leicht and M. E. J. Newman, *Community structure in directed networks*, Phys. Rev. Lett., 100 (2008), p. 118703.

[52] D. Liben-Nowell and J. Kleinberg, *The link prediction problem for social networks*, in CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management, New York, NY, USA, 2003, AMC, pp. 556–559.

[53] J. J. Luczkovich, S. P. Borgatti, J. C. Johnson, and M. G. Everett, *Defining and measuring trophic role similarity in food webs using regular equivalence*, J. Theor. Bio., 220 (2003), pp. 303–321.

[54] S. Mawson, ed., *Thesaurus of English Words and Phrases*, T. Y. Crowell Co., New York, NY, USA, 1911.

[55] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley-Interscience, New York, 1996.

[56] M. McPherson, L. Smith-Lovin, and J. M. Cook, *Birds of a feather: Homophily in social networks*, Ann. Rev. Sociol., 27 (2001), pp. 415–444.

[57] A. Medus, G. Acuña, and C. O. Dorso, *Detection of community structures in networks via global optimization*, Physica A, 358 (2005), pp. 593–604.

[58] M. Molloy and B. Reed, *A critical point for random graphs with a given degree sequence*, Random Structures and Algorithms, 6 (1995), pp. 161–179.

[59] ——, *The size of the giant component of a random graph with a given degree sequence*, Combinatorics, Probability and Computing, 7 (1998), pp. 295–305.

[60] J. Moody, *Race, school integration, and friendship segregation in America*, Am. J. Sociol., 107 (2001), pp. 679–716.

[61] J. L. Moreno, *Who Shall Survive?: A New Approach to the Problem of Human Interrelations*, Nervous and Mental Disease Publishing Co., Washington, D. C. USA, 1934.

[62] M. E. J. Newman, *Mixing patterns in networks*, Phys. Rev. E, 67 (2003), p. 026126.

[63] ——, *Detecting community structure in networks*, Eur. Phys. J. B, 38 (2004), pp. 321–330.

[64] ——, *Fast algorithm for detecting community structure in networks*, Phys. Rev. E, 69 (2004), p. 066133.

[65] ——, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), p. 036104.

[66] ——, *Modularity and community structure in networks*, Proc. Natl. Acad. Sci. USA, 103 (2006), pp. 8577–8582.

[67] M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Phys. Rev. E, 69 (2004), p. 026113.

[68] M. E. J. Newman and E. A. Leicht, *Mixture models and exploratory analysis in networks*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 9564–9569.

[69] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*, Nature, 435 (2005), pp. 814–818.

[70] J. Park and M. E. J. Newman, *A network-based ranking system for American college football*, J. Stat. Mech., (2005), p. P10014.

[71] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse m atrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.

[72] D. J. de S. Price, *Networks of scientific papers*, Science, 149 (1965), pp. 510–515.

[73] ——, *A general theory of bibliometric and other cumulative advantage processes*, Journal of the American Society for Information Science, 27 (1976), pp. 292–306.

[74] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, *Defining and identifying communities in networks*, Proc. Natl. Acad. Sci. USA, 101 (2004), pp. 2658–2663.

[75] A. Rapoport, *Contribution to the theory of random and biased nets*, Bulletin of Mathematical Biology, 19 (1957), pp. 257–277.

[76] E. Ravasz, A. L. Somera, D. A. Mongru, and Z. N. O. A.-L. Barabási, *Hierarchical organization of modularity in metabolic networks*, Science, 297 (2002), pp. 1551–1555.

[77] S. Redner, *How popular is your paper? An empirical study of the citation distribution*, Eur. Phys. J. B, 4 (1998), pp. 131–134.

[78] J. Reichardt and S. Bornholdt, *Detecting fuzzy community structures in complex networks with a Potts model*, Phys. Rev. Lett., 93 (2004), p. 218701.

[79] ——, *Statistical mechanics of community detection*, Phys. Rev. E, 74 (2006), p. 016110.

[80] M. Rosvall and C. T. Bergstrom, *Maps of information flow reveal community structure in complex networks*, Proc. Natl. Acad. Sci. USA, 105 (2007), pp. 1118–1123.

[81] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Auckland, 1983.

[82] G. Schwarz, *Estimating the dimension of a model*, Annals of Statistics, 6 (1978), pp. 461–464.

[83] J. Scott, *Social Network Analysis: A Handbook*, Sage, London, 2 ed., 2000.

[84] P. O. Seglen, *The skewness of science*, J. Amer. Soc. Inform. Sci., 43 (1992), pp. 628–638.

[85] R. Solomonoff and A. Rapoport, *Connectivity of random nets*, Bulletin of Mathematical Biology, 13 (1951), pp. 107–117.

[86] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, *Graphscope: parameter-free mining of large time-evolving graphs*, in KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2007, ACM, pp. 687–696.

[87] M. I. Urofsky and P. Finkelman, *A March of Liberty: A Constitutional History of the United States*, Oxford University Press, New York, 2nd ed., 2002.

[88] S. Wasserman and K. Faust, *Social Network Analysis*, Cambridge University Press, Cambridge, 1994.

[89] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks*, Nature, 393 (1998), pp. 440–442.

[90] H. C. White, S. A. Boorman, and R. L. Breiger, *Social structure from multiple networks: I. Blockmodels of roles and positions*, Am. J. Sociol., 81 (1976), pp. 730–779.

[91] S. White and P. Smyth, *A spectral clustering approach to finding communities in graphs*, in Proceedings of the 5th SIAM International Conference on Data Mining, H. Kargupta, J. Srivastava, C. Kamath, and A. Goodman, eds., Philadelphia, 2005, Society for Industrial and Applied Mathematics.

[92] A. W. Wolfe, *Connecting the dots without forgetting the circles*, Connections, 26 (2005), pp. 105–117.

[93] C. F. J. Wu, *On the convergence of the em algorithm*, Annals of Statistics, 11 (1983), pp. 93–103.

[94] W. W. Zachary, *An information flow model for conflict and fission in small groups*, JAR, 33 (1977), pp. 452–473.