ANALYSIS AND APPLICATIONS OF COMPLEX NETWORKS

by

James Peter Bagrow

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
CLARKSON UNIVERSITY
POTSDAM, NEW YORK 13699, USA
APRIL 2008

CLARKSON UNIVERSITY
DEPARTMENT OF
PHYSICS

The undersigned hereby certify that they have read and recommend to the Faculty of Physics for acceptance a thesis entitled **Analysis and Applications of Complex Networks** by **James Peter Bagrow** in partial fulþllment of the requirements for the degree of **Doctor of Philosophy**.

Dated: April 2008

Research Supervisors: _____
Daniel ben-Avraham

_____
Erik M. Bollt

Examining Committee: _____
Joseph D. Skufca

_____
Lawrence S. Schulman

_____
Brian T. Helenbrook

Table of Contents

List of Tables

List of Figures

Abstract

This thesis is concerned with three main areas of complex networks research. One is on developing and testing new methods to þnd communities, especially methods that do not need knowledge of the entire network. The second is on the application of shells and their usage when characterizing and identifying important network properties. Finally, we oÞer several contributions toward the usage of complex networks as a tool for studying social dynamics.

The study of *communities*, densely interconnected subsets of nodes, is a diŽcult and important problem. Methods to identify communities are developed which have the rare ability to function with only local knowledge of the network. A new benchmarking and evaluation procedure is introduced to compare the performance of both existing and new local community algorithms.

Using shells, we introduce a new matrix structure that allows for quantitative comparison and visualization of networks of all sizes, even extremely large ones. This \portrait" encodes a great deal of information including dimensionality and regularity, and imparts immediate intuition about the network at hand. A distance metric generated by comparing two portraits allows one to test if, e.g., two networks were created by the same underlying generating mechanism. Generalizations to weighted networks are studied, as is applicability to the Graph Isomorphism problem.

We introduce the Patron-Artwork model as a new means of generating a distribution of *fame* or knowledge from an underlying social network, and give a full analysis for a network where all members are neighbors. In addition, the so-called Small World Phenomenon has been studied in the context of social networks, speciþcally that of

Kleinberg navigation. We studied the impact of modifying the underlying Kleinberg lattice by introducing an anisotropy: the lattice is either stretched along one axis or long-distance connections are made more favorable along a preferred direction.

Acknowledgments

I am very grateful to my advisor Dani ben-Avraham for his patience, support, and encouragement, and to my co-advisor Erik Bollt for his generosity, benevolence, and wisdom.

I thank Prof. Schulman, Prof. Skufca, and Prof. Helenbrook for joining my committee and giving their time and eÞort in helping advance my studies.

I wish to thank Pieter Swart, Aric Hagberg, and everyone at T-7 as well as all my friends in Los Alamos for making my time there so much fun.

I also wish to thank my group members Joe Skufca and Abbas Alhakim for their feedback and knowledge and my fellow students Naratip, Chen Yao, Rio, Ryan, Pat, and especially Hernan for their humor, knowledge, and support. I am very grateful to Bonnie Weber for her never-ending patience and support.

I thank my brother Danny, my parents, and all my family for their support and for allowing me so many wonderful opportunities.

Introduction

Complex, non-uniform interconnectedness is present in such diverse areas as human society and social interaction [1, 2, 3]; man-made technology such as the world wide web [4, 5]; and even organic systems including food webs [6, 3], cellular biology [7], and evolutionary relationships [8]. Motivated by the discrete, nonlinear aspects present in such areas, the þeld of Complex Networks has arisen to study these systems with a variety of mathematical tools. All such systems consist of *objects* (people, web pages, chemicals, etc.) and *relationships* (people that are friends, chemicals that react, web pages that link to each other, etc.). Complex networks quantify these structures using Graphs, where nodes and edges represent the objects and their relationships, respectively.

Graph theory has a long, illustrious history, starting with Euler and the bridges of Konigsberg [9], but it hasn't been until the recent availability of fast, cheap computers that graph techniques were applicable to the very large networks of everyday life that are most interesting. This has enabled the use of tools from Statistical Mechanics and other þelds, which are most viable for such (statistically) large systems.

The results presented here cover three broad areas of complex networks interest. One is in the area of þnding **communities**, densely interconnected groups of nodes, inside networks. Another is the application of **shells**, a non-local, non-global means of decomposing a network, to study and characterize network structure. Finally, we present several useful contributions to the area of **social networks**, using complex networks to model intrinsic human behavior (and other phenomena).

We begin in Chapter 1 with a review of the most prominent terminology and

background related to complex networks. This includes graph theoretic background, such as the degree distribution, cycles, shells, etc., as well as the more famous random graphs used to model various large-scale systems. Following this, in Chapter 2 we introduce the problem of þnding communities, covering background material and existing techniques as well as our own contributions (including a new local community detection algorithm) and open questions.

Motivated by their appearance in our local community methods, in Chapter 3 we present our study of *shells*, an interesting property of networks that is neither local nor global. We present a recursive system of equations meant to enumerate quantities of interest related to shells, several new statistics to help eŽciently characterize networks based on shells, and a new measure of *bipartivity*, a means of quantifying how \close" a network is to being bipartite (two-colorable).

Informed by our work on shells, Chapter 4 introduces the network \portrait," an entirely new and very exciting means to visualize and compare networks of any size. We use these portraits to derive a distance metric between networks as well as pose a variety of interesting open questions including generalization to weighted networks, brute-force graph searching, and applicability to the Graph Isomorphism problem.

Finally, in Chapter 5 we present two contributions to the study of social dynamics, one of the key applications of complex networks. We introduce the Patron-Artwork model as a tool to study how objects (movies, art, celebrities, etc.) become famous by means of a simple recommendation mechanism inside a social network. Our second contribution is to the study of Kleinberg navigation, a social network model that exhibits the *small world* phenomenon, the empirical fact that well-separated people tend to be connected by surprisingly short chains of acquaintances. Kleinberg navigation is famous for its richness despite consisting of a single parameter, although it is not the most realistic model for a society. We modify the Kleinberg model by introducing an anisotropy in the underlying lattice. A variety of open questions are posed, including modiþcation of the navigation scheme in the face of a lattice with \voids," which increase the model's realism by representing uninhabitable regions

such as deserts, mountains, and so forth.

A þnal summary of this thesis, including a discussion of contributions as well as open questions and avenues for future work, is presented in Chapter 6.

Chapter 1

Network Deþnitions and Review

The area of Complex networks is a relatively recent þeld of study, but much of the associated terminology comes from Graph theory, which has a much longer history. Due to this close relationship, and the inherent interdisciplinary nature of the þeld, competing, equivalent terms have arisen in many areas. Here we present a brief overview of the most salient terminology, including alternatives when available, but this list is far from comprehensive.

- A **Network** is formally represented by a **Graph** $G = \{V; E\}$ where V is the set of **nodes** (or vertices), representing the objects in the network, and E is the set of **edges** (or links) in the network, representing the connections or relationships between the network's objects. The total number of nodes is $N = |V|$ and the total number of edges is $M = |E|$. The edge connecting nodes i and j is typically denoted e_{ij}, $e(i; j)$, or sometimes just $(i; j)$. A graph is **sparse** if $M = O(N)$.

- The **degree** (or valency) of a node is the number of connections it has to other nodes. For graphs where at most a single edge is allowed between any two nodes, the degree of a node is equivalent to the number of nodes **adjacent** to that node. Adjacent nodes are **neighbors** and the set of nodes adjacent to i is the **neighborhood** of i. The **degree distribution** $P(k)$ gives the probability for a node to have degree k in the network. The number of nodes

with degree k may be denoted as n_k so that the \empirical" degree distribution is $P(k) = n_k/N$. The degree of node i is generally denoted k_i.

- For **undirected** graphs $e_{ij} = e_{ji}$. For **directed** graphs (or digraphs), this is not true, and edges are visualized as \arrows" pointing from the **tail** node to the **head** node. The head node is adjacent to the tail node, but the tail node is not adjacent to the head node (unless there exists another connection).

- A **subgraph** of G is a graph whose edges and vertices are subsets of G's.

- A **path** (or trail) between source node i and target node j is an alternating sequence of nodes and edges, beginning with i and ending with j. The **path length** is the number of edges traversed in the path. The **shortest path** is the path(s) with the smallest number of edges, and is typically the quantity of interest. A **cycle** (or circuit) is a closed path (one with the same source and target). Simple paths and cycles are those where each node and edge is traversed exactly once. Unless otherwise stated, all paths and cycles will be considered simple. The **distance** between two nodes is the length of the shortest (simple) path between them. (This shortest paths distance is sometimes known as chemical distance or chemical space.) A **triangle** is the shortest simple cycle, of length 3.

- The **eccentricity** of a node is the length of the longest of all the shortest paths from that node to any other node. The **diameter** of a graph is the length of the longest of *all* shortest paths; the maximum of all eccentricities.

- If a path exists from every node to every other node, then that graph is **connected**. If this is not true, then the graph consists of two or more **connected components**: subsets of vertices where each node in a component has a path to every other node in that component.

- An acyclic graph is one in which no cycles are present, and is called a **forest**. A connected, acyclic graph is called a **tree**.

- A **weighted graph** is a graph where every edge is labeled with a **weight** (or cost). These weights are typically real numbers, though they are sometimes restricted to rational numbers or even integers. The weight of a path, cycle, or tree is the sum of the weights of the included edges. Shortest paths in weighted graphs are those paths with the minimum weight, not the minimum length. Unless otherwise stated, we consider graphs to be unweighted.

- A **multigraph** (or pseudograph) is one where multiple edges (**multi-edges** or parallel edges) can connect the same pair of nodes. A **self-loop** is an edge that connects a node to itself. In the graphs throughout this work, unless otherwise stated, neither are allowed.

- A **complete** graph of N nodes, often denoted K_N, is the graph where all nodes are adjacent. A complete subgraph is known as a **clique**.

- The **clustering coefficient** C_i of node i is a measure of how well-connected the neighbors of i are to each other [10]. Specifically, for i having degree k_i:

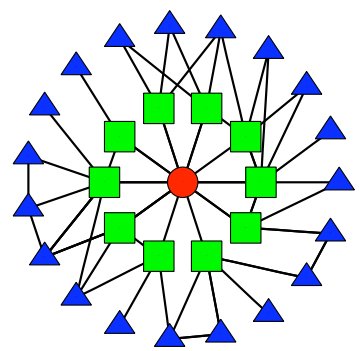
$$C_i = \frac{2E_i}{k_i(k_i - 1)}; \tag{1.0.1}$$

  where E_i represents the number of edges between neighbors of i. The maximum number of edges between neighbors is $k_i(k_i - 1)/2$, therefore $0 \leq C_i \leq 1$. The clustering coefficient of the entire graph is taken as the average clustering over all nodes, $C = \langle C_i \rangle$.

- **Assortativity** measures the degree-degree (and other) correlations between nodes in the network [11, 12]. A network is considered **assortative** when nodes of like degree tend to connect to one another, and **dissortative** (or disassortative) when high-degree nodes tend to connect to low-degree nodes and vice versa. The **assortativity coefficient** is essentially the Pearson correlation coefficient between pairs of nodes:

$$r = \frac{\mathrm{Tr}\,\mathbf{e} - \|\mathbf{e}^2\|}{1 - \|\mathbf{e}^2\|}; \tag{1.0.2}$$

where $e_{i;j}$ denotes the fraction of edges in the network that connect nodes of type i to nodes of type j and $jj\mathbf{x}jj$ denotes the sum of all elements in $\mathbf{x}$. For degree-degree correlations, the type of a node is its degree.



One of the overriding themes of this work is the notion of **shells** (or layers). The l-th shell of a node v is the set of all nodes in G that are at a distance l from v.[1] These shells are denoted as l-shells, or $G_l(v)$, and the node v is sometimes referred to as the **starting node**. The **shell decomposition** of G \about" v is the set $\{G_1(v); G_2(v); \ldots; G_{\ell}(v)\}$, where ℓ is the eccentricity of v. An interesting property of these shell decompositions is that they are neither a local property, since they depend on all of G, nor are they a global property, since they depend on a particular starting node. A decomposition is found in $O(N + M)$ steps using, e.g., the **Breadth First Search** (BFS) or another search algorithm.

Regarding implementation, networks are typically represented in several ways. One is the **adjacency matrix A**, where $A_{ij} = 1$ if edge $e_{i;j}$ exists, and zero otherwise. This matrix has many interesting properties but becomes intractably large for very large networks, since it is of size $N \times N$. Another storage format, which is more efficient than the adjacency matrix, is the **edgelist**, which can be thought of as an $M \times 2$ matrix where row e contains the two nodes composing edge e. This is especially efficient for sparse graphs, since it will only be of size $O(N)$. In addition, a single network can be represented by many adjacency matrices and edgelists, since the ordering of nodes and edges is arbitrary. These data structures are easily extended to directed, weighted, and multi- graphs.

Many types of **random** networks have been proposed, typically consisting of a generating mechanism giving the subsequent degree distribution and other properties. Here we list some of the more famous network models.

[1] Sometimes the shell is defined as the *subgraph* of G consisting of all nodes (and edges between those nodes) that are l steps away from node v. In this work, the definitions are typically interchangeable.

- The earliest random network was the eponymous Erdős-Renyi (ER) graph, proposed in 1959 [13]. This graph consists of N nodes with an edge existing between any two nodes with probability p. The degree distribution is then Binomial (Poissonian as $N \to \infty$). An important characteristic of ER graphs is that they undergo a *phase transition* as p increases past a critical value, p_c. For small values of p, the graph mostly consists of small, separate groups of nodes, but when $p = p_c = 1/N$, the graph becomes connected (a "giant" connected component emerges).

- Watts and Strogatz proposed a famous random network model in 1998, to describe how the *Small World* phenomenon arises [10]. This describes the fact that people tend to be much closer to each other than one would expect, based on the ER model (the famous *six degrees of separation*). Their model consists of a circular graph where nodes are connected to k-nearest neighbors. This network has a very high diameter (large world) until one begins to randomly rewire a few edges, creating long-range contacts. These edges will rapidly collapse the diameter of the graph, illustrating the small-world transition.[2]

- In response to the empirical fact that many real-world networks have a few highly connected nodes (or hubs), but most nodes have low degree, Barabasi and Albert (BA) proposed their seminal model for generating **scale-free** networks: networks with a power-law degree distribution, $P(k) \propto k^{-\gamma}$ [4, 14]. Their model consists of taking an initial seed network of $n \geq 2$ nodes, introducing one or more new nodes of degree m to the network at each time step, and connecting these newcomers to existing nodes based on their degree. This model contains two properties, *growth* and *preferential attachment* (rich-get-richer), that are necessary to generate a power-law degree distribution. The BA model always generates scale-free networks with $\gamma = 3$ but there exist other models that can generate different values for γ.

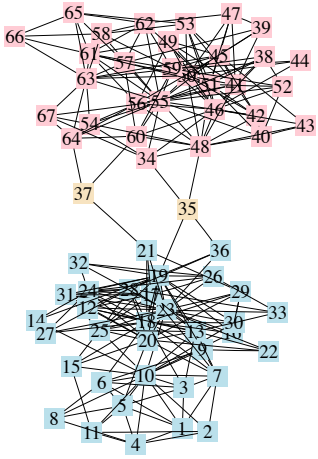
[2] We discuss a similar model illustrating the small world phenomenon, using a two-dimensional lattice, in Chapter 5.

ž One of the most general random network models is the **conþguration model**, sometimes referred to as the Molloy-Reed (MR) or Maximally Random (also MR) model [15, 16, 17, 18]. Unlike the previous networks, this model accepts the degree distribution as an *input*. To build a network, each node i out of N is assigned k_i \stubs," where k_i is drawn from the chosen $P(k)$. Uniformly random pairs of these stubs are chosen and wired together, until all stubs are þlled.[3] For two nodes of degree k_v and k_w, the probability that they are connected is $k_v k_w / 2M$, where $M = \frac{1}{2} \sum_i k_i$. Since edges are placed at random, conþguration model networks exhibit none of the degree-degree correlations inherent in many other models.

[3] One should be concerned if a graph can be made that will satisfy a particular sample sequence: the sum of the sampled degrees must be even, for example, or an empty connection will remain. In general, we limit ourselves to graphs so large that the introduction of a single edge to avoid this will not alter their properties. Self-loops and multi-edges are likewise neglible.

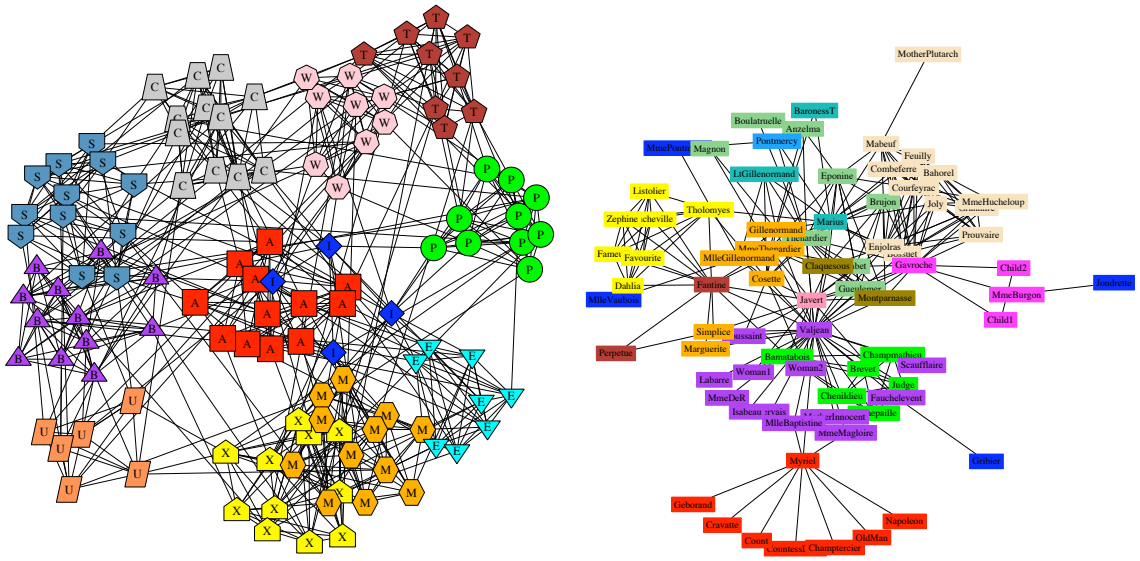
Chapter 2

Communities



A topic of current interest in the area of networks has been the idea of communities and their detection. A **Community** could be loosely described as a collection of vertices within a graph that are densely connected amongst themselves while being loosely connected to the rest of the graph [19, 20, 21]. This description, however, is somewhat vague and open to interpretation. This leads to the possibility that diþerent techniques for detecting these communities may lead to slightly diþerent yet equally valid results. Thus, the community problem becomes that of creating a partitioning which maximally identiþes the community structure.

Many social, technological, and biological networks exhibit community structure. Applications include studying the spread of disease (or, more generally, information) in social and communication networks, since one expects faster transport within communities than between; reducing very large graphs to smaller ones by studying only the community structure (collapsing communities down to a single node); and even the eŽcient routing of both hardware and software within multi-processor computers, since the interconnections between separate CPUs will be slower than internal connections, and their use should be minimized. See Fig. 2.1 for two example networks



(a) (b)

Figure 2.1: Two real-world networks with community structure. Shown is (a) the 2005 NCAA football schedule, and (b) the network of character interactions from Les Miserables by Victor Hugo [23]. The NCAA network is composed of teams who have played against each other in the regular season, and exhibits a community structure based on the *conferences* the teams are organized in, since teams tend to play within their own conference more often. The nodes in (a) are colored according to their conference aŽliation, while the nodes in (b) are colored from the result in Fig. 2.7.

which exhibit such community structure.

The number of ways to partition a graph is extremely large, and it is intractable to enumerate all of them. In fact, it has been shown that the problem of maximizing *modularity* (see Sec. 2.1.3), often taken as equivalent to detecting communities, is NP-Complete, meaning it is easy to check a solution but diŽcult to þnd one [22]. Due to this, many detection methods are approximate, greedy optimizations. Since þnding communities is diŽcult and has many applications, it is an interesting problem.

2.1 Existing Methods

We begin by detailing the most historically important classes of community detection methods. It is a fairly new field; early results date from the 1970s, but it has only recently become popular, in the past six years or so.

2.1.1 Spectral graph partitioning

The earliest form of community detection was the spectral graph partitioning methods due to Fiedler in the 1970s and later Pothen, Simon, and Liou in 1990 [24, 25, 26]. This method splits the network into communities based on the eigenvalues and eigenvectors of the **Graph Laplacian** $L(G)$:

$$L_{ij} \begin{cases} k_i; & \text{if } i = j; \\ 1; & \text{if } i \neq j \text{ and } \exists\, e_{ij}; \\ 0; & \text{otherwise;} \end{cases} \tag{2.1.1}$$

where G is an undirected, unweighted graph and k_i is the degree of node i. This is also known as the Laplacian matrix, Combinatorial Laplacian, Kirchhoff's Matrix, or the Admittance Matrix.

Briefly, some interesting properties of the Graph Laplacian.

- This matrix is symmetric, singular (rows sum to zero), and positive-semidefinite.
- The number of distinct **spanning trees** of G is equal to any cofactor of L [27], [28, page 57] . This is known as the matrix-tree theorem or Kirchhoff's Theorem.
- Since L is symmetric, its eigenvalues are real and its eigenvectors are real and orthogonal.
- The number of connected components in G is equal to the multiplicity of zero as an eigenvalue. This means that $\lambda_1 = 0$ and $\lambda_2 \neq 0$ iff G is connected.
- The eigenvalues are nonnegative: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$.
- The second smallest eigenvalue λ_2 is known as the **algebraic connectivity** of G and acts as a lower bound on the number of edges connecting two partitions in G. In other words, the larger λ_2 is, the more "difficult" it is to cut G into pieces [28].

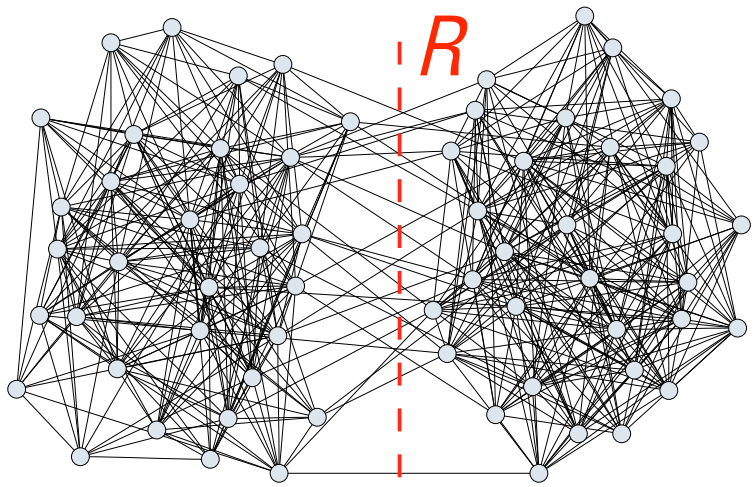


Figure 2.2: The cut number R for a partition of two equally sized groups.

The actual partitioning algorithm is quite simple and exploits the values of the eigenvector $\mathbf{v}_2$ (often known as the *Fiedler Vector*), corresponding to λ_2: for each node n in V, if $v_2(n) < 0$, put n in group N_-; otherwise, put n in group N_+. If one wishes to find further sub-communities (sub-partitions), this algorithm can be repeated by treating each partition as a separate graph.

To see the algorithm, let us derive the *cut number* R [24, 25, 26], [28, pages 268–270]. We wish to divide the graph's nodes into two groups while minimizing the number of edges R that must be cut to disconnect them:

$$R = \frac{1}{2} \sum_{\substack{i,j \text{ in} \\ \text{different} \\ \text{groups}}} A_{ij} = \frac{1}{4} \sum_{i,j} A_{ij} (1 - s_i s_j) ; \tag{2.1.2}$$

where

$$s_i = \begin{cases} +1; & \text{if vertex } i \text{ belongs to group 1,} \\ -1; & \text{if vertex } i \text{ belongs to group 2.} \end{cases} \tag{2.1.3}$$

The spectral bisection method seeks to choose an appropriate partition to minimize R. Rewriting the first sum,

$$\sum_{i,j} A_{ij} = \sum_i k_i = \sum_i s_i^2 k_i = \sum_{i,j} s_i s_j k_i \check{Z}_{ij} ; \tag{2.1.4}$$

gives R solely in terms of the Laplacian:

$$R = \frac{1}{4} \sum_{i;j} s_i s_j \left(k_i \check{Z}_{ij} - A_{ij} \right) = \frac{1}{4} \mathbf{s}^T L \mathbf{s}; \tag{2.1.5}$$

where $\mathbf{s}$ is the vector with elements defined by Eq. (2.1.3). Writing $\mathbf{s}$ in terms of the normalized eigenvectors $\mathbf{v}_i$ of L,

$$\mathbf{s} = \sum_{i=1}^{N} a_i \mathbf{v}_i \tag{2.1.6}$$

where $a_i = \mathbf{v}_i^T \mathbf{s}$, reduces R:

$$R = \frac{1}{4} \sum_i a_i \mathbf{v}_i^T L \sum_j a_j \mathbf{v}_j = \frac{1}{4} \sum_i a_i^2 \lambda_i; \tag{2.1.7}$$

Thus, R for a given partition depends entirely on the eigenvalues and eigenvectors of L.

There is a trivial solution which must be avoided. Since L is singular, $\mathbf{v}_1 = (1;1;1;\ldots;1)/\sqrt{N}$ is always an eigenvector with eigenvalue $\lambda_1 = 0$. If one chooses $\mathbf{s} = \mathbf{v}_1$ then $R = \frac{1}{4} \sum_i a_i^2 \lambda_i = \frac{1}{4} a_1^2 \lambda_1 = 0$, which is certainly minimized. This corresponds to choosing a group of size N and a group of size zero. In other words, the graph has not been partitioned at all. To avoid this solution, fix the two group sizes at N_1 and N_2. This constrains a_1:

$$a_1^2 = \left(\mathbf{v}_1^T \mathbf{s} \right)^2 = \frac{(N_1 - N_2)^2}{N}. \tag{2.1.8}$$

Instead, minimize R by focusing on $\mathbf{v}_2$ (hence it's known as the Fiedler vector). Since λ_2 is the next smallest eigenvalue, we (roughly) minimize R by maximizing a_2. To do this, we want $\mathbf{s}$ to be parallel to $\mathbf{v}_2$ but the elements of $\mathbf{s}$ are constrained to ± 1. The basic method is an attempt to do the best we can:

$$s_i = \begin{cases} +1 & \text{if } v_2(i) \geq 0; \\ -1 & \text{if } v_2(i) < 0. \end{cases} \tag{2.1.9}$$

with some swapping to satisfy a_1. This also justifies the importance of λ_2 (the algebraic connectivity), since it places a lower bound on the value of R.

Overall, this method works very well, but may not always be ideal. A partition is *always* returned, regardless of whether or not the graph naturally possesses communities, and the size of each group is arbitrary. This may or may not be useful, depending on the application. Also, the number of partitions found is always even, which may not be natural for a particular network.

Due to the drawbacks with spectral graph partitioning, and the recent rise in interest in networks in other areas, completely diPerent methods to þnd communities have been investigated. These methods generally fall into one of two categories: **divisive** methods, where (typically) the edges of the graph are cut in a speciþc order, usually based on a centrality measure, or **agglomerative** methods, where all nodes begin in their own community (of size 1), and these communities are then merged step-wise in some prescribed order [29].

2.1.2 Divisive community partitioning

Perhaps the most famous divisive method is the one due to Newman and Girvan [3, 30, 31], it uses **Edge Betweenness**, and is very intuitive. The betweenness of an edge is the number of times that edge appears in the all-pairs shortest paths. Edges that often participate in shortest paths are more \central" in that they are more responsible for transportation across the network. Cutting high-betweenness edges then partitions the network. This method is expensive, however. Finding edge betweenness for a graph scales like $O(N^2)$, since one must þnd all the shortest paths, and this must be recalculated after each edge is cut, for a total cost of $O(N^3)$. This becomes prohibitive for larger networks. For an example of a progressive sequence of divisive partitioning, see Fig. 2.4

In addition to edge betweenness, one can deþne partitioning schemes based on a variety of centrality measures such as closeness centrality,[1] which uses the sum of all

[1] Other centrality measures include Degree centrality, which is just the node's degree; Straightness centrality, which uses the inverse of the shortest path lengths; Eigenvector centrality, which uses the elements of the eigenvector corresponding to the largest eigenvalue of the adjacency matrix; and Information centrality, which uses the change in a quantity similar to straightness under node

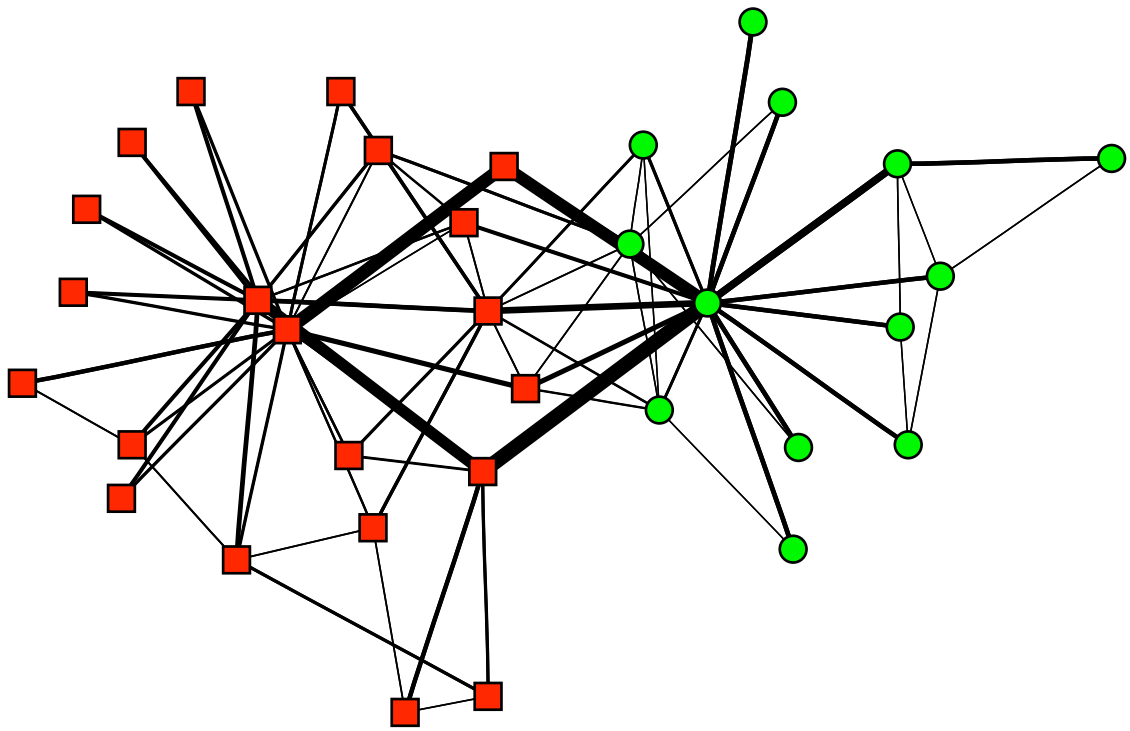


Figure 2.3: The Zachary Karate Club [32], a famous community detection benchmark due to the fact that Zachary observed the club split in half over an argument about membership dues [3]. Edges with higher Betweenness are thicker. Note that betweenness has not been shared equally over paths of equal length, as is usually the case. This is best seen in the two edges leading to the right-most node.

shortest path lengths from a node, and one can use other divisive measures such as node removal [19, 1]. These methods typically scale no better than betweenness with equivalent or lower accuracy (for testable cases such as the Zachary karate club) [2], so betweenness has remained the deþnitive centrality measure used within divisive partitioning algorithms.

deletion.

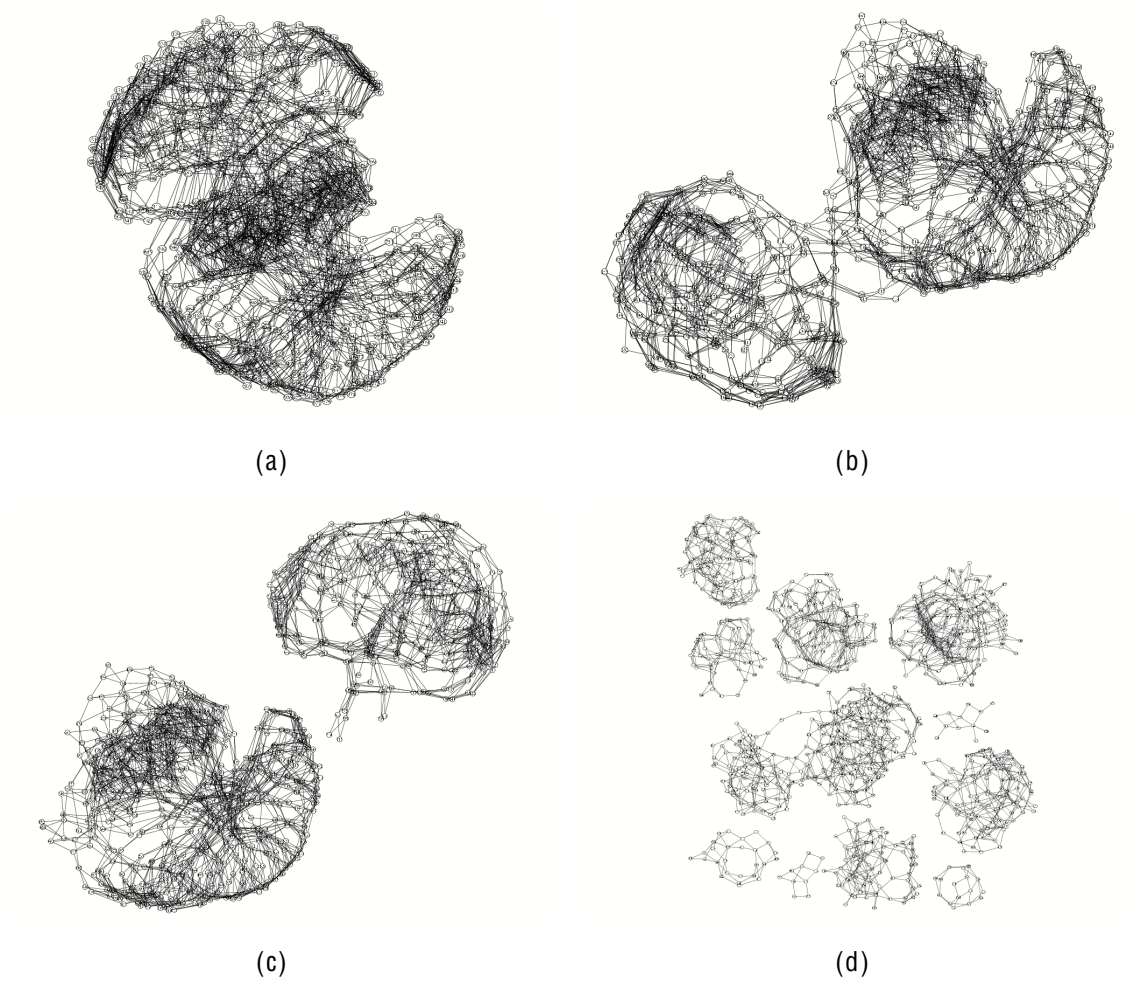


Figure 2.4: An example divisive community partitioning where edges with higher betweenness are cut þrst. Shown is a graph with 0 cuts (a); 100 cuts (b); 120 cuts, where the graph þrst splits (c); and 500 cuts.

2.1.3 Modularity and agglomerative community detection

To combat the expense of betweenness, and to measure the accuracy of a discovered partition, Newman introduced a quantity called **Modularity** [31], and then an agglomerative algorithm using it to þnd communities [33]. This algorithm was then reþned, giving a cost of $O(N \log N)$ [34]. This is one of the most computationally eŽcient algorithm to date and recent improvements [35] have made it up to 70 times faster, allowing analysis of networks with $5 \eth 10^6$ or more nodes.

Modularity is a statistic used to evaluate how \good" a particular community partitioning is; it does not þnd the partitioning. Since a good partition will maximize the number of edges inside each community, it makes sense to deþne a statistic that measures the ratio of intra-community edges to the total number of edges for that partition. For a community partitioning such that vertex v belongs to community c_v, the fraction of edges within communities compared to the total number of edges is

$$\frac{\sum_{v;w} A_{v;w} \check{Z}(c_v; c_w)}{\sum_{v;w} A_{v;w}} = \frac{1}{2M} \sum_{v;w} A_{v;w} \check{Z}(c_v; c_w);$$

where $\check{Z}(c_v; c_w) = 1$ if v and w are in the same community and 0 otherwise. This statistic has its largest value of 1 in the trivial case where all vertices belong to a single community. To correct this, subtract the expected value of the same quantity in the case where edges were randomly placed (no community structure present). The probability of an edge existing between vertices v and w if connections are made at random (and respecting existing vertex degrees) is $k_v k_w / 2M$, where k_i is the degree of vertex i. Then, modularity is deþned to be[2]

$$Q \quad \frac{1}{2M} \sum_{v;w} \quad A_{v;w} \quad \frac{k_v k_w}{2M} \quad \check{Z}(c_v; c_w): \tag{2.1.10}$$

Following modularity, an agglomerative algorithm was introduced [38]. This works by þrst considering each node a community of its own and, at each step, merging the two communities (by re-introducing an edge) that will give the largest (positive)

[2] Further reþnements and a more compact matrix notation are presented in [36] and [37].

change in Q, Ð Q. This is then repeated until only one community remains, and the step with the maximum Q is chosen as the þnal partition. In other words, an approximate (greedy) optimization of Q is employed. Combined with certain data structures,[3] this yields an extremely eŽcient algorithm, with $O(N \log N)$ cost [34].

2.2 A Local Community Detection Method[4]

Often a network is too large to be fully represented or it's too expensive to explore in its entirety. For example, the internet has too many hyperlinks that are changing too much to be succinctly stored in a central location. Another example would be researchers surveying a social population, perhaps in a prison or an isolated tribe. They might not have the time or resources to interview every member of the society, but they might still want to know the community of a particular person such as a leader or authority þgure. Local methods, capable of þnding a particular community within a network without requiring knowledge of every single node and edge are thus of extreme importance.

Here we present a unique community detection algorithm: it uses only local information and is wholly unlike the previously mentioned spectral bisection, divisive, or agglomerative methods. This method þnds a single community inside the network's full structure; global applications capable of þnding the full community partition as well as a hierarchy of sub-communities will also be introduced.

The proposed algorithm consists of an l-shell spreading outward from a starting vertex, $l = 0; 1; \ldots; \ldots$. As the starting vertex's nearest neighbors and next-nearest neighbors, etc., are visited by the l-shell, two quantities are computed: the **emerging degree** and **total emerging degree**. The emerging degree of a vertex is deþned as the number of edges that connect that vertex to vertices the l-shell has not already

[3] Essentially, instead of calculating Q, instead store and update a matrix containing Ð $Q_{i;j}$, the change in modularity when merging communities i and j. This is more eŽcient since Ð Q will always have fewer entries than A and merging two communities will only alter a few elements in Ð Q.

[4] **Published in [39]**

visited as it expanded from the previous $l - 1; l - 2; \ldots; 0$-shells. Edges between vertices within the same l-shell do not contribute to the emerging degree.

Let us introduce the following notation:

$$k_i^e(j) \quad \text{emerging degree of vertex } i \text{ from a shell started at vertex } j; \tag{2.2.1}$$

$$K_j^l \quad \text{total emerging degree of a shell of depth } l \text{ starting from vertex } j. \tag{2.2.2}$$

The total emerging degree of an l-shell is then the sum of the emerging degrees of all vertices on the leading edge of the l-shell. This can also be thought of as the total number of emerging edges from that l-shell [40]. At depth 0, the total emerging degree is just the degree of the starting vertex. At depth l, it is the total number of edges from vertices at depth l connected to vertices at depth $l + 1$. The total emerging degree at depth l is not necessarily the number of vertices at depth $l + 1$, though this approximation is often valid.

It follows from (2.2.1) and (2.2.2) that

$$K_j^l = \sum_{i \in G_l(j)} k_i^e(j); \tag{2.2.3}$$

with $K_j^0 = k_j$. In addition, let us define the *relative change in total emerging degree* ΔK_j^l,

$$\Delta K_j^l = \frac{K_j^l}{K_j^{l-1}}; \tag{2.2.4}$$

for a shell at depth l starting from vertex j.

The algorithm works by expanding outward from some starting vertex j and comparing the change in total emerging degree to some threshold P. When

$$\Delta K_j^l < P; \tag{2.2.5}$$

the l-shell ceases to grow and all vertices covered by shells of a depth $\leq l$ are listed as members of vertex j's community. More specifically:

1. Start at $l = 0$, at vertex j, add j to the list of community members, and compute K_j^0.

2. Spread outward, $l = 1$, add the neighbors of j to the list, and compute K_j^1.
3. Compute Ð K_j^1. If Ð $K_j^1 < P$, then a community has been found. Stop the algorithm.
4. Else repeat from step 2 for increasing l, until P is crossed or the entire connected component of j has been added to the community list.

The total emerging degree of an l-shell started from within a community will tend to increase as l increases, since there tend to be many interconnections within communities. When the l-shell reaches the \border" of the community, the number of emerging edges tends to decrease sharply. This is because, at this point, the only emerging edges are those connecting the community to the rest of the graph which should be, by deþnition, less in number than those within the community.

By introducing a single parameter, P, and monitoring Ð K_j^l, the l-shell's growth can be stopped when it has covered the community. It is this fact that allows for the starting vertex to detect its community locally: at the last depth before P is crossed, it does not matter where the emerging edges lead.

Our method is not perfect, however, and it is possible for the l-shell to \spill over" the community it is detecting. This is dependent on how the starting vertex is situated within the graph: if it is closer (or equally close) to some non-community vertex or vertices than to some community vertices, the l-shell may spread along two or more communities at the same time. To alleviate this eÞect, one can run the algorithm N times, using each vertex as a starting vertex, and then achieve a *group consensus* as to which vertices belong to which communities.

The idea of having an expanding l-shell encompass a community is not in itself new. The algorithm in [40] expands multiple l-shells simultaneously from the n vertices of highest degree (the hubs) until all vertices are within an l-shell. While computationally inexpensive, the number of communities detected is arbitrarily pre-assigned and the possibility of two hubs within the same community is neglected. In addition, it requires simultaneous knowledge of the entire network.

This method has also been generalized to weighted networks, by using summing

the weights on the emerging edges instead of counting them [41].

2.2.1 Global application

The above local algorithm is a method for a single vertex to determine something about its own community membership. It seems reasonable that, by surveying all the locally-determined membership listings, one should be able to generate an idea of the global structure of the network. Here we propose a simple method using a *membership matrix* to obtain such a picture and to overcome membership overlap (vertices claimed by multiple communities; the partition is now a cover) when determining a \consensus" partitioning of the network.

For any given starting vertex j, the algorithm can return a vector $\mathbf{v}_j$ of size N, where the ith component is 1 if vertex i is a member of the starting vertex's community and 0 otherwise. These vectors can be assembled to form an $N \times N$ \membership matrix"

$$\mathbf{M} = (\mathbf{v}_1 | \mathbf{v}_2 | \cdots | \mathbf{v}_N)^{\mathrm{T}}; \tag{2.2.6}$$

where the jth row contains the results from using vertex j as the algorithm's starting point. This allows for a good way to visualize the resultant data when starting the algorithm from multiple vertices.

Unfortunately, this matrix is arbitrarily ordered depending on how vertices are mapped to rows. We introduce a simple sorting step to overcome this. To begin, we define a \distance" $D_{\mathbf{M}}$ between rows i and j as the total number of differences between their elements:

$$D_{\mathbf{M}}(i;j) \equiv \sum_{k=1}^{N} \left[\mathbf{M}(i;k) \neq \mathbf{M}(j;k) \right]; \tag{2.2.7}$$

where $[P] = 1$ if proposition P is true and 0 otherwise. In other words, this is the Hamming distance between rows i and j.

Now we perform a simple sorting algorithm on $\mathbf{M}$. Starting at row $i = 1$:

1. Find $D_{\mathbf{M}}(i;j)$ for all rows $j > i$.

2. Pick the row that is the \closest" to row i (call it row k) and interchange it with row $i + 1$. This requires swapping rows $i + 1$ and k and swapping columns $i + 1$ and k. Columns are swapped because a row interchange is equivalent to a renumbering of the involved vertices, so that new numbering must be kept consistent throughout M.

3. Repeat for row $i + 1$.

Unfortunately, the sorting step can be computationally expensive. Finding each distance costs $O(N)$. When the sorting algorithm begins at the first row, there are $N - 1$ distances to find, so the cost of the first sort is $O(N(N - 1)) \approx O(N^2)$. Sorting the next row requires finding $N - 2$ distances, and so forth. Therefore, since there are N rows, the total cost is

$$\sum_{i=1}^{N} N(N - i) = N - N^2 - \frac{1}{2}N(N + 1) \approx O(N^3). \tag{2.2.8}$$

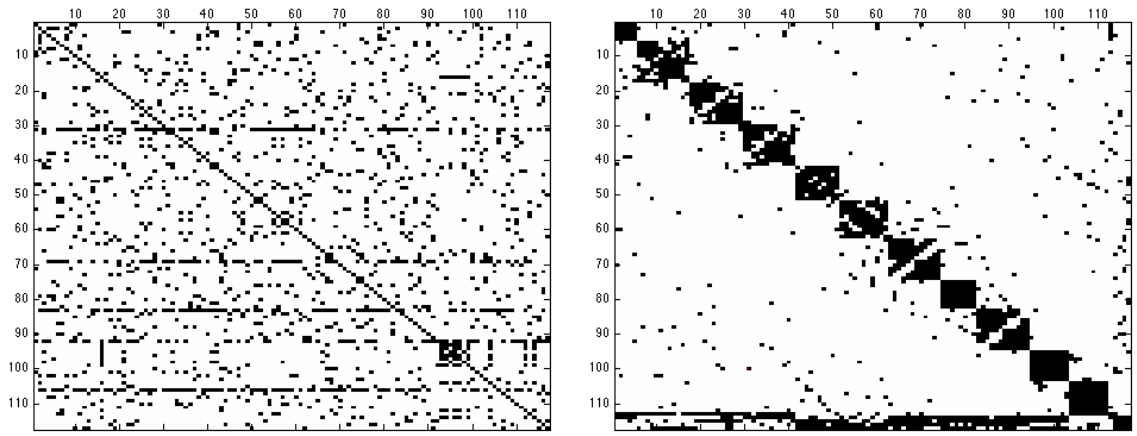
The result of this sorting/renumbering is a matrix that is much more indicative of structure. Well-separated communities appear as blocks along the diagonal and imperfections within the blocks can indicate substructure. See Figs. 2.5 and 2.6(a). The final set of D's can also be used to generate the hierarchy of subcommunities.

2.2.2 Finding a hierarchy of sub-communities

Sorting the membership matrix already provides a useful means of visualizing the results of all the different runs of the local algorithm, but this is not enough to determine how any present sub-communities relate to larger communities. Therefore, we introduce a further operation to apply to M to generate a dendrogram of the community structure.

For row i, we compute a cumulative row distance, CD_i:

$$\begin{aligned} \mathrm{CD}_1 &= 0; \\ \mathrm{CD}_i &= D_{\mathrm{M}}(i; i - 1) + \mathrm{CD}_{i-1} \\ &= \sum_{j=2}^{i} D_{\mathrm{M}}(j - 1; j). \end{aligned} \tag{2.2.9}$$



(a) (b)

Figure 2.5: Unsorted and sorted Membership matrices, NCAA 2005 season. The four bottom rows are the independent teams (Army, Navy, Notre Dame, and Temple).

Plotting the row number i versus the cumulative distance CD_i will yield a collection of points of increasing value falling into discrete bands that indicate the members of each community. See Fig. 2.6(b) for an example. Note that the row number i is the new sorted number i for that vertex: one needs to keep track of all the individual sorting operations to maintain the original number of that vertex. These plots are useful for visualization but are not strictly necessary to generate the sub-community hierarchy. Finally, to yield a dendrogram of the community structure, perform the operation outlined in Table 2.1.

Grouping the rows of the sorted M as per Table 2.1 is equivalent to grouping the vertices of the graph together into a sub-community hierarchy. This is also similar in form to many agglomerative or clustering techniques, with the row distances of M used as a similarity measure. These groupings can then be used to generate a dendrogram of the sub-community structure if we assume that each vertex is a singleton before we started grouping and that after the largest distance is used, all vertices are grouped together. See Fig. 2.7 for an example dendrogram. This algorithm was later applied by Porter, et. al. [41] to þnd the hierarchical structure of the US House of Representatives, based on committee voting patterns, and was shown to be the most

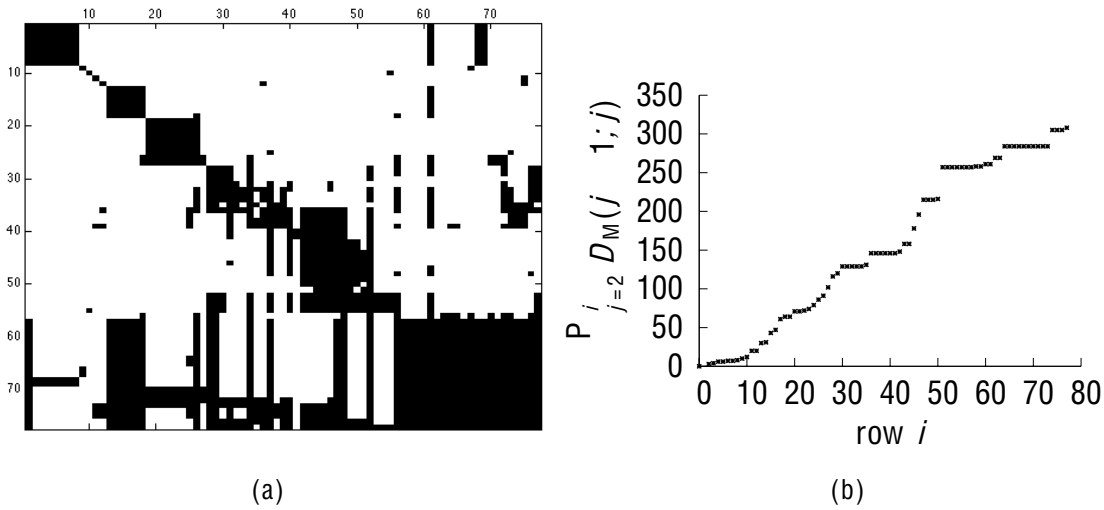


Figure 2.6: (a) The sorted membership matrix for the Les Mis network with $P = 6.9$ and (b) a plot of the cumulative row distances from (a).

accurate of the tested methods.

2.2.3 Conclusions and future work

It is worth pointing out that one can apply the same sorting algorithm to the adjacency matrix, and this will often give a similar block-diagonal structure when communities are present. This does not mean that we do not need the membership matrix. Instead, this is an indication of finite-size effects: for real-world networks such as the NCAA football schedule, nodes within the same community are typically neighbors. This is why the adjacency matrix will exhibit the same block diagonal structure. This should not be expected in general, and thus the local algorithm is still important.

The disadvantage of M is the expense of calculating, storing, and sorting it.[5] In principle, one need not initiate the local algorithm from all N vertices, but instead from just $O(C)$ vertices, where C is the number of communities present, since you really only need one starting node per community. This will be much more efficient, since we typically expect $C \ll N$. Future research will study the effectiveness of the

[5] It is worth noting that the sorting cost in Eq. (2.2.8) is rather naive and improvements, such as using a heap, may reduce this cost to $O(N^2 \ln N)$, for example.

Table 2.1: Clustering the sorted membership matrix to þnd sub-communities. We move downward, grouping together all the vertices whose corresponding rows are closer together than $D_{\min}$ until we arrive at a row that is farther away than $D_{\min}$. Then we start a new group and begin grouping the subsequent vertices together until we *again* þnd a row that is farther away than $D_{\min}$, and so forth. This is repeated using the next smallest $D_{\mathrm{M}}(i \quad 1; i)$ as $D_{\min}$. This has a course-graining eÞect: as we use larger values of $D_{\min}$, farther vertices will start grouping together.

1. $d \quad 1$.
2. For the sorted M, compute $D_{\mathrm{M}}(i \quad 1; i)$ for all $i = 2; \ldots; n$.
3. Choose the smallest $D_{\mathrm{M}}(i \quad 1; i)$ (often zero for identical rows). Call it $D_{\min}$.
4. $C_d \quad$ empty queue. *// clustering queue*
5. enqueue 1st vertex $\quad ! \; C_d$.
6. For $i = 2; \ldots; n$:
   (a) If $D_{\mathrm{M}}(i \quad 1; i) > D_{\min}$:
      i. $d \quad d + 1$.
      ii. $C_d \quad$ empty queue.
   (b) enqueue i-th vertex $\quad ! \; C_d$.
7. Repeat from 3 for next smallest $D_{\mathrm{M}}(i \quad 1; i)$, generating next level of the dendrogram, until all have been used.

discovered partitioning both as a function of P and as a function of the fraction of starting nodes.[6] Another area that must be studied when starting from less than N nodes is the fraction of the network that is *detected*, since it is possible to never visit nodes if no l-shells spread to them. This will be a function of both the number of starting nodes and P, since a very large P will allow even a single l-shell to spread very far.[7] Porter, et al. have also generalized our local algorithm to weighted networks and

[6] Perhaps a three dimensional plot showing modularity as a function of the fraction of starting nodes and P. One expects this to reach a saturation point for some number of starting nodes, after which increasing the number of starting nodes will not improve the detected partitioning. This remains an open question.

[7] Perhaps another three dimensional plot, showing the fraction of the network \discovered" as a function of both P and the number of starting nodes.

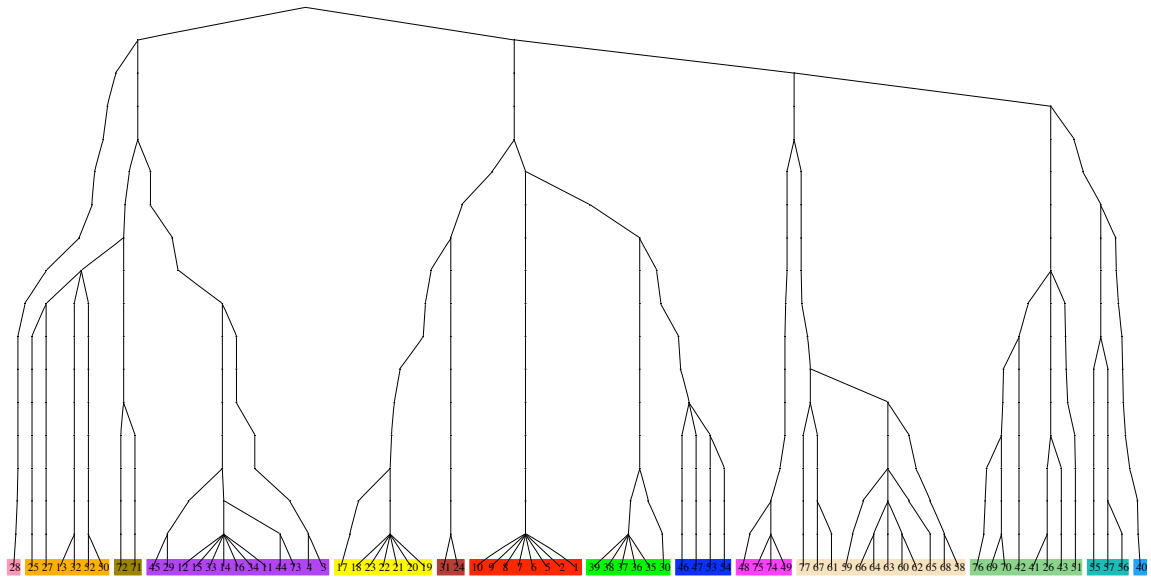


Figure 2.7: The dendrogram of sub-communities for the Les Mis network, calculated using the change in corresponding row distances shown in Fig. 2.6(b). The coloring applied at the bottom was generating by cutting the dendrogram 8 levels from the top, and is also shown in Fig. 2.1(b).

methods have been developed to choose P based on maximizing the modularity [41]. We hope to apply these results in our future work, as well.

2.3 Improved Local Community Methods[8]

Since the publication of the local method described in Sec. 2.2, several new algorithms have appeared. In this section, we brie y detail some of these algorithms, as well as propose a very simple yet surprisingly eÞective new local method. Due to the proliferation of competing methods, an objective **benchmarking** scheme would allow a researcher to compare methods, as well as create and reþne methods for improved accuracy. Here we propose such a scheme.

We will focus our new benchmarking procedure on two existing algorithms, due to **Clauset** [42] and Luo, Wang, and Promislow (**LWP**) [43], as well as a simple new

[8] To appear, *J. Stat. Mech.*

method. Several other local methods exist, including those due to Flake, Lawrence, and Giles [44] and Bagrow and Bollt [39] (Sec. 2.2), but these are either reliant on a priori assumptions of network properties (limiting applicability to speciþc types of networks, such as the WWW), or tend to be accurate only when used as part of a more global method. Other methods (for example, [45, 46, 47, 48]) concern themselves with local community structure, but either require global knowledge to þrst determine this structure, or are deþned locally but do not provide a deþnitive partition necessary for evaluation [49, 47, 50]. Some of these methods may work locally with simple estimates of global values such as the total number of nodes but we neglect these as well, mainly for brevity. Also, some works (e.g. [51]) use similar terminology but are not concerned with local methods in the sense discussed here (they are local in the space of all possible graph partitions, not in the network itself).

All three algorithms begin exploring the network from a starting node s and divide the explored portion into two regions: the community C, and the set of nodes adjacent to the community, B (each has at least one neighbor in C). At each step, one or more nodes from B are chosen and agglomerated into C using some agglomeration scheme, then B is updated to include any newly discovered nodes so that all neighbors of nodes in C are known. This continues until an appropriate **stopping criteria** has been satisþed.[9] When the algorithms begin, $C = \{s\}$ and B contains the neighbors of s: $B = \{n(s)\}$. See Fig. 2.8(a).

2.3.1 Existing local methods

The Clauset algorithm focuses on nodes inside C that form a \border" with B: each has at least one neighbor in B. Denoting this set C_{border}, and focusing on incident edges, Clauset deþnes the following local modularity:

$$R = \frac{\sum_{i,j} b_{ij}[i \in B][j \in B]}{\sum_{i,j} b_{ij}}; \tag{2.3.1}$$

[9] Many \methods" actually consist of just the agglomeration scheme, including most of what is discussed in Sec. 2.3.1. We will discuss stopping criteria later in Sec. 2.3.4.

where β_{ij} is the adjacency matrix comprising only those edges with one or more endpoints in C_{border} and $[P] = 1$ if proposition P is true, and zero otherwise. Each node in B that can be agglomerated into C will cause a change in R, ΔR, which may be computed efficiently. At each step, the node whose agglomeration would give the largest ΔR is agglomerated. This modularity R lies on the interval $0 \quad R \quad 1$ (defining $R = 1$ when $|C_{\text{border}}| = 0$) and local maxima indicate good community separation, as shown in Fig. 2.9. For a network of average degree d, the cost to agglomerate $|C|$ nodes is $O(|C|^2 d)$.

The LWP algorithm defines a different local modularity, which is closely related to the idea of a *weak* community [21]. Define the number of edges inside and exiting C as M_{in} and M_{out}, respectively:

$$M_{\text{in}} = \frac{1}{2} \sum_{i,j} A_{ij}[i \in C][j \in C]; \tag{2.3.2}$$

$$M_{\text{out}} = \sum_{i,j} A_{ij}[i \in C][j \in B]. \tag{2.3.3}$$

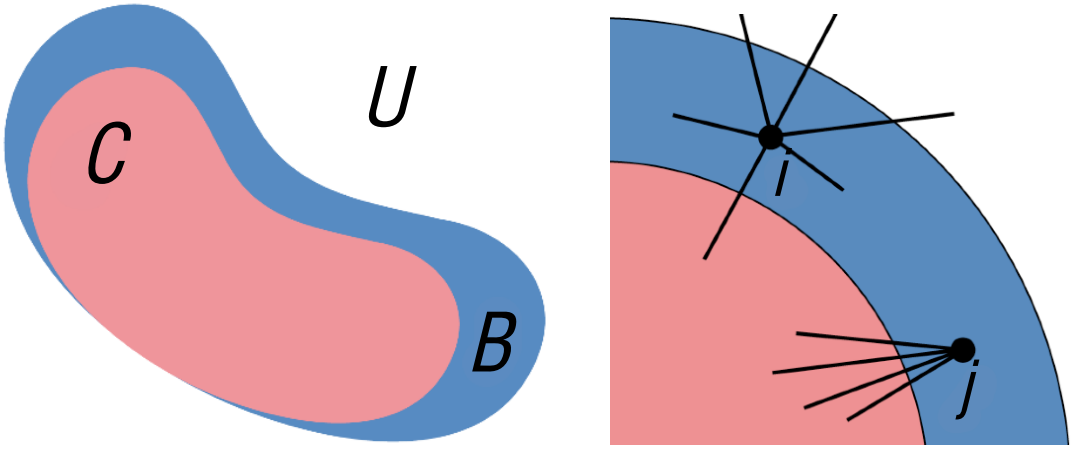
The LWP local modularity M_f is then:

$$M_f(C) = \frac{M_{\text{in}}}{M_{\text{out}}}. \tag{2.3.4}$$

When $M_f > 1/2$, C is a weak community, according to [21]. The algorithm consists of agglomerating *every* node in B that would cause an increase in M_f, $\Delta M_f > 0$, then removing every node from C that would also lead to $\Delta M_f > 0$ so long as the node's removal does not disconnect the subgraph induced by C. (Removed nodes are not returned to B, they are never re-agglomerated.) Finally, B is updated and the process repeats until a step where the net number of agglomerations is zero. The algorithm returns a community if $M_f > 1$ and $s \in C$. Similar to the Clauset method, the cost of agglomerating $|C|$ nodes is $O(|C|^2 d)$.

2.3.2 Outwardness agglomeration

Finally, we present a new (almost toy model) algorithm, as an illustration of how simple a local method can be and as a new test setting for our benchmarking procedure.



(a) (b)

Figure 2.8: (a) The community C is surrounded by a boundary of explored nodes B. This exploration implies an additional layer of nodes that are known only due to their adjacencies with B. (b) Two nodes i and j in B, with $_i = 2=3$ and $_j = 1$. Moving node j into C will give improved community structure, compared to moving i.

Let us deþne the \outwardness" $_v(C)$ of node $v \ 2 \ B$ from community C:

$$_v(C) = \frac{1}{k_v} \sum_{i 2 n(v)} \text{ð} \ i \ 2 \ C \text{Ł} \quad \text{ð} \ i \ 2 \ C \text{Ł} \tag{2.3.5}$$

$$= \frac{1}{k_v} \left(k_v^{\text{out}} \quad k_v^{\text{in}} \right) ; \tag{2.3.6}$$

where $n(v)$ are the neighbors of v. In other words, the outwardness of a node is the number of neighbors outside the community minus the number inside, normalized by the degree. Thus, $_v$ has a minimum value of 1 if all neighbors of v are inside C, and a maximum value of $1 \quad 2=k_v$, since any $v \ 2 \ B$ must have at least one neighbor in C. Since þnding a community corresponds to maximizing its internal edges while minimizing external ones, we agglomerate the node with the smallest at each step, breaking ties at random. See Fig. 2.8(b).

This method is eŽcient for the following reasons. When a node $v \ 2 \ B$ is moved into C, only the neighbors of v will have their outwardness' altered. For a neighbor node $i \ 2 \ n(v)$, the change in $_i$ is just Ð $_i = \quad 2=k_i$ since only a single link can exist

between v and i. If node i was not previously in B, it will now have a single edge to C and $_i = 1 \quad 2/k_i$. Calculating $_i$ at each step thus requires knowing only k_i, which may be expensive (for example, on the WWW), but need only be calculated upon the initial discovery of i.

For eŽciency, one can maintain a min-heap of the outwardness' of all nodes in B then, at each step, extract the minimum with cost $O(\log jBj)$, and update or insert the neighboring 's. For a network with average degree d, the cost of this updating is $O(d^2 \log jBj)$. This is often an overestimate, depending on the community structure, since a node's degree need only be calculated once. Then, the cost of agglomerating jCj nodes is $O(jCjd^2 \log jBj)$. The relative sizes of C and B are highly dependent on the particular network and the current state of the algorithm, but jBj ¾ jCj seems reasonable. A sparse network with rich community structure would give a cost of $O(jCj \log jCj)$.

While seeking to agglomerate the least outward nodes at each step seems natural, it lacks a nicely deþned quality measure, analogous to R in the Clauset agglomeration (or Q for global algorithms). To overcome this we simply track M_{out} during agglomeration. The smaller this is the better the community separation, so we expect local minima in M_{out} when a community has been fully agglomerated. In addition, M_{out} can be easily computed alongside agglomeration. After agglomerating node v, the change in M_{out} is just Ð $M_{\text{out}} = 2k_v^{\text{out}} \quad k_v$: As shown in Fig. 2.10, M_{out} provides useful information about a real-world networks' community structure, in this case the amazon.com co-purchasing network.[10]

Using M_{out} as a measure of quality is not ideal, however: it's not normalized, and (like the Clauset modularity) obtains a trivial value when the entire network has been agglomerated. The latter is less of an issue for local methods. More worrisome is the fact that M_{out} may also be trivially small when C is small. See Fig. 2.9 for a comparison of R and M_{out}. We continue to use M_{out} for the sake of simplicity, but more involved measures may certainly lead to improved results.

[10] This data was generated by crawling the actual links on each amazon product page that point to co-purchased products. This network evolves over time and results are necessarily altered.

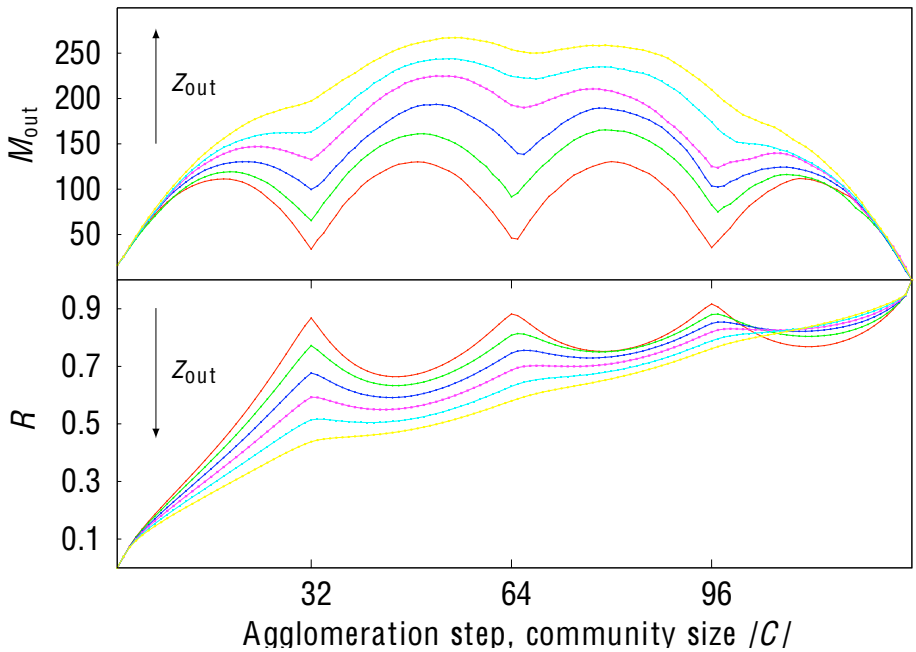


Figure 2.9: Comparison between quality measures for the Clauset algorithm, R, and the method presented here, M_{out}. Shown are the average of 500 realizations of the 128 node ad hoc networks (Sec. 2.3.3), for $z_{out} = 1, 2, \ldots, 6$.

2.3.3 Benchmarking

We now reach the main focus of this section, a specific method for testing the accuracy of local algorithms. We will show that our new method provides insight into how and why a local algorithm performs well or poorly. It will also be shown to be useful for designing new algorithms as well as comparing existing ones.

The procedure consists of two components: the construction of suitable artificial "test" networks, which possess a tunable degree of community structure, and a means of measuring how accurate the algorithm's result is compared to the test network's built in communities. We discuss new and existing test networks and an information theoretic means of comparing the "real" and "found" community partitions.

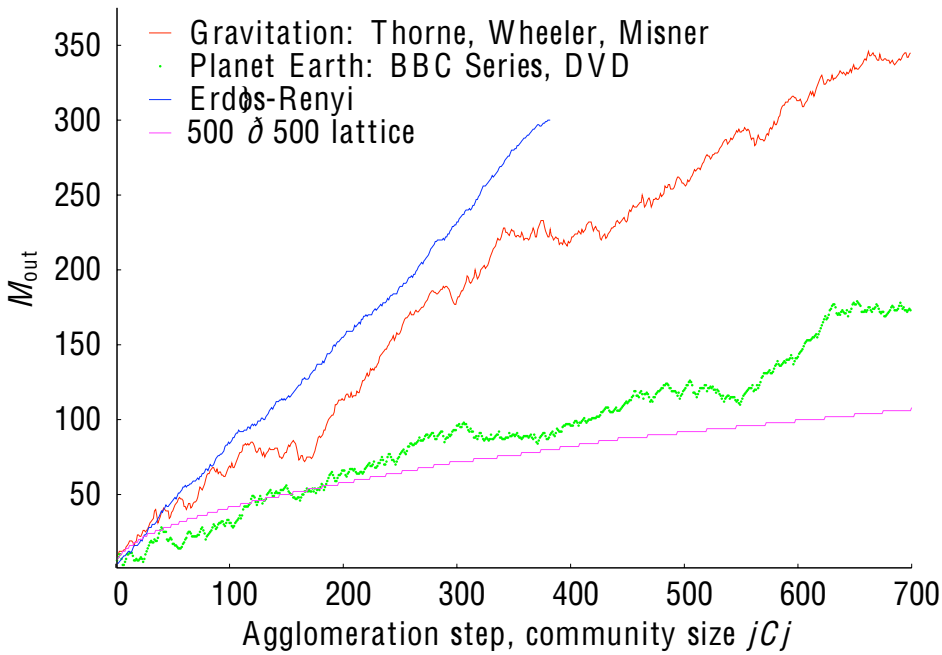


Figure 2.10: Comparison of a seminal physics text and a popular DVD (#1 seller at the time of calculation) on the amazon.com co-purchasing network. Fluctuations in M_{out} in both items indicate the presence of non-trivial community structure. The smooth curve at bottom is for a 2D periodic lattice of 500 ð 500 nodes and the Erdös-Renyi graph has $N = 10^4$ and $hki = 3$.

Test graphs

It has become standard practice to test community algorithms with synthetic networks that possess a given community structure and a parameter to control how well separated the communities are. The traditional example is the so-called \ad hoc" network [31, 52], which typically possess 128 nodes divided into four equally sized communities. Each node has (on average) degree $z = z_{in} + z_{out} = 16$, where z_{out} is the number of links a node has to nodes outside its community. A smaller z_{out} (and correspondingly larger z_{in}) leads to communities that are easier to detect.

These ad hoc networks have a sharply peaked degree distribution. Since local algorithms are dependent on a particular starting node, their accuracy might be

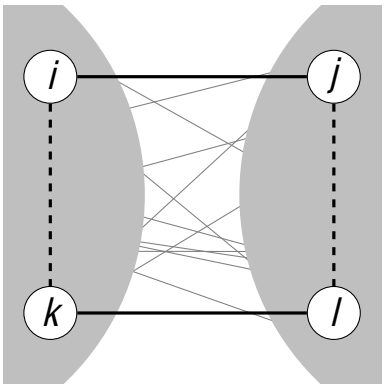


Figure 2.11: The rewiring scheme to build the new artiþcial networks. For two communities (gray), two external edges (solid lines) are removed and two internal edges (dashed) are created, further separating the communities. One must make sure that the dashed edges do not already exist, otherwise edges are being destroyed instead of moved.

aÞected if the starting node is a hub or a leaf.[11] So one would also like more realistic synthetic networks which possess a wider degree distribution, such as a power law. To do this, we propose the following:

1. Build a graph G of N nodes and M edges, perhaps using the conþguration model and a given degree distribution. Throughout this work, we use Barabasi-Albert graphs of $N = 512$, and $m_0 = 8$.[12]
2. Randomly partition the nodes of G into two or more groups. These will serve as the \actual" communities. We limit ourselves to four equally sized partitions.
3. Choose random pairs of edges that are *between* the same two groups and rewire them to be *within* the groups, in such a way that the degree distribution is unaltered.

This rewiring (or switching) technique, replacing edges $(i; j)$ and $(k; l)$ with edges $(i; k)$ and $(j; l)$ [55, 56], has been used in the past to *destroy* the presence of community structure, allowing for a null model to test for false positives [57]. Here we do the opposite, and communities become more sharply separated as the number of rewirings increases. See Fig. 2.11.

Since the partition is random, the initial modularity Q_0 will be very small. As edges are moved within communities, the þrst sum in Eq. (2.1.10) will grow but the second term will remain unchanged, since the degree distribution is unaÞected.

[11] We term the lowest degree node in the network the \leaf," which is not necessarily of degree 1.

[12] These are built quickly by relaxing the constraint on multi-edges, which are then removed [53, 54]. The total number of edges will vary slightly, and the lowest degree nodes often have less than m_0 neighbors.

Therefore, the modularity of the actual partition $Q(t)$ after t pairs of edges have been moved is

$$Q(t) = Q_0 + \frac{2}{M}t: \tag{2.3.7}$$

Rewiring $M/4$ pairs of edges will give $Q \gtrsim 1/2$.

It has been shown that even random networks can possess large values of Q [58, 46]. This is due to the sparsity of such networks when, e.g., $\langle k \rangle \sim 2$. The benchmark networks used here possess much higher $\langle k \rangle$.

Evaluation

Any local method creates a binary partition of the network into the community itself, C, and the remaining non-communnity nodes, $\bar{C} = V - C$. In a realistic setting V is unknown, but synthetic benchmarks allow one to know the full division. In addition, for a synthetic benchmark, the *true* partition $P_R = \{C_R; \bar{C}_R\}$ is already known, while the found partition $P_F = \{C_F; \bar{C}_F\}$ may differ.

Traditionally, the accuracy of the found communities is quantified by the fraction of correctly identified nodes. This has been shown to have drawbacks [52] and the binary partitioning of a local algorithm poses further problems. For example, if the algorithm fails to stop in time, it has still identified every node in the community correctly, there are just additional nodes incorrectly attributed to that community. Should each incorrect node give a penalty? If the algorithm incorrectly finds one community of N nodes, when there were actually K communities of N/K nodes each, one could assign a $+1/N$ for each correct node and $-1/N$ for each incorrect node, giving a composite score of $2/K - 1$. This means that synthetic networks with different K's cannot be directly compared. While scores could be subsequently re-normalized to lie between 0 and 1, we propose an alternative that avoids these problems and is unambiguous. (Sec. IX of [46] provides another alternative.)

Following the application introduced in [59], we use Normalized Mutual Information [60, 61] to measure how well P_R and P_F correspond to each other:

$$I(P_R;P_F) = \frac{-2\sum_i\sum_j X_{ij}\log\frac{X_{ij}N}{X_{i:}X_{:j}}}{\sum_i X_{i:}\log\frac{X_{i:}}{N} + \sum_j X_{:j}\log\frac{X_{:j}}{N}}; \tag{2.3.8}$$

where X is a 2×2 matrix with X_{ij} being the number of nodes from real group i that were placed in found group j, $X_{:j} = X_{1j} + X_{2j}$, and $X_{i:} = X_{i1} + X_{i2}$. In a sense, $I(P_R;P_F)$ is a measure of how much is known about partition P_R by knowing partition P_F, with $I = 1$ corresponding to perfect knowledge, and $I = 0$ to no knowledge at all. A plot of I versus z_{out} or the number of rewirings will give a picture of how accurate an algorithm manages to identify the benchmark's communities as they become more difficult to find.

In general, the *confusion matrix* X is $N_R \times N_F$ where N_R and N_F are the number of real and found communities, respectively. The application of Eq. (2.3.8) is a limiting case corresponding to the binary partitioning inherent to local algorithms. Comparing partitions is a problem more general than the scope presented here: see App. A for other ideas and general background material, including a derivation of Eq. (2.3.8).

In most figures, we have included a "faked" global method, the Clauset-Newman-Moore (CNM) algorithm [33, 34], for comparison. This was done by running CNM to find the partitioning with the highest modularity, one random community was designated C, and the other communities were grouped together in $\mathcal{C}$. A local algorithm is unlikely to match the accuracy of a global method, as shown. More accurate algorithms than CNM exist, meaning the gap between local and global methods is often worse than illustrated.

2.3.4 Stopping criteria

After identifying an appropriate agglomeration scheme, a local method must also be able to correctly *stop* adding nodes. This point is often neglected and, as will be shown, is a critical component in the accuracy of a local algorithm. Here we suggest

two possible schemes and will use the techniques and benchmarks of Sec. 2.3.3 to compare them. It is important that the stopping criteria is also local; a criteria which spreads to the entire network then þnds, e.g., the largest values of Ð M_{out} is no longer a local algorithm.

These stopping criteria are essentially divorced from the agglomeration schemes of most local algorithms, allowing one to mix and match to þnd more accurate methods. We show this with the Clauset and new method from Sec. 2.3. The LWP algorithm already contains a stopping criteria and we use it unaltered.

Strong communities

As per [44, 21], a subgraph $C \ ^{2} \ G$ is a **strong** community (denoted \ideal" in [44]) when every node in C has more neighbors inside C than outside:

$$k_i^{\text{in}}(C) > k_i^{\text{out}}(C); \quad 8i \ 2 \ C: \tag{2.3.9}$$

This local quantity allows for a very simple, natural stopping criteria: agglomerate nodes until the community becomes strong then, at each agglomeration step, check k^{in} and k^{out} for the newly chosen node and stop agglomerating if the community would cease to be strong. If C never becomes strong, the algorithm won't terminate, indicating a possible lack of community structure in the explored region of the network.

As shown in Fig. 2.12, this \strong to not" criteria works well for sharply separated communities, but tends to fail as the contrast decreases. In a sense, a strong community is *too* strong of a requirement: as the distinction between communities blurs, some nodes must fail Eq. (2.3.9), despite probable membership in C.

We generalize the notion of a strong community in the following way. A community is p-**strong** if Eq. (2.3.9) holds, not for all, but only a fraction p (or more) of the nodes:

$$\sum_{i2C} \left[k_i^{\text{in}}(C) > k_i^{\text{out}}(C) \right] \ ½ \ p \, jCj: \tag{2.3.10}$$

Equations (2.3.9) and (2.3.10) are equivalent when $p = 1$, while the requirement becomes increasingly lenient as p decreases. This allows one to tune the sensitivity by varying p. See Fig. 2.13.

An additional benefit of Eq. (2.3.10) is that multiple values of p can be used simultaneously,[13] since a community that is p_1-strong is also p_2-strong ($p_1 > p_2$). More specifically, for the actual fraction p_{eff},

$$p_{\text{eff}} = \frac{1}{|C|} \sum_{i \in C} \left[k_i^{\text{in}}(C) > k_i^{\text{out}}(C) \right]; \tag{2.3.11}$$

C is p-strong for all $p \leq p_{\text{eff}}$, and not p-strong for all $p > p_{\text{eff}}$.

To use, simply choose a set of appropriate parameters, $\{p_1, p_2, \ldots\}$, perform the local algorithm, and maintain the state of C as each p_i stopping criteria is satisfied. One can further use a quality value, such as M_{out} or R, and choose the best corresponding C_i (in this case, that with the smallest M_{out} or largest R[14]). This "best of $\{p\}$" stopping criterion does not entirely negate the introduction of a new parameter; choosing p too small (e.g. $p = 0.1$) can lead to stopping very early. For this work, we use $\{p\} = \{0.75, 0.76, \ldots, 1.0\}$, but this may be worth further exploration. See Figs. 2.14 and 2.15.

In addition to strong communities, **weak** communities have been defined [21]. A community is weak when $M_{\text{in}} > \frac{1}{2} M_{\text{out}}$. We have found the usage of a "weak-to-not" stopping criteria to be problematic. The impact of a single agglomeration is so small that the community will blissfully continue to grow, far past an appropriate stopping point. Just as the strong stopping criteria is too strong, a weak stopping criteria is too weak. See Sec. 2.3.5 for further ideas, however.

Furthermore, it should be kept in mind that these strong communities can be satisfied by random networks [46, 58], so perhaps this is not the best starting point

[13] Indeed, since stopping criteria are often divorced from agglomeration, all manner of criteria may be used simultaneously, to the point where testing to stop can be more expensive than agglomerating.

[14] We limit ourselves to choosing the smallest $M_{\text{out}} > 0$ ($R < 1$), unless *every* C_i has $M_{\text{out}} = 0$ ($R = 1$). This distinction is important for finite graphs, causing a curious (and artificial) increase in accuracy for larger values of z_{out} (smaller numbers of rewirings). This is because inaccurate results that previously spread to *most* of the network now spread to the *entire* network and are subsequently being ignored, raising the average value of $I(P_R; P_F)$.

for a local stopping criteria. Our benchmarking procedure will also show (Sec. 2.3.5) that there is room for improvement, especially when the communities are less clearly separated.

Trailing least squares

Inspired by plots of R and M_{out}, and in an eþort to increase accuracy when community structure is less favorable, we propose another stopping criteria, based on þtting a polynomial to M_{out} (or R) to þnd local minima/maxima. Suppose n nodes have been agglomerated, þt $y = ax^2 + bx + c$ to the þrst $n \quad 3$ values of M_{out}. Then extrapolate y to points $n \quad 2$, $n \quad 1$, n and test the following:

1. parabola opens downward, $a < 0$ **and**,
2. $n \quad 3 > \quad b/2a$ **and**,
3. $M_{\text{out}}(i) > y(i)$, $i = n; n \quad 1; n \quad 2$ **and**,
4. $M_{\text{out}}(n) \text{ ½ } M_{\text{out}}(n \quad 1) \text{ ½ } M_{\text{out}}(n \quad 2)$.

If all are satisþed, stop agglomerating (and remove the þnal three nodes).

As shown in Fig. 2.12's inset, when you pass the border of the community, M_{out} will start to increase, while the parabola, unaware of the next three values, continues downward. This works whether the minima is a cusp or just an in ection point, so one need not resort to testing þrst versus second diþerences in M_{out}, etc. The þtting also provides a degree of smoothing.

This criteria is somewhat involved and has several semi-arbitrary factors: one could extrapolate to a diþerent number of points, relax some of the constraints, þt a diþerent order polynomial, continue þtting until the criteria ceases to be satisþed, etc. These choices (especially criteria 3 and 4) were actually informed by running the benchmarking procedure over multiple possibilities, and choosing the best one, showing that one can use the benchmarks and Eq. (2.3.8) to actually *design* new algorithms. Our results indicate that the criterion as chosen work well, but further reþnement is certainly possible. We also use this with the Clauset method by þtting

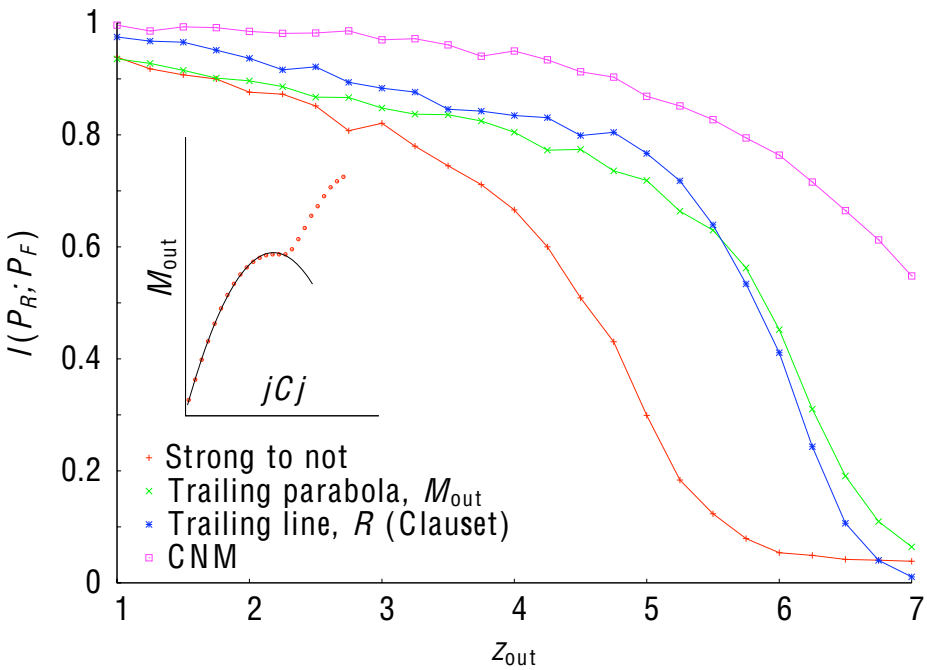


Figure 2.12: The \strong to not" and trailing least squares stopping criteria for the 128-node ad hoc networks using the Clauset method and the new algorithm presented here. Each point is averaged over 1000 realizations. Inset: an example of the trailing least-squares þtting procedure.

a line to R, since Eq. (2.3.1) tends to grow linearly in the þrst community. Both þts have similar accuracy, as shown in Fig. 2.12.

2.3.5 Results and discussion

The results of simulations, shown in Figs. 2.14{2.17, indicate the relative accuracies of the various algorithms and stopping criteria. These þgures show how performance degrades as the communities become less separated (larger z_{out} or smaller number of rewirings). Error bars representing the variance have been omitted for clarity, but note that they are comparable across all algorithms, increase as the communities become more diŽcult to þnd, and are larger than for the global method.

As shown in Figs. 2.14 and 2.17, the LWP method performs extremely well for clearly separated communities, with a rapid decrease in accuracy as the separation

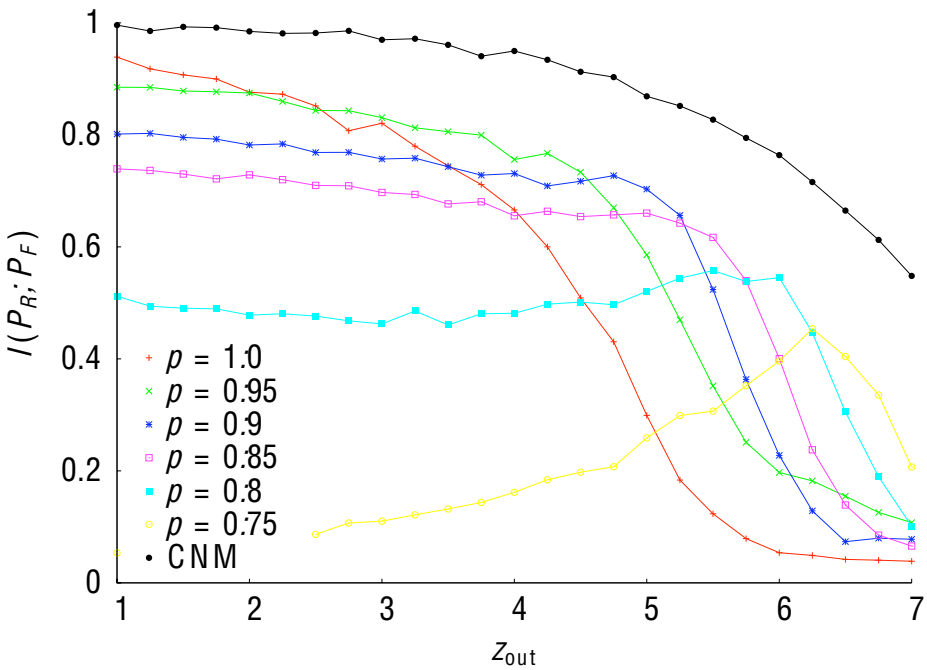


Figure 2.13: Comparison of various p-strong stopping criteria for the 128 node ad hoc networks using the new algorithm shown in Sec. 2.3.

blurs. The \best of *fpg*-strong" (Figs. 2.14 and 2.15) and trailing least-squares (Figs. 2.14 and 2.16) stopping criteria þrst perform at comparable accuracy for both algorithms for the 128-node ad hoc networks, but the trailing least-squares tends to perform better as community distinction blurs. Trailing least-squares outperforms *fpg*-strong in the 512-node networks (Fig. 2.15 vs. Fig. 2.16), suggesting that the size of the community impacts accuracy (which might be expected when þtting data).

Overall, the best of *fpg*-strong has the least accuracy but is also least aÞected by the degree of the starting node. Meanwhile, trailing least-squares performs better overall but is more dependent on the starting node. The LWP algorithm is also quite accurate overall, though trailing least-squares can outperform it when the community separation is less clear.

The \take-home message" from Figs. 2.14{2.17 is this: the performance of a local algorithm is far more dependent on the stopping criteria than the agglomeration scheme. Both the new algorithm and the Clauset algorithm have nearly identical

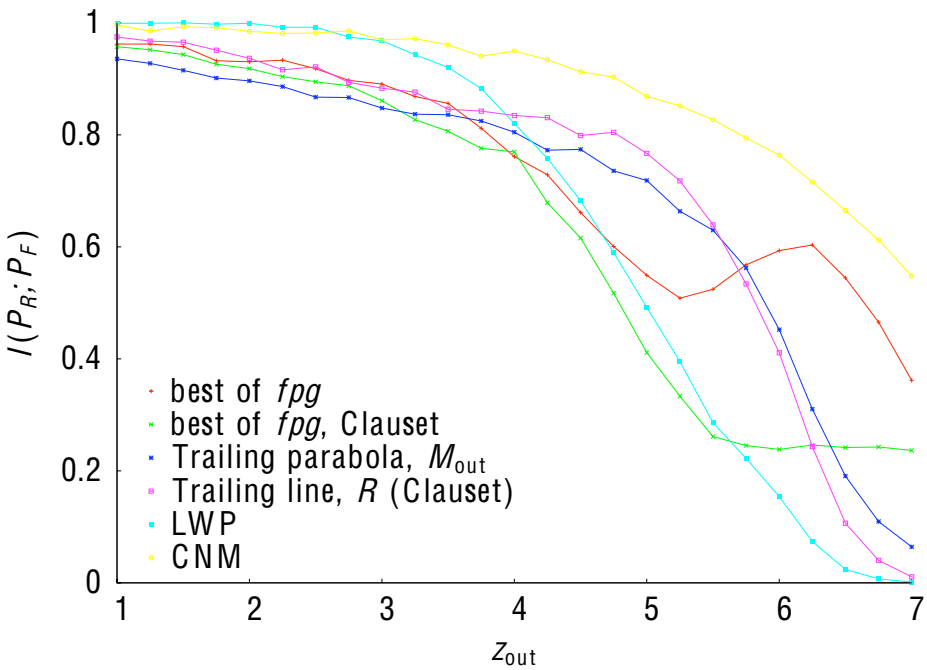


Figure 2.14: An overall comparison of the various methods for the 128-node ad hoc networks, averaged over 1000 realizations. The LWP method is by far the most accurate for low z_{out}, while the trailing least-squares methods offer the best performance at higher values. (The artificial behavior of both `best of *fpg*' criteria for large z_{out} is discussed in Appendix 2.3.4.)

accuracy when using the same stopping criterion. Additionally, there is no clear winner among the algorithms, and they don't perform nearly as well as global methods. The benchmarking procedure shows that these local methods can benefit from improvements.

The agglomeration schemes presented share many similarities, and a certain amount of \cross-pollination" is possible. For example, accuracy may improve if one maintains the outwardness of nodes after agglomeration and, as per LWP, remove every node from C with positive outwardness. Another possibility is simply agglomerating all nodes with the minimum together, instead of breaking ties. This is not necessarily a trivial difference: the agglomeration histories may diverge since the sequence of nodes exposed to B can differ.

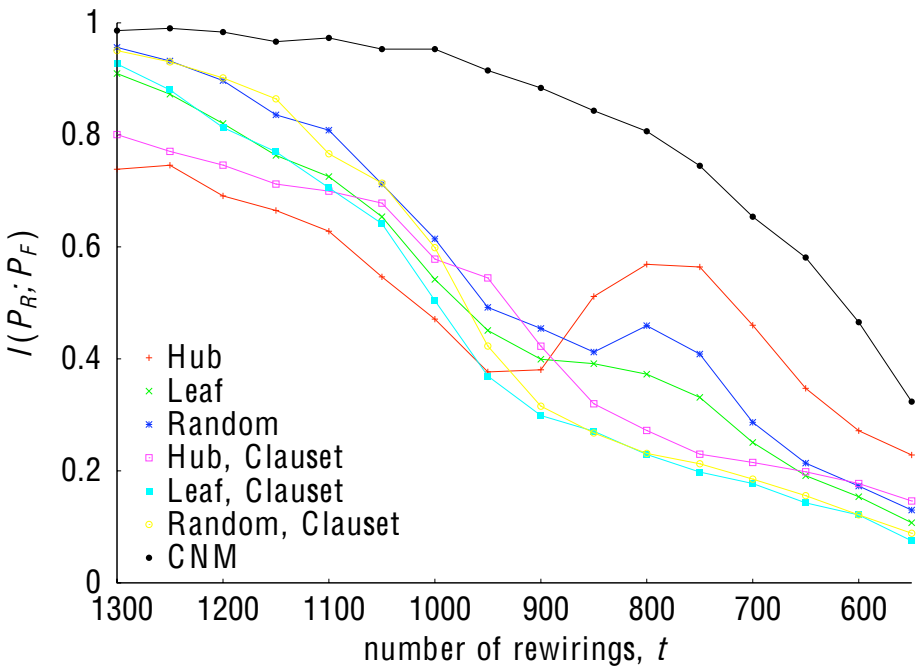


Figure 2.15: Using the \best of *fpg*-strong" criteria on the 512-node rewired scale-free networks, for *fpg* = 0:75;0:76;::::;1. Each point is the average of 500 realizations. The eÞect of rejecting any individual p-strong results where $M_{\text{out}} = 0$ ($R = 1$) (see Appendix 2.3.4) is more apparent for these networks, especially for hub nodes.

There is much room open to develop accurate stopping criteria, and this should be a primary focus of further research. For example, the notion of a weak community can also be generalized to provide a (perhaps improved) stopping criteria. As deþned, a community is weak when $M_{\text{in}} > \frac{1}{2} M_{\text{out}}$. This can be generalized by introducing a parameter to control how strict the constraint is: a community is p-weak when $M_{\text{in}} > pM_{\text{out}}$. Thus, a weak community corresponds to $\frac{1}{2}$-weak, and the LWP stopping criteria is 1-weak. While the introduction of a further parameter is not ideal, and the lack of performance of the p-strong criteria versus the trailing least-squares is not promising, it may still be worth pursuing this and other, similar stopping criteria. Furthermore, stopping criteria using LS-sets and k-cores, as mentioned in [21], may also be worth investigation.

In addition to þnding a single community, any local method could be easily

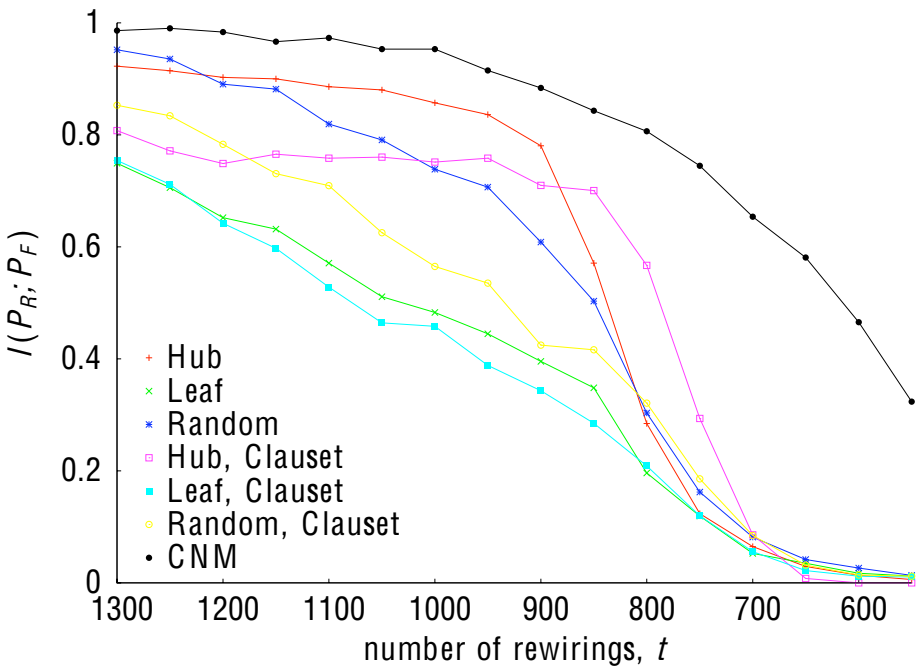


Figure 2.16: A comparison of the trailing least-squares criteria for both the new algorithm and the Clauset method, using the rewired scale-free networks. Starting from a hub tends to be the most accurate, except when the communities are very well separated.

adapted to find more community structure, simply by running the local algorithm multiple times (possibly without repeated agglomeration of nodes or similar modifications). These *quasi-local* methods may not have the same level of accuracy as a global method | agglomerating communities sequentially may lead to compounding errors | but it may still be worth pursuing, even if only as an initialization step for a different algorithm.

There is an implicit assumption, in all these methods, that the underlying network is truly undirected. Of course, this is not generally true. In the WWW it is easy to know what pages an explored web page links to, but it is impossible to know how many other pages may link to the explored page. These *back links* are simply disregarded by the local methods, and it seems a difficult problem to overcome, especially when applying a quasi-local method (with back links continually being discovered as more

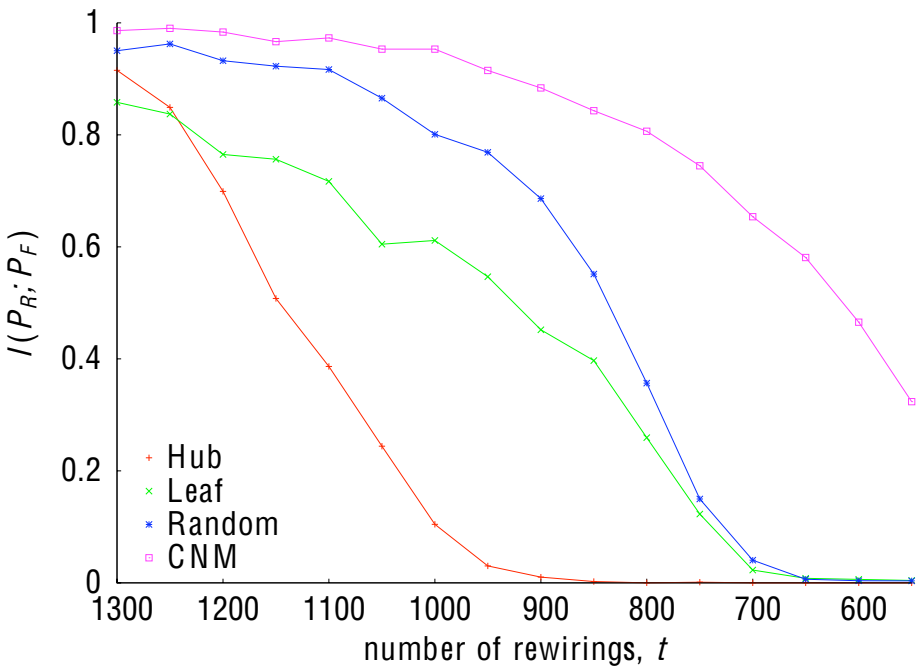


Figure 2.17: The LWP algorithm used on the rewired scale-free networks. LWP performs very well for large numbers of rewirings, but becomes progressively worse as less edges are moved. Both extremes, hubs and leaves, decrease overall accuracy.

communities are found). One possible way to overcome this is to maintain $_V$ after agglomeration, then go through all the found communities, remove nodes with, say, > 0, then re-agglomerate them into the community with the smallest outwardness. Another idea, suggested in [44] is to use a global index, such as a search engine, to list all the back links. It seems that in a different context, such as a partially explored social network, one has no choice but to ignore these back links until they are discovered, then adjust the results accordingly.

2.3.6 Conclusions

Much recent work has been applied to the problem of finding communities in complex networks. We have focused on the idea of finding a particular community inside of a

network without relying on global knowledge of the entire network's structure, knowledge that is unavailable in a variety of areas. We have introduced a new and very simple local method, with a running time of $O(|C| \log |C|)$. Several types of stopping criteria have been introduced, which can be used in conjunction with different agglomeration schemes.

Using Normalized Mutual Information, we have introduced a simple and unambiguous means of quantifying the accuracy of a local algorithm when applied to a synthetic network with pre-defined community structure. Synthetic networks with generalized degree distributions have been used to allow one to test the impact of the starting node's degree, something not possible with existing ad hoc networks.

These techniques have been applied to compare the accuracy of a variety of agglomeration schemes and stopping criteria and we feel they will be of great use when testing newly designed local algorithms. The fact that multiple stopping criteria and algorithms can perform with comparable accuracy shows that the community problem is ill-posed to the point of requiring heuristic methods, and thus it is worth using an evaluation scheme to compare and contrast alternative methods. Developing improved stopping criteria should be a primary goal for future work in this area.

2.4 Conclusions and Open Questions

The problem of identifying communities in complex networks has yielded a diverse collection of possible solutions. From the original spectral bisection methods, through to the divisive and agglomerative techniques, all have provided unique and interesting solutions to this difficult problem. The emergence of modularity as a means of quantifying the *quality* of a discovered partition has allowed for rigorous comparison and evaluation of community detection algorithms.

Here we presented a unique method for detecting communities based on how shells are more interconnected within communities than between them. This also allows for a community to be detected within a network without requiring knowledge

of the entire network, a rare and extremely useful property. A global application of this method was devised as well as a means of identifying the hierarchy of sub-communities and future work will consist of modulating between these two extremes, to þnd a good balance between eŽciency and accuracy.

Much work has been done following the introduction of our original shell-based local algorithm. In response to this, we have created a new benchmarking procedure which allows researchers to create new algorithms and determine quantitatively how they perform. This method used artiþcial benchmark networks and a partition similarity measure built around normalized mutual information and showed that stopping criteria are more important for accuracy than agglomeration schemes. Several stopping criteria were studied, with mixed results. Improvements for these criteria should become the primary focus of research on such local methods. Other problems inherent to local methods, including back links, were discussed, as were new applications for þnding multiple communities (quasi-local methods). There remains many fruitful areas of research worth exploring with this problem.

Chapter 3

Shells

The concept of shells (Ch. 1) and the decomposition of graphs into shells, also known as the inter-vertex distance distribution or shell tomography, has been an important, underlying concept in the majority of the work presented here. The local community detection scheme, presented in Sec. 2.2, used the variation in shell interconnectivity due to the underlying community structure to partition a network without resorting to computationally expensive global statistics or centrality measures. A method to enumerate cycles based on shells (and to relate cycles with community-like structure) is also presented in App. B. Here we delve into the topic directly, deriving new statistics for quantifying networks, estimating said statistics based on assuming an uncorrelated degree distribution, and þnally presenting a new way to measure **bipartivity**, a means of quantifying how \close" a network is to being bipartite (two-colorable). The study of shells will also inform the primary results of Ch. 4.

3.1 Perimetric Edges[1]

We propose that a useful measure of a network is the distribution of what we refer to as **Perimetric Edges.**[2] Edge $e_{j;k}$ is perimetric to a starting node i when $d(i;j) = d(i;k)$,

[1] **Work conducted while visiting Los Alamos National Laboratory, T-7, summer 2005.**
[2] Properly, the decomposition of edges into perimetric and non-perimetric groups.

where $d(x;y)$ represents the length (number of edges) of the shortest path between nodes x and y in G. Perimetric edges always lie within shells.

A perimetric edge inside the d-th shell always participates in at least one odd cycle. One would like to use these perimetric edges to estimate the number of cycles in a network[3] but, while every odd cycle does contain a perimetric edge, cycles can easily share these edges, so the relationship is not clear in general.

Using perimetric edges, we deþne the following statistic:

$$N_{\text{Per}}(i) \qquad \text{the number of edges that are perimetric to node } i. \tag{3.1.1}$$

In addition, we can deþne a similar statistic for edges:

$$E_{\text{Per}}(e_{i;j}) \qquad \text{the number of nodes that edge } e_{i;j} \text{ is perimetric to.} \tag{3.1.2}$$

It seems reasonable to expect that perimetric edges are related to network properties such as feedback and redundancy, due to the relationship with cycles, and to have networks manifest diþerent properties when the distributions of perimetric to non-perimetric edges are diþerent. For example, trees will have no perimetric edges, while large random networks (with uncorrelated degree distributions) should have a very narrow distribution of $N_{\text{Per}}(i)$ and $E_{\text{Per}}(e_{i;j})$. Meanwhile, these statistics are very inexpensive to compute, compared to the full distribution of cycles [62].

Intuitively, we expect that edges with a very large E_{Per} will be less central, since such edges will participate in a lower number of shortest paths. In other words, E_{Per} and edge betweenness should be anti-correlated. Figure 3.1 explores this, and conþrms such intuition, but the relationship is quite weak. Outliers in these plots may be due to regions of peculiar odd-cycle overlap in the network. This remains an open question, however.

[3] The importance of the distribution of cycles is discussed more fully in App. B.

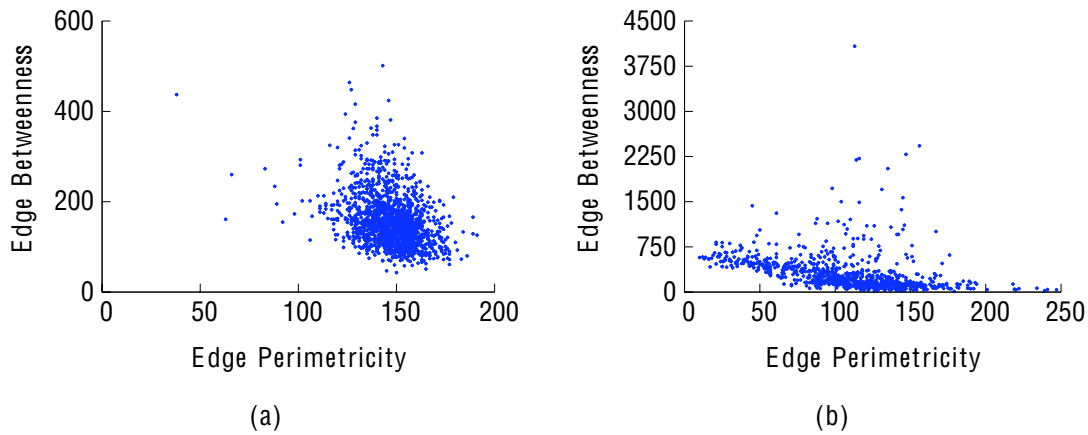


Figure 3.1: Scatter plot of edge perimetricity (horizontal) versus edge betweenness (vertical). Each data point represents an edge in the graph. (a) An Erdős-Renyi graph with 300 nodes and $p = 0.03$. (b) A Barabasi-Albert graph with 300 nodes and $m = 3$. While there is some correlation between the two quantities, it is very weak, especially in (a).

3.2 Shell Distributions

We wish to compute the expected number of edges that are perimetric to a starting node of degree k_s in a random network with a given (uncorrelated) degree distribution $P(k)$. Much work has been done studying the shell decompositions (sometimes referred to as *tomography*) of random networks [63, 64, 65, 66]. We proceed as in [65, 66] with identical notation but several small alterations to improve the results for small networks or small starting node degree. Essentially, we will \build" the network by wiring shells together, one at a time, from a starting node. By keeping track of the distributions of edges within and between shells, we can calculate quantities of interest such as the node perimetricity.

We consider a graph with N nodes of degree given by some distribution $P(k)$ with $k \; 2 \; [m; K]$. At this point, the graph consists of N detached nodes with node i having k_i *open connections* (or stubs), like the conþguration model. We choose a starting vertex with k_s open connections, and we wire those connections to k_s other nodes, thus generating the þrst shell. (See Fig. 3.2 for a helpful illustration of this section's

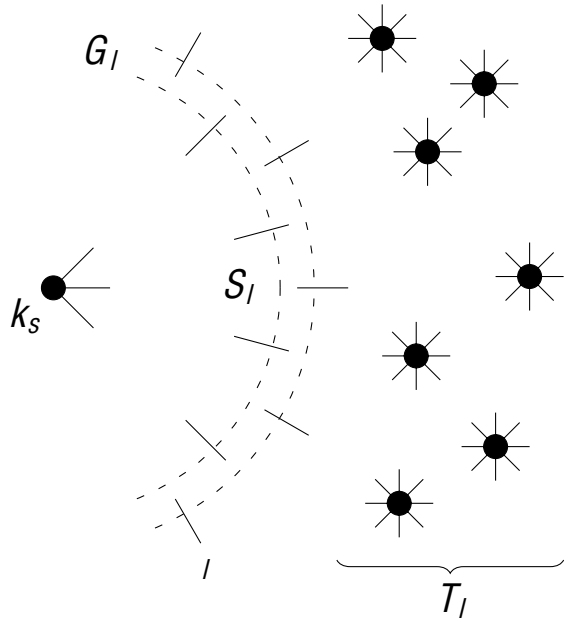


Figure 3.2: The notation used in Sec. 3.2. The dashed semi-circles indicate the l-th shell. Shown is k_s, the degree of the chosen starting node; $_l$, the total number of open connections exiting G_l; S_l, the number of edges connecting $G_{l\ 1}$ to G_l; and T_l, the total number of open connections outside of G_l.

notation.)

All the nodes in shell 1 have one link taken, while the rest of their connections are open. Then, when wiring shell 1 to the remaining open connections, one can wire links to nodes outside of the shell or to other nodes within the shell. Wiring a link back to the same shell generates a perimetric edge. After all the open connections in shell 1 are wired, shell 2 is generated and the process repeats until all connections are closed.

Let us derive the probability that a node with degree k is outside of the þrst l shells, denoted by $P_l(k)$. First, the number of open connections outside of shell l is

$$T_l = N \sum_k kP_l(k): \tag{3.2.1}$$

The probability that an open connection in shell l links to a free node with degree k is $\frac{k}{_l + T_l \quad hk_l i}$, where $_l$ is the number of open connections exiting shell l and $hk_l i\ ^3\ _l/N_l$ is the average out-degree of nodes in shell l, with N_l as the number of nodes in shell l. The $hk_l i$ term tries to account for self-loops.

Now, we can get the conditional probability for a node with degree k to be outside

the first $l+1$ shells, given it is outside of the first l shells. This is the probability that the node does not connect to *any* of the $_l$ open connections exiting shell l:

$$P(k; l+1|l) = \left(1 - \frac{k}{_l + T_l - hk_l i}\right)^{\frac{1}{2}\,_l}. \tag{3.2.2}$$

The probability that a node with degree k will be outside shell $l+1$ is $P_{l+1}(k) = P(k; l+1|l)P_l(k)$. Finally:

$$P_l(k) = P_0(k)\prod_{i=0}^{l-1}\left(1 - \frac{k}{_i + T_i - hk_i i}\right)^{\frac{1}{2}\,_i}. \tag{3.2.3}$$

We can also use P_l to count the number of nodes in a shell:

$$N_l = N\sum_k P_{l-1}(k) - N\sum_k P_l(k). \tag{3.2.4}$$

Now, let's look at the behavior of $_l$ and S_l, where S_{l+1} is the number of links entering the $l+1$ shell. This equals the number of connections exiting shell l minus twice the number of perimetric edges in shell l (since each perimetric edge uses two outgoing connections).

For any given open connection in shell l, there are $_l + T_l - hk_l i$ possible sites to connect to. Of those sites, $_l - hk_l i$ lead to a perimetric edge. Thus, the (approximate) probability for a perimetric edge *within* shell l is

$$P_l(\text{per}) = \frac{_l - hk_l i}{_l + T_l - hk_l i}. \tag{3.2.5}$$

Since there are $_l/2$ possible perimetric edges in shell l, then the number of perimetric edges in shell l, denoted by ł_l, is

$$\text{ł}_l = \frac{_l}{2}\left(\frac{_l - hk_l i}{_l + T_l - hk_l i}\right). \tag{3.2.6}$$

Any open connections in shell l that do not form perimetric edges must then connect to shell $l+1$. Therefore:

$$S_{l+1} = {}_l\left(1 - \frac{_l - hk_l i}{_l + T_l - hk_l i}\right). \tag{3.2.7}$$

The number of connections emerging from all nodes in shell $l+1$ is $T_l \quad T_{l+1}$. This is the number of connections from shell l to shell $l+1$, which is S_{l+1}, plus the number of connections leaving shell $l+1$, which is $_{l+1}$. Rearranging this gives

$$_{l+1} = T_l \quad T_{l+1} \qquad _l \quad 1 \quad \frac{_l \quad \hbar k_l i}{_l + T_l \quad \hbar k_l i} \quad : \tag{3.2.8}$$

Now, Eqs. (3.2.1), (3.2.3), and (3.2.8) form a recursive system that can be iterated numerically with initial conditions $_0 = k_s$ and $P_0(k) \ ^3 \ P(k)$, $k \ 2 \ [m; K]$, with $P(k)$, m, K, and N known. Equations (3.2.4) and (3.2.6) let us compute quantities of interest to our statistics such as the number of nodes per shell and the number of edges perimetric to a starting node i with degree k_s: $N_{\mathrm{Per}}(i) \ ^3 \ \overset{\mathrm{P}}{_l} \ ł_l$.

Figure 3.3 compares these results with simulations. Small networks have been purposefully simulated to illustrate accuracy despite þnite size eÞects.[4] Both of these networks are uncorrelated, however, and Fig. 3.4 illustrates how the theory breaks down when this assumption is no longer true.

3.3 Bipartivity

A network is bipartite (two-colorable) if it can be successfully partitioned (colored) into two groups such that no nodes of the same group are neighbors. Bipartite networks have many applications in areas including social networks [67, 68]. Recently, interest has emerged in **bipartivity**, a quantity measuring how *close* to bipartite a network is [69, 70]. This previous work has introduced bipartivity measures based on frustrated edges of the Ising model [69] or by using spectral measures of the total number of cycles in a graph versus even-cycles [70].

A relationship between bipartivity and perimetric edges is expected, since a network is bipartite when no odd-cycles are present [71]. Motivated by this, and by

[4] For the theoretical result, we have chosen the approximate degree distribution $P(k) \ ^3 \ n_k/N$, taken from a corresponding simulated network.

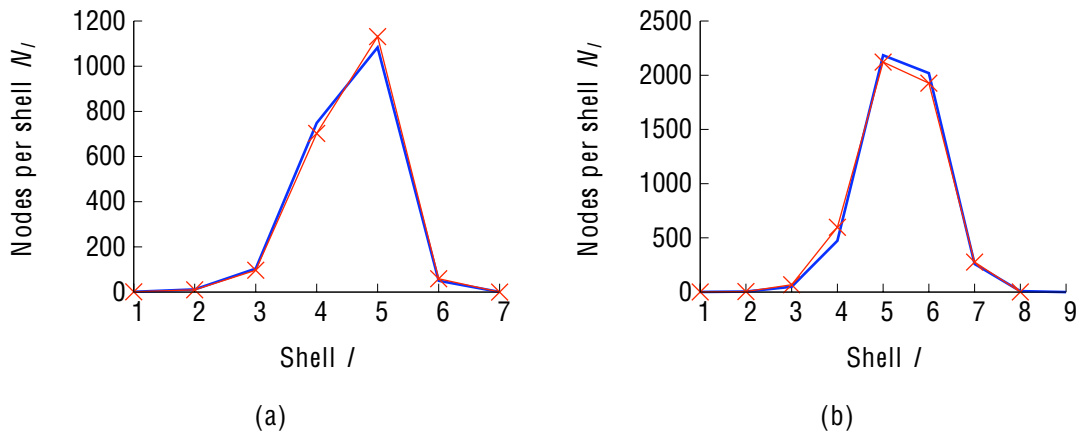


Figure 3.3: The number of nodes per shell, from Eq. 3.2.4 ( ), compared to simulations averaged over 50 runs (∂). Shown is an Erdős-Renyi network of 2000 nodes with p = 0.005 (a) and a Molloy-Reed (configuration model) network of 5000 nodes with $P(k) \propto k^{-2.5}$ (b). For ER graphs, the number of perimetric edges per shell is simply $N_l(N_l - 1)p/2$. A degree-one starting node was chosen for both theory and simulation.

previous measures of bipartivity, we introduce the following related measures:

$$b_{\max} = 1 - \frac{\min(N_{\mathrm{Per}})}{M}; \tag{3.3.1}$$

$$b_{\mathrm{mean}} = 1 - \frac{\mathrm{mean}(N_{\mathrm{Per}})}{M}; \tag{3.3.2}$$

$$b_{\min} = 1 - \frac{\max(N_{\mathrm{Per}})}{M}; \tag{3.3.3}$$

with $N_{\mathrm{Per}}(i)$ given by Eq. (3.1.1). For a bipartite network, no odd-cycles are present and $N_{\mathrm{Per}}(i) = 0$ for all i. Meanwhile, $N_{\mathrm{Per}}(i)$ must grow as odd-cycles are introduced. Therefore, $b = 1$ for bipartite networks and decreases as more edges in the network violate "two-colorability". We expect the differences between $b_{\min}$, $b_{\max}$, and b_{mean} to be minimal, since the distribution of $N_{\mathrm{Per}}(i)$ should be sharply-peaked, especially for larger networks.[5] See Table 3.1 for the bipartivity measures of various networks, for b defined using the min, the mean, and the max of N_{Per}.

[5] This is usually true, but exceptions are possible (see Table 3.1, specifically the airline network). If one consistently uses min, mean, or max when comparing networks, this should not pose a problem, since b is a relative quantity.

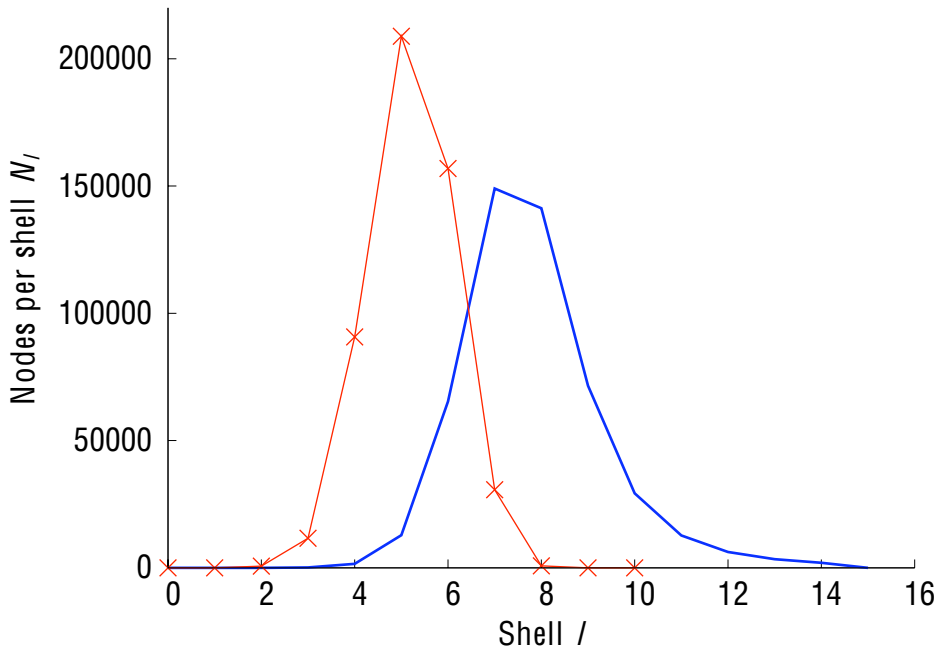


Figure 3.4: The number of nodes per shell, from Eq. 3.2.4 ( ), compared to simulations averaged over 100 runs (δ). Shown is a Barabasi-Albert network of 5 δ 10^5 nodes with m = 2. This network, unlike those shown in Fig. 3.3, has correlations, and this is evident in the lack of alignment between the two curves. These correlations lower the diameter, pushing the curve both leftward and upward, compared to the uncorrelated case.

To þnd a lower bound on b, let us consider the complete graph of N nodes. This graph has $M = N(N \quad 1)/2$ edges. All but $N \quad 1$ of these edges are perimetric to any node, giving $b = 1 \quad (N \quad 2)/N$. It follows that $\lim_{N \to 1} b = 0$, therefore $0 < b \quad 1$ for any þnite network. In practice, $b < 1/2$ or even $2/3$ can be interpreted as being far from bipartite.

3.4 Conclusions and Open Problems

We have presented a recursive analysis of the shell distributions of uncorrelated networks, introduced a new set of statistics to study large networks eŽciently, and applied these statistics to generate a computationally eŽcient calculation of bipartivity.

Table 3.1: Bipartivity for various networks. A network becomes \more bipartite" as b ! 1. In practice, the diPerence between min, mean, and max can be appreciable (although this is rare), but the diPerences decrease as b ! 1.

Network	N	M	$b_{\max}$	b_{mean}	$b_{\min}$
Karate [32]	34	78	0.731	0.664	0.564
Prison [72]	67	142	0.739	0.659	0.585
CS PhD[a] [73]	1025	1043	0.994	0.993	0.988
NCAA 2005[b] [74, 75]	117	616	0.576	0.465	0.386
Grassland [76]	88	137	0.839	0.795	0.730
Scot. Corps.[c] [73, 77]	228	358	1.000	1.000	1.000
Les Miserables [23]	77	255	0.557	0.482	0.392
USAir97 [73, 78]	332	2126	0.602	0.424	0.276
Rogets [73, 79]	994	3640	0.605	0.580	0.555
ODLIS [73, 80]	2898	16376	0.561	0.494	0.410

[a] This network is composed of PhD advisors and their students, and is very nearly a tree (cycles are introduced by students with multiple advisors).

[b] From published schedule at www.ncaa.org.

[c] This network is composed of corporations and their executives as nodes, and is bipartite.

Further study of the fundamental impact of the distributions of perimetric versus non-perimetric edges, including their relationship to the distribution of cycles, is important. One can also improve our current bipartivity measures by using a random sampling of starting nodes, and then study how the \partial" b converges to the actual value. Additionally, larger simulations such as those shown in Fig. 3.3 can be used to judge the accuracy of Eqs. (3.2.1){(3.2.8), especially when computing ł /.

Chapter 4

Network Portraits

Building upon our earlier work with shell distributions, we introduce a new tool for analyzing complex networks. This tool, a network portrait, will be shown to have several unique properties, making it highly useful for both quantitative and qualitative analysis.

4.1 Introduction[1]

A difficult problem when studying networks is that of comparison and identification. Given two networks, how similar are they? Are they identical or, more appropriately, do they arise from the same generating mechanism? Given a real-world network, such as a protein-protein interaction network or an electrical network, how can one determine which random network model most accurately captures the relevant structure?

At the most rigorous level, this is the **Graph Isomorphism** problem: $G = (V; E)$ is isomorphic to $G' = (V'; E')$ if there is a bijection $: V \to V'$ such that $e(x; y) \in E$ iff $e(\ (x);\ (y)) \in E'$ [28]. Like many difficult problems, it is often easier to disprove isomorphism: if $N \neq N'$, then G and G' can never be isomorphic. In addition, discrepancies between the degree distributions would also disprove isomorphism. Of course, these comparisons do not capture all of a graph's structure. In addition,

[1] Published in [81].

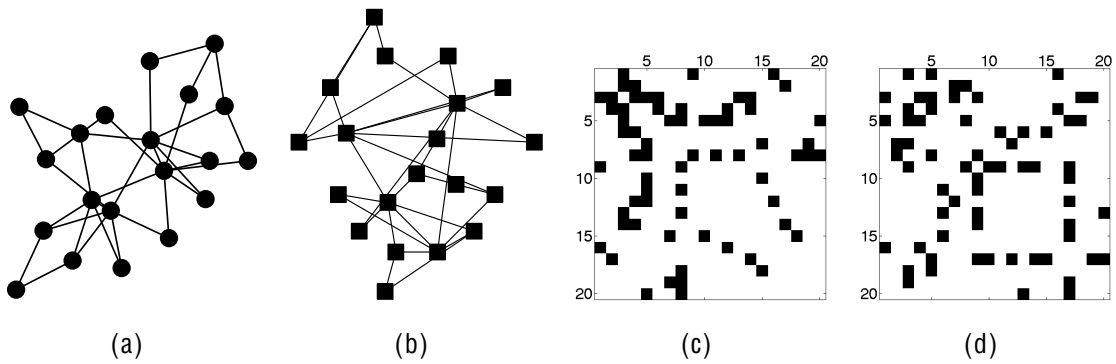


Figure 4.1: Planar embeddings and adjacency matrices for a small network. It is difficult to tell visually that these represent the same network, even at such a small size.

for very large random networks, which are of the most interest, the probability of two networks randomly chosen out of the ensemble of all possible networks being isomorphic is negligible. Graph Isomorphism is, in a sense, *too strict* of a result: we wish to determine if networks are statistically \similar," not identical.

Ideally, one would like to have a data structure that exactly and uniquely encodes the network. Existing structures such as adjacency matrices and edge- or adjacency-lists fail to do this: permutations of rows and columns in the adjacency matrix allow for isomorphic graphs to have different adjacency matrices (though such operations preserve spectra), Meanwhile, edge-lists and other structures are also vulnerable to relabeling, and finding the mapping between two such lists is the entirety of the graph isomorphism problem. See Fig 4.1.

To answer these questions, we propose a new matrix structure B that is truly independent of vertex labeling; it is isomorph-invariant:

$$B_{l;k} \qquad \begin{array}{l}\text{the number of starting nodes with}\\ k \text{ nodes in shell } l.\end{array} \tag{4.1.1}$$

This matrix captures a great deal of structural information about the network, starting with the degree distribution in the first row. Second-, third-neighbors, and so forth, are captured in subsequent rows. In addition, since every node is counted once as a starting node, B is independent of node labeling and permutations: Given a

network G, there is only one B that can be constructed. We term B a **network portrait** due to this invariance, in that it provides a truly unique snapshot of the network and (it will be shown) captures a variety of information about important network properties, similar to how a portrait or photograph contains much information about its subject.

Note that there is some ambiguity regarding how certain quantities are defined. Notably, this matrix has a row 0 and column 0. The zeroth row gives the distribution of nodes in the zeroth shell, which we take to identically be 1 for all nodes: $B_{0;k} = N\,\delta(1;k)$. In addition, the zeroth column contains the distribution of *empty shells*, i.e., how many starting nodes have zero nodes in a shell. The distribution of $B_{l;0}$ increases with l, since a starting node with zero nodes in shell l can not have nodes in shells greater than l. Additionally, we count any starting nodes with zero nodes in shell l as also having zero nodes in all shells l', $l < l' < d$, where d is the diameter of the graph. This normalizes the rows of B, $\sum_k B_{l;k} = N$ for all l, and may have other benefits.

4.2 Examples and Applications

We begin by introducing B for a variety of networks. We begin with a very large real-world network, shown in Figs. Figure 4.2{4.3. (All plots of B are with a logarithmic color scale.) Figure 4.4 shows the portraits of several ER graphs, including how the ensemble average appears and how percolation is readily visible.

Figures 4.5 and 4.6 show B for a variety of periodic and non-periodic lattices. These illustrate the presence of dimensionality with B, the ability to detect defects and imperfections in otherwise homogeneous graphs, and the large-scale impact of boundary conditions on such finite lattices. With periodic boundaries, every node is indistinguishable and this is shown by the single non-zero value in each row of Fig. 4.5(a). The sharp lines illustrate that this representation encodes the dimensionality

of the graph.[2] The non-periodic lattice in 4.5(b) shows a symmetric hierarchy of node types, corresponding to starting nodes' relations to the boundary, yet the dimensionality is still visible. In both, the maximum non-zero value reaches a turning point at a particular row and then decreases; this illustrates finite size effects.

When studying scale-free networks, much focus is placed on the scale-free exponent γ. As shown in Fig. 4.7 however, scale-free networks with the same γ can have radically different properties, and this must be considered when comparing different networks.

Figures 4.9, 4.10, and 4.11 show a variety of real-world networks, including an electric power grid [4.10(a)], the network of airlines inside the United States [4.10(b)], a snapshot of part of the internet backbone [4.9(b)], a collaboration network [4.9(a)], and a variety of cellular metabolic networks [4.10(c-f), 4.11(a)]. Note the distinctive similarity between the various metabolic networks, which is not present in some other networks, such as the power grid. Some of these networks are extremely large; visualizing such networks was previously impossible. Figure 4.8 also shows several sequential illustrations of the emergence of small-world; animations of such quantities are also possible.

4.3 Network Properties

Some network properties are easily calculate from a given B. For example, the number of nodes is $N = B_{0;1} = \sum_k B_{l;k};\ l \in [0; D]$, for an undirected graph with diameter D. Similarly, since the first row of B captures the degree distribution, the number of edges in G is $M = \frac{1}{2}\sum_k B_{1;k}$. Also, $P(k) = B_{1;k}/N$.

Certain mean values are also contained within B. Since the first row contains $P(k)$, we can easily get $\langle k \rangle = \frac{1}{N}\sum_k kB_{1;k}$. We can also use B to calculate the

[2] Indeed, the non-zero values in B form a vertical line (constant slope) for the periodic one-dimensional lattice (the circle graph) and grow quadratically for the three-dimensional periodic lattice.

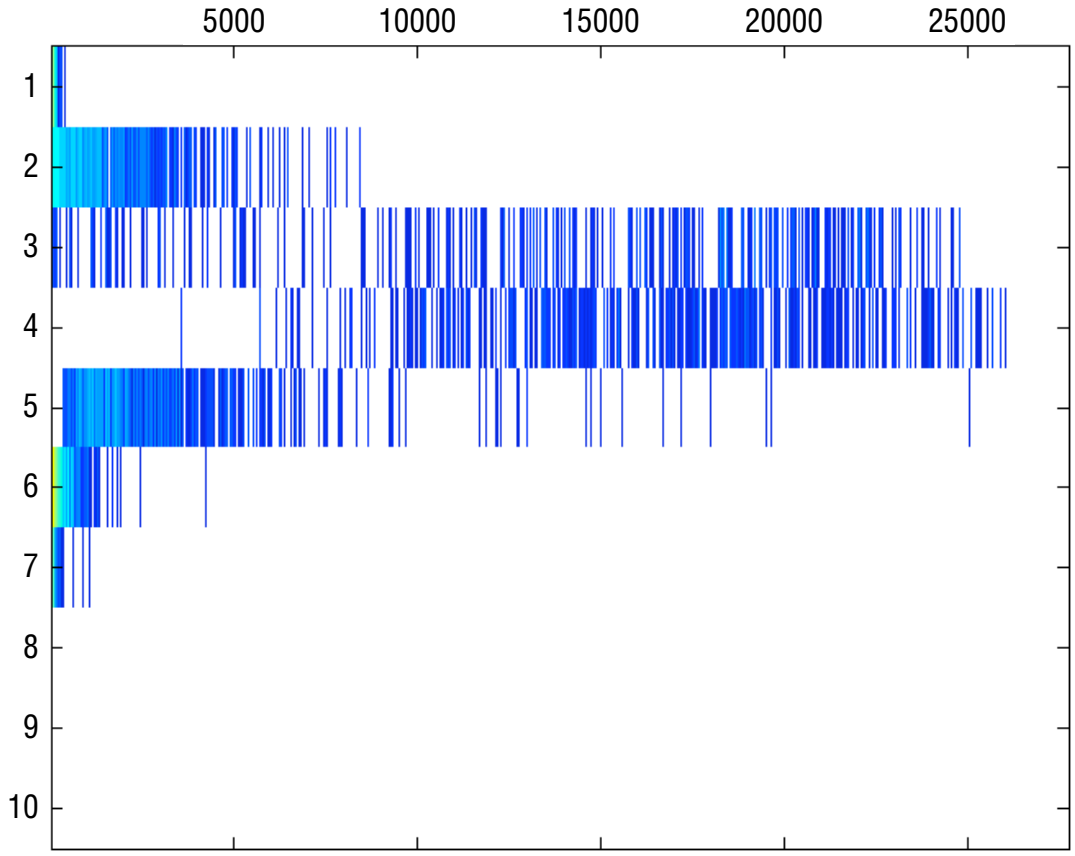


Figure 4.2: A B-Matrix (larger values are darker (brighter), logorithmic color scale, row and column 0 omitted). Note the degree distribution, slightly visible in the þrst row. as well as the turning point about row 4, representing þnite-size eÞects. Shown is the network of the ten percent most connected actors taken from the movie actor collaboration network stored in the Internet Movie Database (www.imdb.com) [82].

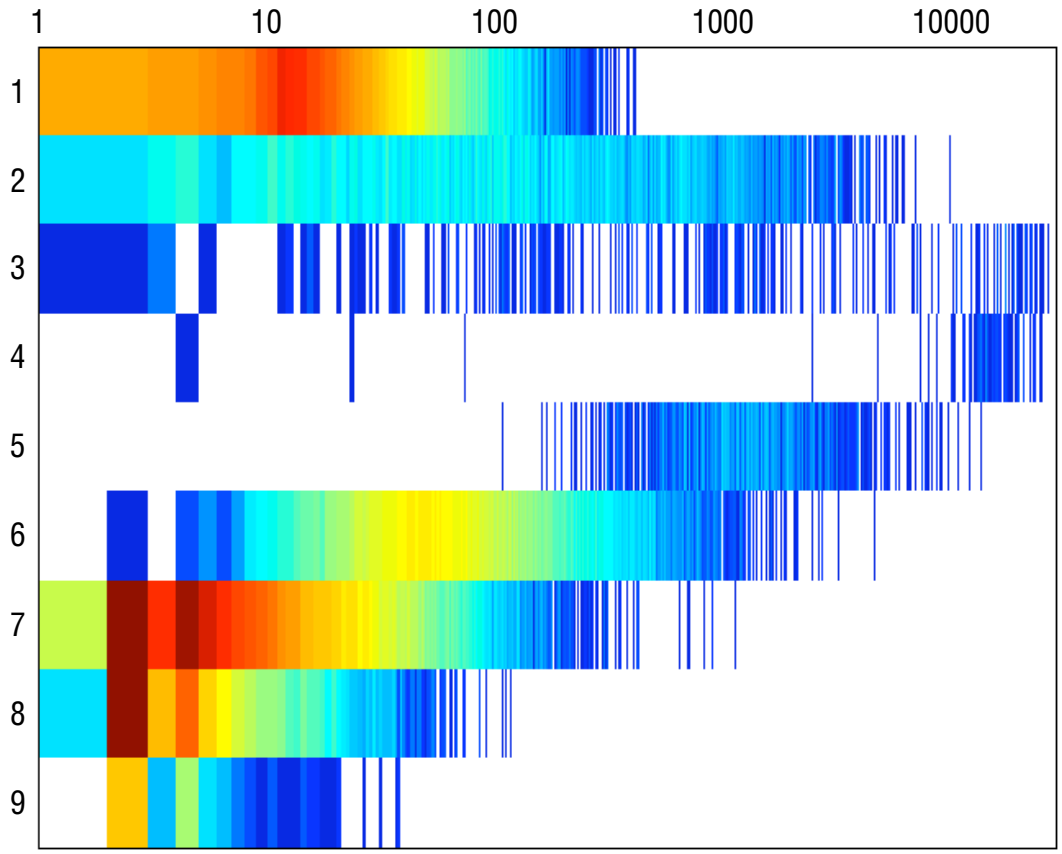


Figure 4.3: The B-matrix from Fig. 4.2 but with a logarithmic horizontal axis. The degree distribution in row 1 is now plainly visible.

average shortest path length:

$$\left\langle \begin{array}{c}\text{all-pairs shortest}\\ \text{path length}\end{array} \right\rangle = \frac{1}{N(N-1)} \sum_l \sum_k k B_{l;k}; \tag{4.3.1}$$

with the denominator being N^2 if including paths of length 0. Meanwhile, the mean eccentricity can be calculated using the zeroth column:

$$\langle \text{eccentricity} \rangle = \frac{1}{D} \sum_{l=1}^{D} l \left(B_{l;0} - B_{l-1;0} \right). \tag{4.3.2}$$

The diameter is also simple to calculate: it's just the number of rows of B minus 1. Or, since every row sums up to N (when counting empty shells as specified), the

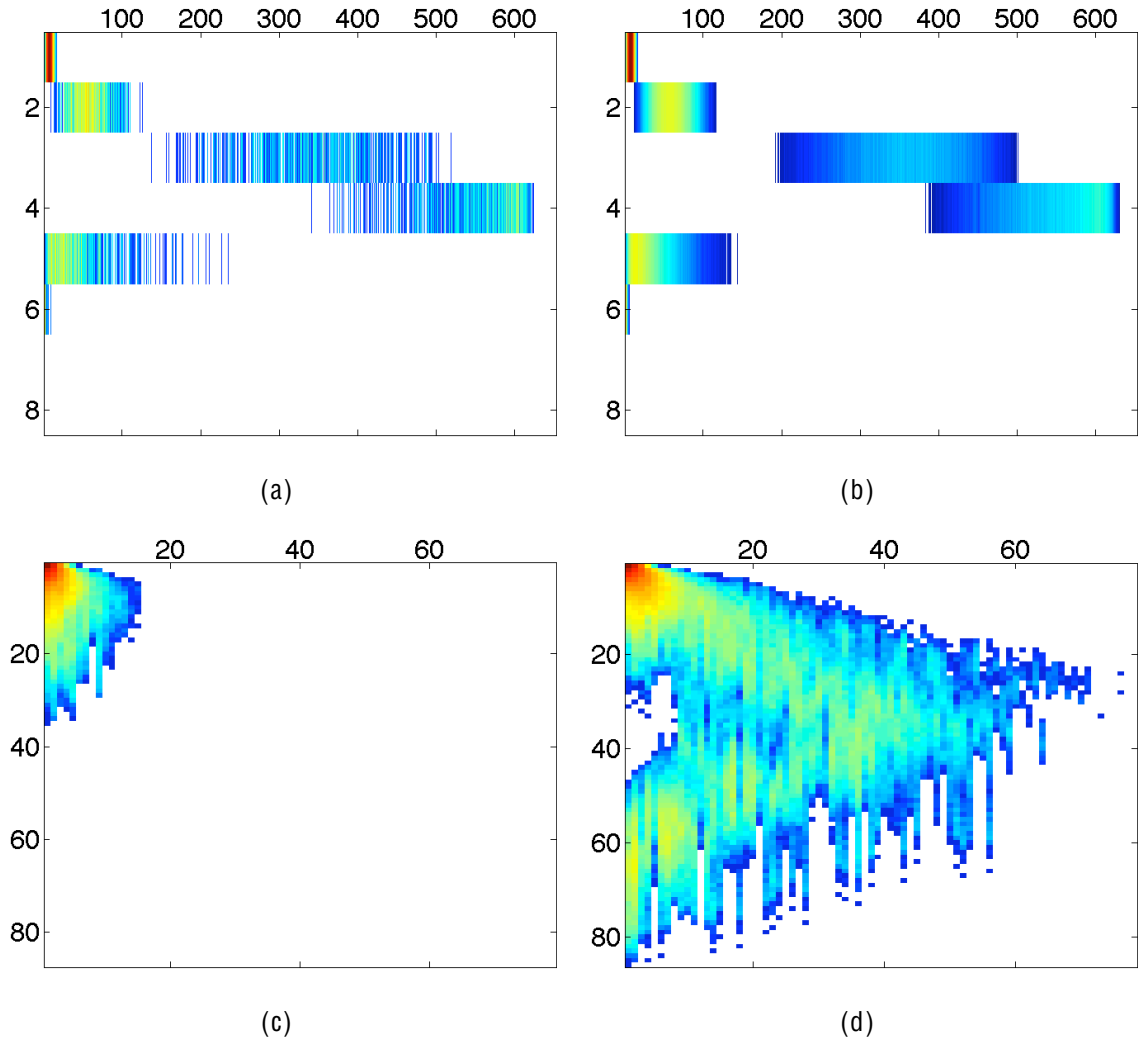


Figure 4.4: Erdős-Renyi (ER) graphs [13]. (a) one graph with $N = 1000$ nodes and $p = 0.008$. (b) The average of 100 graphs from (a). Visualizing percolation: $N = 10^4$ (c) below percolation, $p = (1.1N)^{-1}$; (d) at percolation, $p = N^{-1}$.

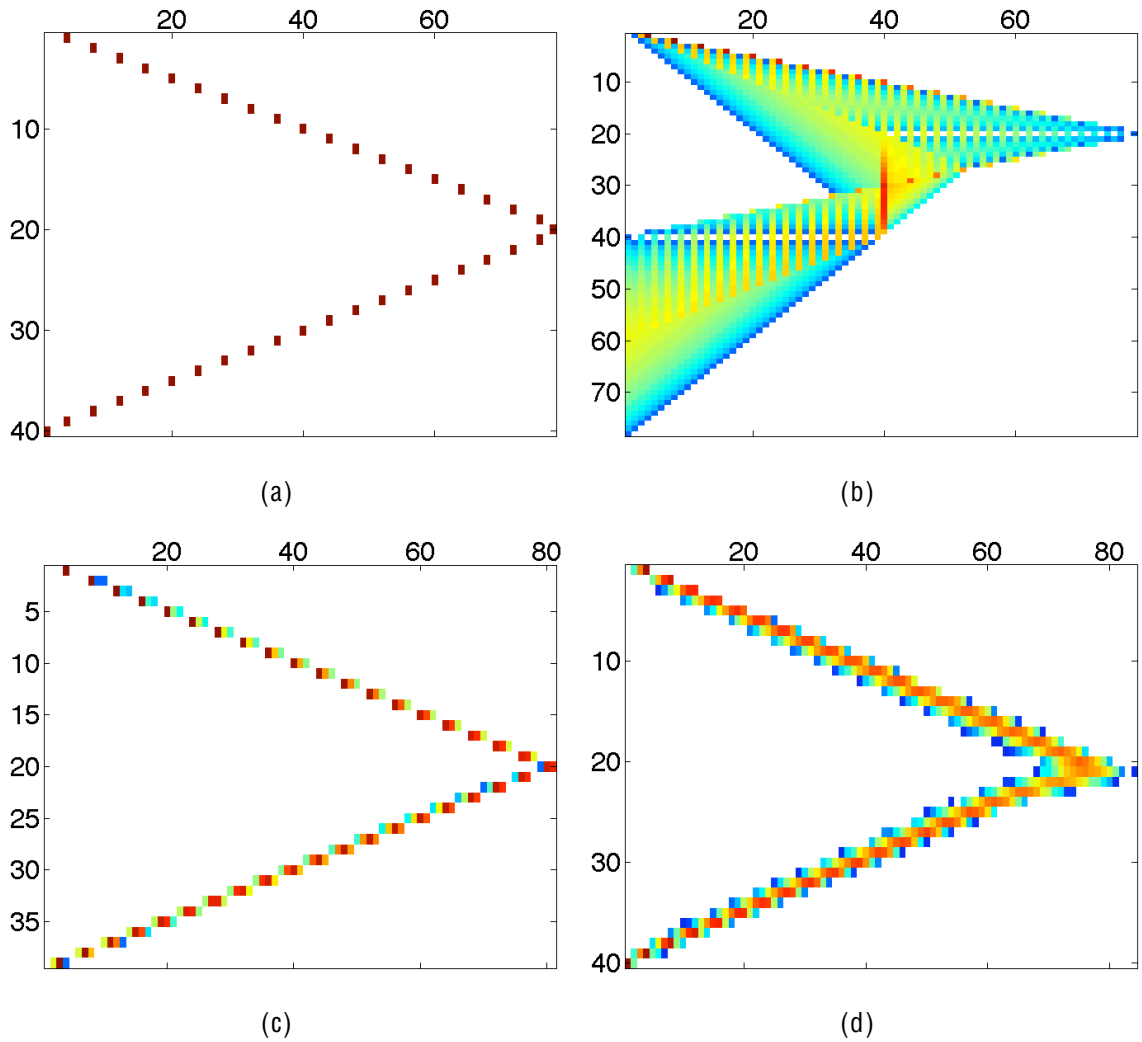


Figure 4.5: Regular 40 ð 40 lattices with defects. (a) A periodic and (b) non-periodic lattice; (c) a lattice with skew-periodic boundaries; and (d) a periodic lattice with a random 5 percent of all nodes missing. Observe the strong linear slope, indicating the underlying two-dimensional lattice, as well as the narrowness of the distributions in (a), (c), and (d), due to the regularity of the periodic lattice.

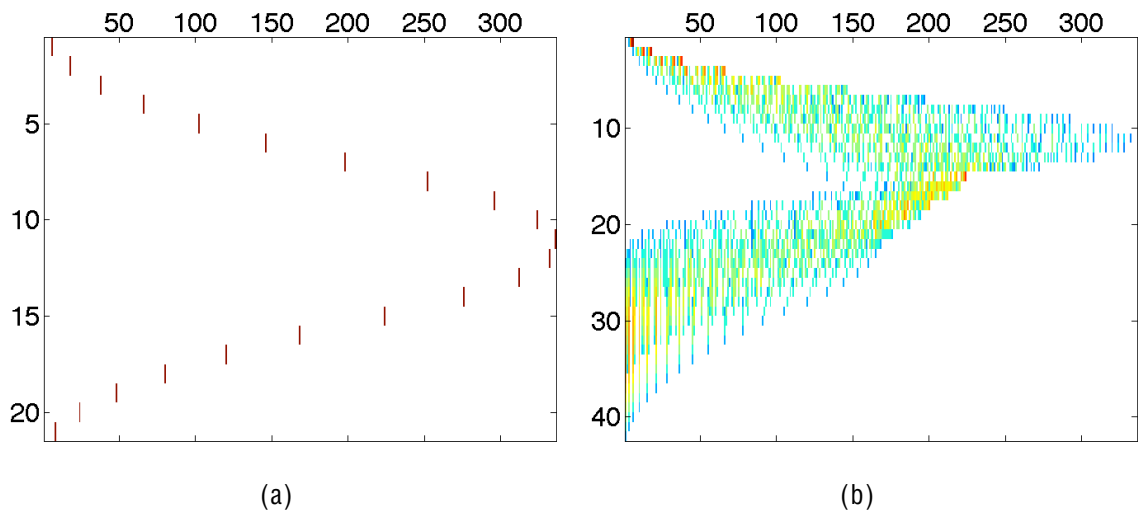


Figure 4.6: Comparison of B for periodic and non-periodic three-dimensional lattices of 15 ð 15 ð 15 nodes. The quadratic growth, present in both matrices, indicates the three dimensions of the underlying networks.

diameter is also

$$D = \frac{1}{N} \sum_{l} \sum_{k} B_{l;k} \quad 1: \tag{4.3.3}$$

Comparing this with the number of rows in B is an easy way to determine whether a graph is directed solely from B, since Eq. (4.3.3) holds only for undirected graphs.

Unfortunately, many quantities that are not directly related to distance currently elude us. For example, how to calculate or even estimate clustering or assortativity remains an open question, since correlations between nodes are mostly lost when creating B. Yet some of these ePects may be indirectly present in B: see Fig. 4.11.

In regards to the graph isomorphism problem, B provides a strong way to *disprove* isomorphism, which appears to be as good or better than known results [86, 87, 88, 89, 90]. A counterexample exists, however, showing that two non-isomorphic graphs can generate the same B. These graphs are the dodecahedron graph [91] and the Desargues graph [92]. See Fig. 4.12 for several planar embeddings of both graphs. Both are cubic distance-regular graphs with $N = 20$ [93] (see also Fig. 4.13) and will

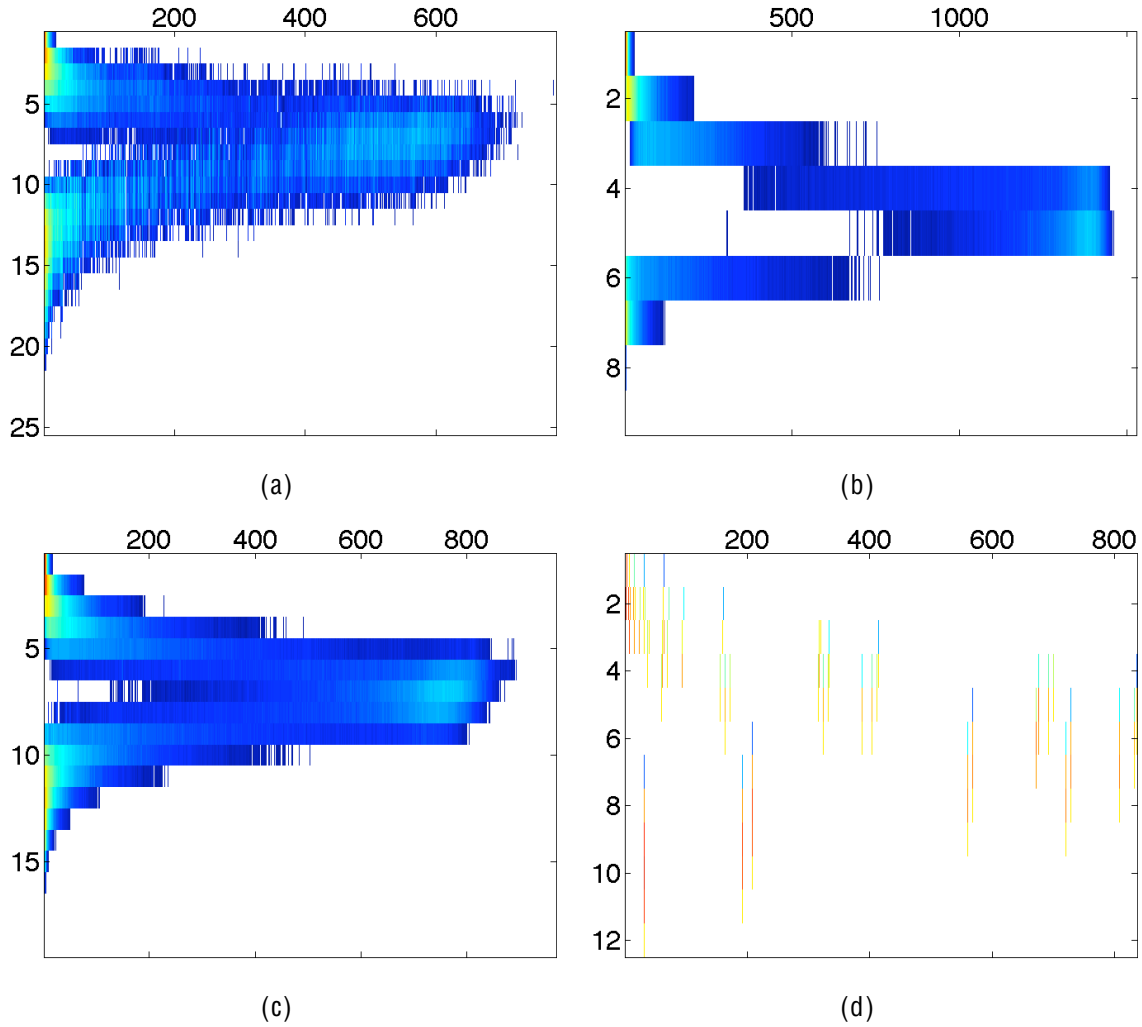


Figure 4.7: Scale-Free models. The average of 100 instances of the (undirected) Krapivsky-Redner (r = 1=2) [83]; Barabasi-Albert (BA) (m = 2) [4]; and Molloy-Reed (MR) (drawn from $P(k)$ ¾ k ^{3}) [18] networks; as well as the (1,3)-Flower at generation 6 [84]; (a){(d), respectively. All have N = 2732, ^{3} 3, but hki varies.

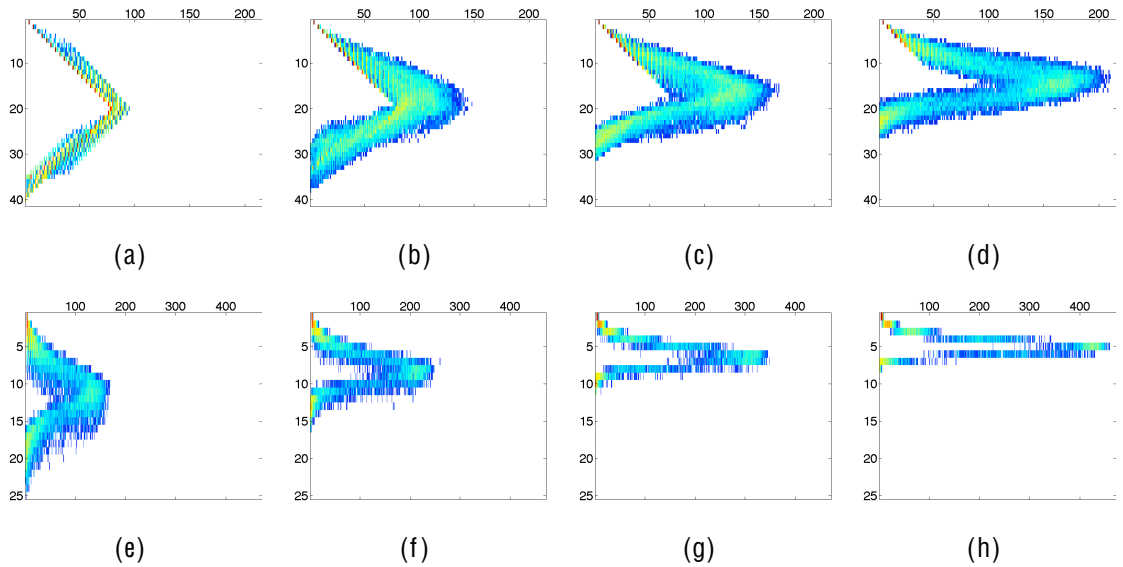


Figure 4.8: Sequential emergence of small-world. (a{d) *B* for a 40 *ð* 40 two-dimensional periodic lattice with 1 random pair of edges permuted, then 4, 5, and 10 more, respectively. The change is drastic when rewiring just 40 out of 3200 edges. The hard edge of slope 4 remains in the þrst shells; it is still possible to identify that this graph is (locally) very lattice-like. (e{h) Newman-Watts-Strogatz graphs [85] with N = 1000; k = 4; and p = 1=20;1=10;1=5, and 2=5, respectively.

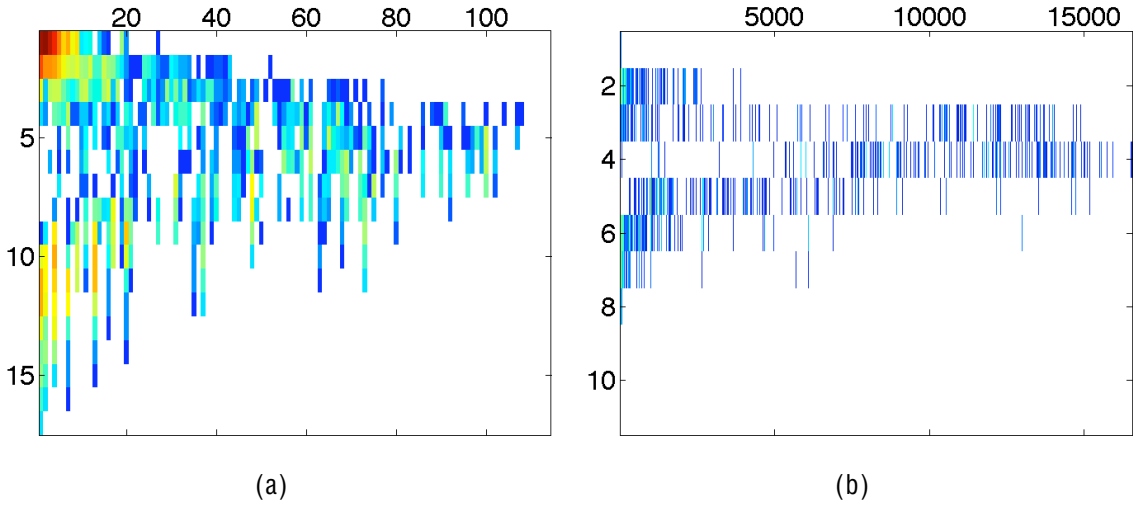


Figure 4.9: Two real-world networks: (a) collaboration network of complex networks researchers [37], and (b) a snapshot of the internet's autonomous systems, taken by Mark Newman on 22 July 2006.

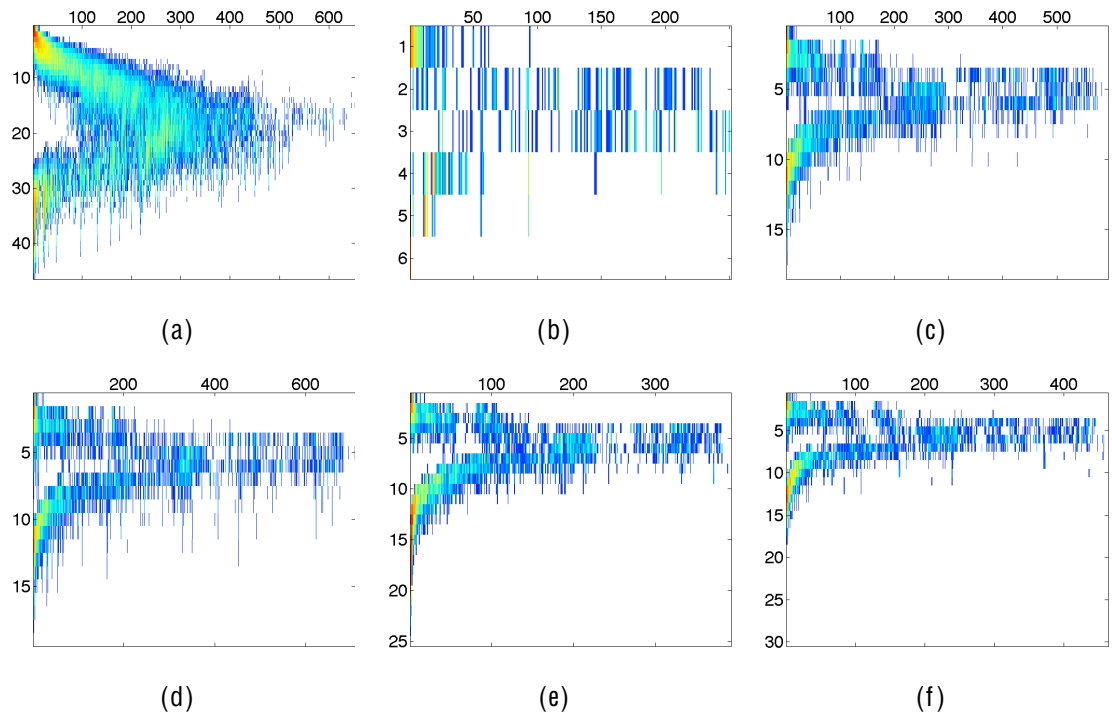


Figure 4.10: Several real world networks. (a) The western states power grid (unweighted) [10], (b) US airlines network [73, 78], and (c){(f) directed metabolic networks for *H. in uenzae*, *R. capsulatus*, *M. jannaschii*, and *C. elegens* [7], respectively. The metabolic networks appear similar to one another yet unlike the power grid and airlines networks.

have exactly one nonzero entry per row in B.[3]

4.4 Network Similarity Testing

To try and tell whether two networks are \alike," or if they come from the same underlying source or generating mechanism,[4] we introduce the following metric[5] between two networks, based on a weighted row-wise comparison of their B matrices.

[3] In principle, this may be exploited to *search* for undiscovered distance-regular graphs by taking a random k-regular graph and rewiring edges along some scheme to minimize the number of nonzero elements per row while respecting node degree. This would likely be cost-prohibitive in practice.

[4] We admit that this deþnition is not rigorous.

[5] It remains an open question whether this is a true topological metric, semi-metric, or neither.

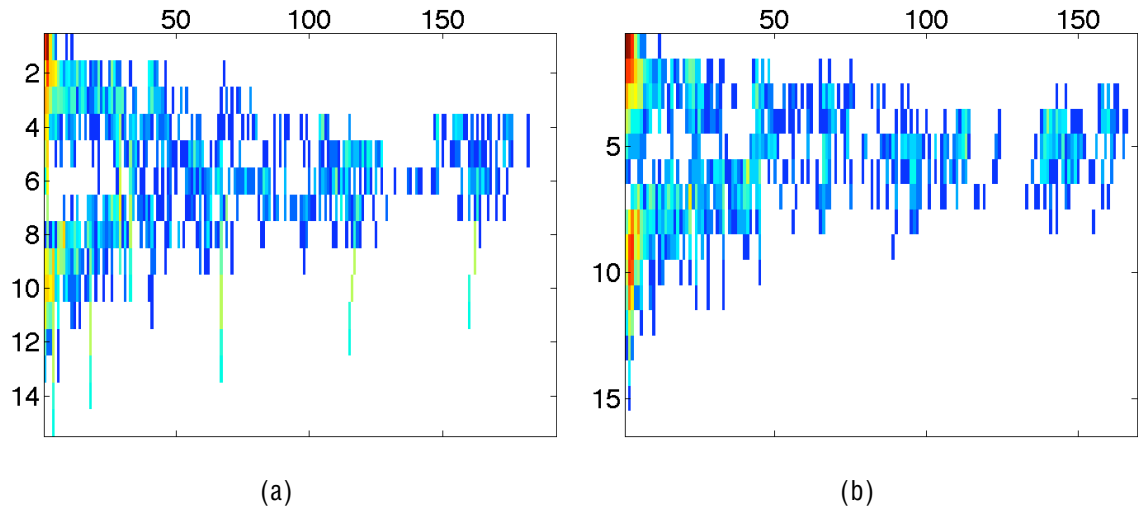


Figure 4.11: (a) The original metabolic network of *M. genitalium* [7] with assortativity $A = 0.174216$ and (b) with $A = 0.000757$ after permuting random edge pairs while preserving the degree distribution. The fine-scale structure in the upper-most shells of (a) is no longer present in (b).

Given two networks G and G', with corresponding matrices B and B', we propose that B and B' encode a great deal of information regarding the generating mechanisms of each network; we expect that B and B' will be similar if G and G' were created using the same generating mechanism, since B captures such a large hierarchy of local and non-local structure. We exploit this in the following way.

The empirical cumulative distribution function (cdf) S_n for n observations x_i is

$$S_n(x) = \frac{1}{n} \sum_{i=1}^{n} [x_i \leq x]; \tag{4.4.1}$$

where $[x_i \leq x] = 1$ if $x_i \leq x$ and 0 otherwise. This is a step-function that increases by $1/n$ at the position of each observation and is constant otherwise. The largest difference between two such distributions is the *test statistic* T for the two-sample Kolmogorov-Smirnov (KS) test:

$$T(X_1; X_2) = \sup_x \left| S(x; X_1) - S(x; X_2) \right|; \tag{4.4.2}$$

where $S(X)$ denotes the empirical cdf for distribution X with the subscript indicating the size of X dropped.

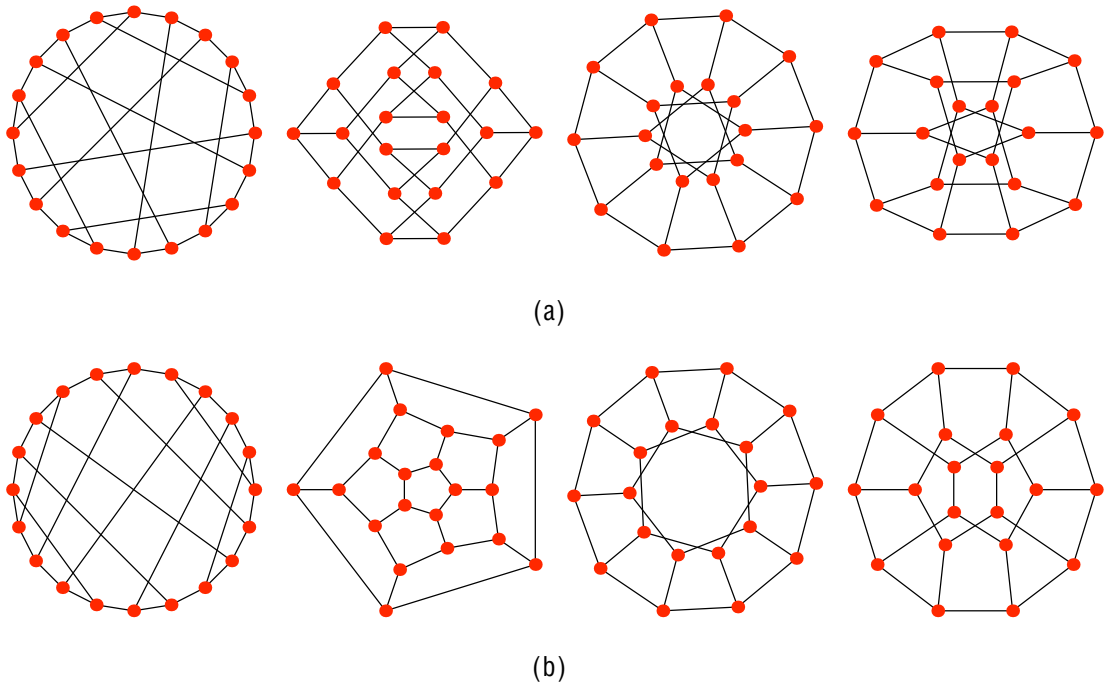


Figure 4.12: Four possible embeddings for both the Desargues graph (a) and the Dodecahedral graph (b) [91, 92]. Both are cubic distance-regular graphs with $N = 20$, $M = 30$, and identical B matrices, from Eq. (4.1.1). The third embedding from left best illustrates the subtle differences between the two.

The two-sample KS test is a useful nonparametric method for comparing two sample sets, due to its sensitivity to changes in both the shape and location of the respective empirical cdfs and the fact that it makes no assumptions about the data's distribution [94, 95]. Motivated by this, we introduce a row-wise statistic K_l, between corresponding pairs of rows l:

$$K_l(B; B') = \max_k \left| C_{l;k} - C'_{l;k} \right|; \tag{4.4.3}$$

where C is the matrix of cumulative distributions of B:

$$C_{l;k} = \sum_{k' \le k} B_{l;k'} = \sum_{k} B_{l;k}. \tag{4.4.4}$$

Thus every row in C is the cdf of the corresponding row's pdf in B.

It has been shown that the lower shells have a greater impact on network properties such as the average path length [96, 64]. This can be considered by weighting shells.

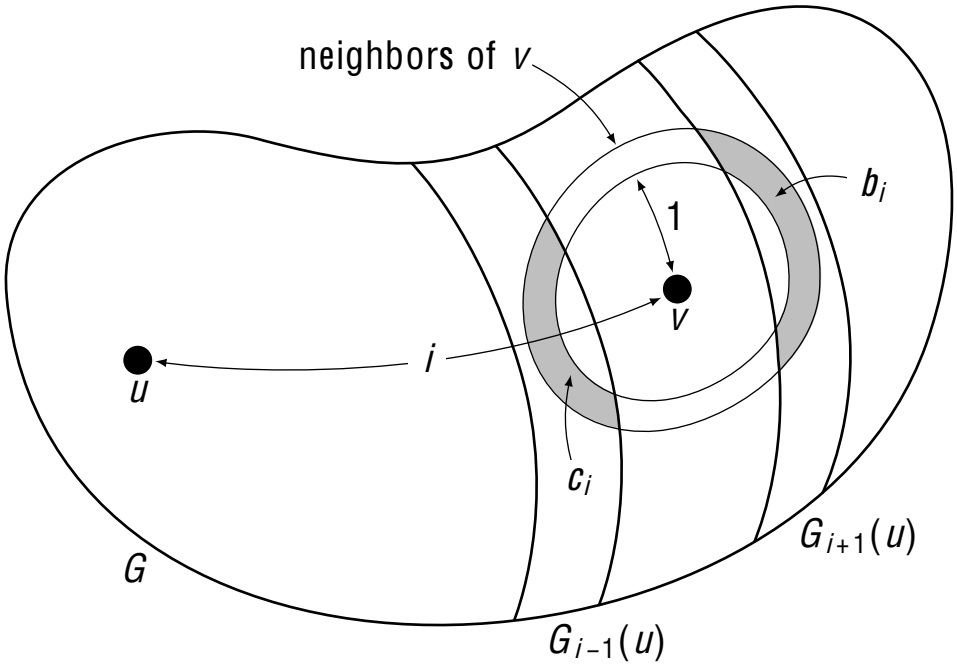


Figure 4.13: A connected graph G is **distance-regular** if it is regular of degree k, and if for *any* two nodes $u; v \; 2 \; G$ at distance $i = d(u; v)$, there are precisely c_i neighbors of v in $G_{i \; 1}(u)$ and b_i neighbors of v in $G_{i+1}(u)$ [93]. Distance-regular graphs possess large amounts of elegant, higher-order symmetries. For example, all of the platonic solids, when represented as graphs, are distance-regular.

One set of weights P, based on shell \mass," could be

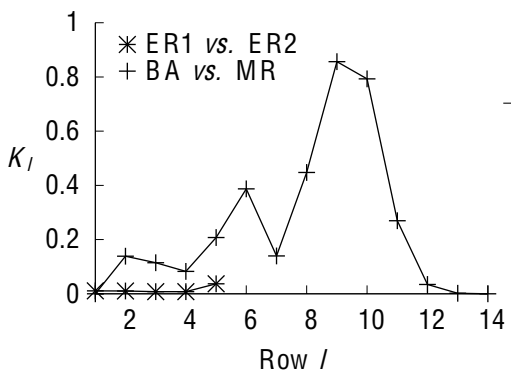
$$P_l = \sum_{k \frac{1}{2} 1} B_{l;k} + \sum_{k \frac{1}{2} 1} B'_{l;k}: \tag{4.4.5}$$

Finally, we choose a scalar \distance" Ð, generated by:

$$Ð(G; G') \quad Ð(B; B') = \left(\sum_{l} P_l K_l \right) = \sum_{l} P_l: \tag{4.4.6}$$

See Fig. 4.14 for examples comparing two Erdős-Renyi graphs against each other as well as a Barabasi-Albert against a Molloy-Reed network.

An open question regarding Ð is whether it can be shown to be a true toplogical metric, semi-metric, or not. The values shown in Fig. 4.14 satisfy the triangle inequality as well as $Ð(x; y) \frac{1}{2} 0$ and $Ð(x; y) = Ð(y; x)$. The last two are both due



	ER1	ER2	BA	MR
ER1	0	0.01223	0.65409	0.61945
ER2	0.01223	0	0.65414	0.61935
BA	0.65409	0.65414	0	0.23157
MR	0.61945	0.61935	0.23157	0

Figure 4.14: (left) Row-wise statistic K_l. Shown are two Erdős-Renyi graphs with $N = 10^4$ and $p = 0.002$; and a Barabasi-Albert (diameter 10) versus a Molloy-Reed network (drawn from $P(k) \propto k^{-3}$, diameter 14), both with $N = 5 \cdot 10^4$. Both the Barabasi-Albert and Molloy-Reed networks have the same degree distribution, so the first few rows are fairly close to one another. Yet differences in, e.g., assortativity, soon become evident: even networks with identical degree distributions may not be similar. (right) Table containing the values of D, given by Eq. 4.4.6, for the four networks shown.

to the absolute value present in K. Discernibility[6] in $D(B; B')$ appears to hold as well, but the dodecahedral and Desargues graphs disprove discernibility in $D(G; G')$, if only because their B-matrices are indiscernable. If metric properties of D can be proven, then this would allow for rigorous comparisons between abstract processes such as graph convergence.[7]

4.5 Conclusions and Open Problems

Equation (4.1.1) encompasses directed graphs and may be generalized to weighted graphs by extending the notion of shells (with shortest paths found by Dijkstra's algorithm [97]). Shells can be defined by introducing a set of weights $W = \{w_1; w_2; \ldots; w_d\}$, defining the shell boundaries. One may also generalize B to *edges* by defining the

[6] $D(x; y) = 0$ iff $x = y$.

[7] For example, comparing a deterministic growth algorithm such as replacing every edge with a 2-path once per time step, versus randomly replacing edges with 2-paths. Can the random graph approach the deterministic graph, as time increases? Despite continued growth in both, D might allow the study of such topological convergence.

distance from a node v_i to an edge $(v_j; v_k)$ as the mean of distances $d(v_i; v_j)$ and $d(v_i; v_k)$.[8] This \edges matrix" has half-integer rows with row 1=2 encoding the degree distribution, $B_{1=2;k} = NP(k)$, and so forth. Both these generalizations will be investigated.

The non-isomorphic dodecahedral and Desargues graphs show that B does not uniquely encode a network but, in practice, the probability of two large non-isomorphic graphs chosen from a statistically-large ensemble having identical B-matrices is vanishingly small. We propose that B is a \very good" answer to graph isomorphism. It is also worth noting that the Desargues and dodecahedral graphs have diþerent edge matrices: we conjecture that graphs are uniquely identiþed with both matrices. In fact, every possible graph of seven nodes or less is uniquely deþned by both. In general, this remains an open question, however, and will be investigated.

In our opinion, the intuition one gains simply by *looking* at these portraits is of great value. Classiþcation and comparison are immediate (Fig. 4.10). Dimensionality and regularity are encoded in the overall slope and row variances (Figures 4.5{4.6), while small-world behavior is displayed in the \aspect ratio" (Fig. 4.8). Even correlation eþects, which one should not expect to be present, may be discernable based on the þne scale structure of the higher rows (Fig. 4.11). Properties such as assortativity were previously impossible to visualize for even moderately sized networks.

The mathematical properties of Ð need to be further explored, and new applications can always be developed. With such a distance metric in hand, it is now possible to apply data clustering methods such as K-means or QT-clustering [98] to *families* of graphs. For example, generating evolutionary (phylogenetic) trees [99, 100] from a collection of metabolic networks. These trees indicate the evolutionary relationships of various organisms and are of great interest to biologists studying the taxonomy and inter-connectedness of Life.

Many physical processes, such as dynamical systems, are captured in stochastic or transition matrices which can be represented as weighted networks, so it is imperative

[8] $B_{l;k}$ is now the number of nodes with k *edges* at distance l.

that we apply our B-matrices to weighted networks. How to best choose the set of shell-deþning weights is the most important open question. These parameters control not only the number of shells present but also their width distribution: should the weights be linearly spaced or logarithmically, for example. Perhaps the number of shells chosen should be based on a binning \rule of thumb" such as the square root of the number of nodes, or perhaps the number of shells should be chosen to equal the diameter of the un-weighted projection of the network?

Chapter 5

Social Networks

One of the most prominent applications of Complex Networks is in the area of modeling human society. Human populations are geographically distributed in highly non-regular, even fractal ways [101, 102]; the distribution of city size, wealth, and other quantities follows a well known power law: *Zipf's law* [103]; and the number of acquaintances people tend to have is also far from uniform [4, 104, 105]. Mechanisms governing the emergence of such structure within populations continue to be analyzed. Understanding population interactions and dynamics has speciþc applications to game theory; epidemiology, including vaccination and disease containment; the spread of information or opinions such as political aŽliations; and improving eŽciency when allocating and disbursing various resources.

We focus our eÞorts on two areas: our newly-proposed Patron-Artwork model, and the study of Kleinberg navigation in the presence of anisotropic underlying lattices.

5.1 The Patron-Artwork Model

The emergence of *fame* appears to be endemic to human societies, yet it is not always fully understood [106, 107, 108, 109]. Simple models, including the Voter [110] and Sznajd [111] model have been introduced to study how opinions and information

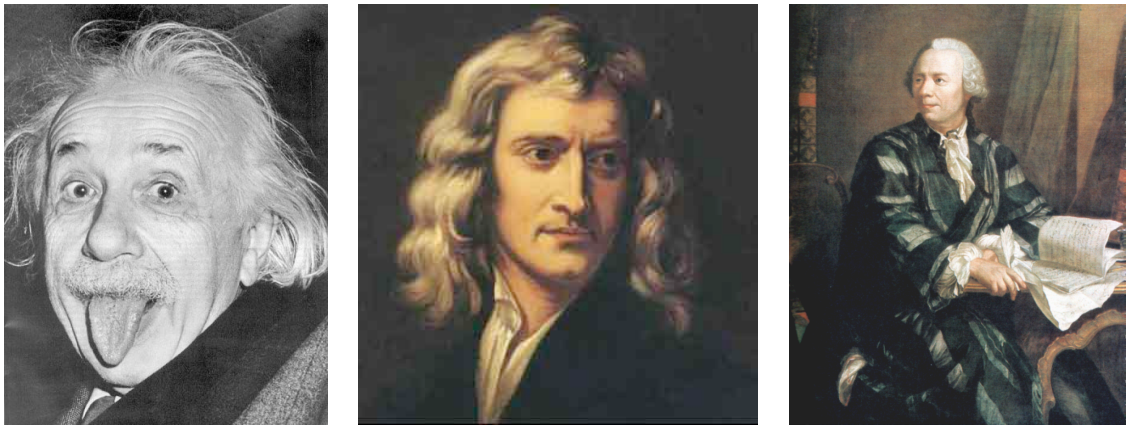


Figure 5.1: Why are these men so famous? Why is Einstein so much more famous than Newton or Euler, besides being so \photogenic"?

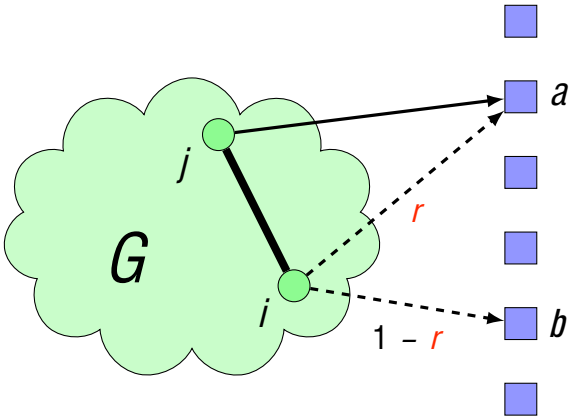


Figure 5.2: Schematic of the Patron-Artwork model. Node *i* is chosen to make a new recommendation. With probability *r*, *i* listens to neighbor *j* and recommends artwork *a*. With probability 1 *r*, *i* instead recommends artwork *b*, chosen uniformly at random. This process is then repeated many times for multiple nodes and the distribution of recommendations per artwork is measured.

ow through a model society, represented as a network. We introduce the **Patron-Artwork** model as a means to study fame directly. This model consists of a dynamics deþned upon a network (of patrons) coupled with and creating a fame distribution on an external population (of artwork).

The model is as follows. Begin with an underlying social network *G* and a line of *A* artworks. At each time step, a randomly chosen node *i 2 G* is allowed to make a new artwork recommendation. With probability *r*, *i* listens to a random neighbor *j* and recommends an artwork that *j* has previously recommended (chosen randomly from *j*'s history of recommendations). With probability 1 *r*, node *i* instead listens

to no one, and recommends an artwork chosen uniformly at random. The fame of an artwork is taken to be the number of recommendations it has received. See Fig. 5.2.

The beauty of this one-parameter model[1] is that it captures important characteristics despite it's simplicity. Every node is \equal" in the sense that they all get the same votes, but authority þgures (hubs in G) have votes that are more important, since nodes are more likely to recommend artwork that the hubs have recommended, meaning artwork lucky enough to be chosen by the hubs garners fame more rapidly. In general, the more fame an artwork has, the more it can gain additional fame, so a \rich-get-richer" mechanism, which is quite popular [112, 113], is naturally built into the model.

5.1.1 The complete graph[2]

We begin our analysis with the simplest type of G, the large complete graph.[3] Here every node is a neighbor of every other node, so the probability of \redirecting" to a particular artwork is entirely proportional to the total number of recommendations that artwork already has (a pure rich-get-richer mechanism). This can be thought of as a mean-þeld approximation, such that the social network ceases to exist in the model, and one can instead envision a large hand doling out packets of fame either proportional to the artwork's current wealth (rich-get-richer process), or uniformly at random (homogenous process), and r governs the relative strengths of these processes.

For the complete graph, we use two diÞerent analytic approaches, as well as numerical simulations, to study both a þnite and an inþnite number of artwork. For the þnite case, at short times, we recover the Pareto law observed for an unbounded number of agents. In later times, the (moving) distribution can be scaled to reveal a phase transition with a Gaussian asymptotic form for $r < \frac{1}{2}$, and a Pareto-like tail

[1] Two parameters, if one considers the dependence on G.

[2] **This special case of the Patron-Artwork model was published in [114].**

[3] This analysis was þrst presented in [114], but the Patron-Artwork model was not directly invoked. The discussion instead focused on \wealth" and \agents," roughly analogous to fame and artwork for this G.

(on the positive side) and a novel stretched exponential decay (on the negative side) for $r > \frac{1}{2}$.

5.1.2 The limit of $A \to 1$

For $A \to 1$, artworks with positive fame form a set of measure zero for any finite time. Thus choosing an artwork uniformly at random is equivalent to the "birth" of a new artwork i with fame $k_i = 1$. We analyze this case with a master equation approach, following the techniques and notation of [83]. The number of artworks with fame $k > 0$ at time t, $N_k(t)$, obeys the master equation:

$$\frac{d}{dt} N_k = (1 - r)\delta_{k,1} + \left[\frac{r}{\sum_{k'} k' N_{k'}}\right] \left[(k-1) N_{k-1} - k N_k\right]. \tag{5.1.1}$$

This limit is characterized by both growth and preferential attachment, hence we expect a power-law distribution of fame.

Since one "fame unit" is disbursed per unit time,

$$\sum_k k N_k = t; \tag{5.1.2}$$

setting the normalization term in (5.1.1). Meanwhile, at each time step a new artwork appears with probability $1 - r$, therefore the total number of artworks with positive fame is $N(t) = (1 - r)t$. The mean fame per artwork is then

$$\langle k \rangle = \frac{1}{1 - r}. \tag{5.1.3}$$

The linear growth in time of artworks and fame suggests a solution of the form

$$N_k(t) = n_k t; \tag{5.1.4}$$

where the n_k are constant. Indeed, upon substituting this *ansatz* into (5.1.1) one obtains a recurrence equation for the n_k, independent of t, whose solution is

$$n_k = \frac{1 - r}{1 + r} \prod_{k'=2}^{k} \frac{r(k'-1)}{1 + rk'}. \tag{5.1.5}$$

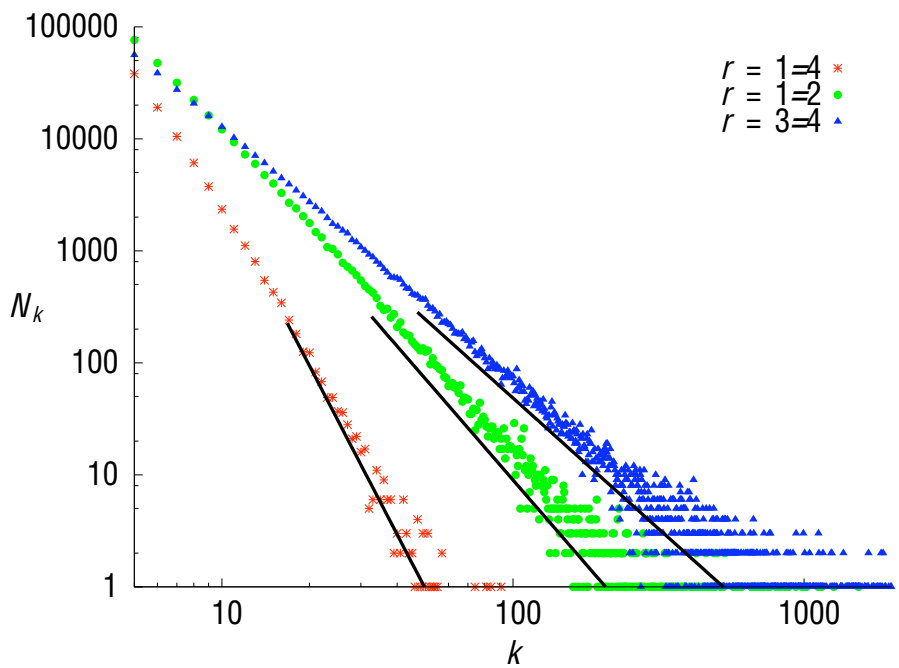


Figure 5.3: Simulations for the case $A \ll 1$, $r = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ (left to right). Each simulation was run until $t = 8 \times 10^6$. Solid lines indicate $\gamma = 1 + 1/r$.

The actual distribution of fame, $P(k)$, is obtained directly from n_k:

$$P(k) = \frac{n_k t}{N(t)} = \frac{n_k}{1 + r}. \tag{5.1.6}$$

The limiting behavior for large k is most easily analyzed by taking the logarithm of (5.1.5), rendering the product into a more manageable sum. Expanding for large k and approximating the sum by an integral we find a power-law tail:

$$P(k) \propto k^{-\gamma}; \qquad \gamma = 1 + \frac{1}{r}. \tag{5.1.7}$$

See Fig. 5.3 for simulations.

It is instructive to obtain this distribution in yet another way [14]. Instead of the master equation we now write the rate equation for the average fame of artwork i:

$$\frac{d}{dt} k_i(t) = \frac{r}{t} k_i(t). \tag{5.1.8}$$

Because $A \ll 1$, the rate at which artwork i is selected by the homogeneous random process is zero, while the rate of selection by the rich-get-richer mechanism is

$k_i = \sum_j k_j$, where $\sum_j k_j$ is simply the total fame, and equals t, see Eq. (5.1.2). Assuming that artwork i got its first unit of fame at time t_i, the initial condition for (5.1.8) is $k_i(t_i) = 1$. Thus,

$$k_i(t_i) = \left(\frac{t}{t_i}\right)^r; \tag{5.1.9}$$

is a monotonically decreasing function of t_i. It follows that the probability that $k_i > k$ is the same as the probability that $t_i < T$, where $k_i(t_i = T) = k$. In other words,

$$(k) \quad \Pr(k_i > k) = \int_k^1 P(k')\,dk' = \Pr(t_i < T): \tag{5.1.10}$$

But $T = tk^{-1/r}$, from (5.1.9), and since the probability that artwork i gets its first unit of fame (by the homogeneous random process) is uniform in time, $\Pr(t_i < T) = T/t = k^{-1/r}$. We then have

$$P(k) = \frac{d}{dk}(k) = \frac{1}{r}k^{-1-1/r}; \qquad k \geq 1; \tag{5.1.11}$$

i.e., a Pareto distribution with the same power-law tail as in (5.1.7). Note that this distribution is properly normalized (taking k to be a continuous variable) and that its first moment agrees with (5.1.3).

The limit of $r \to 0$

When the number of artworks A is finite, the N_k obey the normalization condition

$$\sum_{k=0}^{\infty} N_k(t) = A; \tag{5.1.12}$$

where now we include in the counting artwork with zero fame ($k = 0$), and the distribution of fame is $P(k;t) = N_k(t)/A$. The mean fame per artwork is no longer constant but increases linearly with time:

$$\langle k \rangle = \frac{t}{A}: \tag{5.1.13}$$

Consider the limit of $r \to 0$, where fame is disbursed only by the homogeneous random process. The corresponding master equation is

$$\frac{d}{dt}N_k = \frac{1}{A}\left(N_{k-1} - N_k\right); \tag{5.1.14}$$

with initial and boundary conditions $N_k(0) = A\check{Z}_{k;0}$ and $N_{-1}(t) = 0$. This is a simple Poisson process, as confirmed by the solution of (5.1.14):

$$N_k(t) = A\frac{(t/A)^k}{k!}\exp^{-t/A}. \tag{5.1.15}$$

For $t \gg A$ we apply the Sterling approximation to obtain the distribution

$$P(k;t) = \frac{1}{\sqrt{2\pi(t/A)}}\exp\left[-\frac{1}{2}A\frac{(k-t/A)^2}{t}\right]. \tag{5.1.16}$$

Thus, $P(k;t)$ has a power-law tail in the limit $A \to \infty$ (5.1.2), but is Gaussian when A is finite and $r \to 0$.

5.1.3 Finite A and r

Our analysis proceeds along the same two approaches used for the $A \to \infty$ case. We will show that each works for only certain values of r, and that one approach has problems which are not well understood.

Master equation approach

In the general case of $A < \infty$ and $r > 0$ the master equation for the process is

$$\frac{d}{dt}N_k = \left(\frac{1-r}{A}\right)\left[N_{k-1} - N_k\right] + \frac{r}{t}\left[(k-1)N_{k-1} - kN_k\right]. \tag{5.1.17}$$

The system is then simultaneously pulled toward the two different limiting behaviors analyzed in Sec. 5.1.2. We will show that for $r > \frac{1}{2}$ the rich-get-richer mechanism dominates the process and the fame distribution develops a power-law tail (as for the limit of $A \to \infty$), while for $r < \frac{1}{2}$ the homogeneous random process dominates and the fame distribution tends to a Gaussian (as for $r \to 0$). Because A is finite, $\langle k\rangle = t/A$ increases linearly with time. The width of the distribution of k around the average grows like t^{β}, where the scaling exponent $\beta = r$ for $r > \frac{1}{2}$ and $\beta = \frac{1}{2}$ for $r < \frac{1}{2}$. At the transition point, $r = \frac{1}{2}$, the width scales as $\sqrt{t\ln t}$.

To see these results, begin by approximating the discrete distribution $N_k(t)$ by its continuous counterpart, $P(k;t)$. Expanding to first-order, Eq. (5.1.17) now reads

$$\frac{\partial}{\partial t}P(k;t) = \frac{1}{A}\frac{r}{\partial k}\frac{\partial}{\partial k}P - \frac{r}{t}\frac{\partial}{\partial k}(kP); \tag{5.1.18}$$

and the method of characteristics yields the scaling solution

$$P(k;t) = t^{-\beta} f\left(\frac{k - t/A}{t^{\beta}}\right); \qquad \beta = r: \tag{5.1.19}$$

This, however, cannot be true for all values of r, as it disagrees with the distribution (5.1.16) found for $r = 0$, where the scaling exponent is $\beta = \frac{1}{2}$ instead of $\beta = r = 0$. The reason for this discrepancy is that, in this case, the Kramers-Moyal expansion [115] of (5.1.17) must be carried out beyond the first order. Indeed, upon substituting the scaling form (5.1.19) into the master equation (with unspecified β), and carrying out the expansion to second-order, we find

$$(\beta - r)t^{2\beta} f(x) + (\beta - r)t^{2\beta} x f'(x) + \frac{1}{2A} t f''(x) = 0; \tag{5.1.20}$$

where prime denotes differentiation with respect to $x = (k - t/A)/t^{\beta}$, and we have omitted terms proportional to t^{β} (these are negligible compared to $t^{2\beta}$, as $t \to \infty$). If $\beta > \frac{1}{2}$, the term proportional to t can be neglected in the long-time limit, and (5.1.20) is satisfied provided that $\beta = r$. Thus, the scaling form (5.1.19) is valid only for $r > \frac{1}{2}$. For $r < \frac{1}{2}$, however, the second-order term in (5.1.20) may not be ignored. The only non-trivial way to cancel out the time dependence is then to have $t^{2\beta} = t$. Thus, for $r < \frac{1}{2}$ the scaling exponent is $\beta = \frac{1}{2}$. At the transition point, $r = \frac{1}{2}$, there is no way to get rid of the time dependence in (5.1.20) with the scaling form (5.1.19). Taking a cue from other phase transitions we guess a scaling form with logarithmic dependence:

$$P(k;t) = \frac{1}{(t\ln t)^{\beta}} f\left(\frac{k - t/A}{(t\ln t)^{\beta}}\right); \qquad r = \frac{1}{2}: \tag{5.1.21}$$

On expanding the master equation with this scaling form the leading behavior in time cancels out, provided that the scaling exponent is $\beta = \frac{1}{2}$. The next largest terms (smaller by a $\ln t$ factor), yield the equation

$$f(x) + x f'(x) + \frac{1}{A} f''(x) = 0; \qquad r = \frac{1}{2}; \tag{5.1.22}$$

where now $x = (k - t/A)/\sqrt{t \ln t}$. In all three cases (for r), expanding to third- or higher-order yields additional subdominant terms. From the largest subdominant term one can deduce how fast the system reaches the scaling regime: the transient dies off as $t^{-(2r-1)}$ for $r > \frac{1}{2}$, as $t^{-1/2}$ for $r < \frac{1}{2}$, and as $(\ln t)^{-1}$ for $r = \frac{1}{2}$. Thus at the transition point, $r = \frac{1}{2}$, there occurs a *critical slowing down* as the system creeps into the eventual scaling regime logarithmically slow.

For $r < \frac{1}{2}$ we can use (5.1.20) to find $f(x)$ and show that the limiting form of the fame distribution is Gaussian:

$$P(k;t) \to \sqrt{\frac{A(1-2r)}{2\pi t}} \exp\left[-\frac{1}{2}A(1-2r)\frac{(k-t/A)^2}{t}\right]; \qquad r < \frac{1}{2}; \tag{5.1.23}$$

as $t \to \infty$. The divergence of the width of this distribution as $r \to \frac{1}{2}$ is reconciled with the fact that at the limit $r = \frac{1}{2}$ the scaling parameter picks up a (diverging) logarithmic component. The scaling function at the transition is still Gaussian, as can be deduced from (5.1.22):

$$P(k;t) \to \sqrt{\frac{A}{2\pi t \ln t}} \exp\left[-\frac{1}{2}A\frac{(k-t/A)^2}{t\ln t}\right]; \qquad r = \frac{1}{2}: \tag{5.1.24}$$

For $r > \frac{1}{2}$, Eq. (5.1.20) yields a tautology and one is unable to determine $f(x)$. It is possible, nevertheless, to infer the limiting behavior:

$$f(x) \propto \begin{cases} x^{-1-1/r} & x \to \infty; \\ \exp\left[-(1-r)(A^r|x|)^{\frac{1}{1-r}}\right] & x \to -\infty; \end{cases} \qquad r > \frac{1}{2}: \tag{5.1.25}$$

The limit for $x \to \infty$ follows from comparing the distribution $P(k;t)$ for the case of $A \to \infty$ with $f(x)|_{A\to\infty} = f(kt^{-r})$. For $x \to -\infty$, we observe that the density of artworks with zero fame decays as $N_0 \propto \exp[-(1-r)t/A]$, see Eq. (5.1.28), and we compare to $f(x)|_{k=0} = f(-t^{1-r}/A)$, leading to the second line of (5.1.25). An alternative derivation is presented next, using the rate equation approach.

Rate equation approach

The rate equation for the fame of artwork i, in the general case, is

$$\frac{d}{dt}k_i(t) = \frac{1-r}{A} + \frac{r}{t}k_i(t); \tag{5.1.26}$$

with initial condition $k_i(t_i) = 1$. The solution,

$$k_i(t_i) = \quad 1 \quad \frac{t_i}{A} \quad \frac{t}{t_i} \quad {}^{r} + \frac{t}{A}; \tag{5.1.27}$$

is monotonically decreasing in t_i.

The probability $\quad (t)$ that an artwork still has zero fame at time t satisþes the equation

$$\frac{d}{dt} \quad (t) = \quad \frac{1 \quad r}{A} \quad (t); \tag{5.1.28}$$

so $\quad (t) = \exp[\quad (1 \quad r)t=A]$. It follows that the probability that artwork i has been introduced (gets its þrst unit of fame) by time T, given that it has been introduced by time t, is

$$(T) = \frac{1 \quad e^{\frac{1}{A} \quad r}T}{1 \quad e^{\frac{1}{A} \quad r}t}: \tag{5.1.29}$$

Note that this has the limit $T=t$, as $A \; ! \; 1$, that we used in Sec. 5.1.2.

Finally, $P(k;t) = \quad @ \; (T)=@k$, where $T(k)$ is the solution to $k_i(T) = k$. Since Eq. (5.1.27) cannot be inverted analytically (other than for special values of r), we express P in parametric form: $P(k(T);t) = \quad @ \; (T)=@k = \quad (dk_i=dt_i j_{t_i=T}) \; {}^{1}@ \; (T)=@T$, and $k(T)$ is obtained by putting $t_i = T$ in (5.1.27). The (scaled) fame distribution in parametric form is then

$$x(T) = \quad 1 \quad \frac{T}{A} \quad T \quad {}^{r}; \qquad f(T) = \frac{(1 \quad r)T^{1+r}}{Ar + (1 \quad r)T} e^{\frac{1}{A} \quad r}T; \tag{5.1.30}$$

where we have used the scaled expressions $x = (k \quad t=A)=t^r$ and $f = t^r P$, taking the limit of $t \; ! \; 1$ at the end (the fact that the limit exists and is þnite conþrms this scaling).

It is now easy to verify the asymptotic behavior (5.1.25). The limit $x \; ! \; 1$ corresponds to $T \; ! \; 0$. In this limit, the second equation of (5.1.30) gives $f \; ¾ \; T^{1+r}$. But since $T \; ¾ \; x \; {}^{1=r}$, from the þrst equation, we conclude that $f \; ¾ \; x \; {}^{1} \; {}^{1=r}$. The limit $x \; ! \; 1$ corresponds to $T \; ! \; 1$. In this limit, the second equation of (5.1.30) gives $f \; ¾ \exp[\quad (1 \quad r)T=A]$, while from the þrst equation $x \; ¾ \quad (1=A)T^{1} \; {}^{r}$. We conclude that $f \; ¾ \exp[\quad (1 \quad r)(A^r jxj)^{1=(1 \quad r)}]$.

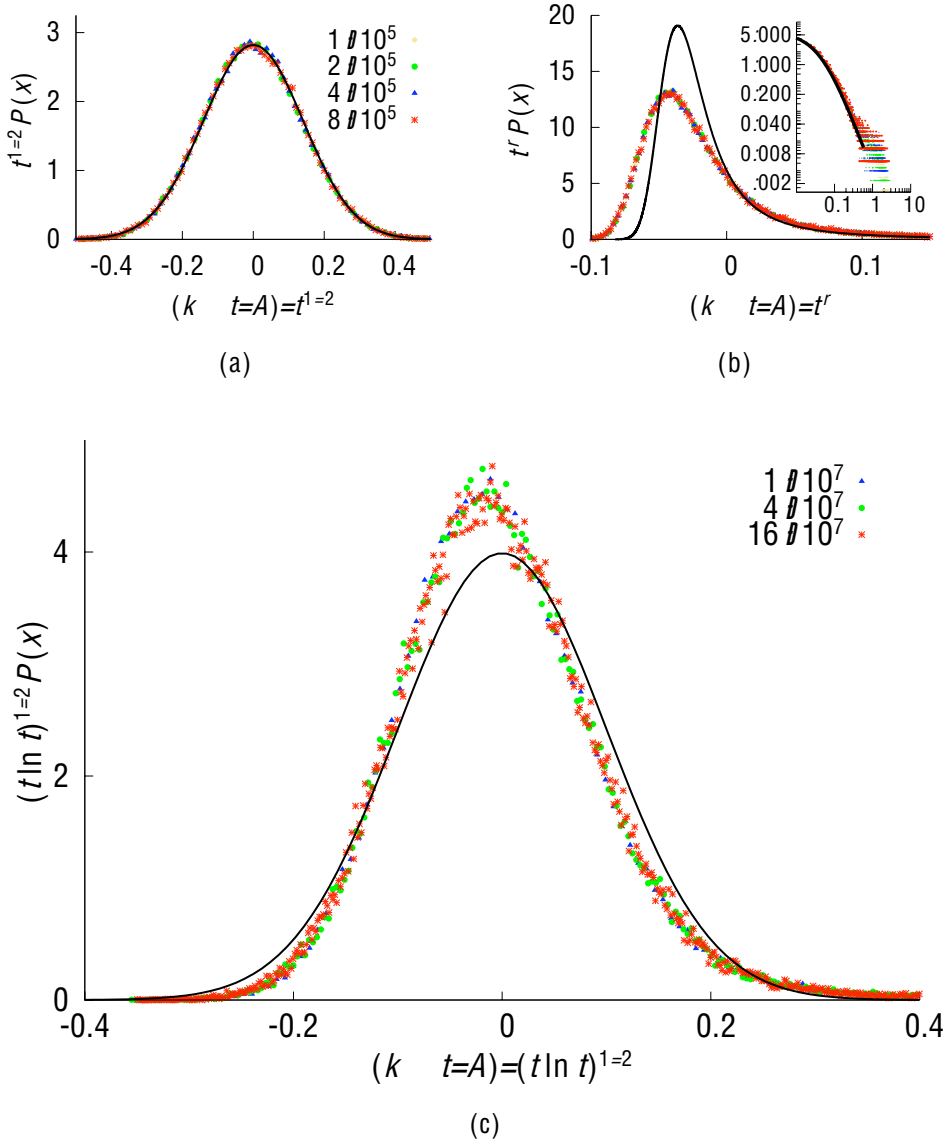


Figure 5.4: Scaling of the fame distribution in each of the two phases at $r = \frac{1}{4}$ (a), $r = \frac{3}{4}$ (b) and at the transition point $r = \frac{1}{2}$ (c). The inset in (b) shows the right-hand tail with logarithmic axes. Convergence to the scaling form is rapid for $r = \frac{1}{4}$ and $r = \frac{3}{4}$ but logarithmically slow for $r = \frac{1}{2}$ | note that in the latter case the data (over exponentially increasing times) is slowly creeping toward the Gaussian limit of (5.1.24) (solid line). The theoretical limit of (5.1.23) (solid line) þts the case of $r = \frac{1}{4}$ perfectly, but the prediction (5.1.30) from the rate equation approach (solid line) þts the case of $r = \frac{3}{4}$ only qualitatively (besides agreeing with the overall scaling).

Clearly, the foregoing rate equation method does not apply to $0 \quad r \quad \frac{1}{2}$, for it fails to reproduce the appropriate scaling forms in this range. Thus the rate equation approach is viable only when the second-order in the Kramers-Moyal expansion of the corresponding master equation may be neglected. In Fig. 5.4 we show numerical simulations for r below, above, and at the transition point. The results conþrm the scaling forms found analytically above. For $r < \frac{1}{2}$ convergence to the Gaussian pdf is relatively fast, while the critical slowing down at the transition point, $r = \frac{1}{2}$, prevents us from attaining the analytical limit (5.1.24) in practice. For $r > \frac{1}{2}$ convergence to the scaling pdf is again quick, however the explicit form predicted by the rate equation approach is correct only qualitatively: we ascribe this to the fact that the second-order is implicitly missing in this approach.

5.1.4 Future work

The case of G as a large complete graph is not a realistic social network; it lacks authority þgures, for example. A natural next choice for G is the complete bipartite graph $K_{H;L}$ consisting of two groups of nodes, one of size H (call them hubs) and the other of size L (call them leaves). Every hub node is connected to every leaf node, and vice versa, while no hubs are connected to other hubs, and similarly for leaves. The limiting case $K_{1;L}$ corresponds to the star graph. The star graph represents a society where there is a single authority þgure that everyone listens to, and no besides the authority þgure listens to anyone else.

As before, one recommendation is made per time step, so the total number of recommendations is $M = t$. Deþne the total number of recommendations made from hubs at time t as $M_H(t)$ and leaves as $M_L(t)$. Then $M = M_H + M_L = t$. A hub (leaf) is selected to make a new recommendation with probability $H/(H + L)$ ($L/(H + L)$), irrespective of r. Therefore

$$M_H(t) = \frac{H}{H + L} t; \tag{5.1.31}$$

$$M_L(t) = \frac{L}{H + L} t: \tag{5.1.32}$$

Define $N(k_1; k_2)$ as the number of artworks with k_1 recommendations from hubs and k_2 recommendations from leaves. This quantity is governed by the master equation:

$$\begin{aligned}\frac{d}{dt}N(k_1;k_2) = \frac{1-r}{A(H+L)}\Big[& H\,N(k_1-1;k_2) - H\,N(k_1;k_2) \\ & + L\,N(k_1;k_2-1) - L\,N(k_1;k_2)\Big] \\ & + \frac{r}{t}\Big[\frac{H}{L}k_2 N(k_1-1;k_2) - \frac{H}{L}k_2 N(k_1;k_2) \\ & + \frac{L}{H}k_1 N(k_1;k_2-1) - \frac{L}{H}k_1 N(k_1;k_2)\Big];\end{aligned} \tag{5.1.33}$$

with the total fame distribution given by

$$N(k) = \sum_{k_2=0}^{k} N(k-k_2;k_2). \tag{5.1.34}$$

For a large star graph ($H = 1$, $L \gg H$) or a *nearly* star graph ($L \gg H$), Eq. 5.1.33 can be greatly simplified with approximations $H/(H+L) \approx 1/L$, $L/(H+L) \approx 1$ and $L/H \approx L$, reducing the master equation to

$$\frac{d}{dt}N(k_1;k_2) = \frac{1-r}{A} + \frac{r}{t}L k_1\Big[N(k_1;k_2-1) - N(k_1;k_2)\Big] \tag{5.1.35}$$

where $O(1/L)$ terms have been dropped. Note that no dependence on k_1 remains; it is a parameter. The remaining k_2 can be dealt with using a generating function:

$$G_{k_1}(z) \equiv \sum_{k_2=0}^{\infty} N(k_1;k_2) z^{k_2}; \tag{5.1.36}$$

giving

$$\frac{d}{dt}G_{k_1}(z) = \frac{1-r}{A} + \frac{r}{t}L k_1 (z-1)\, G_{k_1}(z). \tag{5.1.37}$$

Separation of variables gives

$$\begin{aligned} G_{k_1}(z) &= C t^{rLk_1(z-1)} e^{\frac{1-r}{A}(z-1)t} \\ &= C t^{-rLk_1} e^{-\frac{1-r}{A}t} \exp\Big[\frac{1-r}{A}t + rLk_1 \ln t\, z\Big]. \end{aligned} \tag{5.1.38}$$

Therefore

$$N(k_1;k_2) = \frac{C}{k_2!} t^{\ rLk_1} e^{-\frac{1-r}{A}t} \left(\frac{1-r}{A}t + rLk_1 \ln t\right)^{k_2}; \tag{5.1.39}$$

with C determined from normalization, $\sum_{k_1;k_2} N(k_1;k_2) = A$. Substituting (5.1.39) into (5.1.35) confirms that this is a solution. For the star graph, one need not resort to Eq. (5.1.34) since any individual artwork can have at most a single connection to the hub. Therefore, $N(k) = N(0;k) + N(1;k-1) \approx N(0;k)$. Thus we expect a Poisson distribution with mean $(1-r)t/A$,

$$N(k) \approx \frac{1}{k!} e^{-\frac{1-r}{A}t} \left(\frac{1-r}{A}t\right)^{k}. \tag{5.1.40}$$

The validity of these star graph approximations remains an open question. Simulations will be used to confirm these results.

The next step up the "ladder of heterogeneity" would be a G with an arbitrary, uncorrelated degree distribution.[4] In principle, a master equation can be written governing $N(k_1;k_2;\ldots;k_K)$, the number of artworks with k_1 recommendations from degree 1 nodes, k_2 recommendations from degree 2 nodes, ..., and k_K recommendations from the highest degree K nodes. The total fame would then be $N(k) = \sum_{k_1;k_2;\ldots;k_K} N(k_1;k_2;\ldots;k_K)[k_1 + k_2 + \cdots + k_K = k]$. The question of solvability for such a master equation remains open.

Further generalizations may prove fruitful as well. For example, when G is a complete bipartite graph, one can use a different redirection probability for each of the two groups. Meaning that hubs may be more or less likely to listen to leaves than leaves are to listen to hubs. While this increases the number of parameters in the model, it may lead to improved realism without something as complicated as the aforementioned uncorrelated degree distribution master equation. Using directed or weighted patron networks may also be fruitful.

[4] Or perhaps a binomial or Erdős-Renyi graph.

5.1.5 Summary and discussion

In summary, we have introduced the Patron-Artwork model where A works of art accrue \fame" by a simple one-parameter dynamic on a social (patron) network. Our current analysis has focused on the case where G is an asymptotically large complete graph. We have shown that in the early time regime, or, equivalently, when $A \gg 1$ there results a Pareto distribution for fame k: $P(k) \propto k^{-\nu}$, with $\nu = 1 + 1/r$. In the long time asymptotic limit, the system is attracted to one of two opposite poles, and there is a kinetic phase transition as a function of the parameter r. If $r < \frac{1}{2}$, the distribution tends to a Gaussian of width $t/[(1-2r)A]$. If $r > \frac{1}{2}$, the distribution keeps its power-law tail $\propto k^{-1-1/r}$ for large k.

In all cases the fame distribution tends to an asymptotic scaling form as a function of $x = (k - \langle k \rangle)/w(t)$, where $\langle k \rangle = t/A$ is the average fame amassed by an artwork up to time t, and $w(t) = t^{\beta}$ is a measure of the width of the distribution. The exponent β undergoes a phase transition: $\beta = \frac{1}{2}$ for $r < \frac{1}{2}$, and $\beta = r$ for $r > \frac{1}{2}$. At the transition point, $r = \frac{1}{2}$, there appear logarithmic corrections: $w(t) = (t \ln t)^{1/2}$.

The scaling form of the fame distribution $f(x) = t^{r} P$ in the regime $r > \frac{1}{2}$ is characterized by two more exponents (in addition to the width exponent $\beta = r$): $f(x) \propto x^{-1-1/r}$ for $x \gg 1$, and $f(x)$ decays as a stretched-exponential, with power $1/(1-r)$, as $x \ll -1$. Finally, the approach to the eventual scaling form $\propto t^{-z}$ is characterized by a fourth exponent: $z = \frac{1}{2}$ for $r < \frac{1}{2}$, and $z = 2r - 1$ for $r > \frac{1}{2}$. At the transition point convergence to the scaling form proceeds exceedingly slow, $\propto 1/\ln t$, in a fashion reminiscent of critical slowing down in equilibrium phase transitions.

Several applications come to mind. For example, complex networks could be grown according to this model where the nodes are þxed at the outset (corresponding to the A artwork) and links are connected to the nodes by a proper mix of homogeneous selection and preferential attachment. For $r > \frac{1}{2}$ one could thus create scale-free nets with a þxed degree distribution exponent and a þxed number of nodes, and with a tunable average connectivity $\langle k \rangle = t/A$ that grows linearly with time. Wealth distributions with a stretched-exponential decay on one side and a power-law decay

on the other, such as we þnd for $r > \frac{1}{2}$, are regularly observed in various economic settings [116, 117].

An intriguing þnding concerns the method of rate equations that is often used to obtain the degree distribution of complex networks [14, 118, 119, 120, 121]. Our analysis suggests that this method is only valid when the second-order terms in the Kramers-Moyal expansion of the master equation for the system may be safely neglected. Even then the method yields results that scale correctly but that are otherwise only qualitatively correct, at least in our case. Perhaps the most important open problem is to establish the range of validity of the rate equation approach more rigorously, and to þnd ways to extend it to the cases where it fails.

5.2 Kleinberg Navigation

The small-world phenomenon, one of the most intriguing properties of human society, was touched upon in Ch. 1. This describes the fact that unrelated people in a society, who are a very large geographic distance apart from one another, tend to be connected by surprisingly short chains of acquaintances. This phenomenon was hypothesized in 1929 by Hungarian author Frigyes Karinthy [122, 123] and was þrst observed experimentally in the 1960's with sociologist Stanley Milgram's seminal experiment wherein randomly chosen people were selected to mail a letter to an unknown target person, but were only allowed to send the letter to a friend, who would pass the letter along to another friend, etc., until the target was reached. It ended up taking surprisingly few people to send such letters. Hence the turn of phrase `six degrees of separation,' popularized by Karinthy, was quite accurate. Understanding this phenomenon is an important sociological problem.

To study the underlying mechanism that led to Milgram's results, computer scientist Jon Kleinberg modeled a society as follows [124, 125]. Begin with a large, regular

lattice.[5] Each node is connected to its nearest lattice neighbors and to a single random node a large distance away. The probability of nodes i and j being connected by such a long range connection is

$$P_{ij}(\Phi) = r_{ij}^{\Phi} = \sum_{k \neq i} r_{ik}^{\Phi}; \tag{5.2.1}$$

where r_{ij} is the euclidian distance between nodes i and j and the sum runs over all nodes in the network except i. Physically, the local lattice connections represent associations with immediate neighbors, fellow townspeople, etc., while long-range contacts model friends or relatives in another city or country, for example.

The following algorithm, proposed by Kleinberg, models the message-passing experiment of Milgram on this network [124]. Choose a starting node s and a target node t a distance L apart.[6] The current message-holding node, starting with node s, passes the message along to whichever of its contacts is closest to t, until the message reaches t. We wish to know the number of steps T required to reach the target and what value of Φ gives the lowest T, corresponding to optimally e Ž cient transport. Of great importance is the fact that each node has no information beyond the locations of its contacts and node t; the algorithm is *greedy* in that it seeks to locally minimize the distance to t at each step without regard to the possibility that another node's contact may be closer to t than the current node's contacts.

We begin by reproducing the proof in [126] that navigation is fastest when long-range connections are chosen from Eq. (5.2.1) with $\Phi = d$, where d is the dimension of the underlying lattice.[7]

Since the number of nodes at a distance r scales like r^{d-1} and each node contributes

[5] See [126] for a generalization to underlying fractal lattices.

[6] Existing work has instead chosen s and t at random from within a lattice of size $L \times L$. The average (lattice) distance between a random s and t is $\frac{2}{3} L \approx L$ anyway, so we choose to eliminate this additional randomness.

[7] This includes lattices of non-integer dimension, with the only requirements being that there are no (finite) areas where the message can become trapped and must backtrack, so called "overhangs," and that the lattice distance scales like the euclidean distance.

r^{-p} to the sum, then the normalization term in Eq. (5.2.1) scales like

$$\sum_{k \neq i} r_{ik}^{-p} \propto \int_1^L r^{-p} r^{d-1}\, dr = \begin{cases} \frac{1}{p-d}\left(L^{d-p} - 1\right) \propto (p-d)^{-1}; & p > d; \\ \ln L; & p = d; \\ \frac{L^{d-p}-1}{d-p} \propto L^{d-p}; & p < d. \end{cases} \tag{5.2.2}$$

To show that navigation is most efficient when $p = d$, we proceed by finding the expected speed for $p = d$ and show that it grows more slowly than the best-case expected speed for $p \neq d$, as $L \to \infty$.

For the case where $p = d$, surround the target node with concentric shells of exponentially increasing radii $e^{m-1} < r < e^m$, $m = 1, 2, \ldots, M$.[8] The probability that a message holder in shell m has its long-range connection be to a node in shell $m-1$ scales like, from Eq. (5.2.2),

$$\propto \frac{1}{\ln L} \int_{e^{m-1}}^{e^m} r^{-d} r^{d-1}\, dr = \frac{1}{\ln L}. \tag{5.2.3}$$

If the message holder does not have a long-range connection to the next closest shell, then the message will not reach the next shell within one step (with overwhelming probability). Therefore, the probability that the message will take more than x steps to reach the next shell is $\Phi(x) = (1 - 1/\ln L)^x$, and the expected number of steps spent in the current shell is[9]

$$\langle x \rangle = \int_0^\infty \Phi(x)\, dx = \frac{1}{\ln\left(1 - \frac{1}{\ln L}\right)} \propto \ln L. \tag{5.2.4}$$

The largest shell is of size $e^{M-3} \propto L$, so the number of shells separating the source and target nodes is on the order of $M = \ln L$. It's expected to require $\ln L$ steps to traverse each shell, therefore the total number of steps to reach the target is $\propto \ln^2 L$

[8] These are shells in the Euclidian plane, not the shells in chemical space that we have previously focused on.

[9] This is best seen by working backwards. Let $p(x)\,dx$ be the probability to reach the next shell within x and $x + dx$ steps. Then $\Phi(x) = \int_x^\infty p(x)\, dx$, and $\Phi'(x) = p(\infty) - p(x) = -p(x)$. Integration by parts gives

$$\int_0^\infty \Phi(x)\, dx = x\,\Phi(x)\Big|_0^\infty - \int_0^\infty x\,\Phi'(x)\, dx = \int_0^\infty x p(x)\, dx = \langle x \rangle;$$

since $\Phi(\infty) = 0$, by definition.

When $0 \quad \alpha < d$, begin by surrounding the target node with a ball of radius $l = L^{\check{Z}}$, $0 < \check{Z} < 1$. The probability that a randomly chosen node i has its long-range connection be to a node j within this ball is, from Eq. (5.2.2), ¾ $r_{ij}^{\alpha}/L^{d \quad \alpha} \quad 1/L^{d \quad \alpha}$. Then the probability that node i is connected to **any** node inside the ball will not exceed $l^d/L^{d \quad \alpha} = L^{\check{Z}d \quad d+\alpha}$. Since the source node is not within this ball (for a large enough lattice), then any short path of length ¾ l must contain a long range connection to a node within the ball. The probability that a node with such a connection is encountered within l steps is, at best, l ð $L^{\check{Z}d \quad d+\alpha}$. If this probability can vanish as $L \quad ! \quad 1$, then it will always take more than l steps to reach the target. This happens when $\check{Z} < (d \quad \alpha)/(d+1)$, so the expected number of steps must exceed $L^{(d \quad \alpha)/(d+1)}$.

Meanwhile, for $\alpha > d$, the probability that a node has a long-range connection longer than $r = L \quad$, $0 < \quad < 1$, scales, again from Eq. (5.2.2), as

$$\frac{1}{\alpha \quad d} \int_{L}^{1} r^{\alpha} r^{d \quad 1} dr = \frac{L^{(d \quad \alpha)}}{(\alpha \quad d)^2} \text{ ¾ } L^{(d \quad \alpha)}: \tag{5.2.5}$$

Then the probability to travel a distance greater than $L \quad$ within L^{β} steps ($0 < \beta < 1$) is less than $L^{\beta} L^{(d \quad \alpha)}$. If this probability can vanish as $L \quad ! \quad 1$, then the total distance covered in L^{β} steps will never exceed $L^{\beta} L \quad$. Since we must reach the target eventually, and the source and target are L steps apart, we require $\beta + \quad = 1$. Meanwhile, the probability to make steps longer than $L \quad$ will vanish when $\beta + \quad (d \quad \alpha) < 0$. Both conditions are satisfied when $\beta < (\alpha \quad d)/(\alpha \quad d + 1)$ and it will always take more than $L^{(\alpha \quad d)/(\alpha \quad d+1)}$ steps to reach the target.

In summary, we have shown the expected transit time T to be approximately $\ln^2 L$ when $\alpha = d$; to be more than $L^{(d \quad \alpha)/(d+1)}$ when $\alpha < d$; and to be more than $L^{(\alpha \quad d)/(\alpha \quad d+1)}$ when $\alpha > d$. In the limit $L \quad ! \quad 1$, $\ln^2 L$ will grow more slowly than any positive power of L, therefore the optimum algorithm occurs for $\alpha = d$. See also Fig. 5.5.

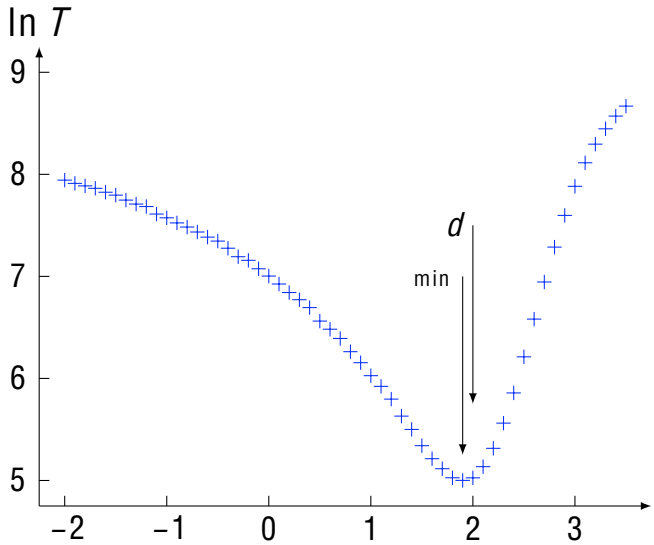


Figure 5.5: Simulations of Kleinberg Navigation on a two-dimensional lattice conþrm that $P_{\min} \; ! \; d$. Shown is the average of 1000 runs where the source and target were positioned $L = 10^4$ lattice steps apart.

5.2.1 Anisotropic lattices

A followup to Kleinberg's original work studied his navigation algorithm upon fractal lattices, in particular the Sierpinski carpet and gasket [126]. The gasket has the shape of an equilateral triangle, but in simulations it was embedded in a square geometry to simplify programming. This distorts the lengths of connections, altering the probability for long-range contacts (nodes in the \stretched" direction were less likely to be connected). They observed an appreciable discrepancy between the ideal $P = d$ (in the limit $L \; ! \; 1$) and the ideal P extrapolated from simulations for þnite L and hypothesized that the anisotropy was responsible.

We wish to study the isolated eÞect of anisotropy on Kleinberg navigation. To do this, we begin with a regular lattice ($d = 2$) and introduce one of two forms of anisotropy.

- **Lattice Anisotropy:** The underlying lattice is stretched horizontally by a factor $b > 0$ such that the area of each cell goes from $1 \; ð \; 1$ to $b \; ð \; 1$. See Fig. 5.6(a){(b).

- **Angular Anisotropy:** Long-range connections are made more probable between nodes separated more horizontally than vertically. To accomplish this, the probability for a long-range connection is not drawn from Eq. (5.2.1) but

instead in the following, essentially equivalent, way: a connection from node i of random length r, chosen from the distribution $P(r) \propto r^{-p}$, and random angle $0 \leq \theta \leq 2\pi$, chosen uniformly, is placed upon the lattice at i. This is connected to node j, the node closest to where it lands. To favor connections along one direction, θ is modified by a factor b:

$$\theta' = \arctan\left(\frac{\sin\theta}{b\cos\theta}\right); \tag{5.2.6}$$

where $b > 0$. See Fig. 5.6(c)–(d) for histograms showing the impact of b.

5.2.2 Simulations

We began our study of these anisotropic effects with simulations, undertaken in the summer of 2006 along with visiting undergraduate Mauricio Campuzano. The source and target nodes are separated by L horizontal lattice steps. Since the underlying lattice has no "voids" and the navigation algorithm is greedy, the message will always progress toward the target. Thus it remains within a disc of radius L centered on the target node. In addition, long-range connections (for *both* anisotropy types) are created based on the previously mentioned scheme of choosing a radius and angle, eliminating the need to compute the normalization term in Eq. (5.2.1). In combination, this allows for an "infinite" lattice to be simulated in that boundary conditions and other concerns can be neglected.

Simulations were performed for various values of b over a large range of p and L, each averaged 1000 times. For each b and L, the minimum p was computed by first fitting a fifth-order polynomial[10] to the averaged data, then using Newton's Method on the polynomial's derivative. Finally $p_{\min}$ was plotted as a function of $1/\ln^2 L$ for each chosen value of b. These are shown in Figs. 5.7–5.8 and indicate that $p_{\min} \to d$ as $L \to \infty$, regardless of b (see also Fig. 5.9).

[10] A parabola could be fitted to the data closest to the minimum, but we must first know what is `closest,' and if we know that then we know the location of the minimum. A higher-order polynomial overcomes this, similar to including higher order terms in a series expansion near the minmum of a function.

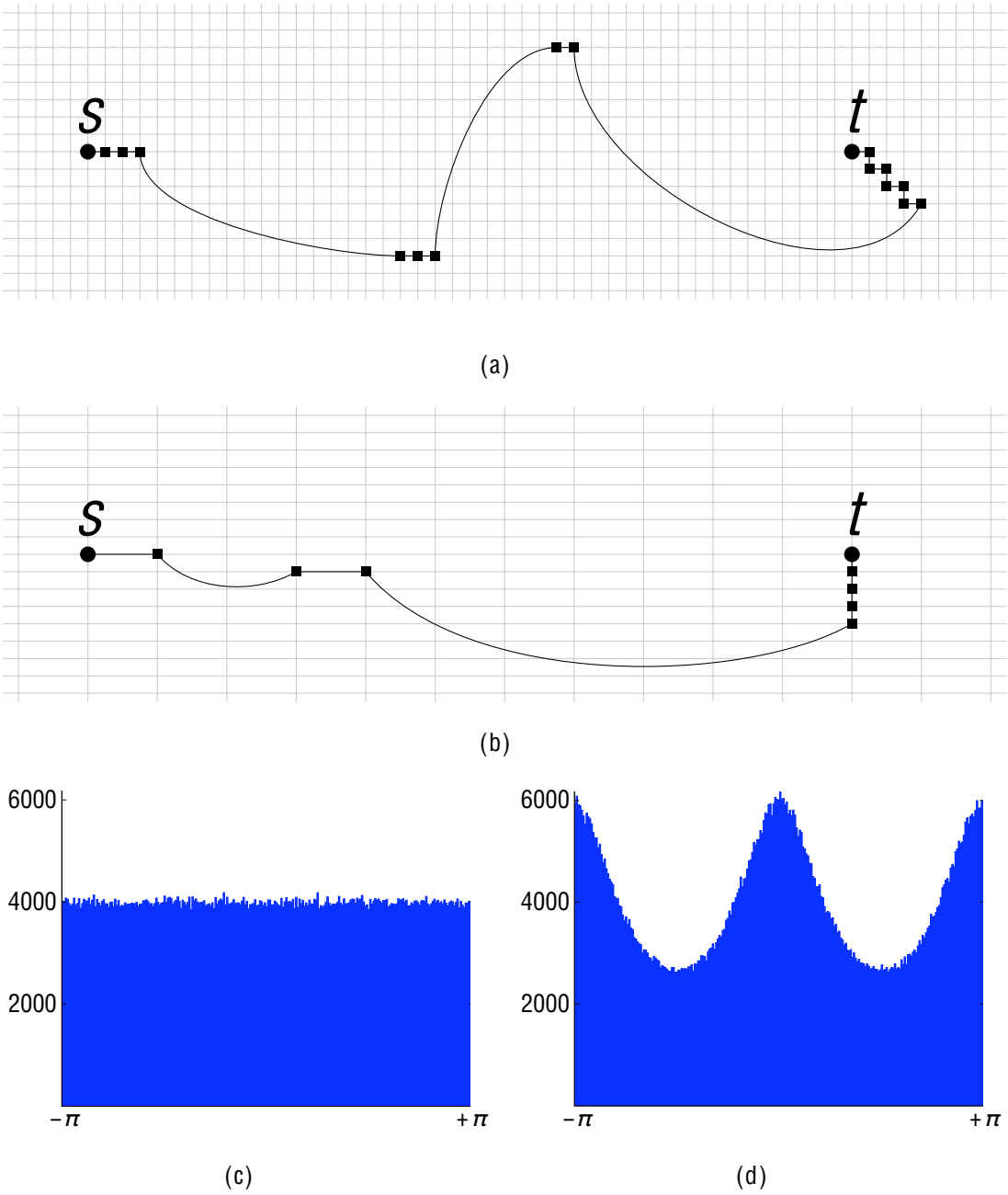


Figure 5.6: Kleinberg Navigation and anisotropy. Example message paths from a source node s to a target node t along intermediary nodes . (Unused long-range connections have been omitted.) The pnal long range connection in (b), despite its length, has only shortened the path by one step, since it lands so far \oP-axis." Note that s and t are closer in (b) than in (a). Angular anisotropy is shown with histograms of 10^6 uniformly random angles in Eq. (5.2.6) with (c) $b = 1$ and (d) $b = 3=2$.

To further clarify the behavior shown in Figs. 5.7{5.8, the following procedure was performed. First þt a cubic polynomial p_b, using least squares, to each b's curve. Then, subtract that polynomial from the isotropic case, $p_b \quad p_1$. This maps $b = 1$ to the horizontal axis and gives the behavior of the $b \neq 1$ curves \relative" to the isotropic curve. These are shown in Figs. 5.10{5.11. The diÞerent behavior for each type of anisotropy is clear: for the lattice case, the $b < 1$ curves converge to $P(1)$ at the same rate as $b = 1$, while $b > 1$ curves eventually converge similarly, but start above the $b = 1$ curve and eventually dip below it. Meanwhile, for the angular anisotropy, Fig. 5.11 shows that the $b > 1$ curves collapse onto the $b = 1$ curve while the $b < 1$ curves approach $P(1)$ at a diÞerent rate than the $b = 1$ curve.

The observed \crossover" present in the lattice anisotropy, especially for large b, is somewhat unexpected. The crossover size, as a function of b, $L_{\text{crossover}}(b)$ is explored by þnding the zero of each $p_b \quad p_1$. These are plotted in Fig. 5.12, and seem to indicate a power law relationship.[11] What is responsible for this remains an open question.

This analysis depends on a three types of least squares þts: the þfth-order polynomial þt to þnd each $P_{\min}$, the linear þt to the tails of the curves of $P_{\min}$ *vs.* $1/\ln^2 L$ to extrapolate $P(1)$, and the cubic þt to the entire $P_{\min}$ *vs.* $1/\ln^2 L$ curves. Least squares þtting is not robust to outliers, nor does it yield optimum estimators in the presence of non-normal errors (Gauss-Markov theorem). Since the minimum was found by þtting plots of $\ln T$ *vs.* P the errors are not gaussian. However, all the þts are reasonable to the eye, especially for very large L. In addition, the data is attest near the minimum, and thus its error distribution is less distorted by the natural log.

Several improvements to this regression analysis are possible, though unlikely to improve the results.[12] More robust techniques, such as weighted least squares (to deal with non-gaussian error) or iteratively reweighed least squares (to mitigate outliers), may be used for the þts. When þnding each $P_{\min}$, one can also weigh the points such that the data with smaller values of $\ln T$ are given stronger weights, emphasizing

[11] The data presented spans roughly a single decade in b, making it less than conclusive. Furthermore, it is possible for other kinds of functions to appear as straight lines on log-log plots, especially when the data doesn't span a suŽcient range.

[12] They would allow for rigorous error analysis, however.

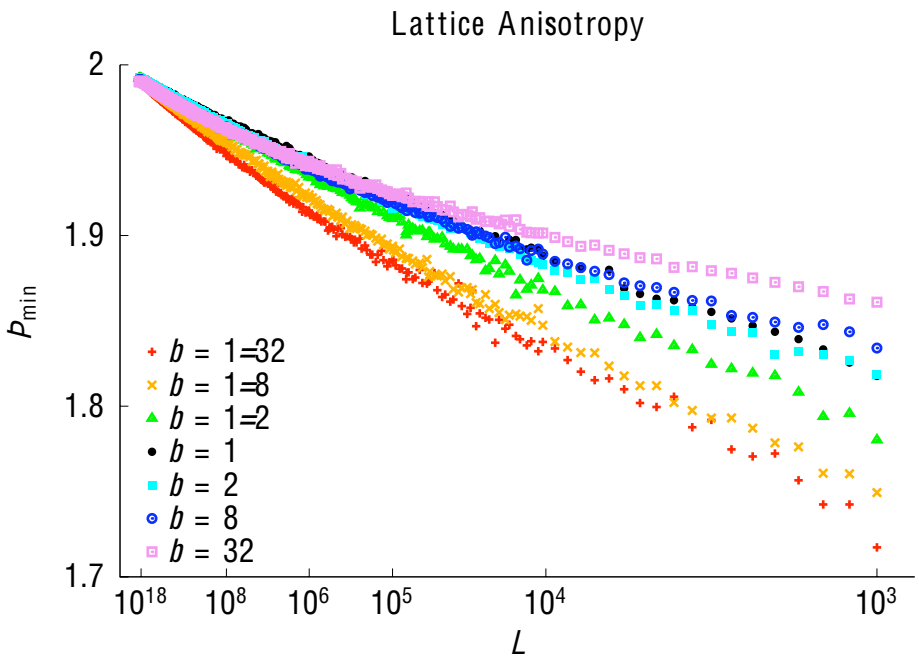


Figure 5.7: Simulations for lattice anisotropy. All curves approach $P(1)$, regardless of b. There is also a crossover eÞect where curves for $b > 1$ dip below the $b = 1$ curve. This is further explored in Fig. 5.10. See Fig. 5.9 for the extrapolated $P(1)$. A horizontal scale of $1=\ln^2 L$ is used throughout.

data closest to the minimum. The presence of non-gaussian errors due to the natural log in the $\ln T$ *vs.* P curves can be removed by computing $\ln T$ for each simulation then taking the average, instead of taking the natural log of the average T. Doing so will not alter the location of P_{min}, only its height. Since the simulation runs are iid (for þxed L and b), the central limit theorem ensures gaussian distributions. An alternative option is to simply þt the polynomials to *all* of the data instead of þrst taking the average. This makes the þtting calculation more expensive, but it can already be done so eŽciently that the increased cost is negligible.

5.2.3 Conclusions and future work

Simulations have shown that $P_{min} \to d$ as $L \to 1$ regardless of anisotropy, but the overall behavior is quite diÞerent for the lattice and angular anisotropies. A variety of

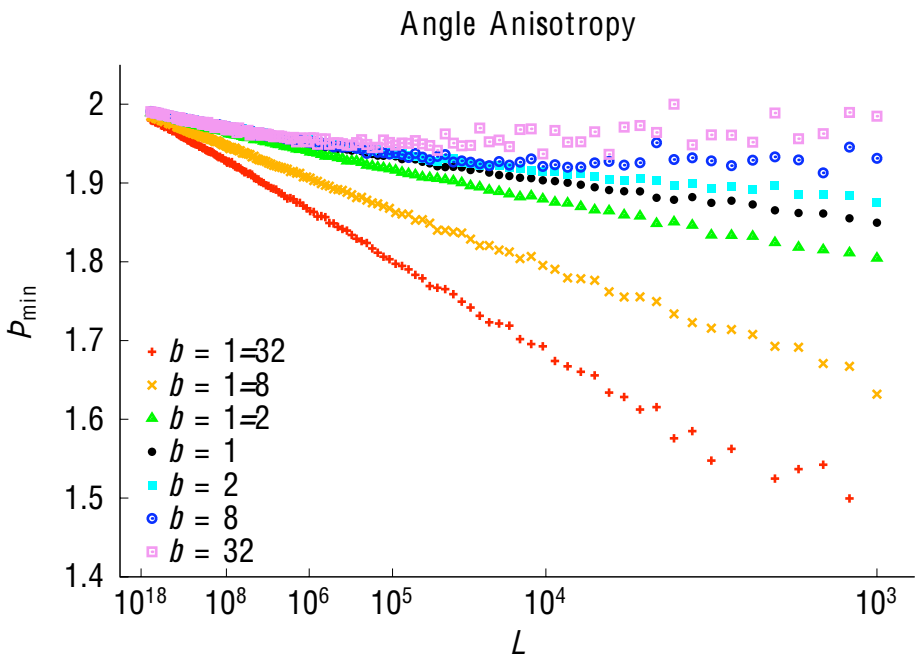


Figure 5.8: Simulations for angular anisotropy. All curves approach $P(1)$, regardless of b. Curves for $b < 1$ approach the inþnite limit at diÞering rates, while curves for $b > 1$ evetually collapse onto the $b = 1$ curve. This is further explored in Fig. 5.11. See Fig. 5.9 for the extrapolated $P(1)$.

open questions remain regarding Kleinberg navigation in the presence of anisotropy. The underlying phenomenon generating the crossover eÞect present in Figs. 5.7 and 5.10 is not well understood. The apparent power-law dependence of $L_{\text{crossover}}$ on b, shown in Fig. 5.12, remains an open question. The apparent lack of similar behavior in Figs. 5.8 and 5.11 is also not well understood. It is also an open question if the power law exponent depends on the dimension of the underlying lattice.

The scaling arguments constituting the proof that $\lim_{L \to 1} P_{\text{min}} = d$ are unable to capture salient details introduced by such anisotropy, since the number of nodes at distance r from the current node continues to scale as r^{d-1}, regardless of b. Thus, one needs an entirely new approach to analytically study the impact of anisotropy.

Regarding the general problem of optimum navigation in social and other networks, there are many avenues of open research to be pursued. The underlying

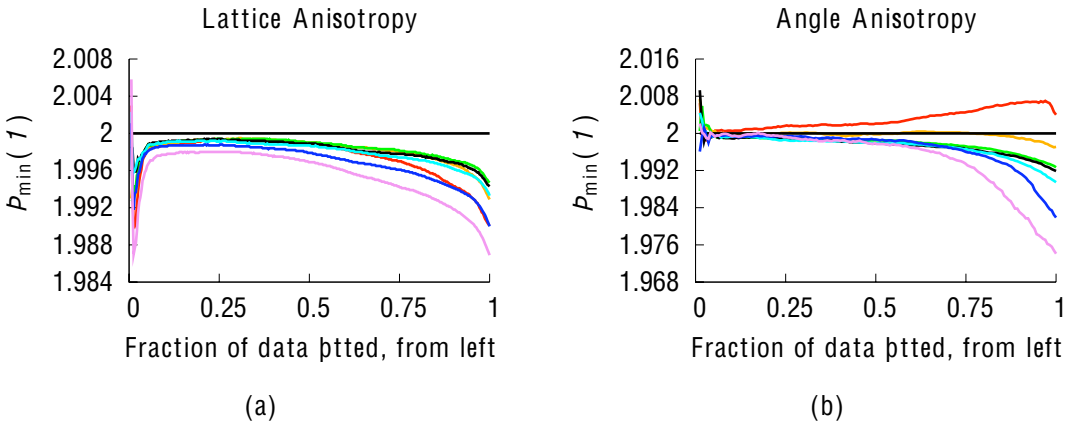


Figure 5.9: Extrapolating to $1/\ln^2 L \to 0$ with a linear least squares ßt to the curves in Figs. 5.7 and 5.8 shows excellent convergence of $P(1)$ to the expected value of $d = 2$. Good values should occur when the curves are flattest, which happens roughly around 0.25. A more robust ßtting procedure could be used, but the accuracy of these results imply that it is unnecessary. The horizontal lines at $P = 2$ provides a guide for the eye.

lattice used in the Kleinberg model has no gaps or holes, therefore the message will never need to \backtrack" during its journey, but this is not generally realistic. Can the Kleinberg navigation scheme be modiþed to account for such dead ends, or can it be shown that an entirely diÞerent procedure allows optimum navigation? If the gaps are large enough,[13] the message may *never* reach its goal: perhaps an optimum navigation scheme can only guarantee successful transport some fraction of the time.

Reasonable alterations to Kleinberg navigation in the face of such adversities include the introduction of randomness, where the current message holder may just randomly pass along the message if it cannot move closer to the target; the message holder may be allowed further knowledge of the network, such as the coordinates of its neighbors' neighbors; or perhaps a node's concept of distance will be altered in the presence of such gaps in the underlying lattice's geography (nodes on the \far shore" of a void may be considered farther away than indicated by their geographic distance alone). Unfortunately, it appears that all of these strategies introduce parameters

[13] If they scale with the size of the lattice or worse.

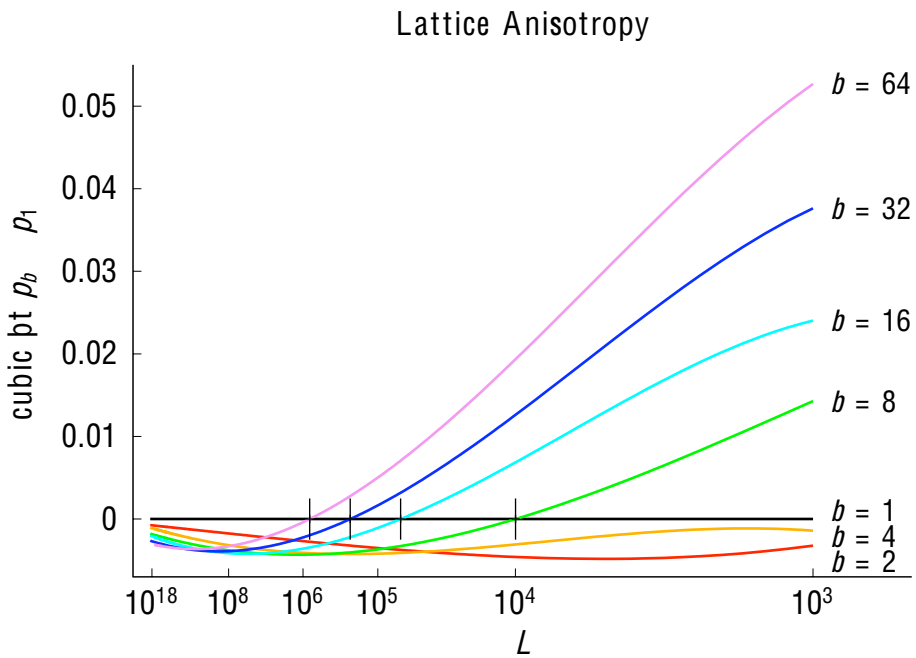


Figure 5.10: To provide a measure of smoothing, cubic polynomials p_b were þtted to the curves in Fig. 5.7. To clarify the impact of anisotropy, we show the behavior relative to the isotropic case, by subtracting p_1 from each p_b. This maps the isotropic curve to a horizontal line and introduces only minor distortion. The crossover behavior for $b > 1$ is clearly displayed. A more robust þtting may be necessary, but these results are still useful.

which must be studied. Kleinberg navigation is so intriguing due to the model's simplicity, it seems that more realistic models must necessitate more complications.

5.3 Conclusions

Complex networks provide an ideal setting for studying the dynamics underlying human society. Our work on modeling social networks can be divided along two main fronts. One is the introduction and analysis of the Patron-Artwork model, which provides a simple mechanism of how \fame" (more generally knowledge) can arise in a þnite population. The other is the study of Kleinberg navigation, which provides a model of the famous small-world `six degrees of separation' phenomenon.

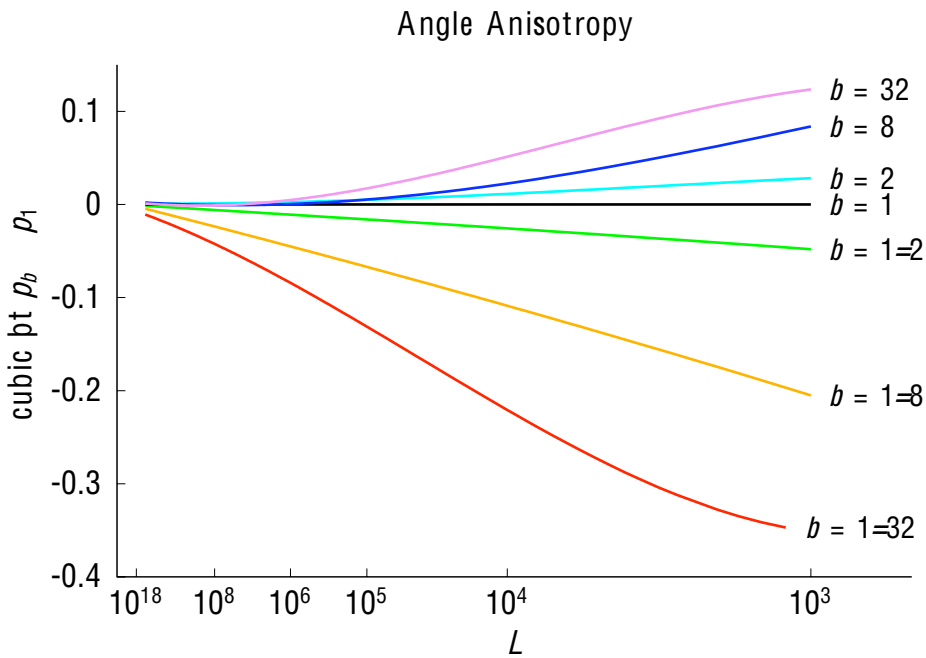


Figure 5.11: Similar to Fig. 5.10 but for angular anisotropy. This clearly shows the $b > 1$ curves collapsing onto the $b = 1$ curve as $L \; ! \; 1$, while the $b < 1$ curves approach $P(1)$ at diPering rates.

The Patron-Artwork model is a very promising mechanism to explain how \fame" or knowledge of an external population (the art) arises by means of a simple recommendation mechanism inside a social network (the patrons). This model has proven tractable when the social network is either a very large complete graph or a very large star graph, and simulations conþrm our results for the former. Our analysis has also served to illustrate some interesting concerns when using the master equation approach versus the rate equation approach, namely that the rate equation seems to only work when second-order terms in the Kramers-Moyal expansion of the master equation can be neglected, and even then it appears to give only qualitatively correct answers. Future work in this area would be to use simulations to conþrm the results in Sec. 5.1.4, to analyze more interesting social networks, to collect real-world data for comparison (such as IMDb votes), and to consider interesting generalizations (such as allowing diPerent values of r for diPerent types of nodes, directed or weighted social

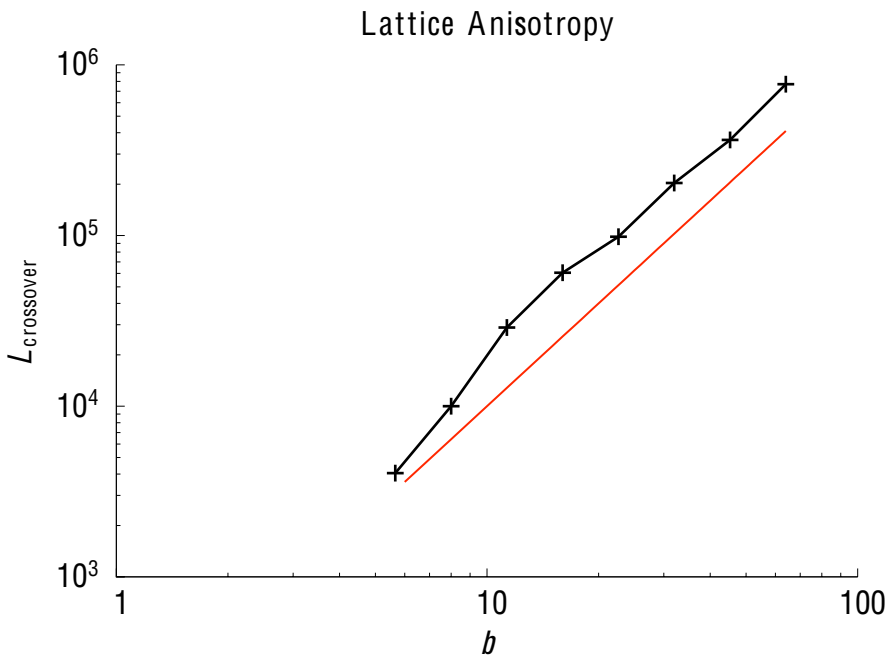


Figure 5.12: Evidence that the crossover locations for $b > 1$ exhibit a power law dependence on b. The straight line is of slope 2. The mechanism generating this behavior remains unknown. It is also an open question whether or not the power law exponent depends on the underlying lattice dimension.

networks, etc.).

Our work on Kleinberg navigation in the presence of anisotropic lattices shows several interesting facts. Simulations conþrmed to high accuracy that $P_{min} \; ! \; d$ as $L \; ! \; 1$, regardless of the amount or type of anisotropy. But the behavior at þnite sizes (some of which are very large) was not well understood. The apparent crossover behavior for the lattice anisotropy is not well understood, nor is it known why the angular anisotropy does not display a similar phenomenon. Future work on Kleinberg navigation may include modiþcations to the algorithm in the face of more realistic networks, such as those with gaps or voids. In such circumstances, convergence to the target is not guaranteed, and a greedy algorithm can get stuck. Whether or not a means of optimum navigation exists under these circumstances remains an open question. It would also be interesting to explore whether there really is a power law

dependence of $L_{\text{crossover}}$ on b, and if the power law exponent is (generically) related to the dimension of the underlying lattice.

Chapter 6

Conclusions

This thesis has focused on two main areas of complex networks research. One has been the development of new analysis tools and techniques, allowing a researcher to study and understand the important properties of a given network, whether it be generated from some model, such as a random network, or from the collecting of real-world data. New methods for detecting communities have been introduced, especially ones capable of detecting a particular community within a network that is too large or too dynamic to be fully explored. Shells, a unique property of a network that is neither local nor global, were studied, leading to several interesting statistics as well as a new measure of bipartivity. These shells have also allowed us to develop a very interesting new tool, the network portrait, capable of capturing a great deal of information in a compact, easy-to-understand representation.

The second main area of this work has been on applications of networks, and has focused on the usage of social networks as a means to study the complex behavior inherent in society. We have introduced the Patron-Artwork model to study how fame can emerge in a population due to a simple recommendation dynamic. Meanwhile, our study of Kleinberg Navigation, an idealized model of the small world phenomenon, has led to several intriguing þndings.

6.1 Contributions

The problem of identifying communities, dense clusters of interconnected nodes, has received much attention [24, 30, 33]. In Ch. 2 we have introduced a new type of community detection algorithm [39], one that is local in the sense that it does not require simultaneous information about every node and edge in the network. This information is often unavailable for networks that are either very large, such as the internet, or very costly to explore, such as some social networks. Yet a researcher may still wish to þnd a community in these networks, perhaps belonging to a particular node. These algorithms begin with such a starting node, and þnd the community containing that node by means of an agglomeration scheme, how nodes are added into the community, and a stopping criterion, how to tell that the entire community has been found and agglomeration should stop.

Alongside our local algorithm, we have developed a global application, using a \membership matrix" to determine the entire community structure. A hierarchy of sub-communities can be generated from this matrix, by means of a simple Hamming distance-based clustering, and this method has been shown to extract more meaningful information than competitors [41]. The method was also generalized to weighted networks [41].

This local method is not ideal however, as it is highly dependent on a starting node's location within a community. Meanwhile, more realistic and accurate methods were subsequently introduced (e.g. [42]). In response to the proliferation of competing techniques, we have introduced a simple benchmarking and evaluation scheme, tailored speciþcally to local algorithms, as a means to both compare and improve the accuracy of these methods. This benchmarking scheme consists of artiþcial test networks possessing a tunable degree of community structure (including our newly-introduced generalized ad hoc networks) coupled with a simple information theoretic partition similarity measure, to determine how \close" an algorithm's partition is to

the test network's pre-built community structure. Using this benchmarking procedure, we have shown that many algorithms perform comparably and, most importantly, that the accuracy of a local algorithm is far more aþected by how the method *stops* growing the community, than by how it grows the community. Several stopping criteria were introduced, often independently of a particular agglomeration scheme, and it was shown that there is room for improvement.

Chapter 3 focused on the study of shells, groups of nodes that are at a þxed distance from a starting node. Our original local community algorithm (Sec. 2.2) relied on the relationship between shells and communities, so further study of shells was worthwhile. We oþered a slight improvement to an existing calculation of the size and distribution of these shells [65, 66], allowing it to be applied more generally, including to smaller networks. We also studied the concept of perimetric edges, edges that are within shells, and their relationship to odd cycles (every perimetric edge participates in at least one odd cycle). Using this relationship, we introduced a new and inexpensive measure of bipartivity, how close a network is to being two-colorable.

Inspired by the distribution of shells, we then introduced the Network Portrait in Ch. 4. These portraits deþne a sort of \joint histogram" over the shell distributions, stored as a matrix. These matrices are unique for a given network, unlike adjacency matrices and edgelists, though we showed that they do not uniquely deþne a network. These portraits encode a great deal of information, however, including dimensionality and regularity, the presence or absence of a small world diameter, and even correlation eþects such as assortativity. Never before have all these quantities been available from a single plot. Quantitative comparison methods were also developed with the introduction of a \distance" metric between graphs, based on their respective portraits. This allows a researcher to, for example, develop a random model and see how well it represents a real-world network, a very useful tool. Finally, and perhaps most promising, a second matrix, describing how *edges* are distributed amongst shells, was introduced and it was shown that every graph of seven nodes or less was uniquely deþned by both of these matrices.

In addition to the previous work, we have also worked on two ways to *apply* complex networks to problems relating to social dynamics and modeling. One was the introduction of the Patron-Artwork model, having some similarities to existing work such as the voter model, to describe how a social network (of patrons) can generate a distribution of fame for an external population (of artwork). This model was studied for the limiting case where the social network was the complete graph, for both an inþnite and a þnite amount of artwork. For the inþnite case, a power law distribution of fame was always generated, but the þnite case led to an interesting phase transition where the distribution went from a gaussian to one with a power law tail on one side and a stretched exponential on the other. Distributions of the latter form are often studied by economists [116, 117], and it is very interesting that they appear naturally in this context. Calculations for the þnite case also illustrate a discrepancy between two diÞering solution techniques, which is an important point as both are heavily relied upon in other problems.

Another problem in the area of social networks was studied, that of Kleinberg Navigation, speciþcally on anisotropic lattices. Kleinberg navigation is an idealized model of the Milgram letter-passing experiment, and consists of a lattice of nodes with each node connected to its nearest lattice neighbors and one additional long-range contact, where the distance to the latter is given by a power law, $P(r) \propto r^{-\Phi}$. It has been shown that greedy navigation (the letter passing) is fastest when $\Phi = d$ (in the limit of an inþnite lattice) [124, 126]. The work in [126] generalized this result to fractal lattices, but showed a curious discrepancy between the Φ extrapolated from þnite simulations and the predicted $\Phi = d$. To ease programming, the fractal lattices were embedded in a square geometry, stretching them slightly in one direction. They hypothesized that this anisotropy was the source of the discrepancy.

In this work, we tested that hypothesis directly, by introducing two types of anisotropy for square lattices. It was shown to high accuracy that Φ does approach d as the lattice becomes very large, but interesting behavior was present \along the way." Speciþcally, a crossover phenomenon was present, where the optimum value of

P for a particular amount of anisotropy coincided with the isotropic P, at a certain lattice size. Furthermore, this behavior was present in one type of anisotropy and not the other. This crossover was observed to have a power law dependence on the amount of anisotropy, with power law exponent ≈ 2. It is worth noting that even though $P \gg d$, the study of this phenomenon is of the utmost importance since it represents more realistic lattices and it still occurs at the sizes of the large networks encountered in everyday circumstances, such as the internet.

6.2 Open Questions and Future Research

All of the areas of this thesis have raised interesting questions and opened new avenues for fruitful study. Here we list some of the more important open questions and further research opportunities.

For the local community methods, it was shown that they do not perform as well as a global method, which is to be expected, but how close to global accuracy can one achieve? Furthermore, these local methods suffer a problem of back links, those links that are later discovered during the process of the algorithm, and it is not clear how (or even if) this problem can be overcome. Finally, it was shown that stopping criteria are a critical component of a method's accuracy and that there is room for further improvement, so developing improved criteria is a prime area for future work. The new benchmarking procedure will be invaluable in this regard.

The network portraits of Ch. 4 open many possibilities. One is simply: what other properties can be understood by looking at the portraits themselves? A distance D was introduced to quantify the similarities and differences between networks, but the metric properties of D remain poorly understood. The fact that the portrait is unique for a given network immediately applies it to the problem of graph isomorphism, but the specifics of this applicability requires further research. It is trivial to construct the portrait for a given network, but the opposite is not true, and a general construction algorithm to generate a graph from its portrait alone would be a great boon.

The conjecture that all graphs are uniquely encoded by both matrices needs to be conþrmed or refuted. A conþrmation seems diŽcult, but a refutation requires only a single counter-example. Finally, generalizing the portraits to weighted networks was brie y discussed, but the best approach to doing this (without introducing a large number of parameters) is not clear.

Our work on social networks has further promise as well. For the patron-artwork model, can solutions be found for more realistic social networks? The discrepancy between the two solution techniques used when analyzing the complete graph is clear, but the underlying cause requires further study. This is very important, since these techniques are in widespread use. Moreover, does a generic solution technique exist? Real-world data on the distribution of fame is available, such as tallies of the number of reviews per movie on the IMDb. Can the distribution of reviews be reconciled with the patron-artwork model? It would be a very important result if one could indicate the general structure of the underlying social network from the fame distribution alone.

For the Kleinberg navigation problem, how and why does the crossover behavior's power law dependence on anisotropy occur, and why is it only present for one type of anisotropy? Does the power law exponent depend on the dimension of the underlying lattice, is it always 2, or neither? Finally, we discussed further generalizations of the navigation problem to non-uniform lattices, such as those with large gaps or voids. Here a simple greedy navigation algorithm will likely fail, with the message becoming trapped and unable to progress toward the target. Can an optimum navigation algorithm be discovered for these circumstances? If so, will delivery of the message be guaranteed or will it be lost some þnite fraction of the time? All of these problems are quite relevant, since real geographically distributed networks are seldom as tidy as a perfect lattice, and so results will have immediate application.

Appendix A

Partition Similarity

The analysis of competing local community algorithms hinges upon a means to compare how \similar" community partitions are. This is a problem more general than partitioning graphs and, for completeness, we present useful background material covering a variety of ways to compare data partitions. We begin by deþning some terminology and other useful quantities, then discuss the strengths and weaknesses of various comparison measures. Our discussion follows those of Meilþ [127] and Karrer [128].

A.1 Partitions

We deþne a **clustering** $C = \{c_1, c_2, \ldots, c_A\}$ as a partition of a set of points (dataset) D into A (mutually disjoint) subsets c_1, c_2, ..., c_A called **clusters**.[1] In other words, $c_k \cap c_l = \emptyset$ when $k \neq l$ and $\bigcup_{i=1}^{A} c_i = D$. Let D contain N points and c_k contain n_k points. Then $n = \sum_i n_i$.

Suppose we are given a dataset D and two partitions $V = \{v_1, v_2, \ldots, v_B\}$, $W = \{w_1, w_2, \ldots, w_C\}$ of that dataset. Our goal here is to measure how *similar* or related V

[1] In terms of þnding communities (Ch. 2) the nodes of the graph form D, the individual communities form the clusters $c_1, c_2, \ldots, c_A$, and one seeks the clustering that maximizes the number of edges between nodes in the same clusters and minimizes the number of edges between nodes in diþerent clusters.

	w_1	w_2	$\cdots$	w_C	
v_1	n_{11}	n_{12}	$\cdots$	n_{1C}	$n_{1:}$
v_2	n_{21}	n_{22}	$\cdots$	n_{2C}	$n_{2:}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
v_B	n_{B1}	n_{B2}	$\cdots$	n_{BC}	$n_{B:}$
	$n_{:1}$	$n_{:2}$	$\cdots$	$n_{:C}$	N

Table A.1: Notation for the confusion matrix n_{ij} of partitions V and W, as well as row and column sums $n_{:j}$ and $n_{i:}$. Both row and column sums themselves sum up to N.

and W are. A useful quantity is the **confusion matrix** (also known as a contingency table):

$$n_{ij} \quad \text{the number of points that appear in both } v_i \text{ in one clustering and } w_j \text{ in the other} \tag{A.1.1}$$
$$= |v_i \cap w_j|.$$

This matrix obeys $\sum_i \sum_j n_{ij} = N$. Defining row and column sums $n_{:j} = \sum_i n_{ij}$ and $n_{i:} = \sum_j n_{ij}$ gives $n_{:j} = |w_j|$ and $n_{i:} = |v_i|$. See also Table A.1.

We seek a means to quantity the differences between V and W, preferably one normalized to $[0,1]$. These measures roughly fall into three categories: pair-counting methods, clustering matching, or information theoretic methods.

A.2 Pair-counting methods

Some measures compare partitions by looking at all possible pairs of points (x,y), $x,y \in D$, and counting how they fall relative to one another in each partition. There are $\binom{N}{2}$ total pairs, and each can be distributed in one of four ways: either x and y are in the same cluster in both partitions, different clusters in one partition but the same cluster in the other, or different clusters in both partitions. Formally, let us

count the number of pairs meeting these descriptions:

a # of pairs such that $x; y \in v_i$ and $x; y \in w_j$;

b # of pairs such that $x \in v_i$, $y \in v_k$ but $x; y \in w_j$;

c # of pairs such that $x; y \in v_i$ but $x \in w_j$, $y \in w_l$;

d # of pairs such that $x \in v_i$, $y \in v_k$ and $x \in w_j$, $y \in w_l$,

where $i \neq k$ and $j \neq l$. These can be calculated directly from the confusion matrix:

$$a = \sum_{i;j} \binom{n_{ij}}{2}; \tag{A.2.1}$$

$$b = \sum_j \binom{n_{:j}}{2} - \sum_i \binom{n_{ij}}{2}; \tag{A.2.2}$$

$$c = \sum_i \binom{n_{i:}}{2} - \sum_j \binom{n_{ij}}{2}; \tag{A.2.3}$$

$$d = \binom{N}{2} - a - b - c. \tag{A.2.4}$$

Several statistics are built using these quantities. Wallace [129] introduced two asymmetric quantities:

$$W_I(V; W) = \frac{a}{a + b}; \tag{A.2.5}$$

$$W_{II}(V; W) = \frac{a}{a + c}. \tag{A.2.6}$$

Since $a + b$ is the number of pairs in the same cluster in W, and a+c is the number of pairs in the same cluster in V, then these are the probability that a pair of points which are in the same cluster in one partition are also in the same cluster in the other.

Fowlkes and Mallows [130] introduced a symmetric criterion, the geometric mean of Eqs. (A.2.5) and (A.2.6):

$$F(V; W) = \sqrt{W_I(V; W)\, W_{II}(V; W)}. \tag{A.2.7}$$

Yet another pair-wise statistic was introduced by Rand:

$$R(V; W) = \frac{a + d}{N(N - 1)/2} \tag{A.2.8}$$

Both R and F need to be renormalized to fall over the range $[0;1]$. This is typically done by subtracting a \null hypothesis" value, assuming clusters are random and independent, and then normalizing the range to give 0 for the null case, and 1 for the maximal case where the clusterings are identical. This procedure is similar to the derivation of modularity given in Ch. 2.

Such adjustments are not ideal for several reasons. Concerns have been expressed as to the plausibility of the null hypotheses [129]. Another issue is that the value of the baseline (before subtracting the null model) can vary considerably depending on the clusterings, and this makes comparing statistics against one another more problematic. For an in-depth discussion, see [130, 127].

Some other pair-wise statistics include the *Jaccard Index* [131],

$$J(V;W) = \frac{a}{a+b+c}; \tag{A.2.9}$$

and the Mirkin metric [132],

$$M(V;W) = \sum_i n_{i:}^2 + \sum_j n_{:j}^2 \quad 2 \sum_i \sum_j n_{ij}^2 \tag{A.2.10}$$

$$= 2(b+c) = N(N \quad 1)\left(1 \quad R(V;W)\right): \tag{A.2.11}$$

Thus the Mirkin metric is just another adjusted form of the Rand index.

A.2.1 Edge counting

In addition to counting *every* pair of points $i;j \ 2 \ D$, one can only count the pairs of points that correspond to edges in the graph. That is, count all pairs $i;j \ 2 \ D$ such that $9e_{ij} \ 2 \ G$. This is, in a sense, a weaker criteria, since one only cares about how edges are distributed amongst clusters, and not how disconnected nodes are situated.

Another possibility is to count the pairs corresponding to neighbors, then next-nearest neighbors, etc. weighing each count less, to account for the increasing distance between the pairs.

Neither of these concepts appear to have been introduced in the literature, but most researchers have turned to alternatives to the pair-counting measures, for various reasons, and therefore the pursuit of these ideas may not be worthwhile.

A.3 Cluster matching

One can also compare clusterings based on various set cardinalities. These avoid assumptions regarding how the clusterings were generated.

Meilă and Heckerman introduced the statistic H as follows [133]. Each cluster in V is given a \closest match" in W. Then H computes the total \unmatched" probability mass in the confusion matrix:

$$H(V;W) = 1 - \frac{1}{N} \max_{\pi} \sum_{i=1}^{B} n_{i;\pi(i)}; \tag{A.3.1}$$

where it is assumed without loss of generality that $B \leq C$, and $\pi(i)$ is an injective mapping of $\{1;\ldots;B\}$ into $\{1;\ldots;C\}$, and the maximum is taken over all such mappings. This statistic is symmetric and has value 1 for identical clusterings. See also [134, 135].

A similar, though asymmetric statistic was also introduced [136]:

$$L(V;W) = \frac{1}{K} \sum_{i} \max_{j} \frac{n_{ij}}{n_{i:} + n_{:j}}: \tag{A.3.2}$$

This asymmetry is less than ideal so van Dongen [137] introduced a related but symmetric statistic:

$$D(V;W) = 1 - \frac{1}{2N} \left[\sum_{i} \max_{j} n_{ij} + \sum_{j} \max_{i} n_{ij} \right]: \tag{A.3.3}$$

Note that D is 0 for identical clusterings and always smaller than 1 otherwise.

All of these statistics suffer from the \matching problem" in that $L; H$, and D all first find a corresponding \best match" for each cluster within the other clustering, then sum the contributions of these matches. They ignore all information related

to the remaining \unmatched" parts of each clustering, and this is not ideal. As an example of this drawback [128], suppose we have three clusterings:

$$C_1 = \{\{a; b; c\}; \{d; e; f\}; \{g\}\}; \tag{A.3.4a}$$

$$C_2 = \{\{a; b; c\}; \{d; e\}; \{f\}; \{g\}\}; \tag{A.3.4b}$$

$$C_3 = \{\{a; b; c\}; \{d\}; \{e\}; \{f\}; \{g\}\}: \tag{A.3.4c}$$

For the van Dongen statistic, $D(C_1; C_2) = D(C_1; C_3)$, despite the claim (and support of other measures) that C_1 is more similar to C_2 than to C_3. For more discussion, see [127].

A.4 Information Theoretic methods

Instead of looking at how pairs of points in the dataset are distributed one can consider the *probability* for points to be placed within clusters in each clustering. Thus one can assume the confusion matrix defines a joint probability $P(v_i; w_j)$ that a randomly chosen point x appears in both v_i and w_j. Formally this means that v and w are assumed to be values of random variables V and W. Then:

$$P(v; w) \quad \Pr(V = v; W = w) \quad \frac{n_{ij}}{N}; \tag{A.4.1}$$

where the suppressed indices i and j are taken to be the indices of the confusion matrix that correspond to v and w, respectively. Following this, the row and column sums then correspond to the marginal distributions:

$$P(v) \quad \Pr(V = v) = \sum_j P_{ij} = \frac{n_{i:}}{N}; \tag{A.4.2}$$

$$P(w) \quad \Pr(W = w) = \sum_i P_{ij} = \frac{n_{:j}}{N}: \tag{A.4.3}$$

Now, consider the *mutual information* between clusterings V and W to be equal to the traditional mutual information of the corresponding random variables:

$$I(V; W) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} P(v; w) \log \frac{P(v; w)}{P(v)P(w)} \tag{A.4.4}$$

Mutual information measures how much knowledge we have about V by having complete knowledge of W, and vice versa. If the clusterings are identical, then we know one completely if we know the other. If there is no correlation then we learn nothing. This can be visualized by using the fact that $P(v;w) = P(v|w)P(w) = P(w|v)P(v)$. Plugging this in reduces Eq. (A.4.4) to:

$$\begin{aligned} I(V;W) &= \sum_{i;j} P(v;w)\log P(v|w) - \sum_{i} P(v)\log P(v) \\ &= H(V) - H(V|W) = H(W) - H(W|V) \end{aligned} \tag{A.4.5}$$

where $H(V)$ is the information (entropy) of V and $H(V|W) = \sum_j P(w)H(V|W = w)$ is the conditional entropy (the additional information needed to know V once W is known). If W tells us nothing about V then the two terms are equal and $I(V;W) = 0$. In essence, I contains the same information as the conditional entropy, but is symmetric, while the conditional entropy is not. This makes I more useful as a measure of distance or similarity.

The values of I do not necessarily fall in the range [0,1] so a normalization is often used. There are several possibilities, one popular choice is the following. Mutual information is bounded by the entropies of the involved random variables:

$$\begin{aligned} I(V;W) &= H(V) - H(V|W) \le H(V) \\ &= H(W) - H(W|V) \le H(W); \end{aligned}$$

thus

$$I(V;W) \le \min\left(H(V); H(W)\right) \le \frac{H(V) + H(W)}{2}.$$

This provides a tight upper bound on $I(V;W)$, giving the normalized form:

$$I_{\text{norm}}(V;W) = \frac{2I(V;W)}{H(V) + H(W)}. \tag{A.4.6}$$

Equation (A.4.6) was used in the local community benchmarking and evaluation method presented in 2.3.3.

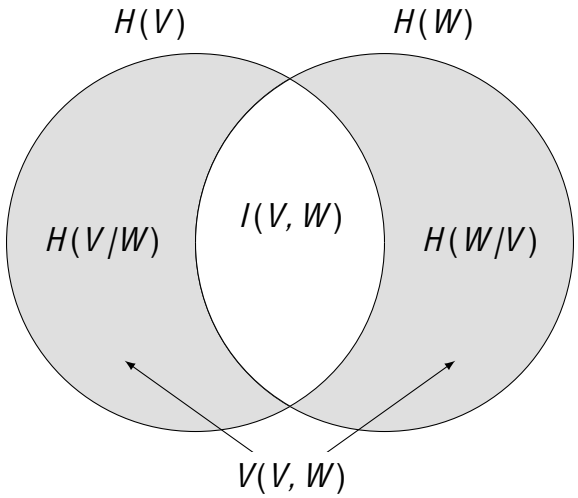


Figure A.1: Diagramming the relationship between $V(V;W)$, the shaded region, and various other quantities. The two circles represent the entropies H, while the overlapping region is the mutual information, and the remaining shaded regions give the conditional entropies. The sum of the conditional entropies is the variation of information. From [127].

Recently, another information theoretic measure has been introduced, the *variation of information* V [127, 128], deþned as:

$$\begin{aligned} V(V;W) &= H(V) + H(W) \quad 2I(V;W) \\ &= H(V|W) + H(W|V) \\ &= \sum_{i;j} P(v;w) \log \frac{P(v;w)}{P(w)} \quad \sum_{i;j} P(v;w) \log \frac{P(v;w)}{P(v)} : \end{aligned} \tag{A.4.7}$$

This is the sum of the information needed to know V given W and the information needed to know W given V. It is a true metric in the space of clusterings, satisfying all the requirements of a proper distance [127]. It is also a local measure in the sense that the distance between two clusterings that only diþer in one \region" of the dataset does not depend on how the rest of the dataset is clustered. Due to these properties, this measure has become quite popular. See Fig. A.1

The maximum value that V takes is $\log N$, which happens when V consists of a single cluster of size N and W consists of N clusters of size 1, or vice versa. This value increases with N because larger datasets intrinsically contain more information, but one can simply renormalize by $\log N$ if this is undesired. When comparing two community partitions of the same graph, N is the same for both, and this normalization is irrelevant.

Appendix B

Shells, Cycles and Communities[1]

In addition to the local community methods presented in Secs. 2.2{2.3, we have also explored how *short cycles* are distributed over community structure. Here we present some background on the importance of cycles, a means to identify and (approximately) enumerate cycles based on how their component edges are distributed within shells, and a simple set of algorithms for identifying which edges in a network form an inter-community \backbone." The identiþcation of shells and cycles is also of interest regarding the material covered in Ch. 3

An important feature of complex networks are the cycles of diÞerent lengths which underlie the patterns of connectivity [62]. The statistical distribution of cycles has been acknowledged as particularly important for deþning not only the topology of the respective networks, but also the dynamics of systems running on such frameworks, due to feedback [138]. The number of cycles in even a moderately sized network is so large that it is intractable to discover all of them. Indeed, many algorithms based on, for example, random walks, have been used to estimate the number of cycles [62, 70].

Generally, the density of cycles tends to increase as more edges are incorporated into a network, with longer cycles emerging before shorter ones [139]. Therefore, the density of cycles of diÞerent lengths can be used as an indicator of the connectivity between any subset of nodes. In other words, the larger the number of short cycles

[1] Published in [75]

amongst a subset of nodes, the more connected such nodes are to one another. Longer cycles tend to grow, \coiled up," alongside these shorter cycles, however, blurring the distinction between nodes based solely on short-cycle participation. We present methods to overcome this.

B.1 Describing Cycles with Shells

For a graph G, we are interested in þnding cycles of length $3 \quad l \quad 5$ containing a particular vertex v. To describe this, we begin by decomposing G into shells G_i about v. Since we are only interested in cycles of length $l \quad 5$, we need only keep $G_1(v)$ and $G_2(v)$.

It is simple to describe short cycles using these shell decompositions. For example, for every edge e_{ij} in $G_1(v)$, there exists a 3-cycle (triangle) $v\{i\{j\{v$. Similarly, for every path of length 2 or 3 in S_1, there exists a 4- or 5-cycle, respectively. Another 4-cycle and two more 5-cycles exist involving both $G_1(v)$ and $G_2(v)$.

One can also describe *all* possible cycles in such a manner. For a cycle of length L ½ 3, the number of such possible \cases" $N(L)$ must rapidly grow with L. Since it requires two edges to visit a shell, any L-cycle can visit at most J shells, where

$$J = \begin{cases} \frac{L}{2}; & L \text{ even}; \\ \frac{L \quad 1}{2}; & L \text{ odd}: \end{cases} \tag{B.1.1}$$

If the farthest shell the cycle visits is G_j (with $j < J$), there are at most $L \quad 2j$ remaining edges that must be distributed between and within the $G_1; G_2; :::; G_j$ shells. The number of ways to distribute $L \quad 2j$ edges over j shells is $\frac{(L \quad 2j+j \quad 1)!}{(L \quad 2j)!(j \quad 1)!}$. Yet it is possible for a cycle to \zig-zag" between shells, using more than the $2j$ necessary edges between shells. Therefore, the total number of possible ways to distribute an L-cycle is at least

$$N_l(L) = 1 + \sum_{j=2}^{J} \sum_{i=0}^{j} \binom{i+j \quad 2}{i} \binom{L \quad 2i \quad j \quad 1}{L \quad 2(i+j)}; \tag{B.1.2}$$

with the outer sum accounting for all the possible shells the cycle can visit, the inner sum for all the optional pairs of edges that can lie between shells and the +1 for the 3-cycle (triangle). Here i is the number of pairs of edges between shells beyond the j necessary to visit the j shells.

Furthermore, splitting the inner sum in Eq. (B.1.2) into cases where extra edges are distributed ($i > 0$) and are not ($i = 0$):

$$N_l(L) = 1 + \sum_{j=2}^{\left\lfloor \frac{L-1}{2} \right\rfloor} \left(\binom{L-j-1}{L-2j} + \sum_{i=1}^{j} \binom{i+j-2}{i} \binom{L-2i-j-1}{L-2(i+j)} \right)$$

$$= \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{L-1} + \sum_{j=2}^{\left\lfloor \frac{L-1}{2} \right\rfloor} \sum_{i=1}^{j} (-1)^{L+i} \binom{1-j}{i} \binom{j}{L-2(i+j)} : \qquad \text{(B.1.3)}$$

This gives a \lower" lower bound of $\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{L-1}$, which is equivalent to neglecting to count those cycles with extraneous edges between shells.

Equation (B.1.3) fails to take into account permutations of the *ordering* of edges between and within adjacent shells. A simple upper \bound" is possible, however, as there are certainly no more than $L!$ possible permutations over the whole network:

$$N_u(L) = 1 + \sum_{j=2}^{\left\lfloor \frac{L-1}{2} \right\rfloor} \sum_{i=0}^{j} \binom{i+j-2}{i} \binom{L-2i-j-1}{L-2(i+j)} L!; \qquad \text{(B.1.4)}$$

with

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{L-1} \leq N_l(L) \leq N(L) \leq N_u(L): \qquad \text{(B.1.5)}$$

The number of possible cycles grows at least exponentially with length. If one were to assume that each particular case has an equal probability of occurring in a given network, which is not generally justified, then the number of cycles present also grows exponentially, as expected.

B.2 Cycles and Communities

Community structure can be studied by comparing the edges covered by certain cycles with the original graph. Let

$$C_l(i) \qquad \text{the set of edges traversed by all } l\text{-cycles starting from vertex } i. \tag{B.2.1}$$

Starting from all vertices and limiting ourselves to only short j-cycles,[2]

$$C \quad \bigcup_{i \in V} \bigcup_{j} C_j(i): \tag{B.2.2}$$

From this, construct a graph

$$H = \{V; E \cap C\} \tag{B.2.3}$$

which is the graph containing only edges that do not participate in j-cycles in G. Separate communities in G will appear as disconnected components in H. We interpret vertices in H with degree zero as communities of size one.

In specifying H, the question of what to choose for j has been left open. Choosing just $j = \{3\}$ will correspond to deleting all edges from G that participate in 3-cycles, generally not a useful result. One may consider j to be a tunable parameter, used to get a desired result when applied to a specific network.

One issue that can occur is that longer cycles often overlap shorter cycles. In terms of communities, most inter-community edges contain few (if any) short cycles, but intra-community edges tend to contain both long and short cycles, since a long cycle can \coil" inside the community. If one were to just delete all 5-cycles in a graph, it is very possible to end up deleting all edges.

There is quite a bit of leeway in how we choose j and build H and we can use this to our advantage. For example, pick two cycle lengths s and t ($s < t$) and compute C_s and C_t. Then, build another set of edges

$$C_{t \setminus s} \quad C_t \cap C_s; \tag{B.2.4}$$

[2] We specify \short cycles" as those of length 3, 4, or 5 but this is not a set rule and, in certain circumstances, it may prove advantageous to consider 4- or 5-cycles, or even just 5-cycles.

containing edges that participate in t-cycles *but not* s-cycles. The graph $H = \{V; C_{tns}\}$ will contain edges that tend to be between communities and not within, for an appropriate choice of t and s. One can think of this as a \backbone" of the network, and deleting these edges may be a useful pre-processing step for applying other community-detection algorithms, including betweenness [39, 3].

B.3 Application Examples

We now apply these cycle-based methods to a network of NCAA Division I-A football games held during the 2005 regular season.[3] This example also helps illustrate the meaning of Eq. (B.2.4). In addition, we discuss how these methods can break down and ways to overcome that.

In NCAA football, teams are grouped into *conferences* based on location. To save on transportation time and cost, more games are played between teams in the same conference than in diPerent conferences. A graph of the game schedule, where nodes are teams and edges connect teams that have played against each other, naturally exhibits community structure based on these conferences [140].

Figure B.1 displays the original football network; the network generated by using $j = \{3\}$ in Eq. (B.2.1); and the network generated by building C_{tns} using $t = 5$ and $s = 3$ in Eq. (B.2.4). The graph $H = \{V; C_{tns}\}$ contains no edges between teams within the same conference.

Choosing $j = \{3\}$ deletes all edges that do not participate in 3-cycles, most of which are between conferences, though some edges remain. This will not split the network into seperate components based on the communities but it may be useful as a preprocessing step for betweenness or another community detection algorithm.

We propose that edges in C_{5n3} comprise the majority of this network's inter-community structure, its \backbone." To test this, one can compare the distributions of edge betweenness for these backbone and non-backbone edges, as shown in Fig. B.2.

[3] Data taken from published schedule at http://www.ncaa.org

Backbone edges tend to carry much higher betweenness values than the more common non-backbone edges.

B.4 Concluding Remarks

The identiþcation and characterization of the communities present in complex networks stands out as one of the most important approaches for understanding their structure and possible formation and evolution. At the same time, the distribution of cycles of various lengths in a complex network has important implications for the connectivity, resilience and dynamics of the respectively studied networks. The current work brought together these two important trends, in the sense of applying short cycle detection as a means to help the identiþcation of communities in complex networks. The relationship between the cycles and communities in the football network has been further investigated in terms of the betweenness centrality measurement, conþrming that the obtained backbone edges tend to exhibit higher betweenness values.

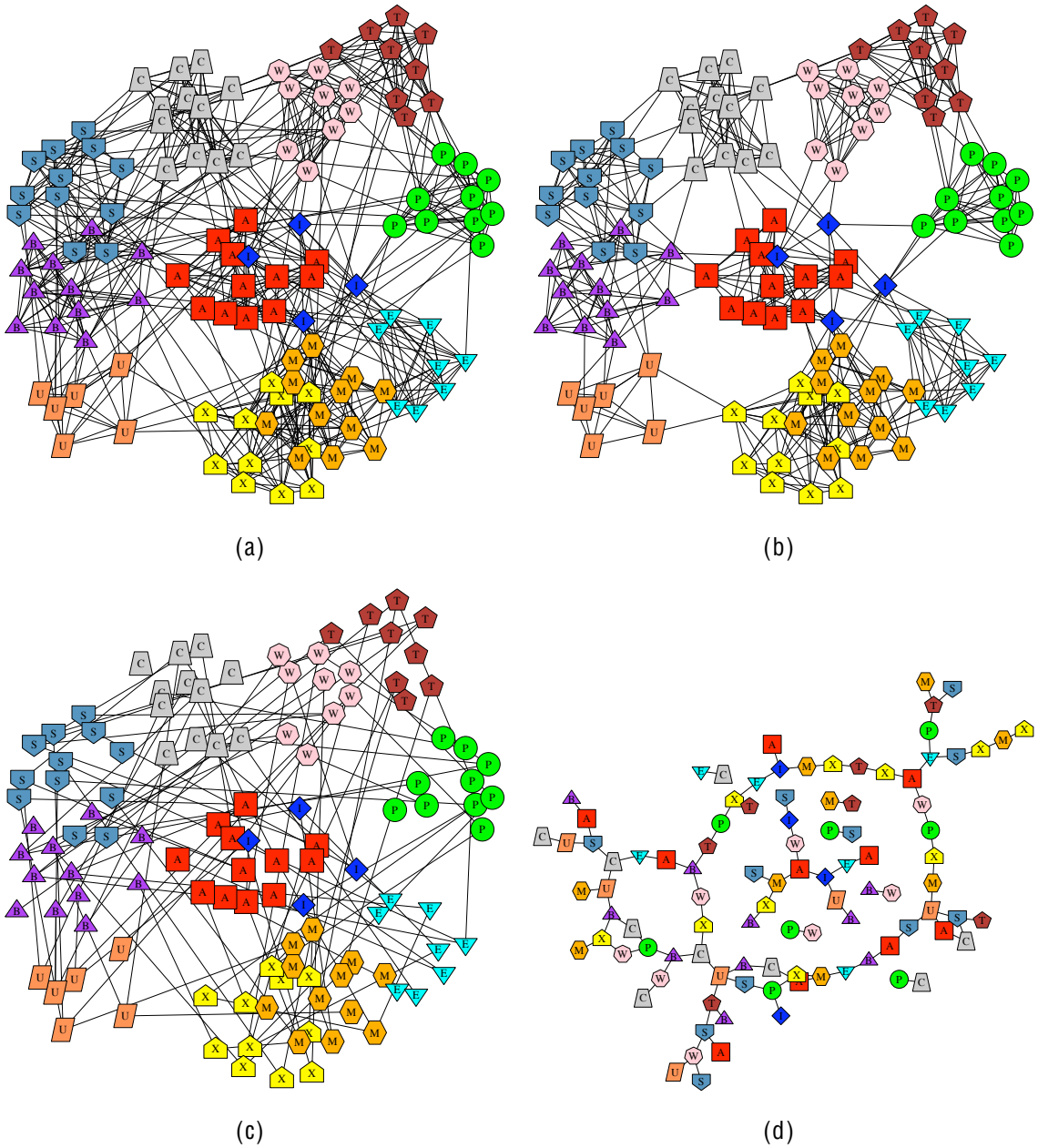


Figure B.1: The NCAA Div I-A 2005 regular season with all edges (a), with 3-cycles only (b), and with just C_{5n3} edges (c). Fig. (d) is the same graph as (c) but with a layout emphasizing that no edges within conferences remain (degree zero nodes omitted). As per [74], the conferences are: A = Atlantic Coast, B = Big 12, C = Conference USA, E = Big East, I = Independent, M = Mid-American, P = Paciþc Ten, S = Southeastern, T = Western Athletic, U = Sun Belt, W = Mountain West, X = Big Ten.

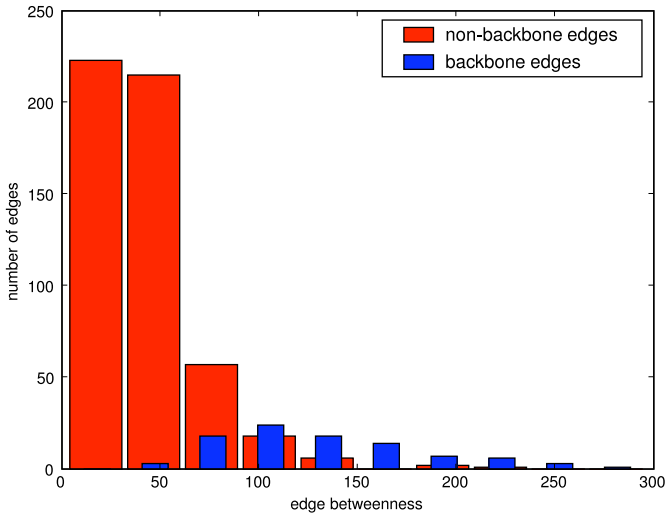


Figure B.2: Histogram of edge betweenness for non-backbone edges (red) and backbone edges (blue) for the NCAA 2005 football network. The mean (unnormalized) betweenness is 42.8 for non-backbone edges and 132.9 for backbone edges. Backbone and non-backbone histograms use the same bins; the frontmost bins have been narrowed for clarity.

Bibliography

[1] John P. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, January 2000. 1, 16

[2] Linton C. Freeman. Centrality in social networks: conceptual clariþcation. *Social Networks*, 1(3):215{239, 1978. 1, 16

[3] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci USA*, 99:7821{7826, 2002. viii, 1, 15, 16, 123

[4] A.-L. Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509{512, 1999. xii, 1, 8, 66, 75

[5] Soon-Hyung Yook, Hawoong Jeong, and A.-L. Barabasi. Modeling the internet's large-scale topology. *Proc Natl Acad Sci USA*, 99:13382{13386, 2002. 1

[6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824{827, October 2002. 1

[7] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabasi. Topological properties of metabolic networks. *Nature*, 407:651, 2000. xiii, 1, 68, 69

[8] J. Podani, Z. N. Oltvai, H. Jeong, B. Tombor, A.-L. Barabasi, and E. Szathmary. Comparable system-level organization of archaea and eukaryotes. *Nature Genetics*, 29:54, 2001. 1

[9] Leonhard Euler. Leonhard Euler and the Konigsberg Bridges. *Scientiþc American*, 189(1):66{70, 1953. 1

[10] D. J. Watts and S. H. Strogatz. Collective dynamics of `small-world' networks. *Nature*, 393:440{442, 1998. xiii, 6, 8, 68

[11] M. E. J. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89:208701, 2002. 6

[12] M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126, 2003. 6

[13] P. Erdős and A. Renyi. On random graphs. *Publ. Math.*, 6:290{297, 1959. xii, 8, 63

[14] Reka Albert, A.-L. Barabasi, and Hawoong Jeong. Mean-þeld theory for scale-free random networks. *Physica A*, 272:173{187, 1999. 8, 79, 90

[15] A. Bekessy, P. Bekessy, and J. Komlos. Asymptotic enumeration of regular matrices. *Stud. Sci. Math. Hungar.*, 7:343{353, 1972. 9

[16] E. A. Bender and E. R. Canþeld. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory A*, 24:296{307, 1978. 9

[17] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161{179, 1995. 9

[18] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability and Computing*, 7:295{305, 1998. xii, 9, 66

[19] S. Wasserman and K. Faust. *Social Network Analysis*. Cambrdige University Press, Cambridge, 1994. 10, 16

[20] G. W. Flake, S. R. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identiþcation of web communities. *IEEE Computer*, 35:66{71, 2002. 10

[21] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Deþning and identifying communities in networks. *Proc Natl Acad Sci USA*, 101:2658{2663, 2004. 10, 29, 37, 38, 43

[22] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner. Maximizing modularity is hard. arXiv:physics/0608255, 2006. 11

[23] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, 1993. viii, 11, 56

[24] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298{305, 1973. 12, 13, 106

[25] Miroslav Fiedler. *Special Matrices and their Applications in Numerical Mathematics*. Martinus NijhoÞ, Dordrecht, 1986. 12, 13

[26] Alex Pothen, Horst D. Simon, and Kan-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430{452, 1990. 12, 13

[27] Gustav KirchhoÞ. Uber die au osung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer strome gefuhrt wird. *Ann. Phys. Chem.*, 72:497{508, 1847. 12

[28] B. Bollobas. *Modern Graph Theory*. Springer, New York, 1998. 12, 13, 57

[29] Narsingh Deo and Hemant Balakrishnan. Bibliometric approach to community discovery. In *ACM-SE 43: Proceedings of the 43rd annual southeast regional conference*, pages 41{42, New York, NY, USA, 2005. ACM Press. 15

[30] M. E. J. Newman and M. Girvan. Mixing patterns and community structure in networks. In R. Pastor-Satorras, J. Rubi, and A. Diaz-Guilera, editors, *Statistical Mechanics of Complex Networks*. Springer, Berlin, 2003. 15, 106

[31] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004. 15, 18, 33

[32] W. W. Zachary. An information ow model for con ict and þssion in small groups. *Journal of Anthropological Research*, 33:452{473, 1977. viii, 16, 56

[33] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004. 18, 36, 106

[34] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004. 18, 19, 36

[35] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1275{1276, New York, NY, USA, 2007. ACM. 18

[36] M. E. J. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci USA*, 103:8577{8582, 2006. 18

[37] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, 2006. xiii, 18, 67

[38] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B*, 38:321{330, 2004. 18

[39] James P. Bagrow and E. M. Bollt. A local method for detecting communities. *Phys. Rev. E*, 72:046108, 2005. 19, 28, 106, 123

[40] L. da F. Costa. Hub-based community þnding, 2004. 20, 21

[41] Mason A. Porter, Peter J. Mucha, M. E. J. Newman, and A. J. Friend. Community structure in the united states house of representatives. *submitted to Social Networks*, 2006. 22, 24, 27, 106

[42] Aaron Clauset. Finding local community structure in networks. *Physical Review E*, 72:026132, 2005. 27, 106

[43] Feng Luo, James Z. Wang, and Eric Promislow. Exploring local community structures in large networks. In *Web Intelligence*, pages 233{239. IEEE Computer Society, 2006. 27

[44] Gary Flake, Steve Lawrence, and C. Lee Giles. EŽcient identiþcation of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150{160, Boston, MA, 2000. 28, 37, 45

[45] V. Farutin, K. Robison, E. Lightcap, Vlado Dancik, Alan Ruttenberg, Stanley Letovsky, and Joel Pradines. Edge-count probabilities for the identiþcation of local protein communities and their organization. *Proteins: Structure, Function, and Bioinformatics*, 62(3):800{818, September 2005. 28

[46] Jeorg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110, 2006. 28, 35, 38

[47] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814{818, 2005. 28

[48] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA*, 100(21):12123{12128, 2003. 28

[49] I. Derenyi, G. Palla, and T. Vicsek. Clique percolation in random networks. *Phys. Rev. Lett.*, 94:160202, 2005. 28

[50] B. Adamcsek, G. Palla, I. J. Farkas, I Derenyi, and T. Vicsek. Cþnder: locating cliques and overlapping modules in biological networks. *BIOINFORMATICS*, 22:1021{1023, 2006. 28

[51] S. MuÞ, F. Rao, and A. Ca isch. Local modularity measure for network clusterizations. *Phys. Rev. E*, 72:056107, 2006. 28

[52] Leon Danon, Albert D. Guilera, and Alex Arenas. The eþect of size heterogeneity on community identiþcation in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(11), 2006. 33, 35

[53] V. Batagelj and U. Brandes. EŽcient generation of large random networks. *Phys. Rev. E*, 71:036113, 2005. 34

[54] Aric Hagberg, Dan Schult, and Pieter Swart. Networkx. High productivity software for complex networks. 34

[55] S. Maslov, K. Sneppen, and A. Zaliznyak. Detection of topological patterns in complex networks: correlation proþle of the internet. *Physica A*, 333:529{540, 2004. 34

[56] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences, 2004. 34

[57] C. P. Massen and J. P. Doye. Identifying communities within energy landscapes. *Physical Review E*, 71(4):046101, April 2005. 34

[58] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral. Modularity from uctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, 2004. 35, 38

[59] Leon Danon, Albert D az-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identiþcation. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005. 36

[60] Alexander Strehl and Joydeep Ghosh. Cluster ensembles { a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)*, 3:583{617, 2002. 36

[61] Ana L.N. Fred and Anil K. Jain. Robust data clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, volume 2, page 128. IEEE Computer Society, 2003. 36

[62] Hernan D. Rozenfeld, J. E. Kirk, E. M. Bollt, and Daniel ben-Avraham. Statistics of cycles: How loopy is your network? *J. Phys. A*, 38:4589{4595, 2005. 49, 119

[63] Reuven Cohen and Shlomo Havlin. Scale-free networks are ultrasmall. *Phys. Rev. Lett.*, 90:058701, 2003. 50

[64] S. N. Dorogovtsev, Jose F. F. Mendes, and A. N. Samukhin. Metric structure of random networks. *Nuclear Physics B*, 653:307, 2003. 50, 70

[65] Reuven Cohen, Danny Dolev, Shlomo Havlin, Tomer Kalisky, Osnat Mokryn, and Yuval Shavitt. On the tomography of networks and multicast trees. 50, 107

[66] T. Kalisky, R. Cohen, Daniel ben-Avraham, and S. Halvin. Tomography and Stability of Complex Networks. In E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors, *LNP Vol. 650: Complex Networks*, pages 3{34, 2004. 50, 107

[67] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proc Natl Acad Sci USA*, 99:2566{2572, 2002. 53

[68] M. Alexander. Using the bipartite line graph to visualize 2-mode social networks. In *NAACSOS conference proceedings*, 2005. 53

[69] Petter Holme, Fredrik Liljeros, Christofer R. Edling, and Beom Jun Kim. On network bipartivity. *Phys. Rev. E*, 68:056107, 2003. 53

[70] E. Estrada and J. A. Rodr guez-Velazquez. Spectral measures of bipartivity in complex networks. *Phys. Rev. E*, 72(4):046105, October 2005. 53, 119

[71] Denes Kφnig. *Theorie der endlichen und unendlichen Graphen*. Akademische Verlagsgesellschaft, Leipzig, 1936. 53

[72] D. MacRae. Direct factor analysis of sociometric data. *Sociometry*, 23:360{371, 1960. 56

[73] Vladimir Batagelj and Andrej Mrvar. Pajek datasets, 2006. http://vlado.fmf.uni-lj.si/pub/networks/data/. xiii, 56, 68

[74] Juyong Park and M. E. J. Newman. A network-based ranking system for american college football. *J. Stat. Mech.*, P10014, 2005. xvii, 56, 125

[75] James P. Bagrow, E. M. Bollt, and L. da F. Costa. Network structure revealed by short cycles, 2006. 56, 119

[76] N.D. Martinez, B.A. Hawkins, H.A. Dawah, and B.P. Feifarek. EÞects of sampling eÞort on characterization of food-web structure. *Ecology*, 80:1044{1055, 1999. 56

[77] John Scott and Michael Hughes. *The anatomy of Scottish capital: Scottish companies and Scottish capital, 1900-1979*. Croom Helm, London, 1980. 56

[78] Collected from North American Transportation Atlas Data (NORTAD), 1997. xiii, 56, 68

[79] Roget's thesaurus of english words and phrases. Project Gutenberg, 2002. http://gutenberg.net/etext/22. 56

[80] ODLIS: Online dictionary of library and information science, 2002. http://vax.wcsu.edu/library/odlis.htm. 56

[81] James P. Bagrow, E. M. Bollt, J. D. Skufca, and Daniel ben-Avraham. Portraits of complex networks. *Europhysics Letters*, 81:68004, 2008. 57

[82] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proc. Natl. Acad. Sci. USA*, 97:11149, 2000. xii, 61

[83] P. L. Krapivsky and S. Redner. Organization of growing random networks. *Physical Review E*, 63:066123, 2001. xii, 66, 78

[84] Hernan D. Rozenfeld, Shlomo Havlin, and Daniel ben-Avraham. Fractal and transfractal recursive scale-free nets, 2006. xii, 66

[85] M. E. J. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Phys. Rev. E*, 60:7332, 1999. xiii, 67

[86] Douglas C. Schmidt and Larry E. DruÞel. A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. *J. ACM*, 23(3):433{445, 1976. 65

[87] P. Foggia, C. Sansone, and M. Vento. A performance comparison of þve algorithms for graph isomorphism. In *Third IAPR TC-15 Int'l Workshop Graph-Based Representations in Pattern Recognition*, pages 188{199, 2001. 65

[88] D. G. Corneil and C. C. Gotlieb. An eŽcient algorithm for graph isomorphism. *J. ACM*, 17(1):51{64, 1970. 65

[89] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42{65, 1982. 65

[90] Brendan D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45{87, 1981. 65

[91] Eric W. Weisstein. Dodecahedral graph. From MathWorld{A Wolfram Web Resource. http://mathworld.wolfram.com/DodecahedralGraph.html. xiii, 65, 70

[92] Eric W. Weisstein. Desargues graph. From MathWorld{A Wolfram Web Resource. http://mathworld.wolfram.com/DesarguesGraph.html. xiii, 65, 70

[93] A. E. Brouwer, A. M. Cohen, and A. Neumaier. *Distance-Regular Graphs*. Springer-Verlag, New York, 1989. xiv, 65, 71

[94] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, December 1998. 70

[95] Sidney Siegel. *Nonparametric statistics for the behavioral sciences*. International Student Edition - McGraw-Hill Series in Psychology, Tokyo: McGraw-Hill Kogakusha, 1956, 1956. 70

[96] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64:026118, 2001. 70

[97] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269{271, 1959. 72

[98] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264{323, 1999. 73

[99] P. Buneman. The recovery of trees from measurements of dissimilarity. In F. R. Hodson, D. G. Kendall, and P. Tautu, editors, *Mathematics in the archeological and historical sciences*, pages 387{395. Edinburgh University Press, Edinburgh, 1971. 73

[100] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406{425, July 1987. 73

[101] Abraham Akkerman. Fuzzy targeting of population niches in urban planning and the fractal dimension of demographic change. *Urban Studies*, 29(7):1093{1114, 1992. 75

[102] Pierre Frankhauser. The fractal approach. a new tool for the spatial analysis of urban agglomerations. *New methodological approaches in the social sciences, Population: an English Selection*, 10(1):205{240, 1998. 75

[103] Martin J. Beckmann. City hierarchies and the distribution of city size. *Economic Development and Cultural Change*, 6(3):243{248, April 1958. 75

[104] Peter S. Dodds, Roby Muhamad, and Duncan J. Watts. An experimental study of search in global social networks. *Science*, 301(5634):827{829, August 2003. 75

[105] Wolfgang Weidlich. Sociodynamics: a systematic approach to mathematical modeling in the social sciences. *J. Artiþcial Societies and Social Simulation*, 7(4), 2004. 75

[106] M. V. Simkin and V. P. Roychowdhury. Theory of aces: Fame by chance or merit? *Journal of Mathematical Sociology*, 30(1):33{42, 2006. 75

[107] James P. Bagrow, Hernan D. Rozenfeld, Erik M. Bollt, and Daniel ben-Avraham. How famous is a scientist? { famous to those who know us. *Europhysics Letters*, 67:511, 2004. 75

[108] James P. Bagrow and Daniel ben-Avraham. On the google-fame of scientists and other populations. In *Proc. Am. Inst. of Physics Conf.*, volume 779, pages 81{89, 2005. 75

[109] Haluk Bingol. Fame emerges as a result of small memory. *Phys. Rev. E*, 77:036118, 2008. 75

[110] Thomas M. Liggett. *Interacting Particle Systems*. Springer-Verlag, New York, 1985. 75

[111] K. Sznajd-Weron and J. Sznajd. Opinion evolution in closed community. *Intl. J. Mod. Phys. C*, 11:1157, 2000. 75

[112] R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47{97, 2002. 77

[113] M. E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46:323{351, 2005. 77

[114] James P. Bagrow, Jie Sun, and Daniel ben-Avraham. Phase transition in the rich-get-richer mechanism due to þnite-size eÞects. *J. Phys. A: Math. Theor.*, 41:185001, 2008. arXiv:0712.2220. 77

[115] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry, Third Edition (North-Holland Personal Library)*. North Holland, April 2007. 82

[116] Arnab Chatterjee and Bikas K. Chakrabarti. Kinetic exchange models for income and wealth distributions. *The European Physical Journal B*, 60:135{149, 2007. 90, 108

[117] B. K. Chakrabarti, A. Chakraborti, and A. Chatterjee, editors. *Econophysics and Sociophysics*. Wiley-VCH, 2006. 90, 108

[118] S. Jung, S. Kim, and B. Kahng. Geometric fractal growth model for scale-free networks. *Phys. Rev. E*, 65(5):056101, Apr 2002. 90

[119] Z. Zhang, L. Rong, and F. Comellas. High-dimensional random apollonian networks. *Physica A*, 364:610{618, 2006. 90

[120] F. Comellas, Hernan D. Rozenfeld, and Daniel ben-Avraham. Synchronous and asynchronous recursive random scale-free nets. *Phys. Rev. E*, 72, 2005. 90

[121] Tao Zhou, Jian-Guo Liu, Wen-Jie Bai, Guanrong Chen, and Bing-Hong Wang. Behaviors of susceptible-infected epidemics on scale-free networks with identical infectivity. *Phys. Rev. E*, 74:056109, 2006. 90

[122] A.-L. Barabasi. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003. 90

[123] Frigyes Karinthy. Chain-links. 90

[124] Jon Kleinberg. Navigation in a small world. *Nature*, 406(6798), August 2000. 90, 91, 108

[125] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000. 90

[126] Mickey R. Roberson and Daniel ben-Avraham. Kleinberg navigation in fractal small-world networks. *Phys. Rev. E*, 74(1):017101, 2006. 91, 94, 108

[127] Marina Meilă. Comparing clusterings | an information based distance. *J. Multivar. Anal.*, 98(5):873{895, 2007. xvi, 111, 114, 116, 118

[128] Brian Karrer, Elizaveta Levina, and M. E. J. Newman. Robustness of community structure in networks. *Phys. Rev. E*, in press, 2007. 111, 116, 118

[129] D.L. Wallace. Comment. *J. Amer. Statist. Assoc.*, 78(383):569{576, 1983. 113, 114

[130] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *J. Amer. Statist. Assoc.*, 78(383):553{569, 1983. 113, 114

[131] A. Ben-Hur, A. Elisseeþ, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacipc Symposium on Biocomputing*, pages 6{17, 2002. 114

[132] B. G. Mirkin. *Mathematical Classipcation and Clustering*. Kluwer Academic Press, Dordrecht, 1996. 114

[133] M. Meilǎ and D. Heckerman. An experimental comparison of modelbased clustering methods. *Mach. Learning*, 42(1/2):9{29, 2001. 115

[134] M. Meilǎ. Comparing clusterings | an axiomatic view. In S. Wrobel and L. De Raedt, editors, *Proceedings of the International Machine Learning Conference (ICML)*, New York, 2005. ACM Press. 115

[135] D. L. Steinley. Properties of the Hubert{Arabie adjusted Rand index. *Psychological Methods*, 9(3):386{396, 2004. 115

[136] B. Larsen and C. Aone. Fast and eƀective text mining using linear time document clustering. In *Proceedings of the Conference on Knowledge Discovery and Data Mining*, pages 16{22, 1999. 115

[137] S. van Dongen. Performance criteria for graph clustering and Markov cluster experiments,. Technical report, INS-R0012, Centrum voor Wiskunde en Informatica, 2000. 115

[138] Alex Arenas, Albert D az-Guilera, and Conrad J. Perez-Vicente. Synchronization reveals topological scales in complex networks. *Phys. Rev. Lett.*, 96:114102, 2006. 119

[139] L. da F. Costa. L-percolations of complex networks. *Phys. Rev. E*, 70:056106, 2004. 119

[140] T. Callaghan, M. A. Porter, and P. J. Mucha. Random walker ranking for NCAA division I-A football. *accepted in American Mathematical Monthly*, 2003. 123