Complex Network Models

D-INFK, ETH Zürich

Johannes Lengler

Stand: January 27, 2023

Contents

Chapter 1

Introduction

In this course, we will study some of the most important random network models. Formally, a (finite) random network model is a probability distribution over all graphs with vertex set $V = [n]$. Practically, it is a randomized procedure to generate a graph on n vertices. We will study these models asymptotically in the limit $n \to \infty$.

There are many reasons to study random network models, from existence proofs in graph theory to the design of efficient data structures and algorithms. But in this lecture, we focus on a different goal: to understand complex real-world networks better.

Complex real-world networks include *social networks*. In these networks, the nodes are usually people, or users. There are online social networks like the facebook graph, in which edges are friendship links in the facebook social network; or a collaboration network, where the nodes are researchers, and two researchers share an edge if they have co-authored a publication; or a mobile phone graph, in which two phones are linked by an edge if there was a phone call between the two phones in a particular month. Or the friendship network, in which two people are connected by an edge if they know each other on a first-name basis.

Another class of networks are *technological networks* like the internet graph, in which the nodes are given by routers and links are given by connecting cables[1]. Another example is the web graph, in which the nodes are the pages of the world wide web, and the edges are given by the hyperlinks. Many other examples of networks can be found in public repositories such

[1]The term *internet graph* is also used to refers only to the highest layer of the physical structure of the internet, where the nodes are the *autonomous systems* of the internet.

as `https://snap.stanford.edu/data/`.

There are several reasons why we want to study real-world networks through models. One is that some of the networks are not directly available. For example, the data of an online social network may not be available because the company does not want to share it. In some cases like for mobile phone data, there are data protection laws preventing them from making the network public. In other cases like the friendship network, the data does not exist at all in computer-readable form.[2] While it is in principle possible to query links in the network by asking the involved people, this allows at best to get tiny samples from the networks. Nevertheless, it is possible to run algorithms on the friendship network, and we will learn about one such algorithm in the lecture.

Another reason to study network models instead of the real networks is that we are able to create *variations* of the networks. For example, assume we want to understand how the efficiency of a routing protocol like the Border Gateway Protocol (BGP) scales if the internet graph grows over time. We have a few data points: we know the internet graph right now, and we know how the internet graph has looked like in the past. So we can run experiments on these real instances, find a trend, and extrapolate to the future. However, network models allow us a different approach: we can choose a model that generates networks like the internet graph, and then we can use the model to generate networks which are twice as large, and run experiments on those networks. Of course, for this approach it is crucial that the network model captures the properties of the internet graph that are relevant for routing and the BGP protocol. So we need to find a good model. This is precisely the goal of this course: to introduce the students to a collection of network models to choose from, and to discuss some network properties that may be relevant for such tasks. Beware that the course can and will not be exhaustive on either side: there are many more interesting network models, and many more important network properties than what we can cover in this course.

A third reason to study network models is rather fundamental: assume we want to understand a real-world phenomenon. Let us take the following empirical fact as an example. We have an unweighted network G and two

[2]This can also be difficult for technological networks. It is surprisingly unclear how big the web graph really is, since search engines only give us access to the parts that are accessible by web crawlers.

vertices u, v, and we want to find a shortest path from u to v. The textbook solution is to use breadth-first search (BFS), starting from u, until v is found. However, in practice, a different algorithm, called *bi-directional BFS* is more efficient: start two BFS in parallel, one from u and one from v. As soon as there is a vertex w that is discovered by both BFS, the path from u to v through w is a shortest path. Why is the second algorithm more efficient than the first one? Classic worst-case or average-case analysis does not give a difference between the two variants, so it must have to do with the networks on which the algorithm is run in practice. What aspects of the networks are important for the runtime? Is it relevant that there are nodes of rather high degree? Does it play a role that those networks tend to be clustered into communities? In such a situation, the best approach is the following: we develop *different levels of abstraction* of the phenomenon. So, we need different network models, some more basic, and some more complex. In the most basic ones, we will not see a difference between uni- and bi-directional BFS. But as we add more and more aspects of real-world networks to the model, at some point the bi-directional variant will start to have an advantage. By studying *when exactly* this happens, we get a much better understanding of why the bi-directional variant is superior in practice. We will come back to bi-directional search for several network models in this course.

This last aspect is a rather fundamental approach in the study of complex system. To understand such a system throroughly, we need to develop different levels of abstraction of the system, and choose the appropriate one to understand different aspects of the system – the level should be as simple as possible, but as complex as necessary.

Scope and limitations of the course

We take a mathematical approach in this lecture: we study random networks models with n vertices in the limit $n \to \infty$. This is a limitation when we want to apply our results to real-world networks. If we find out something for $n \to \infty$, does it also apply for a social network with 100 nodes? With 1000? $10,000$? Or do we need a million or a billion nodes? These questions are a natural second step after understanding the limit, because they are about *speed of convergence* in the limit. Thus, the methods to answer these questions are similar to the methods of obtaining the limit

behaviour in the first place. However, we will not discuss these questions in the course.

We will restrict ourselves to *sparse* network models in this lecture, i.e. network models in which the number m of edges is linear, $m = O(n)$. Equivalently, the average degree is constant, $O(1)$. Very many social and technological networks are sparse.[3] Moreover, we will only cover *undirected* networks. Not because directed networks are less interesting, but rather because they are even more complicated. Also, there is much more data that a network may be equipped with: edge weights (e.g., in a collaboration network the number of co-authored papers), time stamps (e.g., in mobile phone networks), or labels of vertices or edges. We will ignore all these possible extensions in this course.

Literature

The script and lecture will be self-contained, and it is not necessary to read further literature for passing the course. However, for students who want additional material, we want to highlight especially two excellent books on the topic.

- The book *Complex Network: Principles, Methods, Applications* by Latora, Nicosia and Russo gives a gentle introduction into complex networks for readers of all domains. It covers a wider range of topics, in particular more real-world network models and a larger collection of network properties than we can cover in this lecture. It is available through ETH library (vpn required).

  Compared to our course, the book is less math-heavy. It does work with some mathematical concepts, but the authors try to make them accessible for people without a mathematical background.

  The book comes with an excellent homepage, which contains all the network discussed in the book, and C-programs for all the network properties and algorithms that are discussed. This webpage is a great place for anyone who want to play around with some instances of real-world networks.

[3] Though beware that this there is no clear-cut definition for a fixed network of finite size. E.g., the collaboration network on high energy physics (based on arxiv papers) has $n \approx 12,000$ vertices and $m \approx 120,000$ edges. Whether this is sparse or not is interpretation, not math.

- The book *Random graphs and Complex Network, Volume 1* by van der Hofstad is *more* mathematical than our course. The reader can find all the nasty technical details which we skip in our course in this book, and many more. The book covers graph-theoretic aspects (e.g., component structures, typical distances, clustering coefficients) in much more depth and detail. It mostly does not cover the algorithmic aspects that we discuss in this lecture (e.g., routing, bidirectional search). The book (as well as a preliminary volume 2) is freely available at `https://www.win.tue.nl/~rhofstad/NotesRGCN.html`.

Both books do not cover the GIRG model (Geometric Inhomogeneous Random Graphs) that we will discuss in Chapter 5. The results in this chapter are from the frontline of research and are not yet covered by textbooks. For additional literature outside of this script, the only option is to look into the research articles that we link in this script. The article [BKL19] gives an overview of the model and its basic properties.

There are many other sources on complex network models. Students who want to experiment hands-on with real-world networks can find several collection of networks here, here and here, and tools for analyzing networks here and here.

1.1 Preliminaries

Probabilistic tools

Since we consider random network models, we need to analyse random processes. We expect students to be familiar with basics of probability theory like expectations, conditional probabilities, Binomial and Poisson distributions, *with high probability* statements, and so on. For full proofs, one would also frequently need probabilistic bounds like the inequalities of Markov, Chebyshev, Chernoff, Azuma-Hoeffding, and others. However, since we want to cover a variety of different models, we will mostly skip over these technical details. Students with background knowledge on stochastic processes (e.g., who have attended the lecture *Randomized Algorithms and Probabilistic Methods*) will be able to fill out most technical details by themselves if they wish to. Students without this background will have to believe some steps (or look them up in in the RandAlg script or a

textbook [AS16, MR95]), but will be able to follow the main computations and gain the intuition.

We will not give a list of all probabilistic bounds that we use. As mentioned, students who don't know them have to believe them or look them up. But since the Poisson distribution will occur quite often, we make one exception: the following Chernoff-like concentration bound holds for Poisson-distributed random variables [MU17].

Theorem 1.1. Let X be a Poisson random variable $X \sim \mathrm{Po}(\lambda)$. Then

- $\Pr(X > x) \leq \frac{(e\lambda)^x e^{-\lambda}}{x^x}$ for $x > \lambda$.
- $\Pr(X < x) \leq \frac{(e\lambda)^x e^{-\lambda}}{x^x}$ for $x < \lambda$.

In particular, for every $\varepsilon > 0$ there is $\eta < 1$ such that

$$\Pr(X > (1+\varepsilon)\lambda) \leq \eta^\lambda \qquad \text{and} \qquad \Pr(X < (1-\varepsilon)\lambda) \leq \eta^\lambda.$$

Notation and conventions

Basic objects like graphs often have two names, one coming from graph theory and one from network theory. We will use the terms "graph/network", "vertex/node" and "edge/link" interchangeably in this course.

Throughout the course, we will always consider a graph $G = (V, E)$ with vertex set V and edge set E. We use the convention $n = |V|$, $m = |E|$. Unless otherwise mentioned, G is undirected, simple and finite. G is usually obtained from a random graph model where the set of vertices is fixed, but the set of edges is random. We are interested in the behaviour of G for $n \to \infty$. The Landau notation $O(.), o(.), \Theta(.), \Omega(.), \omega(.)$ is always with respect to the limit $n \to \infty$ unless otherwise stated. If $\mathcal{D}$ is a probability distribution, we will slightly abuse notation by writing $\Pr[\mathcal{D} = x]$ as abbreviation for "$\Pr[X = x]$ for a random variable X with distribution $\mathcal{D}$".

Further notation

- $\mathbb{N} := \{1, 2, \ldots, \}$.
- $[k] := \{1, 2, \ldots, k\}$.

- i.i.d. = independently identically distributed.
- whp = with high probability = with probability $1-o(1)$ as $n \to \infty$.[4]
- almost surely = with probability one.
- C_1 = largest connected component, C_2 = second largest component, etc.
- $G[V']$ = induced subgraph with vertex set V'.
- $\deg(v)$ = degree of vertex v
- $E(S_1, S_2) = \{\{v_1, v_2\} \in E \mid v_1 \in S_1, v_2 \in S_2\}$.
- $u \sim v$ = "u is adjacent to v".

[4]This term is sometimes used differently in the literature and may mean "with probability $n^{-\omega(1)}$" or "with probability $n^{-\Omega(1)}$". In this case, probability $1-o(1)$ is also called "asymptotically almost surely" or "aas".

Chapter 2

Erdős-Rényi random graphs

In this section, we will discuss the most basic random network model, the Erdős-Rényi random graph $G_{n,p}$. In particular, we will study three aspects of the component structure: the existence of a giant component, the asymptotic absence of medium-size components and the number and structure of small components.

We will use $G_{n,p}$ to introduce two perspectives on a network: the local and the global perspective. Even though the Erdős-Rényi model is not a good model for real-world networks, the techniques developed here are useful to understand more complex models.

Definition 2.1. The *Erdős-Rényi random graph* $G_{n,p}$ is the random graph on n vertices in which every edge exists with probability p, independently for all edges.

The expected degree of a vertex in $G_{n,p}$ is $(n-1)p$. Since we focus on sparse models, we will assume that $p = \mu/n$ for a constant $\mu > 0$. Therefore, the expected degree is $\mu\frac{n-1}{n} \to \mu$ for $n \to \infty$. Moreover, the degree of a vertex is binomially distributed, $\deg(v) \sim \mathrm{Bin}(n-1, \mu/n)$. It is a basic fact about the binomial distribution that this distribution converge to a Poisson distribution $\mathrm{Po}(\mu)$ for $n \to \infty$. Therefore, in the limit the degrees follow a Poisson distribution with parameter μ. Hence, for all $k \geq 0$,

$$\Pr[\deg(v) = k] \overset{n\to\infty}{\to} \Pr[\mathrm{Po}(\mu) = k] = \frac{e^{-\mu}\mu^k}{k!} \text{ in } G_{n,p} \text{ with } p = \frac{\mu}{n}. \quad (2.1)$$

2.1 The local perspective and the Galton-Watson branching process

We want to know how many vertices of $G_{n,p}$ with $p = \mu/n$ are contained in connected components of size $s = 1, 2, \ldots$, or in larger components. The model $G_{n,p}$ is so symmetric that we could compute this directly. For example, the expected number of components of size $s = 2$ is

$$\begin{aligned}
&\sum_{u \neq v \in V} \Pr[\{u, v\} \text{ is a component of size 2}] \\
&\quad = \sum_{u \neq v \in V} \Pr(uv \in E, \text{ none of the other } 2(n-2) \text{ incident edges exist}) \\
&\quad = \binom{n}{2} \frac{\mu}{n} \left(1 - \frac{\mu}{n}\right)^{2n-4} = (1 - o(1)) \frac{\mu e^{-2\mu}}{2} \cdot n.
\end{aligned}$$

Each component of size $s = 2$ contains two vertices, so the expected number of vertices in such components is $\approx \mu e^{-2\mu} n$. It is an easy exercise to show that the true value is concentrated around its expectation. The calculation becomes slightly more complicated for components of larger sizes, but can be done.

However, we want to use a different approach, which generalizes better to more complex situations. To this end, we fix a vertex $v \in V$ and compute the probability that v is in a component of size s. We explore the graph starting from v in a breadth-first search (BFS). So, we take the local perspective of vertex v.

In the first step, we uncover the number of neighbours of v. This is distributed as $\mathrm{Bin}(n-1, p) \to \mathrm{Po}(\mu)$, since we assumed $p = \mu/n$. Assume v has $X_1 = o(n)$ neighbours. In the next step, for every neighbour w we reveal how many edges go out from w. One edge clearly goes to the parent v, and we ignore this edge at this point. For the other $n-2$ potential edges, each of them has the same probability p to be present, so the number of outgoing edges is distributed as $\mathrm{Bin}(n-2, p) \to \mathrm{Po}(\mu)$. So the number of new edges that we find per neighbour w follows essentially again the same distribution.

Of course, not every new edge necessarily leads to a new vertex. It could also lead to a vertex that we have already found before. However, if we process the vertices one by one, as long as the number of vertices that we have found is $x = o(n)$, the number of *new* vertices that we find from

w is $\mathrm{Bin}(n-x,p) \to \mathrm{Po}(\mu)$. So, this effect only starts to matter when we have discovered a linear fraction of the graph.

Summarizing, exploring the graph resembles the process of growing a tree with parent v, where each node has a random number of children, distributed as $\mathrm{Po}(\mu)$. This process is known as *Galton-Watson branching process*.

Definition 2.2. Let $\mathcal{D}$ be a probability distribution over $\mathbb{N}_0$. A *Galton-Watson branching process* with offspring distribution $\mathcal{D}$ is the process for generating a random rooted tree in which the number of children of each vertex (including the root) follows independently a $\mathrm{Po}(\mu)$-distribution.

The resulting random tree T is called *Galton-Watson tree*. It can be either finite or infinite. The *size* $|T|$ of the Galton-Watson tree is the number of nodes, and can be finite or infinite. We denote $p_s := \Pr[|T| = s]$, where $s \in \mathbb{N} \cup \{\infty\}$.

We say that the Galton-Watson branching process *survives* if the resulting tree is infinite, and that it *becomes extinct* otherwise. We also call p_∞ the *survival probability* and $1-p_\infty$ the *extinction probability*.

We have informally argued that Galton-Watson trees with Poisson offspring distribution are related to how an Erdős-Rényi graph looks like from the perspective of a single vertex. Indeed, the following theorem makes this precise.

Theorem 2.3 (Small components in Erdős-Rényi graphs.). Let $G \sim G_{n,p}$ be an Erdős-Rényi graph with $p = \mu/n$, and let T be a Galton-Watson tree with offspring distribution $\mathrm{Po}(\mu)$. For $s \in \mathbb{N}$, let n_s be the number of vertices of G in components of size s. Then for all $s \in \mathbb{N}$, almost surely,

$$\lim_{n\to\infty} \frac{n_s}{n} = p_s = \Pr[|T| = s].$$

Proof. We will only show that $\mathbb{E}[n_s/n] \to p_s$. For the theorem, one also needs to show concentration, which can be obtained by probabilistic tools like Azuma's inequality.

Fix $s \in \mathbb{N}$ and $v \in V$. Let $\mathcal{E}(v)$ be the event that v is in a component of size s. By symmetry, we have $\mathbb{E}[n_s] = \sum_{u \in V} \mathcal{E}(v) = n\Pr[\mathcal{E}(v)]$. Thus we need to show that $\Pr[\mathcal{E}(v)] \to p_s$ for $n \to \infty$. Let $\varepsilon > 0$.

Consider a BFS starting in v. We run the BFS until either we have explored the component, or we have uncovered more than s vertices. When we uncover the neighbours of a vertex u, then the number of *new* vertices that we find from u follows the binomial distribution $\mathrm{Bin}(n-x, p)$, where x is the number of neighbours that we have uncovered before that point. By definition of the process, we have $1 \leq x \leq s$. Therefore, we can sandwich the distribution $\mathrm{Bin}(n-x, p)$ between the two distributions $\mathcal{D}_1 := \mathrm{Bin}(n-s, p)$ and $\mathcal{D}_2 := \mathrm{Bin}(n-1, p)$.[1] Thus, we can sandwich the BFS tree between two Galton-Watson branching processes T_1 and T_2 with respective offspring distributions $\mathcal{D}_1$ and $\mathcal{D}_2$. Note that this coupling only works while the BFS still runs (i.e., only until we have explored the whole component or uncovered more than s vertices), but this is fine for our purposes.

The distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ both converge to $\mathcal{D} := \mathrm{Po}(\mu)$ for $n \to \infty$. In particular, let $X_1 \sim \mathcal{D}_1$, $X_2 \sim \mathcal{D}_2$ and $X \sim \mathcal{D}$. If n is large enough, then $|\Pr[X_1 = i] - \Pr[X = i]| \leq \frac{\varepsilon}{2s}$ for all $i \in \mathbb{N}_0$, and similarly for X_2.[2] This allows us to couple X_1 and X_2 to X as follows. We first draw X. Then we flip a coin with $\Pr[\text{tail}] = \frac{\varepsilon}{2s}$. If the coin comes up head, then we set $X_1 := X$. Otherwise we set X_1 to whatever value we need to obtain the correct distribution of X_1. We also perform a second coin flip with $\Pr[\text{tail}] = \frac{\varepsilon}{2s}$ to couple X_2 to X.

Now consider the Galton-Watson trees T_1, T_2, and a third Galton-Watson tree T with offspring distribution $\mathcal{D}$. We use the above coupling for the first s nodes of the branching processes (or until the processes die out): we flip two coins for each node, and if they both come up head then all three processes have the same number of children for this node. By a union bound, the probability that we ever see tail in these s steps is at most $2s \cdot \frac{\varepsilon}{2s} = \varepsilon$. If we never see tail, then the three Galton-Watson-trees are identical until this point. Since the BFS tree is sandwiched between the

[1] This is called *domination*. $\mathrm{Bin}(n-x, p)$ dominates $\mathrm{Bin}(n-s, p)$ means that for two random variables $X \sim \mathrm{Bin}(n-x, p)$ and $X' \sim \mathrm{Bin}(n-s, p)$ we have $\Pr[X \geq i] \geq \Pr[X' \geq i]$ for all $i \in \mathbb{N}_0$.

[2] It is obvious that we can achieve this for a constant number of values of i, for example for all $0 \leq i \leq s$. This would also suffice for our purposes, since any larger number of offspring terminates the BFS immediately. The stronger version for all $i \in \mathbb{N}_0$ needs a proof, but the proof is not hard.

trees T_1 and T_2, it must then also be identical. Thus, with probability at least $1 - \varepsilon$, all four processes either terminate with the same number of vertices, or all four processes result in more than s nodes. Since one of them (the process T) has probability p_s to terminate with exactly s vertices, the other processes have probability $p_s \pm \varepsilon$ to terminate with s vertices, if n is large enough. For the BFS tree, this event was called $\mathcal{E}(v)$, and hence $p_s - \varepsilon \leq \Pr[\mathcal{E}(v)] \leq p_s + \varepsilon$ for n large enough. This implies $\Pr[\mathcal{E}(v)] \to p_s$ for $n \to \infty$ and concludes the proof. □

Remark 2.1 (Structure of small components). In fact, the connection between Galton-Watson trees and components in Erdős-Rényi graphs goes much further than Theorem 2.3. This theorem tells us that the *number* of components is given by Galton-Watson trees. But the *structure* of components is also given by such trees. First of all, it is easy to see that small components are unlikely to contain any cycles, so they are trees. This is because starting from a fixed vertex v, it is very unlikely to discover the same vertex twice. By the "birthday paradox", we only find the same vertex twice when we uncover $\Theta(\sqrt{n})$ vertices.

Now consider components of size 4. We know that they are trees, but there are two different types of trees on 4 vertices: the path P_3 of length 3 (three edges and four vertices), and the star S_4 of size 4. How often do they occur? Exactly as often as a Galton-Watson process produces P_3 and S_4, respectively. These probabilities are not hard to compute. Denoting $\mu = pn$ as before, the probability that a Galton-Watson process gives a star S_4 where the root v has degree 3 is

$$\begin{aligned}\Pr[S_4 \text{ with } \deg(\text{root}) = 3] &= \Pr[\mathrm{Po}(\mu) = 3] \cdot (\Pr[\mathrm{Po}(\mu) = 0])^3 \\ &= \frac{e^{-\mu}\mu^3}{3!} \cdot \left(\frac{e^{-\mu}\mu^0}{0!}\right)^3 = \frac{e^{-4\mu}\mu^3}{6}\end{aligned}$$

and the probability to obtain a path P_3 where the root has degree 1 is

$$\begin{aligned}\Pr[P_3 \text{ with } \deg(\text{root}) = 1] &= (\Pr[\mathrm{Po}(\mu) = 1])^3 \cdot \Pr[\mathrm{Po}(\mu) = 0] \\ &= \left(\frac{e^{-\mu}\mu^1}{1!}\right)^3 \cdot \frac{e^{-\mu}\mu^0}{0!} = e^{-4\mu}\mu^3.\end{aligned}$$

Note that by this calculation we count every component S_4 exactly once (because there is only one vertex of degree 3), while we count every P_3 twice. On the other hand, every component has four vertices. Therefore, in $G_{n,p}$ the fraction of vertices in S_4-components is $\frac{2}{3}e^{-4\mu}\mu^3$, and the fraction of vertices in P_3-components is $2e^{-4\mu}\mu^3$. Thus there are three times as many vertices in S_3-components than in P_4-components.[3]

Now that we have established Theorem 2.3, we can derive the number of small (constant-size) components in Erdős-Rényi graphs by analyzing

[3] General rules can be obtained here. It can be shown that the fraction of vertices is inversely proprtional to the number of automorphisms of the structure.

the properties of branching processes. First we will estimate how the probability p_s scales with s, and in particular, when a Galton-Watson process has extinction probability of one.

Theorem 2.4. Let T be a Galton-Watson tree with offspring distribution $\mathcal{D}$, and let μ be the mean of $\mathcal{D}$.

(a) If $\mu < 1$, then $\Pr[|T| = \infty] = 0$.

(b) If $\mu > 1$, then $\Pr[|T| = \infty] > 0$.

Moreover, if $\mathcal{D} = \mathrm{Po}(\mu)$ for $\mu \neq 1$, then there is $\eta < 1$ such that $\Pr[|T| = s] \leq \eta^s$ for all $s \in \mathbb{N}$.

Proof. Let $Z_1, Z_2, Z_3, \ldots$ be an infinite sequence of i.i.d. copies of a random variables with distribution $\mathcal{D}$. We can construct T as follows. We enumerate all vertices of T in the order in which we create them. The root has number 1. Then we use Z_i to determine the number of children of node i. So the root has Z_1 children, its first child (if it exists) has Z_2 children, and so on. Note that it may happen that we don't use all the Z_i. For example, if $Z_1 = 0$, then the root has no children, and we never use Z_2.

In general, assume that the Galton-Watson tree has size at least s. Then it stops with size exactly $|T| = s$ if and only if $\sum_{i=1}^{s} Z_i = s - 1$: then after inspection the first s nodes, we have altogether created s nodes (the root and $s-1$ offspring), so there is no $(s+1)$-st node to continue with. Therefore, the condition $\sum_{i=1}^{s} Z_i = s - 1$ implies $|T| \leq s$. But note that it does not imply $|T| = s$; the process may already have stopped at an earlier stage. Conversely, the condition $\sum_{i=1}^{s} Z_i > s - 1$ is necessary, but not sufficient for $|T| > s$.

Now let us assume $\mu < 1$. The *strong law of large numbers* states that whp the average of s i.i.d. random variables converges to the mean of the distribution for $s \to \infty$, as long as this distribution is positive and has finite mean. Hence,

$$\Pr[|T| > s] \leq \Pr\Big[\sum_{i=1}^{s} Z_i > s - 1\Big] = \Pr\Big[\sum_{i=1}^{s} Z_i \geq s\Big] = \Pr\Big[\frac{1}{s}\sum_{i=1}^{s} Z_i \geq 1\Big] \stackrel{s\to\infty}{\longrightarrow} 0, \tag{2.2}$$

where the last step holds by the law of large numbers. This shows in

particular that $\Pr[|T| = \infty] = 0$ for $\mu < 1$. If the Z_i are Poisson distributed $\mathrm{Po}(\mu)$, then the sum $\sum_{i=1}^{s} Z_i$ is also Poisson distributed with mean $s\mu$, and is thus concentrated with exponential tail bounds (Theorem 1.1). Hence we can strengthen the last step of (2.2) by the exponential bound $\leq \eta^s$.

On the other hand, assume $\mu > 1$. We would like to use the strong law of large numbers again, but this only holds if $\mu < \infty$, while we also allow $\mu = \infty$. So in the latter case we use a little trick: we truncate the offspring distribution (and thus the Galton-Watson tree) at some value k, so we set $Z_i' := \min\{Z_i, k\}$. If we choose k large enough, then the expectation μ' of Z_i' is still larger than one, but it is also finite by construction. If the truncated Galton-Watson tree is infinite, then so is the original tree. So, we only need to show that the truncated Galton-Watson tree is infinite with positive probability. It thus suffices to prove the theorem in the case $\mu < \infty$.

Let $\mathcal{E}(s_0)$ be the event "$\forall s \geq s_0 : \frac{1}{s}\sum_{i=1}^{s} Z_i \geq 1$". By the strong law of large numbers, almost surely $\frac{1}{s}\sum_{i=1}^{s} Z_i \to \mu$. Hence, there exists s_0 such that $\Pr[\mathcal{E}(s_0)] \geq 1/2$. On the other hand, consider the event $\mathcal{F}(s_0)$ that the Galton-Watson tree reaches size s_0. Then $\Pr[\mathcal{F}(s_0)] \geq \Pr[\forall i \in \{1, \dots, s_0\} : Z_i \geq 1] > \Pr[Z_1 > 0]^s > 0$. Since the event $\mathcal{F}(s_0)$ only gives lower bounds on the Z_i, it cannot decrease the probability of $\mathcal{E}(s_0)$. Thus

$$\Pr[|T| = \infty] \geq \Pr[\mathcal{E}(s_0) \wedge \mathcal{F}(s_0)] \geq \tfrac{1}{2}\Pr[\mathcal{F}(s_0)] > 0. \tag{2.3}$$

It remains to show the tail bound in s for $\mu > 1$. Recall that a necessary condition for $|T| = s$ is $\sum_{i=1}^{s} Z_i = s - 1$. The left hand side is a sum of independent Poisson distributed random variables, so it also follows a Poisson distribution with mean $s\mu$. Hence,

$$\Pr[|T| = s] \leq \Pr[\sum_{i=1}^{s} Z_i = s - 1] \leq \Pr[\sum_{i=1}^{s} Z_i < s] \leq \eta^s, \tag{2.4}$$

where the last step again follows from concentration of the Poisson distribution, Theorem 1.1. This concludes the proof. □

Remark 2.2. It can be shown that $\Pr[|T| = \infty] = 0$ also holds for $\mu = 1$, except for the trivial case that the distribution is constant with $\Pr[Z_i = 1] = 1$. However, the tail bounds in s are no longer true, not even in the case $\mathcal{D} = \mathrm{Po}(1)$. We will generally ignore such threshold cases in this lecture. We are interested in models for real-world networks. If some properties of the model only hold for a parameter μ which is exactly on the threshold, then this means that they are not robust against tiny parameter changes. It is then hard to argue that the model is relevant.

There is an important implication of the different cases (a) and (b) of Theorem 2.4 for Erdős-Rényi graphs. Obviously, if we consider any *finite* n and sum up the number of vertices in components of size s, then we obtain $\sum_{s=1}^{\infty} n_s = n$ because we count every vertex exactly once. Dividing by n gives the *fraction* of vertices in components of size s,

$$\sum_{s=1}^{\infty} \frac{n_s}{n} = 1. \tag{2.5}$$

What happens if we take the limit for $n \to \infty$ of each summand? (Recall from your analysis classes that there is no guarantee that equality still holds afterwards, since we are exchanging the limit with the infinite sum here.) By Theorem 2.3, the left hand side then gives $\sum_{s=1}^{\infty} p_s$. In the subcritical case $\mu < 1$, the Galton-Watson tree is finite almost surely, and thus $\sum_{s=1}^{\infty} p_s = 1$. Hence, we may indeed exchange limits if we take $n \to \infty$ in (2.5). However, in the supercritical case $\mu > 1$, we have $p_\infty = \Pr[|T| = \infty] > 0$, and thus $\sum_{s=1}^{\infty} p_s = 1 - p_\infty < 1$. So, we may *not* exchange limits in (2.5) in the supercritical case!

What happens here? Where do the vertices disappear to when we take the limit for $n \to \infty$? The answer is that p_s is asymptotically the fraction of vertices in components of size s, but we only count vertices in components *of constant size* here. If there is a component of size $n/2$, then for every n it will be counted by a different s, and thus there is no s for which it will leave a trace in the limit.

This is exactly what actually happens. As we will show, there is a single component C_1 of linear size, the *giant component* of the graph. Intuitively, this contains all vertices for which the Galton-Watson tree has infinite size. (Recall that the Galton-Watson process stops to be an good description of a BFS exploration when it reaches size $\Omega(n)$, because then a relevant fraction of discovered vertices in the graph are already known.) In fact, what is true, but what we will not show, is that

$$\lim_{n\to\infty} \frac{|C_1|}{n} = p_\infty = 1 - \sum_{s=1}^{\infty} p_s. \tag{2.6}$$

Informally, this says that all vertices (except for an asymptotically negligible number) are either in the giant component or in components of constant size. Indeed, it can be shown that for every function $f(n)$ with $\lim_{n\to\infty} f(n) = \infty$, whp $\lim_{n\to\infty} \sum_{s=f(n)}^{\infty} n_s = |C_1| + o(n)$. In other words,

there are only $o(n)$ vertices outside of the giant component and outside of constant-size components.

Equation (2.6) raises the question how we can compute p_∞. Of course, we could compute the first few of $p_1, p_2, \ldots$, and thus approximate p_∞. But there is a more elegant way, as the following proposition shows.

Proposition 2.5. Consider a Galton-Watson process with offspring distribution $\mathcal{D}$, and let μ be the mean of $\mathcal{D}$. Then the extinction probability $p_{\text{ext}} := 1 - p_\infty$ satisfies the following equation.

$$p_{\text{ext}} = \sum_{i=0}^{\infty} \Pr[\mathcal{D} = i] p_{\text{ext}}^i. \tag{2.7}$$

If $\mathcal{D} = \text{Po}(\mu)$ for $\mu > 1$, then the survival probability is the unique positive solution of

$$1 - p_\infty = e^{-\mu p_\infty}. \tag{2.8}$$

Proof. Equation (2.7) follows directly from the law of total probability, where we discriminate between the number i of children of the root. The Galton-Watson tree is finite if and only if the subtrees below all children of the root are finite. Since all children are the root of independent Galton Watson trees, the probability that all of them are finite is p_{ext}^i. Hence,

$$p_{\text{ext}} = \Pr[\text{GW tree becomes extinct}] = \sum_{i=0}^{\infty} \Pr[\text{root has } i \text{ children}] p_{\text{ext}}^i,$$

which implies (2.7).

For (2.8), we could simply plug $\Pr[\mathcal{D} = i] = e^{-\mu}\mu^i/i!$ into (2.7) and simplify, but there is also a way without calculation. The number of children of the root is $\text{Po}(\mu)$-distributed. Each child independently has probability p_∞ to be the root of an infinite tree, and we call such a child *surviving*. Hence, we may obtain the number of surviving children by first drawing the number of children, and then flipping a coin for each of them, keeping it with probability p_∞. This process yields again a Poisson distribution $\text{Po}(p_\infty \mu)$.[4] The Galton-Watson tree becomes extinct if and only if it has

[4] If this is not clear to you, remember that the Poission distribution $\text{Po}(\mu)$ is approx-

no surviving children. Hence, the survival probability satisfies

$$1 - p_\infty = \Pr(\text{no surviving child}) = \Pr(\mathrm{Po}(p_\infty \mu) = 0) = e^{-p_\infty \mu}.$$

We already know by Theorem 2.2 that $p_\infty > 0$ for $\mu > 1$, and it is easy to see that the function $f(x) = 1 - x - e^{-\mu x}$ has a unique positive solution. (E.g., because f is concave, starts at $f(0) = 0$ with a positive slope $f'(0) = \mu - 1 > 0$, and becomes negative since $f(1) < 0$.) Therefore, p_∞ must be the unique root of the equation $1 - x = e^{-\mu x}$. □

Note that the solution of (2.7) is not necessarily unique. In general, we can only say that the extinction probability is one of the solutions, but it may be non-trivial to decide which one it is. Even in the special case $\mathcal{D} = \mathrm{Po}(\mu)$ there are really *two* solutions of $1 - x = e^{-\mu x}$, namely $x = 0$ and some positive x. In this case, we could rule out the solution $x = 0$ by Theorem 2.2.

2.2 The global perspective: sprinkling

Surprisingly, Theorem 2.4 tells us that for constant s, components of size s are quite rare as s gets large. This holds both for the subcritical case $\mu < 1$ and for the supercritical case $\mu > 1$. Indeed, this is a property that holds in many network models, though not always as extreme as for Erdős-Rényi graphs. It also holds in many real-world networks. The Facebook network with $n \approx 10^9$ nodes has many isolated nodes, but its second-largest component is of size $\approx 2000 \ll n$ [UKBM11].

Recall the proof idea of Theorem 2.3: a BFS exploration from a fixed vertex v resembles closely a Galton-Watson branching process. If the Erdős-Rényi graph has edge probability $p = \mu/n$, and the BFS tree has already size x, then the number of new neighbours that we find from an unexplored vertex v is $\mathrm{Bin}(n - x, p)$-distributed, and this is almost the same as $\mathrm{Po}(\mu)$. This connection can be pushed further. In Theorem 2.3 we only considered components of constant size, but the connection stays tight while $x = o(n)$.

imated by the binomial distribution $\mathrm{Bin}(n, \mu/n)$ for large n. The binomial distribution is obtained by flipping n coins with probability μ/n, and counting the number of heads. Now for each head, we flip another coin and keep it with probability p_∞. Then the whole procedure is the same as flipping n coins, each with probability $p_\infty \mu/n$. Hence, the result is $\mathrm{Bin}(n, p_\infty \mu/n)$-distributed, which approximates $\mathrm{Po}(p_\infty \mu)$.

In particular, we have shown in Theorem 2.4 that $p_s \leq \eta^s$ holds for all $\mu \neq 1$. We have only shown this statement for constant s, but let us just pretend for a moment that we were allowed to plug in larger values of s. Then plugging $s = C \log n$, we obtain $p_s \leq \eta^{C \log n} = n^{-C|\log \eta|}$. (Mind that $\eta < 1$ means $\log \eta < 0$.) If C is large enough, the number of vertices in components of size $C \log n$ or larger should be $n \cdot n^{-C|\log \eta|} = o(1)$. In other words, we should not expect to see any vertices at all in components of size between $C \log n$ and $o(n)$. Indeed, this can be formally proven.

Lemma 2.6 (Medium-size components in Erdős-Rényi graphs). Let $G \sim G_{n,p}$ be an Erdős-Rényi graph with $p = \mu/n$. Then there exist $C, \delta > 0$ such that with high probability the following holds.

1. If $\mu < 1$, then G does not contain any component of size $s \geq C \log n$.
2. If $\mu > 1$, then G does not contain any component of size $s \in [C \log n, \delta n]$.

Proof. We omit the proof. It follows the same line of argument as Theorems 2.3 and 2.4, except that the coupling in Theorem 2.3 needs to be made a bit more carefully. As key insight for the case $\mu > 1$, note that we want to couple the BFS in G up to size δn with a Galton-Watson tree with offspring distribution $\mathrm{Bin}(n - \delta n, p)$. If δ is sufficiently small then this distribution still has expectation $(1 - \delta)np = (1 - \delta)\mu > 1$, so the Galton-Watson tree will be very unlikey to have size s for any large s. □

Lemma 2.6 tells us that there are no medium-size components. Moreover, we understand very well the constant-size components. But what about linear-size components? Could there be several of them? The answer is No, there is never more than a single linear-size component. The reason is that any configuration with several large component is *unstable*. Imagine a graph in which two components C_1, C_2 have size at least δn, and imagine that an *additional edge* is randomly inserted into the graph. Then the new edge has a constant probability of at least δ^2 to have the first endpoint in C_1 and the second endpoint in C_2. If this happens, then the two components are merged into a single, bigger component. If we not just insert a single edge, but many edges, then it is very unlikely that C_1

and C_2 survive this. Hence we obtain the following theorem.

Theorem 2.7 (Uniqueness of the giant). Let $G \sim G_{n,p}$ be an Erdős-Rényi graph with $p = \mu/n$ for some constant $\mu > 1$, and let $\delta > 0$ be sufficiently small. Then with high probability G contains a unique component with more than δn vertices. This is called the *giant component*.

Proof. We construct G in two steps. In the first step, we insert every edge with probability $p_1 := p - n^{-3/2}$. In a second step, we insert every edge with probability p_2 such that $(1-p_1)(1-p_2) = 1-p$, or equivalently

$$p_2 := \frac{p-p_1}{1-p_1} = (1+o(1))n^{-3/2}. \tag{2.9}$$

Note that this gives exactly the correct probability for the Erdős-Rényi graph, because the probability that an edge is *not* present after both rounds is $(1-p_1)(1-p_2) = 1-p$, independently for all edges. The second round is also called *sprinkling*.

Now we consider the graph G_1 after the first round. Since there are vertices for which the corresponding Galton-Watson process has infinite size, there must be super-constant components,and it is easy to see that there must also be components larger than $C \log n$, where C is the constant from Lemma 2.6. (Because the coupling from Theorem 2.3 still works after the BFS has discovered $O(\log n)$ vertices.) By the same lemma, there are no medium-sized components in G_1.[5] Therefore, there must be components of size at least δn in G_1.

In the second round, we add at most $O(n^2 p_2) = O(\sqrt{n})$ edges in expectation, and also with high probability by the Chernoff bound. In particular, we can not not create a *new* linear-size component, since r edges can join at most $r+1$ components. Since all non-linear components have size $O(\log n)$ in G_1, the additional edges could create at most a component of size $O(\sqrt{n}\log n)$. Thus every linear-size component in G must contain a linear-size component from G_1.

[5] We cheat very slightly here, since Lemma 2.6 a constant μ, while the value for G_1 is $\mu_1 := \mu - n^{-1/2}$, which depends on n. However, μ_1 stays bounded away from one, and is also bounded from above, and Lemma 2.6 also holds under this weaker condition.

Assume that there are two linear-size components C_1 and C_2 in G_1. Since $|C_1|, |C_2| \geq \delta n$, there are at least $\delta^2 n^2$ pairs (v_1, v_2) with $v_1 \in C_1$ and $v_2 \in C_2$. The probability that none of these pairs is hit by an edge is

$$\Pr[\text{no pair hit}] \leq (1-p_2)^{\delta^2 n^2} \leq e^{-p_2 \delta^2 n^2} = e^{-\Omega(\sqrt{n})}. \tag{2.10}$$

So whp at least one such pair is hit, and the components C_1 and C_2 are joined in G.

Let k be the number of linear-size components in G_1, and call them $C_1, \ldots, C_k$. Then $k \leq 1/\delta$, since $|C_i| \geq \delta n$ for all i, and since the C_i are disjoint. By a union bound and (2.10), the probability that there is a $i \in \{2, \ldots, k\}$ that is not joined with C_1 in the second round is $O((k-1)e^{-\Omega(\sqrt{n})}) = o(1)$. Hence, with high probability all components are joined with C_1 in the second round, and G only has a single giant component. □

2.3 Shortcomings of $G_{n,p}$: components, degrees, clustering, communities, distances

2.3.1 Component structure

In the previous sections, we have studied the component structures of $G_{n,p}$. We have seen that in the supercritical regime, there is a single giant component and a few smaller components, but no medium-size components. The smaller components are indeed very small: the number of vertices in components of size s drops exponentially with s. This is a pretty good match for real-world networks, where we also usually have a giant component and otherwise only very few small components. For example, the only neural network that we know completely is that of C. elegans.[6] It has a system of 282 somatic neurons, with 514 connections between them. It has a giant component of size 248, two smaller components of size 2 and 3, and 29 isolated nodes [VCP+11].

A completely different example is the facebook network, where nodes consists of (active) profiles, and links are given by facebook friendships. In

[6] Caenorhabditis elegans is a tiny worm, and its connectome is hard-coded in its genes. So all adult worms of the same sex have the same set of neurons, and mostly the same set of connections between them. For our discussion we only count the (undirected) electrical connections, not the (directed) chemical ones.

an analysis from 2011 [UKBM11], when the network had about $n = 7 \cdot 10^8$ nodes (users) and $m = 7 \cdot 10^{10}$ links, it was found that 99.91% of the users were part of a single giant component.[7] Most of the other components were very small. The second-largest component had size 2000.

Is this facebook data a good match for the Erdős-Rényi model? Partly. On the one hand, qualitatively we get the right behaviour: a single giant component, and most of the other vertices concentrating in very small components. On the other hand, quantitatively the match is not so good. In Erdős-Rényi graphs, the number of components of size s shrinks exponentially in s and is bounded by $n \cdot \eta^{-s}$. Note that it hardly matters that the factor n is pretty big, since s enters in the exponent. It is hard to argue for a realistic value of η for which η^{-2000} is not astronomically small. For example, $n \cdot 0.95^{-2000} \approx 10^{-36}$. Only for $\eta = 0.99$, we start to get $n \cdot 0.99^{-2000} \approx 1$. But this would be a pretty extreme value of η. Since $0.99^{100} \approx 1/e \approx 0.37$, it would mean that we should only see a small difference (a factor of three or less) between the number of components of size $s = 2$ and of size $s = 102$.

Indeed, a closer analysis shows that the number of components of size s in the facebook network does not decrease exponentially in s, but rather polynomially. This means that the Erdős-Rényi model is missing something about the structure of small components in the facebook graph. If we want to understand the details of the component structures in the facebook graph, then we will need to look for explanations which are not captured by Erdős-Rényi graphs.

2.3.2 Degree distribution

The degree of a vertex in $G_{n,p}$ is $\mathrm{Bin}(n, p)$-distributed, which for $p = \mu/n$ is approximately $\mathrm{Po}(\mu)$. The Poisson distribution has exponentially decaying tails, so the degrees are concentrated in a small range. In particular, it is very unlikely for a vertex to have degree larger than $O(\log n / \log\log n)$.[8]

This is not a good match for most real-world networks. One of the most prominent properties is the existence of nodes with very different degrees.

[7] The analysis excluded users without facebook friends, so by definition there were no isolated nodes. The relative size of the giant component would probably be smaller if those nodes were counted as well.

[8] This can be computed by a direct calculation or by using Chernoff bound. For Chernoff bounds, mind that there are weaker and stronger versions. The weaker versions may only give you $O(\log n)$.

Usually the degrees follow at least partly a power-law distribution. We will return to this point in Chapter 3.

2.3.3 Clustering

We start with a definition.

Definition 2.8. For a graph $G = (V, E)$, the *clustering coefficient* of a vertex v is defined as follows.

$$CC(v) := \begin{cases} \frac{|\{\{u,u'\}\in\binom{V}{2} | \{u,v\},\{u',v\},\{u,u'\}\in E\}|}{|\{\{u,u'\}\in\binom{V}{2} | \{u,v\},\{u',v\}\in E\}|} & \text{if } \deg(v) \geq 2 \\ 0 & \text{if } \deg(v) \leq 1. \end{cases}$$

Note that the denominator in the first case is simply $\binom{\deg(v)}{2}$. The (local) clustering coefficient of G is then defined as

$$CC(G) := \frac{1}{n} \sum_{v \in V} CC(v).$$

The clustering coefficient can be interpreted in the following way. Choose a vertex v uniformly at random. Afterwards, select two different neighbours u, u' of v. Then the clustering coefficient is the probability that u and u' are adjacent, see also Figure 2.1. (We count the probability as zero if $\deg(v) \leq 1$.)

Some authors use an alternative definition, where the clustering coefficient of G is three times the number of triangles in G, divided by the number of paths of length 2. This is also sometimes called the *global clustering coefficient*. The global clustering coefficient puts more weight on large-degree vertices than the local clustering coefficient. In fact, the global clustering coefficient can be written as the weighted average of all $CC(v)$, where vertex v is weighted by $\binom{\deg(v)}{2}$.

For $G_{n,p}$ with constant $\mu = pn > 1$, the clustering coefficient is easy to compute. If we pick a vertex v, then with constant probability it has degree at least two. (In the limit the probability is $\Pr(\mathrm{Po}(\mu) \geq 2) = 1 - e^{-\mu}(1+\mu)$.) Conditional on that, after picking two random neighbours u, u' of v, the probability that they are connected is exactly $p = \mu/n$. Thus the local clustering coefficient is $\Theta(1/n)$.

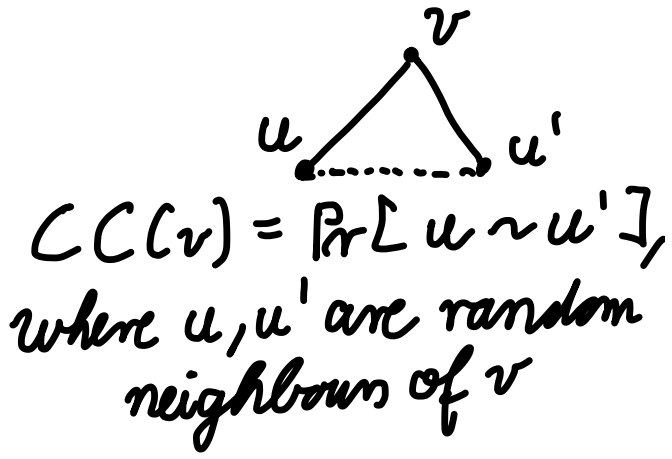

Figure 2.1: The clustering coefficient $CC(v)$ is the probability that two random neighbours of v are adjacent.

For many real-world networks, this is a poor match. They have rather high clustering coefficient. This is not surprising. For example, in the friendship network, if you pick two random friends of yourself, then what is the probability that they know each other? It's not one, but it is not $\Theta(1/n)$ either, where $n = 8 \cdot 10^9$ is the number of people on earth. In fact, taking $G_{n,p}$ as a model for the friendship network, this would predict that the probability that two random friends of yours know each other is exactly the same as the probability that two random people on earth know each other. This is clearly a mismatch between Erdős-Rényi networks and reality.

2.3.4 Communities

There are several definitions of *communities*, none of them very formal. One is that communities are subgraphs that are much denser than the total graph. Another definition is that a community is a subgraph which has much more internal edges (within the subgraph) than external edges (from the subgraph to the remainder). It can be tricky to find an appropriate definition of "much" more. If we pick the densest subset S of size k, then we have $\binom{n}{k}$ options for S. This is a huge number of options to pick from, especially if k is large. Even random fluctuations can create a large

difference in such a case.

Let us look at a few examples to see how tricky the intuitive definition can be. The model $G_{n,p}$ is considered as a graph without comunities simply due to its definition: the edges are all completely independent of each other. Thus it is sometimes used as a baseline for the definition of communities, i.e., communities are subgraphs that are denser than anything that we typically find in $G_{n,p}$. But what *do* we find in $G_{n,p}$?

Start in any vertex v in the giant component of $G_{n,p}$ and let S be the set of the first k vertices that we find via a BFS. For small k the induced subgraph will typically be a tree , so it has $k-1$ edges and average degree roughly two. For $\mu = 1+\varepsilon$, this is almost twice as dense as the whole graph! Superficially, it looks like a community. Even more extreme, if we choose S to be the giant component itself, then it has average degree > 2, but zero external edges! Again, this can easily be mistaken for a community. Or take S as as the union of all components of size exactly 2. Then S has still a linear number of internal edges, but zero external edges. Nevertheless, we would hardly want to classify S as a community.

While $G_{n,p}$ does have sets which superficially look like communities, the picture changes a bit if we restrict to *connected subgraphs* of $G_{n,p}$. This restricts the number of options that we can choose from. For example, for small k (e.g., constant), almost all connected subgraphs of k vertices are trees, so they only have $k-1$ edges, which is the minimal density possible among all connected graphs. Globally, we would consider the giant component C_1 of $G_{n,p}$. This graph does not have subgraphs which are much denser than C_1 itself, and no subgraphs with much more internal than external edges. For example, it is impossible to split the giant component into two subsets of linear size such that the number of edges between both sides is $o(n)$ [LM01].[9] On the other hand, real-world networks usually have communities of all sizes, from small ones to linear-size ones, even within the giant component. Thus Erdös-Rényi graphs are not a good model for community aspects of graphs.

2.3.5 Distances

If we pick two nodes u, v uniformly at random, what is the graph distance between them, i.e., what is the length of a shortest path from u to v? This

[9] Another application of the sprinkling technique.

is also called the *typical distance* of the graph.[10] To answer this question for $G_{n,p}$, we can use the following algorithm for finding a shortest path. Assume we start a BFS from u and at the same time a BFS from v. Then in depth d we find all vertices which have distance exactly d from u and v respectively. Let us call the set of these vertices the d-th level. We continue with the two BFS level by level (alternating between the two) until we find a vertex w that belongs to both BFS trees. Then a shortest path from u to v runs through w, and its length is given by the sum of the depths of w in the two BFS tree, see Figure 2.2.

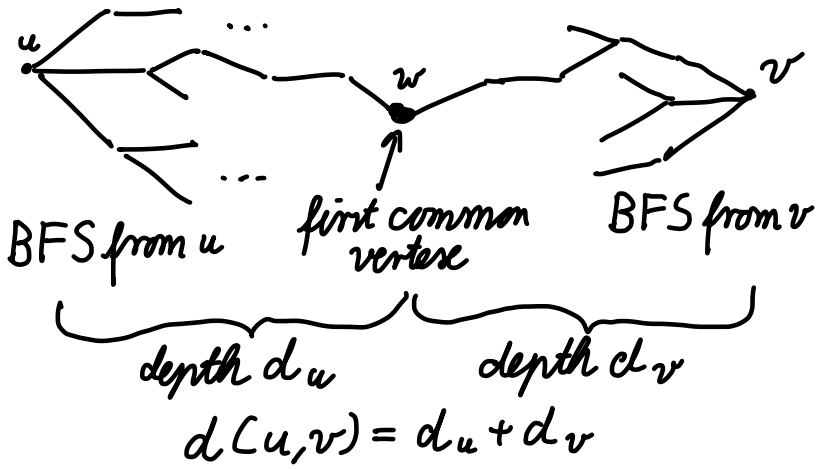

Figure 2.2: Shortest paths can be found by bidirectional breadth-first search (BFS) from both endpoints.

Now let us analyze the process for $G_{n,p}$. We will only give an informal argument, but it can be turned into a formal one with standard probabilistic tools like the Chernoff bound. For each BFS, we know that it branches like a Galton-Watson tree T. We have two possibilities for T. Either it dies out quickly, and then the vertex is in a small component. Or it grows to infinite size. In this latter case, it is not hard to see that it grows rather reliably. Every vertex has offspring distribution $\mathrm{Po}(\mu)$, which has expectation μ. If we have x nodes in depth d, then in expectation the next level has size

[10] We do not give a formal definition of typical distances, as the term is used in different ways. The most common situation is that one can show that for two random vertices u, v, their distance is whp in some small interval. Then this interval is called the set of typical distances.

μx. Since each node draws the number of its children independently, the actual size of the next level is sharply concentrated around μx if x is large. So after some initial phase (where x is still small), once x becomes large, it will realiably grow by a factor of μ in each level.

One can indeed show that, if the process survives, whp the trees grow like $\Theta(\mu^d)$, up to some small fluctuations in the beginning that essentially add or subtract a constant number of rounds. In other words, the number of vertices in distance d from u is roughly $\Theta(\mu^d)$, and similarly for v.

When is the first time that we find a shared vertex w in both search trees? You man have encountered a variation of this question as *birthday paradox*. If we have two random subsets[11] of V of size s, then the expected number of collisions (elements which appear in both subsets) is $s^2/|V|$. So if $s = o(\sqrt{n})$, then by Markov's inequality whp we will not see a collision, but if $\omega(\sqrt{n})$ then whp we do see collisions. In other words, the first time when we encounter a collision is when $s = \Theta(\sqrt{n})$. This happens after $d = \log_\mu(\Theta(\sqrt{n})) = \frac{1}{2}\log_\mu n \pm O(1)$ rounds. Since d is the distance from u to w and from w to v, the distance from u to v is $2d \approx \log_\mu n = \Theta(\log n)$. Summarizing, for two vertices u and v, conditional on being in the giant component, the distance between them is typically $\log_\mu n \pm O(1) = \Theta(\log n)$.

Is this a good match for real-world networks? As for the component structures, the answer is mixed. On the one hand, $\log n$ is a pretty small function, and we do see that real-world networks tend to have very small typical distances. Graphs with typical distance $O(\log n)$ are called *small-world graphs*. On the other hand, many real-world networks have extremely small typical distances, so small that $\log n$ might still be considered large. For example, the facebook graph was studied in 2016 with $n = 1.6 \cdot 10^9$ nodes [EDF+16]. It has average distance (in the giant component) of 4.57, which is a surprisingly small number for more than a billion user. It might still be compatible with average distances of $\approx \log_\mu n$ (especially because the facebook graph has a large average degree of $\mu \approx 200$), but we will learn later about models which have even smaller typical distances.

More importantly, Erdős-Rényi graphs completely miss two aspects which are crucial for average distances: on the one hand, they lack cluster-

[11] In a formal proof one would need to argue why the nodes in the search trees are random subsets, but this follows from the symmetry of $G_{n,p}$.

ing and communities, which increase typical distances. To see this, consider the BFS search from a vertex v. For Erdős-Rényi graphs, we could assume that all found vertices are new vertices, which we have not encountered before. But in a graph with large clustering coefficient, this already goes wrong in the second step. When we explore the first neighbour u of v, then a large clustering coefficient means that many of the neighbours of u were already neighbours of v, and thus have already been revealed in the first step. Thus the BFS tree is much smaller than a corresponding Galton-Watson tree. The existence of communities causes similar problems later in the BFS search. Since the BFS trees grow slower with clustering and communities, these phenomena increase typical distances.

On the other hand, $G_{n,p}$ has a very homogeneous degree distribution. As we will see for other models, the degree distribution plays an important role for typical distances, and heterogeneous distributions can massively decrease them. Thus Erdős-Rényi graphs lack two aspects that are both important for typical distances, one which increases distances, and another one which increases them. So we should be careful to draw conclusions about graph distances from this model.

Chapter 3

Inhomogeneous Degree Distributions

For Erdős-Rényi random graphs, the degrees are concentrated in a small interval, and it is unlikely to have degrees larger than $O(\log n / \log \log n)$. Many real-world networks are not like this. In a train network, hubs like Zürich HB have much more connections than the average Swiss station. In social networks, the number of friends or followers can span many orders of magnitude. In the web graph, some pages have a tremendous number of incoming links. In many of these cases, the degree distribution for many such networks follows a *power-law*. This observation was first made by Barábasi and Albert [BA99]. By now there are many good and thorough discussions of this phenomon, for example Chapters 1.6 in the book by van der Hofstad [VDH09] or the classical accounts in [BB03] and [New03].[1] In this chapter we will study models which have power-law degree distributions built into them.

[1] There are exceptions, of course. For example, the street network of a city, where roads are edges and road intersections are vertices, do not have a power-law degree distributions. Also, there is ongoing debate how good power-laws approximate the real degree distribution for various networks, see for example the discussion in [VDH09, Chapter 1.6]. This issue can be circumvented by working with more general classes of degree distributions, but we will ignore them for this lecture.

3.1 Power-laws

3.1.1 Power-law probability distributions

We start with a general introduction into power laws. They are also sometimes called *scale-free*.[2] Throughout the section, $\mathcal{D}$ will denote a probability distribution either on $\mathbb{N}_0$ or on $[1, \infty)$. Let $X \sim \mathcal{D}$, and let $f_{\mathcal{D}}(x)$ be the probability density function of $\mathcal{D}$ (if it exists). By abuse of notation, we will usually write $\Pr[X = x]$ instead of $f_{\mathcal{D}}(x)$. This notation is not problematic if the distribution lives on $\mathbb{N}_0$, but for continuous distributions on $[0, \infty)$, the term $\Pr[X = x]$ does not literally mean "the probability of the event $X = x$", which has probability zero for decent continuous distributions. However, there should never be room for confusion, since we will never consider zero-probability events of this type in this lecture. This convention allows us to treat distributions over $\mathbb{N}_0$ and over $[1, \infty)$ at the same time. For the cumulative distribution function $\Pr[X \geq x]$ of $\mathcal{D}$, the notion is not problematic in either case.

Definition 3.1. Let $\mathcal{D}$ be a probability distribution on $[1, \infty)$, and let $X \sim \mathcal{D}$. We say that

(i) $\mathcal{D}$ follows a *(strict density) power law* with exponent $\tau > 1$ if

$$\Pr[X = k] = \Theta(k^{-\tau}) \qquad \text{for } k \to \infty.$$

(ii) $\mathcal{D}$ follows a *weak density power law* with exponent $\tau > 1$ if

$$\Pr[X = k] = k^{-\tau \pm o(1)} \qquad \text{for } k \to \infty.$$

(iii) $\mathcal{D}$ follows a *cumulative power-law* with exponent $\tau > 1$ if

$$\Pr[X \geq k] = \Theta(k^{1-\tau}) \qquad \text{for } k \to \infty \qquad \text{(strict power law);}$$
$$\Pr[X \geq k] = k^{1-\tau \pm o(1)} \qquad \text{for } k \to \infty \qquad \text{(weak power law).}$$

In this definition, $\Theta()$ and $o()$ are taken with respect to $k \to \infty$.

For our purposes, the differences between the four different versions of

[2] Some authors reserve the phrase *scale-free* to power-laws with exponent $\tau \in (2, 3)$.

power-laws do not matter much. A strong/weak density power-law implies a strong/weak cumulative power-law, but not vice versa. In this lecture, we will usually assume strict density power-laws, i.e., we make the strongest possible assumption. We do this to simplify calculations, not because it is necessary. Without further specification, "power-law" means "strict density power-law".

3.1.2 Power-law sequences

We have defined power-laws for probability distributions, but what does it mean that the degrees of a random network model $G = G_n$ follow a power-law? This question is a bit more tricky. The first attempt would be to draw a random vertex v from G_n, compute the limiting probability $p_k := \lim_{n\to\infty} \Pr_{G_n}[\deg(v) = k]$, and require that the p_k converge to the density function of a power-law probability distribution. For example, for Erdős-Rényi graphs $G_{n,p}$ with $\mu = np = \Theta(1)$, we know that $p_k = e^{-\mu}\mu^k/k!$. This is indeed the density of a probability distribution (not a power-law), namely of $\mathrm{Po}(\mu)$. However, this approach does not work well in general, for two reasons.

First, we require that the p_k form a probability distribution. This does not have to be the case in general. For example, consider $G_{n,p}$ with any $p = \omega(1/n)$. Then $p_k := \lim_{n\to\infty} \Pr_{G_n}[\deg(v) = k] = 0$ for every fixed k. Of course, the function $\forall k : p_k = 0$ does not correspond to a probability distribution. The second issue is that the p_k only carry information about *constant* k. Remember the fraction p_s of vertices in components of size s? They do not sum up to one if there is a giant component. That is exactly the same problem as here. The p_s do not account for components of growing sizes, and the p_k do not account for growing degrees. But the whole point of power-law distributions is that we find vertices of very large degree in the graph, e.g., vertices of degree $\Theta(\sqrt{n})$ if $\tau < 3$ (more on that below). We cannot account for that by the limits p_k.

Instead, we need to make a more direct approach. We require that the number of vertices of degree k (at least k) is roughly the same as predicted by a power law, up to some point which depends on n.

Definition 3.2. Let $G(n)$ be a sequence of random graphs, let $N_{d,n}$ be

the number of vertices of degree d in $G(n)$, and let $D = D(n) \in [1, n]$ be a function of n. We say that $G(n)$ follows a (strict, density) power-law with exponent τ up to D if there are constants $c_1, c_2 > 0$ such that with high probability

$$\forall 1 \leq d \leq D : N_{d,n} = \Theta(d^{-\tau} n), \tag{3.1}$$

where the hidden constants are uniform over all d. In other words, we require that there are $c_1, c_2 > 0$ such that with high probability the following holds.

$$\forall 1 \leq d \leq D : c_1 d^{-\tau} n \leq N_{d,n} \leq c_2 d^{-\tau} n. \tag{3.2}$$

We can analogously define weak power-laws, and (strict or weak) cumulative power-laws.

We say that the power-law has *negligible cut-off error* if with high probability $\sum_{d=D+1}^{n} N_{d,n} = o(n)$.

There are some differences between power-laws for probability distributions and for degree sequences that are worth pointing out. Most obviously, Definition 3.2 only requires that degrees up to D are well-behaved. For a finite graph, we can not expect (3.1) to hold without restriction on D. If we would plug in $d = \omega(n^{1/\tau})$ then $d^{-\tau} n = o(1)$, so the integer $N_{d,n}$ can not be in $\Theta(d^{-\tau} n)$. Many models have (density) power-law degrees up to $D = n^{1/\tau - \varepsilon}$ for any fixed ε. We remark that the requirement for cumulative power-law degrees is $N_{\geq d,n} = \Theta(d^{-\tau+1} n)$, where $N_{\geq d,n}$ is the number of vertices of degree *at least* d in $G(n)$. This requirement can hold a bit longer, and in many models it holds until $D = n^{1/(\tau-1)-\varepsilon}$. This is why one can sometimes prove stronger results by working with cumulative power-laws.

A second difference between power-laws of distributions and degree sequences is that (3.1) can actually be checked for a concrete network. In practice, power-laws (of degrees, but also of any other quantity) can be checked by plotting $N_{d,n}$ on a log-log-plot. On such plots, a power-law corresponds to a straight line: if $N_{d,n} = c \cdot d^{-\tau} n$, then $\log(N_{d,n}) = \log(cn) - \tau \log d$, so there is a linear relation between the quantities $\log d$ and $\log(N_{d,n})$ that are used in the axes of a log-log-plot. Moreover, the slope of the line is $-\tau$, so we can recover the power-law parameter by estimating the slope of the line in the log-log-plot. In practice one indeed often

finds a linear relationship in log-log-plots, though often with a cut-off point that is earlier than the theoretically achievable value. We refer the reader to Figure 5.5 in [LNR17] and Chapter 1.6 in [VDH09] for log-log-plots of the degree distributions of various real-world networks.

3.1.3 Properties of power-laws

The most important case for us will be power-law exponents $\tau \in (2,3)$. For most real-world networks, the degree sequences have power-law exponents in this range (though there are exceptions). During this course we will understand why this is the interesting case. As the next lemma shows, this is the range where the distribution has finite expectation, but infinite second moment.

Lemma 3.3. Let $\mathcal{D}$ be a distribution which follows a power-law with exponent $\tau > 1$ for any of the four variants in Definition 3.1, and let $X \sim \mathcal{D}$.

(i) If $\tau < 2$ then $\mathbb{E}[X] = \infty$.

(ii) If $\tau > 2$ then $\mathbb{E}[X] < \infty$.

(iii) If $\tau < 3$ then $\mathbb{E}[X^2] = \infty$.

(iv) If $\tau > 3$ then $\mathbb{E}[X^2] < \infty$.

Proof. We only give the calculation in the simplest case of a strong power-law for densities. Then

$$\mathbb{E}[X] = \sum_{k=0}^{\infty} X \cdot \Pr[X = k] = \Theta(\sum_{k=1}^{\infty} k \cdot k^{-\tau}) = \Theta(\sum_{k=1}^{\infty} k^{-\tau+1}),$$

and the sum is finite for $\tau > 2$ and infinite for $\tau < 2$. Note that $\Theta()$ was taken with respect to the limit $k \to \infty$ and thus the hidden factors are independent of k (and not, as usual, independent of n). This is why we could take $\Theta()$ out of the sum.

For the second moment, the calculation is almost identical:

$$\mathbb{E}[X^2] = \sum_{k=0}^{\infty} X^2 \cdot \Pr[X = k] = \Theta(\sum_{k=1}^{\infty} k^2 \cdot k^{-\tau}) = \Theta(\sum_{k=1}^{\infty} k^{-\tau+2}),$$

which is finite for $\tau > 2$ and infinite for $\tau < 2$.

For power-laws of the cumulative distribution, a sum like $\sum_{k=0}^{\infty} X^2 \cdot \Pr[X = k]$ can be linked to the sum $\sum_{k=0}^{\infty} X \cdot \Pr[X \geq k]$ via *Abel summation*, the discrete version of integration by parts. We do not give the details. □

We remark that for the threshold cases $\tau = 2$ and $\tau = 3$ in Lemma 3.3, it depends on the exact distribution whether the corresponding moments are finite or infinite. As usual for this course, we do not go into details for such threshold cases.

Lemma 3.3 has an important implication. Let us forget momentarily about the cutoff in Definition 3.2. If $X \sim \mathcal{D}$ is the distribution of degrees in a network model, then $\mathbb{E}[X]$ corresponds to the expected degree, and $\mathbb{E}[X] \cdot n/2$ is the expected number of edges in the graph. More formally, let us consider the case $\mathbb{E}[X] = c < \infty$ and let us assume that the number of edges is concentrated around its mean and that the power-law has negligible error. Then whp the number of edges is $c/2 \cdot n \pm o(n)$, and the average degree is $c \pm o(1)$. On the other hand, if $\mathbb{E}[X] = \infty$, then we can not ignore the cutoff in Definition 3.2, since the number of edges in an n-vertex graph can not be infinite. However, the next lemma tells us that this corresponds to an average degree of $\omega(1)$, or equivalently to $\omega(n)$ edges.

Corollary 3.4. Let $G(n)$ be a sequence of random graphs, $D = D(n) = \omega(1)$ and assume that the degrees in $G(n)$ follow a power-law with exponent τ up to D as in Definition 3.2.

(i) If $\tau < 2$, then whp the number of edges in G is $\omega(n)$.

(ii) If $\tau > 2$, let $m_{\leq D}$ be the number of edges which have at least one endpoint of degree at most D. Then $\mathbb{E}[m_{\leq D}] = O(n)$. Moreover, if the power-law has negligible cut-off error then whp $m = m_{\leq D} \pm o(n)$.

Proof. (i). Let $N_{d,n}$ be the number of vertices of degree d in $G(n)$. Let $C > 0$ be arbitrary. We will show that whp $m > Cn$ if n is sufficiently large.

By (3.2), the number of edges is at least

$$m = \tfrac{1}{2}\sum_{v\in V}\deg(v) = \tfrac{1}{2}\sum_{d=0}^{n} d\cdot N_{d,n} \overset{\text{whp}}{\geq} \tfrac{1}{2}\sum_{d=0}^{D} d\cdot c_1 d^{-\tau} n = \tfrac{c_1 n}{2}\sum_{d=0}^{D} d^{1-\tau}. \quad (3.3)$$

Since $1-\tau > -1$, the sum over $d^{1-\tau}$ diverges. Hence there is a constant k such that $\sum_{d=0}^{k} d^{1-\tau} > 2C/c_1$. If we make n large enough, then $D = D(n) \geq k$, and (3.3) implies $m > Cn$.

(ii). We make a similar computation to (3.3). Since we want to estimate $m_{\leq D}$ instead of m, we may use the following upper bound. (It is not an equality because we count edges twice if both their endpoints have degrees at most D.)

$$m_{\leq D} \leq \sum_{d=0}^{D} d\cdot N_{d,n} \overset{\text{whp}}{\leq} \sum_{d=0}^{D} d\cdot c_2 d^{-\tau} n = c_1 n \sum_{d=0}^{\infty} d^{1-\tau}. \quad (3.4)$$

Other than in (i), the sum in (3.4) converges to a constant C since $1-\tau < -1$. Hence $m_{\leq D} \leq C\cdot c_1 n$. The second statement in (ii) follows directly from the definition of negligible cutoff error since $m - m_{\leq D} \leq \sum_{d=D+1}^{n} N_{d,n} \overset{\text{whp}}{=} o(n)$. □

We have found out that $\tau > 2$ leads to sparse graphs (constant degrees), while $\tau < 2$ does not. Since we are interested in sparse graphs in this course, we will from now on restrict to $\tau > 2$. Let us compute how many edges come from vertices of degree between K and D, for some large K. By a similar calculation as above, the number of incident edges to such vertices is at most

$$\sum_{d=K}^{D} d\cdot(c_2 d^{-\tau} n) \leq \sum_{d=K}^{\infty} c_2 d^{1-\tau} n = \Theta\Big(\int_K^{\infty} c_2 x^{1-\tau} n\, dx\Big) = \Theta(K^{2-\tau} n).$$

Since $\tau > 2$, the exponent of K is negative. So if K is large, then this is only a small fraction of all edges. In other words, most edges will not have an endpoint of degree larger than K. A common way to phrase this is: "Most edges run between vertices of small degrees". Yet another formulation of the same fact is that if we pick a random edge, and then a random endpoint of this edge, then the endpoint will likely have small degree. It is easy to see that the opposite is true for $\tau < 2$. There, a random endpoint of a random edge has likely very large degree.

3.2 The Chung-Lu model

In the following sections we will discuss several random graph models that have power-law degree distributions. The easiest one is the *Chung-Lu model*, named after Fan Chung and Linyuan Lu, who analyzed the model in detail [CL02].[3] It is also known as the *weighted Erdős-Rényi model* or simply as *weighted random graphs* or *generalized random graphs*.

Definition 3.5. Let $\mathcal{D}$ be a power-law distribution on $[1, \infty)$ with exponent $\tau > 2$. A *Chung-Lu random graph* on n vertices is obtained by the following two-step procedure.

1. Every vertex $v \in V$ draws i.i.d. a *weight* $w_v \sim \mathcal{D}$.
2. For any two distinct vertices $u, v \in V$, we insert the edge $\{u, v\}$ with probability

$$p_{uv} := \min\left\{1, \frac{w_u w_v}{n}\right\}, \tag{3.5}$$

independently for all $\{u, v\}$.

3.2.1 Degree distribution

The minimum in (3.5) needs to be present so that p_{uv} is a probability. A fundamental fact about this formula is that it converts weights into expected degrees, as the following lemma shows.

Lemma 3.6. Let v be a vertex of weight $w_v \leq n$ in the Chung-Lu model. Then

$$\mathbb{E}[\deg(v)] = \Theta(w_v). \tag{3.6}$$

The hidden constants are independent of v and w_v.

Proof. Let $u \in V \setminus \{v\}$. Assume first that $w_v \leq n/2$, or rephrased that $n/w_v \geq 2$. We can calculate the expected degree of v as follows. (See also

[3] The original model of Chung and Lu differed in some details, see Section 3.2.4.

Excursion 3.1 below.)

$$\begin{aligned}
\mathbb{E}[\deg(v)] &= (n-1)\cdot \Pr[v \sim u] \\
&= (n-1)\int_1^\infty \Pr[w_u = w]\Pr[v \sim u \mid w_u = w]\mathrm{d}w \\
&= \Theta(n)\cdot \int_1^\infty w^{-\tau}\min\left\{1, \frac{w\cdot w_v}{n}\right\}\mathrm{d}w \\
&= \Theta(n)\cdot\left(\int_1^{n/w_v} w^{-\tau}\frac{w\cdot w_v}{n}\mathrm{d}w + \int_{n/w_v}^\infty w^{-\tau}\cdot 1\mathrm{d}w\right) \\
&= \Theta(w_v)\cdot\int_1^{n/w_v} w^{1-\tau}\mathrm{d}w + \Theta(n)\cdot[w^{1-\tau}]_{n/w_v}^\infty \\
&\overset{\tau>2}{=} \Theta(w_v) + \Theta(n\cdot(n/w_v)^{1-\tau}) \\
&= \Theta(w_v) + \Theta((n/w_v)^{2-\tau}\cdot w_v) = \Theta(w_v),
\end{aligned}$$

where the last step uses that $n/w_v \geq 1$ and $\tau > 2$. One checks that the same end result also holds for $n/2 < w_v \leq n$. □

The assumption $w_v \leq n$ in Lemma 3.6 is not problematic. Let us call $n_{\geq w}$ the number of vertices of weight at least w. Then $\mathbb{E}[n_{\geq w}] = \Theta(n\cdot w^{1-\tau})$. In particular, for $w_{\max} := n^{1/(\tau-1)}$ this expectation is $\Theta(1)$. This means that the largest weight among the n vertices is roughly $w_{\max}$. For $w = \omega(w_{\max})$ we have $\mathbb{E}[n_{\geq w}] = o(1)$, so such vertices are unlikely to exist by Markov's inequality. On the other hand, if $w = o(w_{\max})$ then $\mathbb{E}[n_{\geq w}] = \omega(1)$, and whp $n_{\geq w} > 0$ by the Chernoff-Hoeffding bound. For $\tau \in (2,3)$, we can bound $n^{1/2} \leq n^{1/(\tau-1)} \leq n$, so whp the largest weight is larger than $\sqrt{n}$, but smaller than n.

We can actually say a bit more about the exact distribution of degrees. Consider a vertex of weight w_v, i.e., we fix the weight of this vertex, but consider the other weights still as random. Then every other vertex has the same probability $p_v = \Theta(w_v/n)$ to connect to v, so the degree of v is $\mathrm{Bin}(n-1, p_v)$-distributed. If $p_v = o(1)$ (or equivalently $w_v = o(n)$, which whp holds for all vertices), then this converges to $\mathrm{Po}((n-1)p_v)$. We say that $\deg(v)$ converges to a *mixed Poisson distribution*, which is a Poisson distribution $\mathrm{Po}(X)$, where X is again a random variable. In this case, $X = \mathbb{E}[\deg(v) \mid w_v] = \Theta(w_v)$.

Recall that the Poisson distribution is highly concentrated around its expectation if the expectation is large. This means that the degree of a

vertex of large weight is concentrated around its expectation, which is (up to a constant factor) the same as its weight. For example, it can be shown that whp all vertices of weight at least $\log n$ have degrees which match their expectations up to a constant factor. As a consequence, the degree sequence follows a power-law with the same exponent τ as the weight distribution. This is the main purpose of the model: to generate graphs with a power-law degree distribution of exponent τ.

Excursion 3.1 (Smart with integrals). We will see computations as in the proof of Lemma 3.6 over and over again, so let us make a detour to discuss the structure of the calculation a bit more in detail. It looks messy at first glance, but it becomes much easier if one knows what to look for.

We obtained a sum of two integrals,

$$\underbrace{\int_1^{n/w_v} w^{-\tau}\frac{w\cdot w_v}{n}\,dw}_{=:I_1}+\underbrace{\int_{n/w_v}^{\infty} w^{-\tau}\cdot 1\,dw}_{=:I_2}.$$

In the proof we evaluated both of them, and it was a bit tricky to see that the second one is negligible. However, *none of this was actually necessary.* With the right perspective, it was a priori clear that the second integral was negligible.

Let us first summarize what we have already used in the computation. Both integrals are over polynomials in w. Integrating over polynomials is easy. Since we do not consider threshold case, we only have exponents $s \neq -1$, and the inverse derivation of w^s is $\frac{1}{1+s}w^{1+s}$. The integral $\int_{w_0}^{w_1} w^s\,dw$ can thus be easily evaluated and is $[\frac{1}{1+s}w^{1+s}]_{w_0}^{w_1}$. If $s > -1$, then w^{1+s} is increasing, and we get $\Theta(w_1^{1+s})$. (We assume here that w_1 is at least by a constant factor larger than w_0.) If $s < -1$, then w^{1+s} is decreasing, so we obtain $\Theta(w_0^{1+s})$. Note that the signs also work out in the second case, since there are two canceling minus signs: from $1/(1+s) < 0$ and from evaluating the lower boundary. So the integral is always dominated either by the upper boundary term or by the lower boundary term.

Let us call I_1^{low} and I_1^{upp} the two terms needed to evaluate I_1, i.e., I_1^{low} and I_1^{upp} are obtained by plugging the lower (the upper) boundary into the inverse derivative $\frac{1}{2-\tau}w^{2-\tau}w_v/n$, and similarly for I_2. In the proof of Lemma 3.6, the dominating term was given by the lower boundary for both integrals, and we evaluated both I_1^{low} and I_2^{low}. However, a more clever argument uses the observation that $|I_1^{upp}| = \Theta(|I_2^{low}|)$. Before we argue why this is generally true, let us check this by hand. For $|I_1^{upp}|$, we need to plug n/w_v into the inverse derivative $w^{2-\tau}w_v/n$, and obtain up to constant factors

$$|I_1^{upp}| = \Theta((n/w_v)^{2-\tau}w_v/n) = \Theta((n/w_v)^{1-\tau}).$$

For $|I_2^{low}|$, we plug n/w_v into $w^{1-\tau}$, and obtain the same term.

This is an incredibly helpful observation. Knowing this relation, we know that

- I_1^{low} dominates I_1^{upp}.
- I_1^{upp} is of the same order as I_2^{low}.

- I_2^{low} dominates I_2^{upp}.

From these observations it is obvious that I_1^{low} dominates everything else. Even more, in the proof of Lemma 3.6 we had to give special treatement to the case $n/2 < w_v \leq n$, because our estimation of I_1 may fail if w_v is very close to n. This is because if the upper and lower boundary (1 and n/w_v) are too close to each other, the terms I_1^{low} and I_1^{upp} are very similar to each other any may cancel. (They have opposite signs.) But in fact, we have nothing to fear: if I_1^{low} and I_1^{upp} are of the same order, then we have a third term I_2^{low} which is also of the same order and which can not be canceled out. So by our meta-argument it becomes obvious that the asymptotics do not change in this case either.

Why is it generally true that $|I_1^{\text{upp}}| = \Theta(|I_2^{\text{low}}|)$? This is because the two integrals I_1 and I_2 were obtained by splitting a single integral $I = \int_1^\infty f(w)dw$ into two parts. But I was taken over a *continuous* function f. (The min function is not smooth, but it is continuous.) This means that the functions in I_1 and I_2 take the same values when the splitting point $w_{\text{split}} = n/w_v$ is plugged in; it is just the same as plugging w_{split} into f. How do we get I_1^{upp} and I_2^{low} from $f(w_{\text{split}})$? Since we integrate over polynomials in both cases, we just increase the exponent by one and plug in w_{split}, so in both cases we obtain $w_{\text{split}} \cdot f(w_{\text{split}})$ up to constant factors.

The argument may seem a little magic, so let us rephrase it in terms of the quantities that we compute. We compute how many neighbours a vertex v of weight w_v has in expectation. The integral I has a natural interpretation: the range from w_0 to w_1 gives us the number of neighbours with weight between w_0 and w_1 (all in expectation). The fact that the first integral has exponent < -1 tells us that there are more neighbours of constant weight than of larger weight, for example of weight in the interval $[w_{\text{split}}/2, w_{\text{split}}]$. The key insight is that the latter number is essentially the same as the number of neighbours with weight in $[w_{\text{split}}, 2 \cdot w_{\text{split}}]$, due to three ingredients:

1. Both intervals have the same length, up to a constant factor of 2.
2. The probability density is the same up to a constant factor: increasing w by a constant factor κ decreases the probability density $\Pr[w_u = w]$ by the constant factor $\kappa^{-\tau}$.
3. The connection probability $\Pr[u \sim v \mid w_u = w]$ only changes by a constant factor if we vary w within $[w_{\text{split}}/2, 2 \cdot w_{\text{split}}]$. This is because the connection probability is continuous and piecewise smooth in w_u.

So let us summarize how we should actually think about Lemma 3.6. (All statements about expectations.)

(a) Since I_1 has exponent < -1, there are more neighbours of constant weight than of larger weights, in particular than weights in $[w_{\text{split}}/2, w_{\text{split}}]$.

(b) There are the same number of neighbours with weights in $[w_{\text{split}}/2, w_{\text{split}}]$ and with weights in $[w_{\text{split}}, 2 \cdot w_{\text{split}}]$, up to constant factors.

(c) Since I_2 has exponent < -1, there are more neighbours of weights in $[w_{\text{split}}, 2 \cdot w_{\text{split}}]$ than of larger weights.

Thus, most neighbours have constant weight, and we can neglect any term that comes from neighbours of larger weight. It thus suffices to compute how many neighbours of

constant weight there are. This is easy to compute. There are $\Theta(n)$ vertices of constant weight, and each of them has probability $\Theta(w_v/n)$ to connect to v. Hence, w_v has $\Theta(w_v)$ neighbours.

As a final exercise, let us try to apply the same reasoning for $1 < \tau < 2$. How many neighbours does w_v have in this case? We still obtain the same integrals I_1 and I_2, but now the exponent $1-\tau$ of I_1 is larger than -1. This means that v has more neighbours of weight in $[w_{\text{split}}/2, w_{\text{split}}]$ than of smaller weights. On the other hand, the exponent $-\tau$ in I_2 is still smaller than -1, so there are more neighbours with weight in $[w_{\text{split}}, 2 \cdot w_{\text{split}}]$ than neighbours with larger weight. So we only need to evaluate either I_1^{upp} or I_2^{low} (both automatically give the same value up to constant factors). The term I_2^{low} looks a bit simpler, so we plug $w_{\text{split}} = n/w_v$ into the inverse derivative $w^{1-\tau}$ and obtain that v has $\Theta((n/w_v)^{1-\tau})$ neighbours in expectation.[4]

3.2.2 Friends of your friends

We have seen that the degree sequence follows a power-law with exponent τ, because degrees are tightly coupled to the weights. In this section we will ask a perhaps surprising question. If we pick a random vertex v, and then a random neighbour u of v (assuming $\deg(v) \geq 1$), what is the weight (or degree) distribution of u? It is not the same distribution as the unconditional distribution $\Pr[w_u = w]$, since the event "$u \sim v$" is positively correlated to large weights, so the distribution $\Pr[w_u = w \mid u \sim v]$ will also be skewed towards larger weights. This has the psychologically surprising consequence that the neighbours of a typical node v tend to have higher degrees than v. It is also known as the *friendship paradox*: your friends have more friends than you! [Fel91]

How many more friends do your friends have compared to yourself? In an Erdős-Rényi random graph, there is no correlation between edges, so the number of neighbours of u is $1 + \text{Bin}(n-2, p)$: they have neighbour v, all other neighbours are as likely as before. So the asymptotic answer is $1 + \text{Po}(\mu)$ versus $\text{Po}(\mu)$, and neighbours of a random node have their degree increased by exactly one in the limit. But the degree distribution in real-world network is a power-law, and there the difference is much more dramatic.

[4] This calculation tells us a lot: most of the neighbours have weight roughly w_{split}. Note that the connection probability is one if $w_u \geq w_{\text{split}}$, and $\Theta(w_{\text{split}}^{1-\tau})$ is simply the number of vertices of weight at least w_{split}. So we have found out that v connects to all vertices of weight larger than w_{split}, there are $\Theta(w_{\text{split}}^{1-\tau})$ such vertices, and this is more than the number of neighbours of smaller weight.

Theorem 3.7. Let G be a Chung-Lu random graph with power law exponent $\tau > 2$. Let v be a vertex of weight w_v, and let u be a uniformly random neighbour of v, conditioned on $\deg(v) \geq 1$. We denote $W_2 = w_u$. Then W_2 follows a power-law distribution with exponent $\tau - 1$ up to cut-off $D = n/w_v$.

Proof. We need to compute $\Pr[w_u = w \mid u \sim v]$ for $w \leq D = n/w_v$. Note that this is a density, not a probability. Nevertheless, the usual formula for conditional probabilities still applies:

$$\begin{aligned}\Pr[w_u = w \mid u \sim v] &= \frac{\Pr[w_u = w \text{ and } u \sim v]}{\Pr[u \sim v]} \\ &= \Theta(1)\frac{w^{-\tau}\min\{1, \frac{ww_v}{n}\}}{w_v/n} = \Theta(w^{1-\tau}),\end{aligned}$$

since the minimum is taken by the second term for $w \leq D$. □

Of course, since degrees and weights are tightly coupled, a similar statement would be true for the distribution of the *degree* of u instead of the *weight* of u. Theorem 3.7 is rather remarkable since it says that the distribution of w_u does not depend on w_v. This property is also called *neutral assortativity* (or no assortativity). Assortativity is a measure for how much the distribution of $\deg(u)$ depends on the value of $\deg(v)$, and in which direction this connection goes. We do not give a formal defintion (since there are several competing ones), but informally speaking a graph has *positive assortativity* if the distribution of $\deg(u)$ is more skewed towards larger values if $\deg(v)$ is large, and more skewed towards smaller values if $\deg(v)$ is small. If the connection goes into the opposite direction, we speak of *negative assortativity*. As a rule of thumb, social networks tend to have positive assortativity (nodes of large degree connect especially well to other nodes of large degree), while many technological networks have negative assortativity (nodes of large degree connect well to nodes of small degree).

Note that in the most common case $\tau \in (2, 3)$, the random variable W_2 in Theorem 3.7 has infinite expectation. A bit more precisely, the limiting distribution for $n \to \infty$ has infinite expectation, while for finite n there is a cut-off point that goes to infinity. Pointedly speaking, while you have a constant number of friends, your friends have in expectation an infinite

number of friends.[5]

Of course, it is prevented by the cut-off point and the finite size of the universe that your friends actually have infinitely many friends. But the cut-off point goes to infinity as a polynomial in n, so it grows rather fast. The expectation of w_u is really large even for finite networks. On the other hand, this is one of the cases where the expectation is dominated by low-probability events (the rare event that you have a superstar as friend: for most people it does not happen, but the slim chance that it happens dominates the expectation). So for most people the situation looks a bit less depressing. A more accurate estimation of the *typical* most popular of your friends is given by Theorem 3.8.

Theorem 3.8. Let $G = (V, E)$ be a Chung-Lu random graph with power-law exponent $\tau > 2$, and let $v \in V$ be a vertex of weight $w_v = w$ with $2 \leq w \leq n^{(\tau-2)/(\tau-1)}$.

Let $\varepsilon > 0$ be a constant, and let $w_{\max}$ be the highest weight among the neighbours of v. Then for sufficiently large w,

(i) $w_{\max} \geq w^{1/(\tau-2)-\varepsilon}$ with probability $1 - e^{-w^{\Omega(1)}}$.

(ii) $w_{\max} \leq w^{1/(\tau-2)+\varepsilon}$ with probability $1 - w^{-\Omega(1)}$.

The hidden constants are uniform over all w.

Proof. We know by Theorem 3.7 that the neighbours of v follow a power-law with exponent $\tau - 1$. A density power-law implies a cumulative power-law. Therefore, if u is a neighbour of v then $p_x := \Pr[w_u \geq x \mid u \sim v] = \Theta(x^{2-\tau})$. Hence, each of the $n - 1$ other vertices has probability $q_x := \Pr[u \sim v] \cdot \Pr[w_u \geq x \mid u \sim v] = w/n \cdot p_x$ to be a neighbour of v of weight at least x. This is independent for all u since we have fixed the weight $w_v = w$. Hence, the number of such vertices is $\mathrm{Bin}(n-1, q_x)$ distributed.

Now we compute the expectation of this binomial distribution. For (i), we plug in $x := w^{1/(\tau-2)-\varepsilon}$. The condition $w_v \leq n^{(\tau-2)/(\tau-1)}$ ensures that x is smaller than the cut-off point $D = n/w$ in Theorem 3.7, since

[5] This is true for all social networks with power-law exponent $\tau \in (2, 3)$. A Swedish study found that the network of sexual contacts has a power-law exponent of 2.6 [LEA+01]. Draw your own conclusions.

$x \cdot w = w^{(\tau-1)/(\tau-2)-\varepsilon} \leq n$. We obtain an expectation of

$$(n-1)q_x = \Theta(wp_x) = \Theta(w) \cdot w^{(1/(\tau-2)-\varepsilon)\cdot(2-\tau)} = \Theta(w) \cdot w^{-1+\Omega(1)} = \Theta(w^{\Omega(1)}).$$

Since we assumed $w \geq 2$, we can omit the Θ in the last expression:[6] for any constant $C > 0$, we can simplify $Cw^{\Omega(1)}$ into $w^{\Omega(1)}$. Hence the number of neighbours of v of weight at least x is binomially distributed with expectation $w^{\Omega(1)}$. By the Chernoff bounds, the probability to have no such neighbour is exponentially small in $w^{\Omega(1)}$.

For (ii), we plug in $x := w^{1/(\tau-2)+\varepsilon}$, and perform an analogous calculation. We obtain that the expected number of neighbours of v with weight at least x is $w^{-\Omega(1)}$. By Markov's inequality, the probability that there exists such a neighbour is also at most $w^{-\Omega(1)}$. □

In Theorem 3.8, note that the exponent $1/(\tau-2)$ is larger than one for $\tau \in (2,3)$. Hence, v has a neighbour of much larger weight than v itself. On the other hand, if $\tau > 3$ then for a large-weight vertex v typically all its neighbours have much smaller weight than v. However, the model is generally less interesting for $\tau > 3$. The large-degree vertices are then negligible for most questions since there are too few of them, for example for the small-world properties discussed in the next section. In such respects, the Chung-Lu model for $\tau > 3$ behaves just like the Erdős-Rényi model.

For the probabilities in Theorem 3.8, both terms approach 1 as w increases. However, the probability in (i) approaches 1 very rapidly as w increases ("stretched exponentially"), while the probability in (ii) approaches 1 more slowly (polynomially fast in w).

3.2.3 Ultra-small worlds

A remarkable consequence of Theorem 3.8 is that the typical distances in Chung-Lu graphs are even smaller than in Erdős-Rényi random graphs if $\tau < 3$. They are $O(\log\log n)$. Graphs like this are also called *ultra-small worlds*.

Theorem 3.9. Let G be a Chung-Lu random graph on n vertices with power-law exponent $\tau \in (2,3)$, and let x, y be two vertices drawn

[6] This is the only reason for assuming $w \geq 2$, to get rid of some notational ballast. This simplification would not be true for $w = 1$.

uniformly at random from the giant component. Then the graph distance $d(x, y)$ satisfies

$$d(x, y) = \frac{2 \pm o(1)}{|\log(\tau - 2)|} \log\log n \tag{3.7}$$

in expectation and with high probability.

Proof (sketch). We will only show the upper bound, and only sketch the main idea. Let $\varepsilon > 0$, and let us write $\eta := 1/(\tau - 2) - \varepsilon$ for brevity. Since $\tau < 3$, we may choose ε so small that $\eta > 1$.

Consider a vertex v_0 of large constant weight w. Then by Theorem 3.8 v_0 has a neighbour v_1 of weight $w_1 = w^\eta$. Applying the same theorem again, v_1 has a neighbour v_2 of weight $w_2 = w_1^\eta = w^{\eta^2}$. Iterating, we find a vertex v_i in distance i of v_0 of weight $w_i = w^{\eta^i}$. We proceed this way until we find a vertex v_k of weight $w_k \geq n^{1/2}$, which happens if $w^{\eta^k} \geq n^{1/2}$, which gives the condition $k \approx \log_\eta \frac{1}{2} \log_w n = \frac{\log\log n}{\log \eta} - O(\log\log w)$.

We apply this reasoning for both x and y. From each of them, we find paths of length k to vertices x' and y' of weight at least $n^{1/2}$. By definition of the connection probability in (3.5), x' and y' are connected with probability one. Thus we have found a path of length $2k+1$ from x to y. Since $k \approx \frac{\log\log n}{|\log(\tau-2-\varepsilon)|}$, and since we can find this bound for any constant ε, this gives the upper bound of the theorem, see also Figure 3.1.

For a complete proof, one would need to analyze the failure probabilities in each step. Moreover, one would need to adapt the argument for the first steps when the weights are still small, since the failure probabilities are large during this phase. This is where the condition enters that x and y are from the giant component. Finally, the lower bound can be achieved by a careful first-moment argument showing that the expected number of shorter paths is $o(1)$. □

3.2.4 Variations of the Chung-Lu model

Constant factor deviations

It is easy to see that all computations go through if the formula (3.5) is relaxed a bit. In particular, constant factor deviations do not play a role,

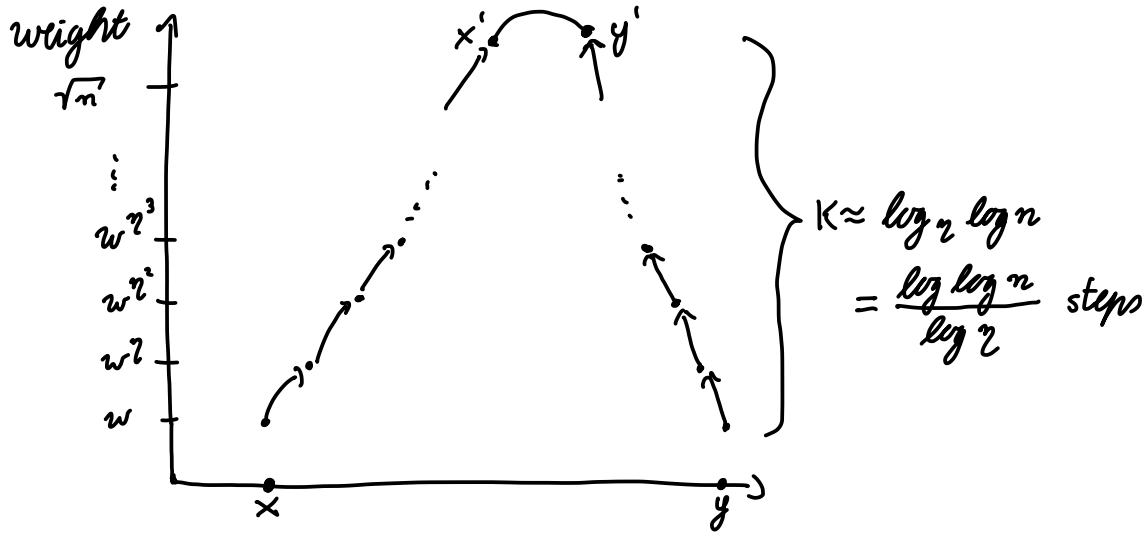

Figure 3.1: In Chung-Lu graphs with $\tau \in (2,3)$, it only takes $(1 \pm o(1))\frac{\log\log n}{\log \eta}$ steps to reach a vertex in the inner core, and all vertices in the inner core are connected to each each other.

so all our calculations remain valid if we only require

$$c_1 \min\left\{1, \frac{w_u w_v}{n}\right\} \leq p_{uv} \leq c_2 \min\left\{1, \frac{w_u w_v}{n}\right\} \tag{3.8}$$

for two universal constants c_1 and c_2. For most of the arguments presented here, constant factors do not matter. An exception is the last step in the proof of Theorem 3.9, where we used that any two vertices of weight at least $n^{1/2}$ are connected with probability one, so they form a single, very large clique. Equation (3.8) only ensures a connection probability of $\Omega(1)$ among them. But this is enough to make this set of vertices so densely connected that any two vertices share a common neighbour. So instead of a clique, we obtain a dense subset of diameter 2.

There are many formulas in the literature that look different but sat-

isfy (3.8), for example

$$p_{uv} = \frac{w_u w_v}{n + w_u w_v}, \tag{3.9}$$

$$p_{uv} = 1 - e^{-w_u w_v / n}, \tag{3.10}$$

$$p_{uv} = \min\left\{1, \frac{w_u w_v}{W}\right\}, \quad \text{where } W = \sum_{z \in V} w_z. \tag{3.11}$$

The last formula falls essentially into this category since there are constants c_1, c_2 such that whp $c_1 n \leq W \leq c_2 n$. There are some subleties: it is no longer true that the events "$u \sim v$" and "$u' \sim v'$" are independent of each other if we condition on the four weights $w_u, w_v, w_{u'}, w_{v'}$. They only become independent after conditioning on *all* weights, since the connections probability also depends on W. In practice, these subtleties do not make a difference.

Other distributions

It is possible to define the Chung-Lu model with a distribution $\mathcal{D}$ which is not a power-law distribution. For example, we can retain the Erdős-Rényi model as the special case where all weights are one. However, it is important that $\mathcal{D}$ has finite expectation, since otherwise the expected number of edges is no longer linear, and it is no longer true that $\mathbb{E}[\deg(v)] = \Theta(w_v)$. In this case, formula (3.11) is often used, but that solves neither of the two problems.

Deterministic Weight Sequences

Instead of drawing the weights randomly in step 1 of Definition 3.5, it is also possible to use a fixed weight sequence. For example, a popular power-law sequence with exponent τ is given by $w_i := (n/i)^{1/(\tau-1)}$. This is also possible for sequences which are not necessarily power-law, and this is the framework in the original work of Chung and Lu. However, the caveats from the last paragraph still apply.

Multigraph Variation

It is sometimes helpful to compare the Chung-Lu model with the following *multigraph variation*. Recall that in a multigraph, we may have several

edges between the same pair of vertices. In a multigraph version, we can get rid of the minimum; we simply require that the *expected number* of edges between u and v is $w_u w_v/n$, and we don't need to cap this formula at one. Since we have only specified the expectation, there are many ways of realizing this variation. The most natural one puts randomly either $\lfloor w_u w_v/n \rfloor$ or $\lceil w_u w_v/n \rceil$ edges between u and v, where the two probabilities are chosen such that the expectation is $w_u w_v/n$. If $w_u w_v/n \leq 1$ (which is true for the vast majority of vertices), then the number of edges is the same Bernoulli random variable as in the Chung-Lu model. Of course, the multigraph variation has more edges (counted with multiplicities), but it can be computed that the number of multi-edges (even with multiplicities) is rather small. Since the formula for the multigraph variation is simpler, it is sometimes easier to understand. For example, the computation in Lemma 3.6 becomes much easier in the multigraph variation:

$$\begin{aligned}
\mathbb{E}[\deg(v)] &= (n-1) \cdot \mathbb{E}[\#\text{ edges between } u \text{ and } v] \\
&= (n-1) \int_1^\infty \Pr[w_u = w] \mathbb{E}[\#\text{ edges between } u \text{ and } v \mid w_u = w] dw \\
&= \Theta(n) \cdot \int_1^\infty w^{-\tau} \frac{w \cdot w_v}{n} dw \\
&= \Theta(w_v) \cdot [w^{1-\tau}]_1^\infty = \Theta(w_v).
\end{aligned}$$

3.3 Perfect sampling: the configuration model

There is another very popular option to obtain random graphs with *any* prescribed degree distribution. Given a sequence $d_1, \ldots, d_n$, we may abstractly consider the set of all graphs on n vertices with $\deg(v_i) = d_i$ for $1 \leq i \leq n$, and draw a graph uniformly at random from this set. Of course, this abstract description by itself is not very helpful. However, there is a simple way to generate such graphs, which can be surprisingly efficient in some situations.

We call a degree sequence $d_1, \ldots, d_n$ *valid* if there is a graph with this degree sequence.[7] Then Algorithm 1 is a way to draw a graph uniformly

[7] Obviously, a necessary condition is that $0 \leq d_i \leq n-1$ and that $\sum_{i=1}^n d_i$ is even. However, this alone is not sufficient. For example, there is no graph which contains at the same time degrees 0 and $n-1$. It is an exercise in graph theory to show that a degree sequence $d_1 \geq \ldots \geq d_n$ is valid if and only if the sum is even and $\sum_{i=1}^k d_i \leq$

at random from that distribution, see also Figure 3.2.

Algorithm 1: Drawing Configurations

Input: valid degree sequence $d_1, d_2, \ldots, d_n$
repeat
  Let G be the empty multigraph on n vertices.
  For $i = 1, \ldots, n$, create d_i half-edges ("stubs") $s_{i,1}, \ldots, s_{i,d_i}$.
  Pick a random perfect matching among the stubs.
  For every matched pair $(s_{i,x}, s_{j,y})$, insert an edge between i and j in G.
  if G *does not have loops or multiple edges* **then**
    return G
  else
    discard G and restart
until;

Obviously, the efficiency of this method depends on the probability that the produced multi-graph G is a simple graph, i.e., that G does not have loops or multiple edges. It turns out that this probability is pretty big if the second moment of the degree distribution is bounded. More precisely, let D be the degree of a random element of the degree sequence (i.e., the degree of a random vertex). If the second moment $\mathbb{E}[D^2]$ is bounded by a constant, then $\Pr[G \text{ simple}] = \Omega(1)$. Intuitively, this makes sense: each stub of vertex i has probability $\frac{d_i - 1}{\#\text{ stubs} - 1}$ to choose another stub of i as partner, so the expected number of loops is given by $\nu := \frac{1}{\#\text{ stubs} - 1} \sum_{i=1}^{n} \binom{d_i}{2}$. If $\mathbb{E}[D^2] = O(1)$, then also $\nu = O(1)$. It can be shown (under some technical conditions) that the number of loops is $\mathrm{Po}(\nu)$-distributed in the limit, so there is a constant probability of having zero loops. A similar argument holds for multiple edges.

While this is a really nice result, for power-laws it excludes the most interesting case. Power-laws have finite second moment if $\tau > 3$, but the more interesting case is $2 < \tau < 3$. In this case, it is an open problem whether a graph with this degree distribution can be sampled efficiently uniformly at random. However, it is possible to just ignore loops and multiple edges (i.e, just delete them from the multigraph). Then we do not get the exact degree sequence that we desire, but one can show that

$k(k-1) + \sum_{i=k+1}^{n} \min\{d_i, k\}$ for all $1 \le k \le n$ (Erdős-Gallai theorem).

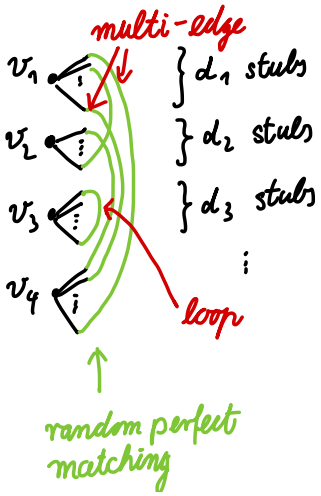

Figure 3.2: In the configuration model, every vertex v_i gets d_i stubs (half-edges), and the stubs are connected via a random perfect matching. It may happen that loops or multi-edges are created.

the number of loops and multiple edges is very small, so we are still close to the target degree sequence. Moreover, by the same argument as above, loops and multiple edges mostly affects vertices of large degree, where one edge fewer might be tolerable.

One reason why the configuration model is popular is that in the case of finite second moments $\mathbb{E}[D^2]$, it can be analyzed with amazing precision. For example, let us call D_2 the degree of a random neighbour u of a random vertex v. It is not hard to see that D_2 only depends on the degree sequence, and to compute its expectation:

$$\mathbb{E}[D_2] = \frac{\mathbb{E}[D^2]}{\mathbb{E}[D]}.$$

This means that we can employ all the machinery that we have learned for Erdős-Rényi graphs in this more general case. In particular, we can couple a local exploration of the configuration model with a Galton-Watson tree with offspring distribution $D_2 - 1$. The "-1" accounts for the fact that the parent is also a neighbour that must not be counted as offspring. In particular, the Galton-Watson tree has positive survival probability if and only if $\mu = \mathbb{E}[D_2] - 1 > 1$, or equivalently

$$\mathbb{E}[D^2] > 2\mathbb{E}[D], \tag{3.12}$$

which is known as the *Molloy-Reed criterion*. In particular, it can be shown that the graph has a giant component if and only if (3.12) is satisfied [Dur07]. Moreover, the expected number of vertices in distance k grows by a factor of μ in each step, starting with $\mathbb{E}[D]$ at the first step. Thus the number of vertices in distance k is in expectation $\mathbb{E}[D] \cdot \mu^{k-1} = \Theta(\mu^k)$. From this equation, it can be derived that the typical distance in the graph model is $(1 \pm o(1)) \log_\mu n$ if $1 < \mu = O(1)$ [vdHHVM05].[8] Other quantities like the component structure can also be derived just as for Erdős-Rényi graphs.

However, as mentioned, all this is restricted to the case $\mathbb{E}[D^2] = O(1)$, and many phenomena are specific to this situation. For example, for finite $\mathbb{E}[D^2]$ the Molloy-Reed criterion (3.12) is a condition about *constant* degrees. If $\mathbb{E}[D^2] > 2\mathbb{E}[D]$, then there exists a constant $C > 0$ such that even the truncated distribution $D_{\leq C} := \min\{D, C\}$ satisfies $\mathbb{E}[D^2_{\leq C}] > 2\mathbb{E}[D]$. Hence, the giant component would also form in the truncated graph where we reduce all large degrees to C (and even if we just remove them from the graph). Likewise, since we can approximate μ to arbitrary precision by a truncated distribution $D_{\leq C}$, the typical distance $\log n / \log \mu$ is a property that arises from vertices of degree at most C. Vertices of larger degree play essentially no role for shortest paths in the graph. Note how different this is from Chung-Lu graphs with exponent $\tau \in (2, 3)$, where shortest paths were obtained by reaching vertices of very weight in $O(\log \log n)$ steps.

A particular case for which $\mathbb{E}[D^2] = O(1)$ are power-law networks with exponent $\tau > 3$ (Chung-Lu or configuration model). In both models, the high-weight vertices do not even help to cut the lengths of shortest paths. Moreover, they do not even play a role for the formation of a giant component: a giant exists if the low-weight vertices are dense enough to form it on their own, and otherwise it does not exist. For most purposes, it is adequate to consider power-law networks with $\tau > 3$ as networks in which there are too few large-degree vertices to affect the global structure. Of course, some things *do* change. For example, power-law networks with $\tau > 3$ still contain vertices of polynomial degree $\approx n^{1/(\tau-1)}$. As a trivial consequence, in the subcritical regime without a giant component, they still contain *components* of size at least $n^{1/(\tau-1)}$, even if they are little more than a star around a central vertex of that degree. Recall that such com-

[8] The reference uses a slight variation of our model here.

ponents do not exist in Erdős-Rényi networks by Lemma 2.6. On the other hand, the supercritical regime (both for $\tau \in (2,3)$ and for larger τ if the Molloy-Reed criterion (3.12) is satisfied) is similar to Erdős-Rényi graphs with $\mu > 1$: the fraction of components of size s decays exponentially in s as in Theorem 2.4. The reason is the same as for the Erdős-Rényi case: in the corresponding Galton-Watson process, if we have k vertices in some layer, then each of them has the same positive probability to become the root of an infinite subtree, and this is independent for all k vertices. So the probability of staying finite is exponentially small in k. (For the configuration model, it is still true that a BFS can be coupled to a Galton-Watson process, though the proof is a bit harder than for the Chung-Lu model.)

3.4 Preferential attachment

There is yet another way to obtain power-law random graphs. This model goes back to Barabási and Albert [BA99] and is known as *preferential attachment* (or simply as Barabási-Albert model). Let $M \geq 1$ be an integer, and $\delta > -M$. We start with a complete graph G_{M+1} on $M+1$ vertices $v_1, \dots, v_M$. Then the remaining vertices are added one by one to the graph, and we denote by G_k the graph of k vertices. In order to go from G_k to G_{k+1}, we add the $(k+1)$-st vertex v_{k+1}, and add exactly M edges from v_{k+1} to the previous vertices $v_1, \dots, v_k$ in G_k. Those M neighbours are chosen randomly, and the probability to choose v_i as a neighbour of v_{k+1} is proportional to $\deg_{G_k}(v_i) + \delta$. I.e., we assign to v_i a probability

$$\frac{\deg_{G_k}(v_i) + \delta}{\sum_{j=1}^{k} (\deg_{G_k}(v_j) + \delta)},$$

and we draw M neighbours from this distribution. If we draw the same vertex twice, then we repeat the second drawing until we find a new vertex.[9]

The idea behind this model is that a new node is more likely to join it as neighbour of popular nodes. The principle is also called *Rich-get-Richer* or the *Matthew effect*.[10] Preferential attachment has become extremely

[9] Some variants also allow multiple edges in this case, which does not make a big difference.

[10] From the bible verse "For to every one who has will more be given, and he will have abundance; but from him who has not, even what he has will be taken away." Matthew 25:29.

popular since it gives an intuitive explanation for where heterogeneous degrees may come from. (Though it is not the only possible explanation.) Intriguingly, the above rule yields a power-law degree distribution.

Theorem 3.10. Let G_n be given by the preferential attachment model with parameters M and δ. Then with high probability the degree sequence of G_n follows a power-law with exponent $\tau = 3 + \delta/M$.

Note that we allow any $\delta > -M$, so we can achieve any power-law exponent $\tau > 2$.

We will not give a proof of Theorem 3.10, but in the following we will try to make it plausible. Firstly, it may seem that the random process is rather unpredictable, but it is not. Rather the opposite, it can be shown that the time of "birth" (i.e., the index k of a vertex) plays almost the same role as the weight for Chung-Lu random graphs, where the "weight" of v_k is $(n/k)^{1/(\tau-1)}$. I.e., whp the number of neighbours of vertex v_k is proportial to $(n/k)^{1/(\tau-1)}$ if n/k is large. This is the same deterministic formula that is sometimes used as a fixed weight sequence in the Chung-Lu model to generate power-law graphs of exponent τ. However, there are also some systematic deviations from the Chung-Lu model. In particular, recall that every vertex receives M edges at birth, so there are no isolated vertices. Moreover, by induction the graph is connected at all times.[11]

So where does the power-law come from? Let us denote the degree of vertex v_i at time t (i.e., when the graph has t vertices) by $D_{i,t}$, and assume that $D_{i,t}$ is large for some t. At this time, the total number of edges is tM. Therefore, when an edge chooses a random endpoint, then the probability $p_{i,t}$ that it chooses v_i is

$$p_{i,t} = \frac{\deg_{G_t}(v_i) + \delta}{\sum_{j=1}^{t}(\deg_{G_t}(v_j) + \delta)} = \frac{D_{i,t} + \delta}{(2M + \delta)t}. \tag{3.13}$$

Now consider the next εt rounds. A total of εtM edges will be added during this time. This means that the denominator of (3.13) changes little

[11] There are also variants where small components can form, similar to Erdős-Rényi graphs. For example, we may not equip a new vertex with *exactly* M edges, but only with M edges *in expectation*, where the exact number may be zero with positive probability. It is mostly a matter of taste whether generating only connected graphs is a bug or a feature of the model.

during this time. Let us momentarily assume that $p_{i,t}$ also stays almost constant in this period. (We can achieve this by choosing ε small enough.) Then the expected number of edges that hit v_i during this time is roughly

$$\varepsilon t M \cdot \frac{D_{i,t} + \delta}{(2M + \delta)t} \approx \varepsilon t M \cdot \frac{D_{i,t}}{(2M + \delta)t} = \frac{\varepsilon D_{i,t}}{2 + \delta/M}. \tag{3.14}$$

So during this phase, the total number of vertices and edges in the graph increases by a factor of $(1 + \varepsilon)$, while the degree increases by a factor of $(1 + \frac{\varepsilon}{2+\delta/M}) = (1 + \frac{\varepsilon}{\tau-1}) \approx (1 + \varepsilon)^{1/(\tau-1)}$. By iterating the argument, we see that this is true for any factor: when the number of vertices increases by a factor C, then the degrees increase by a factor $C^{1/(\tau-1)}$. In particular, if vertex v_k is inserted at time k, then its degree may grow in the time interval from k to n. In this time interval, the number of vertices grows by a factor of n/k (it grows from k to n), and therefore the degree grows from $\Theta(1)$ to $\Theta((n/k)^{1/(\tau-1)})$. In particular, how many vertices are there with degree at least x? Assuming concentration and ignoring the hidden constant factors, it is exactly those v_k for which $(n/k)^{1/(\tau-1)} \geq x$, or equivalently those v_k for which $k \leq x^{1-\tau}n$. There are exactly $x^{1-\tau}n$ such integers k, so the fraction of vertices of degree at least k is $x^{1-\tau}$. This is the cumumlative power-law condition with exponent τ.

Of course, a full proof needs a lot of concentration bounds to make the argument precise, but not much more than that.

The preferential attachment model is compelling for two reasons. Firstly, it gives a possible explanation for the origin of the power law. Secondly, it is dynamic and models how the graph changes over time. For some questions, we would like to work with such models. For example, for some networks there is birth time data available, i.e., data about the time when nodes joined a network. If we want to study related to questions like this, then dynamic models like preferential attachment are the models of choice.

If one is only interested in the *static* network of size n that is generated in the end, and not in the history of the process, then the preferential attachment model is less attractive. The process introduces dependencies and thus makes it very technical to analyze rigorously. In fact, most analyses of the model show that the resulting graph is very similar to a Chung-Lu model or configuration model, and use this connection to prove that the corresponding statements transfer to the preferential attachment model. (Just mind the obvious differences in component structure and low-degree

vertices since the minimum degree is M.) As usual, there is no "best" model, and it depends on the question of interest which model to choose.

Preferential attachment networks have a rather rigid dynamics. On the one hand, this is very helpful for analyzing them. On the other hand, the dynamics are too rigid to match real evolving networks well, like the web graph (webpages and hyperlinks) or citation networks (scientific papers and citations). To understand the problem, let us assume in the preferential attachment model that the degrees of two vertices u, v differ by a constant factor at time t, e.g. $\deg_t(u) \geq 2\deg_t(v)$. If both u and v are large[12] then this will stay true throughout the whole process: it is very unlikely that v will ever overtake u. In particular, the final degree is mostly determined by the age of the vertex, and the highest degrees are obtained by the oldest vertices. In real-world networks, the age of a node (a webpage, a paper, ...) is certainly *correlated* with its degree, but the correlation is much weaker than in preferential attachment networks. This limitation can be overcome by combining the ideas of Chung-Lu random graphs and preferential attachment. In this model, the graph is still generated vertex by vertex, but each vertex also obtains a weight. The connection probability to vertex v is then a function of the degree of v (at time t) and of the weight of v. The resulting graphs are similar to ordinary preferential attachment graphs, but the dynamics are more realistic. We do not go into further detail. A discussion can be found in [LNR17, Chapter 6.5].

3.5 Strengths and weaknesses of the Chung-Lu model

3.5.1 Strengths

The obvious advantage of the Chung-Lu model and the configuration model is that they can model the power-law degree distributions that we see in real-world networks. We have seen that there are strong structural difference between the case $\tau \in (2,3)$ and $\tau > 3$. In the latter case, the large-weight vertices are too sparse to change the global structure of the network. Most real-world networks seem to have exponents in the range $\tau \in (2,3)$.

[12] More precisely, the error probability of the following statement is exponentially small in $\deg_t(u)$.

The models can help us understand some effects that are closely related to the degrees. For example, we understand better why our friends are much more popular than we are. While this effect is hardly existent in Erdős-Rényi graphs, it becomes extreme for power-law exponents $\tau \in (2,3)$: while we have a constant number of friends, our friends have in expectation an infinite number of friends, but this expectation is dominated by low-probability events. This also helps us understand some confusing effects: if we try to empirically investigate a distribution with infinite mean, then averages from finite samples tend to be dominated by a few outliers (the largest-degree nodes in the sample), and are often rather inconsistent with each other.

We have also seen that a power-law with exponent $\tau \in (2,3)$ leads to *ultra-small worlds*. It is up to debate whether typical distances are rather like $\log n$ or like $\log\log n$, but we have also seen that the structural properties of shortest paths change dramatically, since all shortest paths go through a small set of large-degree vertices. These properties are important. For example, routing protocols which use shortest paths will put a very high load on those high-degree vertices in Chung-Lu random graphs, while the load is rather uniformly distributed in Erdős-Rényi random graphs.

3.5.2 Weaknesses

The biggest weakness of Chung Lu and configuration models is that they have no clustering or community structure. The clustering coefficient is easily seen to be $o(1)$. It is slightly larger than for Erdős-Rényi graphs because of the skewed neighbourhood degree distribution: the neighbourhood of a vertex v has an increased probability to contain vertices of large weight, and those are more likely to connect to each other. However, the effect is not large: most neighbours of v are still of constant weight, and those still have probability $\Theta(1/n)$ to connect to each other.

This lack of triangles extends to larger cycles, to cliques, and to other small and dense subgraphs. The networks look locally tree-like (which is essentially the same as saying that they are well described by Galton-Watson processes). In particular, if we start from a random vertex and explore the graph, usually we obtain in the first rounds a subgraph of k vertices which is a tree. Thus is has only $k-1$ edges and has minimal density. However, it can be tricky to find the proper statistics here. For

example, if we try to simply count the number of connected subgraphs of size k which have more than $\alpha \cdot k$ edges for some $\alpha > 1$, then this may yield a large number. The vertices of weight at least $\sqrt{n}$ form a gigantic clique, and this contains very many subcliques of size k, all of which are counted in the statistics. Still, the networks are considered not to have community structures, even though measurement can be tricky.[13] This lack of communities is the weakest point of the models.

[13] In fact, the configuration model is sometimes used as baseline, and other networks are said to have community structure if they have more densely connected subgraphs than the corresponding configuration model with the same degree distribution.

Chapter 4

Geometric Graphs

We have seen models that can create a power-law degree distribution, or even an arbitrary degree distribution. However, the models so far could not provide high clustering or community structures. The most classical example of graphs with strong community structures are *grids*. The simplest one is $\Gamma_{s,d} := \{1, \dots, s\}^d$, where each vertex is adjacent to the $2d$ vertices of Euclidean distance 1. (It is exactly $2d$ neighbours if we use the *torus topology* by treating the numbers 1 and s of $\{1, \dots, s\}$ as neighbours. Otherwise boundary vertices have fewer neighbours.) The dimension d is usually considered a constant, most often a small constant like $d = 1, 2, 3$. In a grid it is easy to find "communities" of K vertices such that there are only $O(\sqrt{K})$ edges out of the community: just pick a ball or cube of volume K, and declare the grid points inside the box as a comunity.

The grid defined above leads to a *bipartite* graph. This is awkward since the most prominent measure for locality is the clustering coefficient, which is zero for bipartite graphs. (In bipartite graphs, two neighbours of the same vertex are never adjacent.) To circumvent this, two options are commonly used: either we use a *hexagonal grid* instead of a *square grid*. Or we use an ordinary square grid $\Gamma_{s,d}$, but connect any two points in Manhattan distance[1] at most r for some constant $r > 1$. This is also called the r-*th power* of the grid.

Beware that such patches are rather a proof-of-concept that should not be over-interpreted. The clustering coefficient is one way to measure a much

[1]The Manhattan distance between two vertices x and y, also called L_1-distance, is the graph distance between x and y in the grid graph without random edges. If we don't use torus topology, then it can be computed as $\sum_{i=1}^{d} |x_i - y_i|$.

more fundamental property of real-world networks, *locality*. The problem is that we do not really know what exactly we mean by locality, so we measure it by auxiliary measures. The clustering coefficient is one of them, and the hexagonal lattice happens to behave much better with respect to this measure than the square lattice. However, this does not necessarily mean that the hexagonal lattice is a better representative of real-world networks than the square lattice. We should be careful not to overfit to a single auxiliary measure. In this case, we can easily see this by switching to other auxiliary measures. For example, real-world networks have also many K_4 as subgraphs, and those exist neither in the square lattice nor in the hexagonal lattices. Still, we will see some applications where either type of lattice models locality just fine.

As an alternative, for $d \geq 2$ it is possible to use *Random Geometric Graphs* instead of grids. In this setting, n vertices are randomly placed in a d-dimensional cube of volume n (with or without torus topology). Then two vertices are connected if and only if they have distance at most r, where r is a parameter of the model. If $d \geq 2$ and r is a sufficiently large constant, then one can show that the graph has a giant component, and the remaining components show a stretched exponential tail bound as for Erdős-Rényi graphs.[2]

4.1 Weak Ties and the Watts-Strogatz Model

The Watts-Strogatz model was designed to demonstrate that even a minimal change to a rigid geometric model can lead to small worlds.

Definition 4.1. The *Watts-Strogatz random graph model* starts with a d-dimensional grid with torus topology for some constant $d \geq 1$, i.e., the vertex set is $\{1, \dots, s\}^d$ and the edges wrap around from s to 1 in each coordinate. Then for every pair u, v of different vertices,

[2] *Stretched exponential* means that the fraction of vertices in component of size s is at most $\eta^{(s^\varepsilon)}$ for some constants $\eta < 1$ and $\varepsilon > 0$. This is still a very fast decaying function in s, though not quite as fast as an exponential function. Arguably, this type of tail bound is even more plausible than a proper exponential, since the non-giant components in real-world networks are small, but not quite as small as an exponential tail would suggest.

with probability p we add the edge from u to v.

The original construction by Watts and Strogatz [WS98] differed in some detail. It started with the r-th power of a grid and *redirected* edges instead of adding them. This makes the analysis a bit more complicated, but does not change the result. Also, the original model was for dimension $d = 1$. We refer to the additional edges as *random edges* and to the original edges as *grid edges*. For symmetry reasons, it is convenient to also make a coin flip for grid edges, even though "adding" an already existing grid edge does not change the graph. So, an edge can be both a random edge and a grid edge at the same time.

The general idea of the Watts-Strogatz model (and of similar models that we will see later) is very popular because it matches an important concept in sociology: *weak ties*.[3] This concept describes edges which exist without obvious reasons: the endpoints seem rather unrelated, and they are not part of the same clusters and communities. Sociological experiments have shown that such weak ties are very important for the connectivity of social networks, for work careers and for processes like the spread of behaviour through the networks [Cen10]. The distinction into strong and weak ties matches nicely with the grid edges and random edges in the Watts-Strogatz model.

We will be interested in typical distances in the Watts-Strogatz model. Obviously, for $p = 0$, without random edges, typical distances are of order $\Theta(n^{1/d})$.[4] The surprising insight is that even for very small values of p, the distance decreases dramatically.

Theorem 4.2. Let G be a Watts-Strogatz random graph on n vertices with parameter $p = p(n)$ satisfying $p = \omega(1/n^2)$ and $p = O(1/n)$. Then the typical distances in G are of order $\Theta(\log(n^2p)/(np)^{1/d})$.

Proof. We will only prove the upper bound. Partition the grid into $n' := n^2p/2$ cubes of volume $U := 2/(np)$, i.e., the side length of the cubes is $U^{1/d}$. We will ignore rounding issues and assume for simplicity that $U^{1/d}$

[3] The paper *The strength of weak ties* by Mark Granovetter [Gra73] is the most cited paper in sociology of all times.

[4] In the sense that for every $\varepsilon > 0$ there are $c_1, c_2 > 0$ such that two randomly chosen vertices u, v have distance $c_1 n \leq d(u, v) \leq c_2 n$ with probability at least $1 - \varepsilon$.

is an integer. By construction, U is the number of vertices in each block, and by the assumption on p we have $U = o(n)$ and $U = \Omega(1)$.

Consider the graph $G' = (V', E')$ where the vertex set is the set of cubes, and there is an edge between two cubes $C_1, C_2 \in V'$ if in G at least one random edge was added between a vertex $u \in C_1$ and a vertex $v \in C_2$, see also Figure 4.1. Note that the edges in G' are formed independently of each other, since the random edges in G are placed independently. Thus G' is an Erdős-Rényi graph $G_{n',p'}$, where $n' = n/U$ is the number of vertices in G'. We will next show that $p' \geq 1.5/n'$. Note that this is plausible since the expected number of random edges in G that are added to vertices in a cube C is $\approx U \cdot (n-1) \cdot p = 2$, and these edges target random cubes. If you are already convinced by this argument, you can skip the next paragraph.

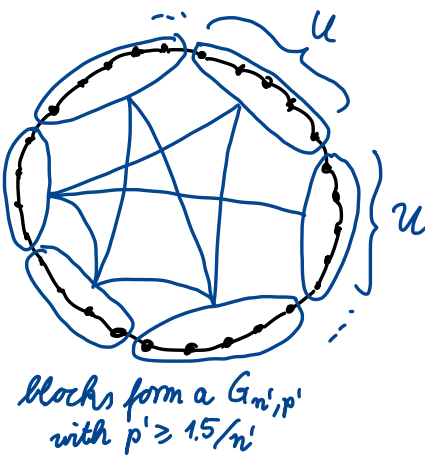

Figure 4.1: Partitioning into blocks for $d = 1$. The induced graph between blocks is a supercritical Erdős-Rényi graph on n' vertices. Thus it has typical distance $O(\log n')$.

Otherwise, for two blocks $C_1, C_2 \in V'$, there are U^2 pairs of vertices (v_1, v_2) with $v_1 \in C_1$ and $v_2 \in C_2$. Hence, the probability that no random edge is added between C_1 and C_2 is

$$\Pr[C_1C_2 \notin E'] = (1-p)^{U^2} \approx e^{-pU^2} = e^{-4/(pn^2)} \overset{(*)}{\approx} 1 - \frac{4}{pn^2} = 1 - \frac{2}{n'},$$

where in step $(*)$ we have used that $pn^2 = \omega(1)$. Hence, for large n we have $p' \geq 1.5/n$ and the Erdős-Rényi graph G' is *supercritical*. In particular,

this means that the typical distances in the giant component of G' are $\Theta(\log n')$.

Now we consider two random vertices $v_1, v_2 \in V$ and bound their distance. For simplicity, let us assume that their corresponding cubes C_1, C_2 are in the giant component in G'. (Otherwise we can walk along the grid to cubes which are in the giant.) Then we need to use $O(\log n')$ edges in G' to connect C_1, C_2 via a path π'. This is not yet a path in G, since we typically enter and leave a cube C on π' in two different vertices u_1, u_2. But since all the cubes have side-length $O(U^{1/d})$, we can walk from u_1 to u_2 along the grid in $O(U^{1/d})$ steps. Thus we can walk from v_1 to v_2 in G in $O(U^{1/d} \cdot \log n')$ steps. □

Let us take a moment to appreciate how quickly the distances in Theorem 4.2 decrease even for small p. For example, if $p = n^{-2+\varepsilon}$ then we only add $O(n^{\varepsilon})$ random edges to the graph, a ridiculously small number compared to the $\Theta(n)$ grid edges. But the typical distance is already significantly reduced, from $\Theta(n^{1/d})$ to $\Theta(n^{(1-\varepsilon)/d} \log n)$. As a second example, for $p = 1/(n \log n)$ we add only $O(n/\log n)$ vertices to the graph, so only $O(1/\log n) = o(1)$ edges per vertex. So most vertices do not receive an extra edge. Nevertheless, these few extra random edges are sufficient to bring down the typical distance to $\Theta((\log n)^{1+1/d})$. In general, we see that even if we start with a very rigid graph with high distances (for $d = 1$ we have the cycle and distances of $\Theta(n)$!), we only need to add a tiny number of random edges to decrease the distances dramatically.

4.2 Navigatibility and the Kleinberg model

We have seen several network models which are small or even ultra-small worlds, in particular the Erdős-Rényi, Chung-Lu and Watts-Strogatz model. However, a serious of famous experiment by psychologist Stanley Milgram[5] in the 60s and 70s showed that social networks are not only small worlds, it is also possible to *navigate efficiently* with local knowledge in these

[5]The same Stanley Milgram who performed the *Milgram experiment* in which the majority of test subjects continued to seemingly torture a fellow test person to death, simply because a scientist in a lab told them that this is how the protocol goes. If you don't know about this experiment, you should read about it: `https://en.wikipedia.org/wiki/Milgram_experiment`.

networks. In the most famous experiment, Milgram gave a letter to some person A in the US Midwest that was addressed to some person B at the US East Coast. However, A was not allowed to send the letter directly to B. Instead, A was only allowed to send the letter to a personally known contact A', defined as someone whom A knew on a first-name basis. A was supposed to pick a neighbour A' who was more likely to know the target B. Then A' continues the process in the same manner, i.e., A' sends the letter to a personal contact whom she knew on a first-name basis, and so on until the chain fails or the letter reaches a person who knows B personally. Thus the participants were only allowed to navigate within the friendship graph, and they had to make their decision without global knowledge of the network.

The experiment showed that a portion of letters reached their targets,[6] and reached them within just a few steps. Since one of the experiments found an average of less than 6 intermediate steps, the result of the experiment became widely known as *six degrees of separation*.

So, it is possible to navigate the friendship network using only local information. I.e., every node v only knows the location of the target[7] and some basic information about the direct neighbours of v (what you know about your friends; address, popularity, profession, ...). This information suffices to navigate the network efficiently. In abstract terms, we would assume that a vertex knows the geometric location and possibly some intrinsic properties of its neighbours, but nothing else about the network.

This type of efficient navigation is not possible in any of the networks that we have seen so far. In Erdős-Rényi graphs, none of your neighbours is better than any other, so by symmetry it is clear that $\Omega(n)$ nodes need to be visited. In Chung-Lu graphs, the shortest path runs through vertices of very high weight, and it *is* possible to find those efficiently: simply go to the neighbour of largest weight. But for the second half of the path, one would need to navigate from large-degree vertices down to small-degree vertices, and this can not be done efficiently. Finally, the shortest paths in the Watts-Strogatz model use that the random edges turn the graph G' of supernodes into a supercriticial Erdős-Rényi graph, which has small diameter. However, the following theorem shows that these shortcuts are

[6] The success rate in the first experiments was only 5-30%, but could be increased to up to 85% in later variations where letters were replaced by phone and email.

[7] In the original experiment, the participants were also told the job of the target.

not too helpful for navigation.

Theorem 4.3. Consider the Watts-Strogatz model on n vertices with dimension d and parameter p, where $p = \omega(1/n^2)$ and $p = \Omega(1/n)$. Then any local navigation algorithm that routes from a random vertex s to a random vertex t takes at least $\Omega(p^{-1/(d+1)})$ steps in expectations, which is in particular $\Omega(n^{1/(d+1)})$. Moreover, the greedy routing algorithm which always proceeds to the/a neighbour closest to t needs $O(p^{-1/(d+1)})$ steps in expectation.

Proof. For the lower bound, we may assume that s and t have Manhattan distance at least $\Delta := \frac{1}{2}p^{-1/(d+1)}$, since this is asymptotically smaller than the side length of the grid. Consider the event $\mathcal{E}$ that during the first Δ steps, the algorithm does not uncover a random edge whose endpoint has Manhattan distance at most Δ from t. First we show that conditional on $\mathcal{E}$, it is impossible for the algorithm to reach t in Δ steps. Consider the last random edge e that the algorithm takes during those Δ steps. Then after taking this edge, the algorithm has Manhattan distance more than Δ from t. By definition of t, it only uses grid edges afterwards, so it does not reach t in Δ steps. The same applies if the algorithm does not take any random edges at all during the first Δ steps.

Next we show $\Pr[\mathcal{E}] \geq 1/2$. Note that this will conclude the proof of the lower bound, since it implies that the expected number of steps is at least $\Delta/2$, as required. The crucial insight is that since the random edges are uniformly at random, it does not matter which path the algorithm takes during the first Δ steps. By symmetry, the probability of $\mathcal{E}$ does not depend on the *set* of explored vertices, but only on the *number* of explored vertices. So let us compute $\Pr[\mathcal{E}]$.

Let us consider the ball around t of radius Δ with respect to the Manhattan distance. It contains less than Δ^d vertices. Thus, when we explore a new vertex v, the probably to find a random edge into this Manhattan ball is at most $p \cdot \Delta^d \leq \Delta^d/n$ by a union bound. By another union bound over the first Δ steps, the probability that this happens in any of those steps is $\Pr[\mathcal{E}] \leq p\Delta^{d+1} = 2^{-d-1} \leq 1/2$. This concludes the proof.

For the upper bound, we compute the time until we reach Manhattan distance Δ from t. We pessimistically assume that we need to wait for a random edge into that region. Since the Manhattan ball has size $\Omega(\Delta^d)$,

in each step we have probability of $\Omega(p\Delta^d)$ of discovering such an edge. Hence, the expected time until we find such an edge is $O(p^{-1}\Delta^{-d}) = O(\Delta)$. Afterwards, we need at most Δ more steps to proceed to the target. Thus the expected number of steps is $O(\Delta)$. □

To make Theorem 4.3 more concrete, for $d = 1$ and $p = 1/n$ the lower bound is $\Omega(\sqrt{n})$ steps, even though the typical distances are only $O(\log n)$. Random edges are not completely useless. Without them, the typical distance would be $\Omega(n)$. However, navigation is much less efficient than shortest paths. In general, for any dimension d and any p in the specified range, the time for local navigation is always polynomial in n, while the typical distance may be polylogarithmic for some values of p.

In 2000, Jon Kleinberg proposed a model in which routing is possible in poly-logarithmic time [Kle00]. The main difference to the Watts-Strogatz model is that edges are no longer placed uniformly at random, but rather the probability for placing an edge depends on the distance of the two endpoints.

Definition 4.4. The *Kleinberg random graph model* starts with a d-dimensional grid with torus topology for some constant $d \geq 1$. Then for every pair u, v of different vertices, with probability

$$p_{uv} := \frac{1}{\log n} \cdot \frac{1}{d_1(u,v)^d} \tag{4.1}$$

we add the edge from u to v, where $d_1(u,v)$ denotes the Manhattan distance between u and v.

As for the Watts-Strogatz model, we call the additional edges *random edges*, and the original edges *grid edges*.

To understand the model a bit better, let us first compute how many edges we add to a fixed vertex v. We fix an exponentially growing set of radii $R = \{1, 2, 4, 8, \ldots, 2^s\}$ and sort the other vertices into annuli $A_r(v)$ for $r \in R$, where $A_r(v) := \{u \in V \mid r \leq d_1(u,v) < 2r\}$. Since the diameter of the grid is $dn^{1/d}/2$, we need $s = \lfloor \log_2(dn^{1/d}/2) \rfloor = \Theta(\log n)$ radii.

The annulus $A_r(v)$ contains $\Theta(r^d)$ vertices. Moreover, the vertices in $A_r(v)$ have distance $\Theta(r)$ from v. Therefore, the expected number of neigh-

bours of v in $A_r(v)$ is

$$\mathbb{E}[u \in A_r(v) \mid u \sim v] = |A_r(v)| \cdot \frac{1}{\log n} \cdot \frac{1}{\Theta(d_1(u,v)^d)} = \Theta(1/\log n). \quad (4.2)$$

So v has the same expected number of neighbours "in each distance", namely $1/\log n$. (Plus the $2d$ neighbours from the grid.) Since we have $\Theta(\log n)$ different distances r, the total number of random edges incident to v is $\Theta(1)$. In particular, the expected degree is still $O(1)$.

The neighbours of v are not uniformly at random over all vertices, but they are (approximately) uniform over distances. Every distance range has the same probability $\Theta(1/\log n)$ to provide a neighbour. Moreover, if we fix an annulus $A_r(v)$, then all vertices in $A_r(v)$ have roughly the same probability to connect to v, up to a factor of $2^d = \Theta(1)$. Rephrased, if v has a neighbour in $A_r(v)$, then this neighbour is uniformly distributed in $A_r(v)$ up to constant factors, meaning that for each potential neighbour, the probability to be the actual neighbour is the same as in the uniform distribution up to a constant factor. This double form of equidistribution allows for efficient greedy routing.

Theorem 4.5. Consider the Kleinberg model on n vertices with dimension d. Consider the greedy routing algorithm with start vertex s and target vertex t, which always proceeds to the/a neighbour closest to t. Then for random s and t the the greedy routing algorithm takes $O(\log^2 n)$ steps in expectations.

Proof sketch. Assume that the algorithm is at a vertex with Manhattan distance Δ from t. We will show that after an expected $O(\log n)$ steps, we find a random edge that reduces the distance from t to $\frac{3}{4}\Delta$. We call such an edge a *shortcut*. Repeating the same argument $O(\log n)$ time, we reduce the Manhattan distance to a constant, which yields the runtime $O(\log^2 n)$.

So we need to show that we quickly find a random edge that reduces the distance to $\frac{3}{4}\Delta$. We will ignore some technical details, in particular that the distance to t can change while we search for the shortcut. Moreover, we assume that Δ is a power of 2. Let $r := \Delta/2$. Then by (4.2), each explored vertex v has probability $\Theta(1/\log n)$ to have a neighbour $u \in A_r(v)$. Note that a constant portion of $A_r(v)$ lies in the Manhattan ball $B_{3\Delta/4}(t)$ around t, as illustrated in Figure 4.2. Since the location of u within $A_r(v)$ is

uniform up to a constant factor 2^d, the probability that u lies in $B_{3\Delta/4}(t)$ is $\Omega(1)$. Summarizing, each explored vertex v has probability $\Theta(1/\log n)$ to have a neighbour $u \in A_r(v)$, and this neighbour has constant probability to be in $B_{3\Delta/4}(t)$. Therefore, we need to wait $O(\log n)$ in expectation to find a neighbour in $B_{3\Delta/4}(t)$, as required. □

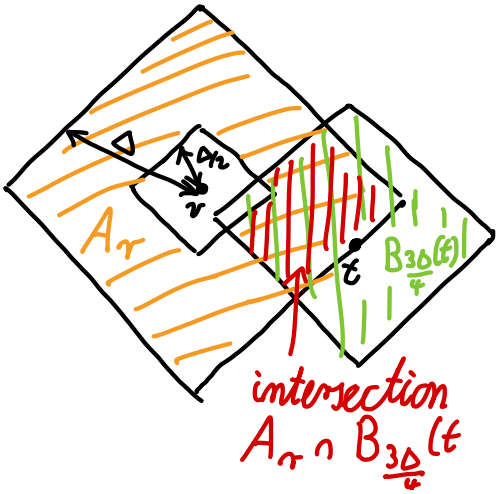

Figure 4.2: The distance between v and t is Δ. The annulus $A_r(v)$ contains all points in distance $[\Delta/2, \Delta)$ from v, and the ball $B_{3\Delta/4}(t)$ contains all vertices in distance at most $3\Delta/4$ from t. Both have volume $\Theta(\Delta^d)$, and their intersection has also volume $\Theta(\Delta^d)$. (Balls in Manhattan distance look like diamonds, but the conclusion would also hold for any other.)

It can be shown that the bound in Theorem 4.5 is tight, so navigation in the Kleinberg model takes time $\Theta(\log^2 n)$.

4.2.1 Shortcomings of the Kleinberg model

The Kleinberg model gives a nice and simple model for a network that can be efficiently navigated. Moreover, it can be navigated by the *greedy algorithm*, which is rather similar to the description of Milgram's small-world experiments. However, there are also some shortcomings of the model.

Firstly, as Kleinberg himself pointed out, the model is rather vulnerable with respect to the exponent d. Any other exponent than d will lead to a highly imbalanced distribution of neighbours over different distances. An

exponent $d + \varepsilon$ leads to a much smaller number of long edges, so that the algorithm can not cover the long distances at the beginning of routing in reasonable time. An exponent of $d - \varepsilon$ gives too many long edges; in this case the normalization factor $1/\log n$ in (4.1) needs to be replaced by a polynomial factor, or the degrees would become polynomial in n. But then the number of short random edges is too small, so that greedy routing does not accelerate the second part of the routing process, as in the Watts-Strogatz model. For either $d + \varepsilon$ and $d - \varepsilon$ (and every other choice except d), we end up with a *polynomial* runtime for greedy routing. Thus the model is rather brittle with respect to the scaling in (4.1).

Secondly, the model relies quite heavily on the underlying grid structure. With a grid, it can never happen that the greedy algorithm gets stuck. If we replace the grid with a Random Geometric Graph, or if some grid edges are missing, then it may happen that the algorithm enters a vertex v that has no neighbour which is closer to the target than v. If v still forwards the message to the best neighbour, then this could result in v sending the message *back* to its predecessor, and the algorithm might enter an infinite loop. It can be seen that such problems occur in each step with constant probability, so the probability that they happen in at least one of the $\Theta(\log^2 n)$ steps is very large. So once we replace the grid with a Random Geometric Graph, the success rate of the algorithm becomes $o(1)$. (In fact, it goes to zero very fast, polynomially in n.) On the other hand, it could be argued that this problem might not show in real social networks due to the large average degree. Either way, the theoretical model and analysis relies rather heavily on the perfect grid structure.

Thirdly, the while the runtime $O(\log^2 n)$ is poly-logarithmic, it still seems rather large compared to the very small paths found in real networks. The famous six degrees of separations correspond to a path length of seven, and other experiments found even shorter paths. One possible explanation is that the degrees in real social networks are much larger than the degrees in Kleinberg's model, so that the leading constants are rather small. Alternatively, in the final part of the lecture we will see a model in which greedy routing succeeds in even shorter time.

Finally, there is a non-trivial asymptotic *stretch*, which is defined as the ratio between the path that greedy routing finds, and the graph-theoretic shortest path. The reason is that greedy routing only finds shortcuts by accidentally discovering them. We have argued that it needs to

explore $\Theta(\log n)$ vertices to find a shortcut. However, for constructing shortest paths, with global knowledge of the network, we can simply go to the nearest shortcut. The nearest shortcut is typically in Manhattan distance $\Theta((\log n)^{1/d})$, because there are $\Theta(\log n)$ vertices in this Manhattan distance, and each has probability $\Omega(1/\log n)$ to provide a shortcut. So, instead of blindly exploring $\Theta(\log n)$ vertices, global knowledge enables us to immediately go to a shortcut within $\Theta(n^{1/d})$ steps. We still need to repeat this $O(\log n)$ times, so shortest paths are of length $O((\log n)^{(1+1/d)})$. Thus the stretch is at least $\Omega((\log n)^2/(\log n)^{1+1/d}) = \Omega((\log n)^{1-1/d})$. Rephrased, this means that if the shortest path has length $\ell = O((\log n)^{1+1/d})$, then greedy routing needs $\Omega(\ell^{2d/(d+1)})$ steps. For example, for $d = 2$ greedy routing needs $\ell^{4/3}$ steps, where ℓ is the length of a shortest path. The stretch is hard to measure for real-world networks since we can not take a limit $n \to \infty$, and constants are hard to distinguish from logarithmic factors or even from n^{ε}. Still, it seems that the Kleinberg model has a rather large stretch, especially when phrased in terms of the distance ℓ.

Chapter 5

Geometric Inhomogeneous Random Graphs (GIRGs)

5.1 The GIRG model: basic properties

In this chapter we will introduce a model which combines the Chung-Lu model with geometry. This allows us to get the best aspects from both models. However, as we will see, we also actually get additional features from the combination that are explicitly build into the model.

Definition 5.1. Let $\alpha > 1$, $d \in \mathbb{N}$ and let $\mathcal{D}$ be a power-law distribution on $[1, \infty)$ with exponent $\tau \in (2, 3)$. Let $\mathcal{X}$ be a d-dimensional cube of volume $n \in \mathbb{N}$ with torus topology. A *Geometric Inhomogeneous Random Graph (GIRG)* $G = (V, E)$ on n vertices is obtained by the following three-step procedure.

(a) Every vertex v draws independently a weight w_v from distribution $\mathcal{D}$.

(b) Every vertex v draws indepdendently a uniformly random position $x_v \in \mathcal{X}$.

(c) Every two different vertices $u, v \in V$ are independently connected by an edge with probability

$$p_{uv} := \min\left\{1, \frac{w_u w_v}{\|x_u - x_v\|_\infty^d}\right\}^\alpha. \tag{5.1}$$

The idea behind the model is that the geometric position captures properties and categories of the nodes. In social networks, this might be profession, place of living, or hobbies. The weight captures the popularity of a node. The connection probability increases with the popularity of the nodes, and is larger for nodes which are geometrically close to each other.

There are some choices in (5.1) which do not immediately have an obvious reason. One is the appearance of the exponent α. Another is why the distance $\|.\|$ should have exactly exponent d. We will return to both of these question later in more detail (Sections 5.1.1 and 5.5). We will also see that we could have used any other norm. The only reason to use the ∞-norm is that formulas look a little bit nicer, since any two points in $\mathcal{X}$ have distance at most $n^{1/d}$. In the following, we will omit the index ∞ from the norm and simply write $\|x_u - x_v\|$.

But first we will show that formula (5.1) yields the same marginal probabilities as the Chung-Lu model.

Lemma 5.2. Let $u, v \in V$ for a GIRG. Assume we have drawn w_u, w_v and $x_u \in \mathcal{X}$, but that x_v is still random. Then the probability that u and v are connected is

$$\Pr[u \sim v \mid w_u, w_v, x_u] = \Theta\Big(\min\Big\{1, \frac{w_u w_v}{n}\Big\}\Big).$$

This probability is also known as the *marginal connection probability* of u and v.

Proof. First note that if $w_u w_v \geq n$, then $w_u w_v \geq \|x_u - x_v\|^d$ for all $x_v \in \mathcal{X}$ since the diameter of $\mathcal{X}$ is $n^{1/d}$. Hence, no matter which x_v we draw, u and v are always connected, and $\Pr[u \sim v \mid w_u, w_v, x_u] = 1$ as required. So, in the following we may assume $w_u w_v \leq n$ and the minimum on the right hand side is taken by the second term.

The probability density of the event "$\|x_u - x_v\| = r$" is $\Pr[\|x_u - x_v\| = r] = \Theta(r^{d-1})$, since the ball of radius r around x_u has surface area $\Theta(r^{d-1})$. (This is also true for the ∞-norm.[1]) Therefore, we can compute the marginal

[1] We are slightly imprecise when r approaches the diameter $n^{1/d}$ of $\mathcal{X}$, since there are wraparound effects. We will ignore this complication here.

probability as

$$\begin{aligned}\Pr[u \sim v \mid w_u, w_v, x_u] &= \int_0^{n^{1/d}} \Pr[\|x_u - x_v\| = r] \cdot \Pr[u \sim v \mid w_u, w_v, r] dr \\ &= \int_0^{n^{1/d}} \Theta(r^{d-1}) \cdot \min\left\{1, \frac{w_u w_v}{r^d}\right\}^\alpha dr \\ &= \int_0^{(w_u w_v)^{1/d}} \Theta(r^{d-1}) \cdot \min\left\{1, \frac{w_u w_v}{r^d}\right\}^\alpha dr \\ &\quad + \int_{(w_u w_v)^{1/d}}^{n^{1/d}} \Theta(r^{d-1}) \cdot \min\left\{1, \frac{w_u w_v}{r^d}\right\}^\alpha dr. \quad (5.2)\end{aligned}$$

Note that the splitting point $(w_u w_v)^{1/d}$ lies in the integration range $[0, n^{1/d}]$ because $w_u w_v \leq n$. In the first integral, the minimum is taken by 1, in the second integral the minimum is taken by $w_u w_v / r^d$. Hence, (5.2) simplifies to

$$\int_0^{(w_u w_v)^{1/d}} \Theta(r^{d-1}) \cdot 1 dr + \int_{(w_u w_v)^{1/d}}^{n^{1/d}} \Theta(r^{d-1-d\alpha}) \cdot (w_u w_v)^\alpha dr.$$

The integration variable is r, and it has exponent $d - 1 > -1$ in the first integral and exponent $d(1-\alpha) - 1 < -1$ in the second integral. By Excursion 3.1, this means that we only need to evaluate the first integral at the upper boundary and the second integral at the lower boundary. Moreover, from the excursion we also know *a priori* that both evaluations will give the same term up to constant factors.[2] Since the first integral looks simpler, we only evaluate that and obtain

$$\Pr[u \sim v \mid w_u, w_v, x_u] = \Theta([r^d]_{(w_u w_v)^{1/d}}) = \Theta(w_u w_v),$$

as required. □

Lemma 5.2 has important consequences. It means that when we fix all information about a vertex u (its weight and position), then all other vertices v have the same probability of connecting to u as in the Chung-Lu model, up to constant factors. And of course, once the weight w_u is fixed, any two different vertices v_1 and v_2 have *independent* chances of connecting to u. In particular, this means that the degree of u in the GIRG model follows exactly the same distribution as in the Chung-Lu model (all up to

[2] If you don't believe it, compute both and see the magic happen.

constant factors, which we ignore for now), namely a Binomial distribution with expectation $\Theta(w_u)$. In the limit for $n \to \infty$, this converges to a Poisson distribution.

Moreover, the weight distribution of a random neighbour of u is also the same as in the Chung-Lu model, so it follows a power-law with exponent $\tau - 1$. Since this is independent of w_u, the GIRG model has, as the Chung-Lu model, neutral assortativity with respect to vertex weights. However, this does not directly translate into neutral assortativity with respect to *degrees*, especially for low-degree vertices. Consider two neighbours u and v of (constant) weights w_u and w_v. This yields expected degrees $E_u = \Theta(w_u)$ and $E_v = \Theta(w_v)$, respectively. If someone tells us $\deg(u)$, and this happens to be untypically large compared to E_u, what can we infer about $\deg(v)$? Of course, nothing is certain. But one potential reason for the large degree of w_u is that it has many *strong ties*, i.e., that an untypical number of vertices have positions close to x_u. In this case, *if* x_v is also close to x_u then x_v also has untypically many vertices nearby, and thus x_v might also have an untypically high degree. There are many if's in this reasoning, but we will see later that indeed most neighbours are geometrically close to each other, so that this does give a notable positive correlation between $\deg(u)$ and $\deg(v)$, at least for small degrees. So while GIRGs have neutral assortativity with respect to *weights*, they have positive assortativity with respect to *degrees*.

Of course, for large weights the degree is highly concentrated around its expectation, because a Poisson distribution with large expectation is concentrated. Hence, the *heaviest* neighbour of u has the same typical weight $w_u^{1/(\tau-2)\pm o(1)}$ as in the Chung-Lu model. The discussed properties are so important that we collect them in a corollary.

Corollary 5.3. In a GIRG G with parameters $\alpha > 1$, $d \in \mathbb{N}$ and $\tau \in (2,3)$, let u be a vertex of weight w_u.

(a) The degree of u follows a Binomial distribution with $\mathbb{E}[\deg(u)] = \Theta(w_u)$.

(b) The degree distribution in the neighbourhood of u follows a power law with exponent $\tau - 1$.

(c) For every $\varepsilon > 0$, if w_u is sufficiently large then the heaviest neighbour v of u has weight $w_v \in [w^{1/(\tau-2)-\varepsilon}, w^{1/(\tau-2)+\varepsilon}]$, with the same error probabilities as in Theorem 3.8.

(d) There are $n^{\Omega(1)}$ vertices of weight at least $n^{1/2}$, which form a single clique. We call this set of vertices the *inner core*.

(e) With high probability, G has a giant component with typical distance $\frac{2\pm o(1)}{|\log(\tau-2)|} \log\log n$.

Proof. We have already argued that (a)-(c) are immediate consequences of Lemma 5.2. For the number of vertices of weight at least $w = n^{1/2}$, this has nothing to do with connection probability, so the number of such vertices is $\Theta(w^{1/(\tau-1)}) = n^{\Omega(1)}$ just as in every power law with exponent $\tau \in (2,3)$. It follows immediately from (5.1) that any two such vertices are connected since the maximal distance in $\mathcal{X}$ is at most $n^{1/d}$.

For (e), we just need to recall how we obtained a short path from u to v in a Chung-Lu random graph: starting in u, we greedily go to the heaviest vertex in the neighbourhood, until we reach the inner core. This has constant success probability, which already shows that a constant fraction of the vertices is connected to the inner core.[3] Then we do the same from v. By (c) the lengths of these greedy paths are the same in Chung-Lu graphs and in GIRGs. Finally we use that any two vertices in the inner core are connected by (d). We have not discussed how to obtain the lower bounds for typical distances in Chung-Lu graphs, but these arguments also make only use of the marginal probabilities (they compute the expected number of shorter paths), so they also transfer to GIRGs. □

5.1.1 Variations and extensions

In this section we will discuss the formula (5.1) for the connection probability in GIRGs. We will also mention some variations of the GIRG model, and show that the definition of GIRGs is rather robust against small changes.

[3] Strictly speaking, it shows that the number of such vertices is $\Omega(n)$ *in expectation*, and an (easy) extra argument is required to obtain the whp statement.

Constant factors

In the connection probability (5.1) we could allow any $\Theta(1)$-factors. Consequently, it does not matter which norm we take in (5.1), since all norms on $\mathbb{R}^d$ only differ by constant factors. However, this also includes models where p_{uv} never exceeds some constant $c < 1$ even if $w_u w_v / \|x_u - x_v\|^d > 1$. This allows for less rigid models. One of the main differences is that then the inner core does not form a single huge clique, but rather a dense Erdős-Rényi graph with constant connection probability. Such a graph does not have a large clique – the largest clique has size $O(\log n)$. But it is still extremely well-connected. For example, any two vertices have a common neighbour, i.e., the diameter of the inner core is 2. So the typical distances remain the same.

Let us make a short historic excursion. The first version of a GIRG were *Hyperbolic Random Graphs* [KPK+10]. Those had a very different description: place n points uniformly at random in a two-dimensional disc of radius $R = 2\log n + O(1)$ in hyperbolic space, and connect any two points of distance at most 1. This is a very simple description *if* one knows what the hyperbolic plane is. Analyzing these graphs is also not so simple because one needs to understand hyperbolic distances. However, it turns out that Hyperbolic Random Graphs are just a special case of the GIRG model, where one of the two hyperbolic dimensions corresponds to the weight, and the other hyperbolic dimension corresponds to a Euclidean circle. Thus (two-dimensional) Hyperbolic Random Graphs correspond to one-dimensional GIRGs. The only difference is that the former have a very complicated $\Theta(1)$-factor in the connection probability that comes from the hyperbolic geometry. In fact, GIRGs were developed as simplification and generalization of Hyperbolic Random Graphs.

The role of the exponent d

The GIRG model is less fragile than the Kleinberg model with respect to the exponent d in the term $\|x_u - x_v\|^d$ in (5.1). If instead we choose any different exponent $d' > d/\alpha$, then the resulting graph still has a power-law degree distribution. If additionally $d' < 2d/(\tau-1)$, then it is still an ultra-small world. Thus we do not rely on d' being one specific value. There is a whole range $[d/\alpha, 2d/(\tau-1)]$ of possible exponents, and one easily checks that this interval is non-empty. However, the case $d' = d$ has a convenient

parametrization: for other values of d' we do not have $\mathbb{E}[\deg(v)] = \Theta(w_v)$, but instead we have $\mathbb{E}[\deg(v)] = \Theta(w^{d/d'})$. Since weights and degrees fall apart, the power-law exponent τ' of the *degree* distribution no longer coincides with the power-law exponent τ of the *weight* distribution. Instead we have $\tau' = \frac{d'}{d}(\tau - 1) + 1$. So other values of d' would give us models which are less convenient to work with.

The role of α

The terms *weak ties* and *strong ties* also make sense for GIRG. However, other than for the Watts-Strogatz model, in GIRGs there is a continuous spectrum between strong and weak ties. For an edge uv, if $w_u w_v / \|x_u - x_v\|^d \geq 1$ then the edge is a strong tie, and if $w_u w_v / \|x_u - x_v\|^d$ is "much" smaller than one, then the edge is a weak tie. It is a matter of taste where to draw the line. In order to have a clear distinction, we define an edge uv to be a strong tie if and only if $w_u w_v / \|x_u - x_v\|^d \geq 1$. Informally, a more natural convention might be that uv is a strong tie if $w_u w_v / \|x_u - x_v\|^d = \Omega(1)$, and that it is a small tie if $w_u w_v / \|x_u - x_v\|^d = o(1)$. However, this would not give us a clear distinction between strong and weak ties for a fixed edge in a fixed graph for some concrete value of n, which is why we do not use this convention.

The exponent α ensures that "most" edges are strong ties.[4] For illustration, let us focus on vertices of constant weight. Recall that those form the majority of vertices. Fix a vertex v of weight $w_v = O(1)$. We want to study the number of neighbours of weight $O(1)$ of v, and we want to understand how this is affected by the exponent α. As in the analysis of the Kleinberg model, Theorem 4.5, we want to understand the number N_r of neighbours of constant weight in distance $[r, 2r]$ from v. There are $\Theta(r^d)$ vertices in this distance range. If we would omit the exponent α, then the connection probability would be $w_u w_v / r^d = \Theta(r^{-d})$, so $\mathbb{E}[N_r]$ would be $\Theta(1)$. Thus we would have exactly the same situation as in the Kleinberg model, and a vertex would have the same number of neighbours (of constant weight) in every distance range. In total, this would lead to a degree of $\Theta(\log n)$. (Or we could put a factor $1/\log n$ in front of the probability as in the Kleinberg

[4] This is to be taken with a grain of salt. With our strict definition of strong ties, still a $\Theta(1)$-fraction of all edges are weak ties, and it could even be more than half. With the informal alternative, it would be a $o(1)$-fraction.

model, to get constant degrees.) However, with the exponent α we obtain $\mathbb{E}[N_r] = \Theta(r^d \cdot (r^{-d})^\alpha)$. Thus, for $\alpha > 1$ the number of neighbours per distance range decreases with r, and most neighbours are close to v. In fact, there is nothing special about the function x^α that we applied here. It can be shown that we could take any non-negative increasing function $f(x)$ with $f(1) = 1$ and $\int_{x=1}^{\infty} f(1/x)dx < \infty$, and define

$$p_{uv} := \min\left\{1, f\left(\frac{w_u w_v}{\|x_u - x_v\|^d}\right)\right\}.$$

The resulting graph model would work just as well as the GIRG model. The function f determines how quickly the number of weak ties decays with increasing distance.

An important special case is known as the *threshold GIRG model* or as $\alpha = \infty$. In this case, we set $f(x) := 0$ for all $0 \le x < 1$, and $f(x) := 1$ for $x \ge 1$. So, we connect two vertices if and only if $\frac{w_u w_v}{\|x_u - x_v\|^d} \ge 1$. This is an important extreme case because it is a model which behaves as GIRGs in many aspects, but without any weak ties.

Poisson point processes and grids

An interesting property of the GIRG model is *self-similarity*. Pick any cube $\mathcal{X}'$ of radius R, and let V' be the set of vertices in $\mathcal{X}'$. Then the induced subgraph $G' := G[V']$ is itself again almost a GIRG with $|V'|$ vertices. The vertices $v \in V$ which draw a location in $\mathcal{X}'$ land in a uniform location within $\mathcal{X}'$, and the distribution of weights and connection probabilities remain the same. The only minor difference to a GIRG is that there we place exactly $n = \mathrm{Vol}(\mathcal{X})$ points in $\mathcal{X}$. For G', every vertex has chance $\mathrm{Vol}(\mathcal{X}')/n$ to land in $\mathcal{X}'$, so $\mathbb{E}[|V'|] = n' := \mathrm{Vol}(\mathcal{X}')$. So the *expected* number of vertices in G' is n', but in a GIRG we would place *exactly* n' vertices. However, $|V'|$ is binomially distributed $\mathrm{Bin}(n, n'/n)$ and thus highly concentrated when its expectation n' is large, so the difference to a GIRG is small.

There is a way to remove even this small difference: if in the definition of GIRG we place $\mathrm{Po}(n)$ vertices instead of *exactly* n vertices, then the model becomes perfectly self-similar, and the induced subgraph G' is again a GIRG. The method of placing $\mathrm{Po}(n)$ points in a space of volume n is also known as *Poisson point process* and has a nice mathematical consequence: for two disjoint regions $A, B \subseteq \mathcal{X}$, the number of vertices in A and B is *independent* of each other.

Yet another possibility is to use a grid instead of placing the points randomly. This does not yield a perfectly self-similar model (though it is still *approximatively* self-similar). But it gives a connected graph, since all grid edges are present. This is sometimes mathematically convenient, though one should be aware that such prefectly regular structures are usually not present in real-world networks.

5.2 Neighbours and Communities

In this section we will study the community structure of GIRGs. As we will see, despite the high-degree vertices and the small-world property, GIRGs are strongly governed by geometry.

5.2.1 Most neighbours are close

In this section we will study how far away the neighbours of a vertex v are from v. To this end, let us define the *ball of influence* $I(v)$ of v as the ball around x_v of radius $r_I(v) := w_v^{1/d}$, with respect to $\|.\|_\infty$. The ball of influence has volume $\Theta(w_v)$. Every other vertex u in $I(v)$ has connection probability $p_{uv} \geq \min\{1, w_v w_u / r_I(v)^d\} = \min\{1, w_u\} = 1$. Hence, v connects to all vertices in its ball of influence.

Recall that in the GIRG model, we throw n vertices into a box of volume n. Therefore, for any region R of volume x, the expected number of vertices that land in R is exactly $n \cdot x/n = x$. For $R = I(v)$, this means that in expectation $\Theta(w_v)$ vertices land in $I(v)$. Hence, v has in expectation $\Theta(w_v)$ neighbours in its ball of influence. On the other hand, we know that the *total* number of neighbours of v is also $\Theta(w_v)$. Therefore, at least a constant fraction of the neighbours of v are in $I(v)$. Note that all of these are strong neighbours of v.

The next lemma quantifies how many weak and strong neighbours v has outside of the ball of influence. Recall our definition:

$$u \text{ is a strong neighbour of } v \iff \frac{w_u w_v}{\|x_u - x_v\|^d} \geq 1. \tag{5.3}$$

Lemma 5.4. In the GIRG model, consider a vertex v of weight $w_v \geq 1$ and position $x_v \in \mathcal{X}$, and let $r_I(v) \leq r \leq n^{1/d}$. Let $N^{\text{strong}}_{[r,2r]}$ and $N^{\text{weak}}_{[r,2r]}$ be respectively the number of strong and weak neighbours u of v with distance $\|x_u - x_v\| \in [r, 2r]$, and let analogously $N^{\text{strong}}_{\geq r}$ and $N^{\text{weak}}_{\geq r}$ for neighbours in distance $\|x_u - x_v\| \geq r$. Then

(a) $\mathbb{E}[N^{\text{strong}}_{[r,2r]}] = \Theta(\mathbb{E}[N^{\text{strong}}_{\geq r}]) = \Theta(r^d \cdot (r^d/w_v)^{1-\tau})$.

(b) Assume that $\alpha \neq \tau - 1$, and let $\mu := \min\{\alpha, \tau - 1\}$. Then $\mathbb{E}[N^{\text{weak}}_{[r,2r]}] = \Theta(\mathbb{E}[N^{\text{weak}}_{\geq r}]) = \Theta(r^d \cdot (r^d/w_v)^{-\mu})$.

In particular, if $\alpha > \tau - 1$ then

$$\mathbb{E}[N^{\text{strong}}_{[r,2r]}] = \Theta(\mathbb{E}[N^{\text{strong}}_{\geq r}]) = \Theta(\mathbb{E}[N^{\text{weak}}_{[r,2r]}]) = \Theta(\mathbb{E}[N^{\text{weak}}_{\geq r}]).$$

Proof. We first consider the interval $[r, 2r]$. The number of vertices with distance in $[r, 2r]$ from v is $\Theta(r^d)$. We will only give the calculation under the simplifying assumption that all those vertices have distance *exactly* r from v. Then a vertex u in distance r yields a strong tie with v if and only if it has weight $w_u \geq r^d/w_v$. Moreover, we have the probability density $\Pr[w_u = w] = \Theta(w^{-\tau})$, and hence

$$\mathbb{E}[N^{\text{strong}}_{[r,2r]}] = \Theta(r^d) \cdot \int_{r^d/w_v}^{\infty} w^{-\tau} dw = \Theta(r^d \cdot (r^d/w_v)^{1-\tau}) = \Theta(r^{d(2-\tau)} w_v^{\tau-1}). \tag{5.4}$$

On the other hand, a vertex u in distance r forms a weak tie with v if and only if *i)* it has weight $w_u < r^d/w_v$, and *ii)* it forms an edge with v. The probability density that it has weight w is $\Theta(w^{-\tau})$, and therefore

$$\begin{aligned}
\mathbb{E}[N^{\text{weak}}_{[r,2r]}] &= \Theta(r^d) \cdot \int_1^{r^d/w_v} w^{-\tau} \cdot \left(\min\left\{1, \frac{w w_v}{r^d}\right\}\right)^{\alpha} dw \qquad (5.5)\\
&= \Theta(r^d) \cdot \int_1^{r^d/w_v} w^{-\tau} \cdot \left(\frac{w w_v}{r^d}\right)^{\alpha} dw \\
&= \Theta(r^{d-d\alpha} w_v^{\alpha}) \int_1^{r^d/w_v} w^{\alpha-\tau} dw.
\end{aligned}$$

If $\alpha - \tau < -1$, then we have to evaluate the integral at the lower boundary. Moreover, $\mu = \alpha$ in this case, and hence

$$\mathbb{E}[N^{\text{weak}}_{[r,2r]}] = \Theta(r^{d-d\alpha} w_v^{\alpha}) = \Theta(r^d \cdot (r^d/w_v)^{-\mu}).$$

If instead $\alpha - \tau > -1$, then we have to evaluate the integral at the upper boundary. Since this is cumbersome, we cleverly observe that the integral (5.5) is in fact the same integral that we evaluated in (5.4), only with a different integration range. Since we evaluate the integral both times at the same value (the lower boundary in (5.4), the upper boundary in (5.5)), by Excursion 3.1 they must give the same value up to constant factors. Hence,

$$\mathbb{E}[N^{\text{weak}}_{[r,2r]}] = \Theta(\mathbb{E}[N^{\text{strong}}_{[r,2r]}]) = \Theta(r^d \cdot (r^d w_v)^{1-\tau}).$$

Since $\mu = \tau - 1$, this proves the claim for the interval $[r, 2r]$.

For the distances $\geq r$, since $\tau - 1 > 1$ and $\mu > 1$, we observe that $\mathbb{E}[N^{\text{strong}}_{[r,2r]}]$ and $\mathbb{E}[N^{\text{weak}}_{[r,2r]}]$ are decreasing in r. If we start at $r' := r$ and sum over all values of $\mathbb{E}[N^{\text{strong}}_{[r',2r']}]$ or $\mathbb{E}[N^{\text{weak}}_{[r',2r']}]$ for $r' := r, 2r, 4r, 8r, \ldots$, then the summands form a geometric series, which is dominated by the first term. This proves the statements about "$\geq r$". □

Let us take a moment to understand Lemma 5.4. Firstly, $\mathbb{E}[N^{\text{strong}}_{\geq r}]$ and $\mathbb{E}[N^{\text{weak}}_{\geq r}]$ are decreasing in r. Hence, the further away we go from v, the fewer neighbours we find. So, "most" edge are in or close to the ball of influence. The term "most" is slightly imprecise because there is a soft transition as we increase the distance from v. Moreover, within distance $\Theta(r)$, strong neighbours need to have weight at least $\Omega(r^d/w_v)$ by definition of strong ties. In the proof, we evaluated the integral in (5.4) at the lower boundary. This means that "most" of the strong neighbours in distance r have the minimal possible weight $\Theta(r^d/w_v)$.

For weak ties, there are two different regimes: for $\alpha > \tau - 1$ there are few weak ties, and most weak neighbours in distance r have so large weight that they almost qualify as strong ties, i.e., their weight is only a constant factor below the threshold r^d/w_v. In this case, the majority of strong ties and weak ties look rather similar to each other. For $\alpha < \tau - 1$ there are many weak ties. In particular, there are asymptotically more weak neighbours than strong neighbours in distance r as $r \to \infty$, and "most" of the weak neighbours in distance r have weight $\Theta(1)$. Rephrased, in the case $\alpha > \tau - 1$, a random neighbour in distance $r \gg r_I(v)$ typically has large weight $\Theta(r^d/w_v)$, while it typically has small weight $\Theta(1)$ in the case $\alpha < \tau - 1$. Note that we can only observe this distinction outside of the ball of influence, so only for radii $r > r_I(v)$. Since v connects to all vertices

inside the ball of influence, picking a random neighbour is the same as picking a random vertex inside the ball, which will likely have weight $\Theta(1)$, regardless of the values of α and τ.

As an interesting corollary of Lemma 5.4, we obtain that GIRGs have a large clustering coefficient.

Corollary 5.5. Let G be a GIRG. Then the clustering coefficient of G satisfies $CC[G] = \Omega(1)$ in expectation and with high probability.

Proof. We only sketch the argument. Let v be a random vertex. With probability $\Omega(1)$, the vertex has constant weight, say for concreteness $w_v \in [1, 2]$. Assume that $\deg(v) \geq 2$, and pick two random neighbours u_1, u_2 of v. We need to show that u_1 and u_2 have probability $\Omega(1)$ to be adjacent.

Since $w_v = O(1)$, the ball of influence of v has radius $r_I(v) = O(1)$. One consequence of Lemma 5.4 is that a constant fraction of the neighbourhood of v lies in its ball of influence.[5] Hence, with probability $\Omega(1)$, both u_1 and u_2 lie in the ball of influence, and thus have distance at most $r_I(v)$ from v. By the triangle inequality, then u_1 and u_2 have distance at most $2r_I(v)$ *from each other*. In this case, their connection probability is $p_{uv} \geq w_{u_1} w_{u_2}/(2r_I(v))^d = \Omega(1)$. So, with constant probability, the vertices u_1 and u_2 are "close by", i.e., they have locations such that $p_{uv} = \Omega(1)$. This means that overall $\Pr[u_1 \sim u_2] = \Omega(1)$. Hence, $\mathbb{E}[CC(v)] = \Omega(1)$ for vertices v of constant weight and degree at least 2, which implies $\mathbb{E}[CC(G)] = \Omega(1)$. The whp statement can be obtained by standard concentration inequalities. □

5.2.2 Boundaries and communities

Equipped with Lemma 5.4, we will now prove that there are strong communities in GIRGs. Pick any cube $\mathcal{X}'$ of radius R, and let V' be the set of vertices in $\mathcal{X}'$. Then $|V'|$ has expectation $n' := \mathrm{Vol}(\mathcal{X}') = \Theta(R^d)$. Since the induced subgraph G' is again (almost) a GIRG, G' also inherits the properties that we know about GIRGs. For example, G' has a giant component of size $\Theta(n')$. Also, the number of edges within G' is $\Theta(n')$. The

[5] The formal statement is a bit more technical, since Lemma 5.4 only makes a statement about the *expected* number of neighbours in distance $\geq r$.

next theorem shows that this is much larger than the number of edges in G that connect V' to the rest of the graph. Thus V' forms a community.

Theorem 5.6. Let G be a GIRG. Let $\mathcal{X}' \subseteq \mathcal{X}$ be a cube of radius $1 \leq R \leq n^{1/d}/4$, and let V' be the set of vertices in $\mathcal{X}'$. Let $E(V', V \setminus V)$ be the set of edges from V' to $V \setminus V'$. Let $\nu := \max\{1-1/d, 3-\beta, 2-\alpha\}$. Then

$$\mathbb{E}[E(V', V \setminus V)] = \tilde{\Theta}((R^d)^\nu),$$

where the notation $\tilde{\Theta}(.)$ hides polylogarithmic factor in R, i.e., $\mathbb{E}[|E(V', V \setminus V)|] = O((R^d)^\nu (\log R)^{c_1})$ and $\mathbb{E}[|E(V', V \setminus V)|] = \Omega((R^d)^\nu (\log R)^{c_2})$ for two constants $c_1, c_2 \in \mathbb{R}$.

Proof. The proof involves the most complex calculation in this course, and we will ignore some borderline cases. We will go over the vertices $v \in V'$ and compute how many neighbours in $V \setminus V'$ they have in expectation. To do this, we let $\partial\mathcal{X}'$ be the boundary of $\mathcal{X}'$, and we write $d(x, \partial\mathcal{X}')$ for the distance of x from $\partial\mathcal{X}'$. Then we will use a double integral of the following form.

$$\int \Pr[\exists v : d(x_v, \partial\mathcal{X}') = r] \int \Pr[w_v = w] \cdot \mathbb{E}[\#\{\text{nbs of } v \text{ in } V \setminus V'\} \mid r, w] dw dr.$$

In the outer integral, we integrate over the possible distances r that v may have from $\partial\mathcal{X}'$. In principle, this distance may be anything between 0 and R. However, we will ignore distances $r \in [0, 1]$. Vertices in distance $r \in [0, 1]$ have a constant fraction of their neighbours in $V \setminus V'$. But the same is true for vertices in distance $r \in [1, 2]$, and there are about as many vertices with distance $r \in [1, 2]$ as vertices with $r \in [0, 1]$, up to a constant factor. Thus, we will lose at most a constant factor by omitting $r \in [0, 1]$. On the other side, we will also ignore distances $r \geq R/2$ from $\partial\mathcal{X}'$, i.e., we ignore the central subcube of radius $R/2$ of $\mathcal{X}'$. This part is negligible: it contributes only a constant factor to the total volume, and it is not hard to see that vertices in the center have less expected neighbours in $V \setminus V'$ than vertices which are closer to the boundary. So, we will only consider vertices in distance $r \in [1, R/2]$ from $\partial\mathcal{X}'$.

For $1 \leq r \leq R/2$, the probability density of having a vertex at distance exactly r from $\partial\mathcal{X}'$ corresponds to the surface area of a ball of radius $R-r$,

which is $\Theta((R-r)^{d-1}) = \Theta(R^d)$. Conveniently, this is independent of r.

For the inner of the two integrals, we may now assume that we have a vertex v with $d(x_v, \partial\mathcal{X}') = r$. Then we integrate over the possible weights $w_v = w$ that v may have, and count how many neighbours v has for these values of r and w.

We will compute the integral in two steps. In the first step, we will only consider vertices v for which the ball of influence $I(v)$ has non-empty intersection with $\mathcal{X} \setminus \mathcal{X}'$. This is the case if and only if $r_I(v) \geq r$, or equivalently $w \geq r^d$. Hence, we integrate w over the range $[r^d, R/2]$. By a similar argument as before, we may leave out weights in the range $[r^d, 2r^d]$, because the range $[2r^d, 4r^d]$ contributes the same amount. For weights $w \geq 2r^d$ we have $r_I(v) \geq cr$ for a constant factor $c = 2^{1/d} > 1$. Hence, a constant portion of $I(v)$ lies in $\mathcal{X} \setminus \mathcal{X}'$, so the intersection $I(v) \cap (\mathcal{X} \setminus \mathcal{X}')$ has volume $\Theta(\mathrm{Vol}(I(v))) = \Theta(w)$. Since every vertex in the intersection is a neighbour of v, we have $\mathbb{E}[\#\{\text{nbs of } v \text{ in } V \setminus V'\} \mid r, w] \geq \mathrm{Vol}(I(v) \cap (\mathcal{X} \setminus \mathcal{X}')) = \Theta(w)$ in this case.

So, we can finally compute the contribution of vertices v for which $I(v)$ intersects $\mathcal{X} \setminus \mathcal{X}'$ as

$$\begin{aligned} I_1 &:= \Theta(1) \int_1^{R/2} R^{d-1} \int_{2r^d}^{\infty} w^{-\tau} \cdot w \, dw dr \\ &= \Theta(R^{d-1}) \int_1^{R/2} [w^{2-\tau}]_{2r^d} \, dr \\ &= \Theta(R^{d-1}) \int_1^{R/2} r^{d(2-\tau)} \, dr. \end{aligned} \tag{5.6}$$

Now we need to distinguish two cases. Let us first assume that $d(2-\tau) \neq -1$, so that we need to evaluate the function $[r^{d(2-\tau)+1}]$. For $d(2-\tau) > -1$ we need to evaluate the upper boundary, and for $d(2-\tau) < -1$ the lower boundary. For $d(2-\tau) = -1$, the inner integral simply gives $\log(R/2)$, which we may swallow by a $\tilde{\Theta}(.)$ notation. Hence,

$$I_1 = \begin{cases} \Theta(R^{d-1}) \cdot R^{d(2-\tau)+1} = \Theta(R^{d(3-\tau)}) & \text{, if } d(2-\tau) > -1, \\ \tilde{\Theta}(R^{d-1}) & \text{, if } d(2-\tau) \leq -1. \end{cases} \tag{5.7}$$

The condition $d(2-\tau) > -1$ is equivalent to $d(3-\tau) > d-1$, so we can summarize (5.7) as follows.

$$I_1 = \tilde{\Theta}(R^{\max\{d-1, d(3-\tau)\}}) = \tilde{\Theta}((R^d)^{\max\{1-1/d, 3-\tau\}}). \tag{5.8}$$

The second step is to compute the contribution of vertices v for which $I(v)$ is disjoint from $\mathcal{X} \setminus \mathcal{X}'$, so assume from now on that v is such a vertex. This is equivalent to the condition $w_v < r^d$. We split this in yet two subcases: strong ties and weak ties. Note that a vertex can have strong ties outside of its ball of influence, if the neighbour has large weight. For this case, we will just show an upper bound, and find that this contribution is negligible. We will use the same integration method as before. Consider a vertex v of weight $w_v = w$ in distance r from the boundary. In order to form a strong tie with a vertex in $\mathcal{X} \setminus \mathcal{X}'$, the weight of the neighbour must be at least $w' := r^d/w$. We can easily compute the expected number of neighbours of weight at least w' of v: the *total* number of neighbours of w is $\Theta(w)$, and the degree distribution in its neighbourhood is a power-law with exponent $\tau - 1$. Therefore, the expected number of neighbours of weight at least w' is $\Theta(w \cdot (w')^{2-\tau}) = \Theta(w \cdot (r^d/w)^{2-\tau})$. Hence, $\mathbb{E}[\#\{\text{strong nbs of } v \text{ in } V \setminus V'\} \mid r, w] = O(w \cdot (r^d/w)^{2-\tau})$. The contribution of strong neighbours is thus upper bounded by

$$\begin{aligned} I_2^{\text{strong}} &:= \Theta(1) \int_1^{R/2} R^{d-1} \int_1^{r^d} w^{-\tau} \cdot w \cdot (r^d/w)^{2-\tau} \, dw dr \\ &= \Theta(R^{d-1}) \int_1^{R/2} r^{d(2-\tau)} \int_1^{r^d} w^{-1} \, dw dr \\ &= \tilde{\Theta}(R^{d-1}) \int_1^{R/2} r^{d(2-\tau)} \, dr. \end{aligned} \tag{5.9}$$

This is the same integral that we have already evaluated in (5.6). Therefore, $I_2^{\text{strong}} = O(I_1)$, and we may ignore this term.

Finally, we come to the last integral I_2^{weak}. This covers weak ties of vertices v for which $I(v)$ is disjoint from $\mathcal{X} \setminus \mathcal{X}'$. So let v be such a vertex. Then in particular $w_v < r^d$. If $\alpha > \tau - 1$, then by Lemma 5.4 the number of weak and strong neighbours of v in any distance range is asymptotically the same. Hence, in this case we have $I_2^{\text{weak}} = \Theta(I_2^{\text{strong}})$, and we can also ignore this term. Thus we may restrict to the case $\alpha < \tau - 1$.[6] By Lemma 5.4, for a vertex v in distance r from the boundary, the number of all weak neighbours in distance at least r is $\mathbb{E}[N_{\geq r}^{\text{weak}}] = \Theta(r^d \cdot (r^d/w_v)^{-\alpha})$. Again, it is not hard to argue that a constant portion of those weak neighbours are in $\mathcal{X} \setminus \mathcal{X}'$: the asymptotics does not change if we consider weak neighbours in

[6] We omit the case $\alpha = \tau - 1$. This case gives another $\log R$-factor, but is otherwise identical to the other cases.

distance at least $2r$, and those have a constant probability to be in $\mathcal{X} \setminus \mathcal{X}'$. Hence, we have $\mathbb{E}[\#\{\text{weak nbs of } v \text{ in } V \setminus V'\} \mid r, w_v] = \Theta(r^d \cdot (r^d/w_v)^{-\alpha})$. Thus the last integral is

$$\begin{aligned} I_2^{\text{weak}} &:= \Theta(1) \int_1^{R/2} R^{d-1} \int_1^{r^d} w^{-\tau} \cdot r^d \cdot (r^d/w)^{-\alpha} \, dw dr \\ &= \Theta(R^{d-1}) \int_1^{R/2} r^{d(1-\alpha)} \int_1^{r^d} w^{\alpha-\tau} \, dw dr \end{aligned} \tag{5.10}$$

Since we are in the case $\alpha < \tau - 1$, the inner integral is $\Theta(1)$, and the outer integral has exponent $d(1-\alpha)$. Similar as in (5.7), we need to make a case distinction, depending on whether $d(1-\alpha) > -1$ or not. Note that we can also write this condition as $d(2-\alpha) > d-1$. Thus we get

$$\begin{aligned} I_2^{\text{weak}} &= \begin{cases} \Theta(R^{d-1}) \cdot R^{d(1-\alpha)+1} = \Theta(R^{d(2-\alpha)}) & \text{, if } d(2-\alpha) > d-1, \\ \tilde{\Theta}(R^{d-1}) & \text{, if } d(2-\alpha) \le d-1 \end{cases} \\ &= \tilde{\Theta}(R^{\max\{d-1, d(2-\alpha)\}}) = \tilde{\Theta}((R^d)^{\max\{1-1/d, 2-\alpha\}}). \end{aligned} \tag{5.11}$$

Now we just need to collect the results. For $\alpha \ge \tau - 1$, we may ignore the terms I_2^{strong} and I_2^{weak}, since they are asymptotically dominated by I_1. Thus we obtain

$$\mathbb{E}[|E(V', V \setminus V')|] = \tilde{\Theta}(I_1) = \tilde{\Theta}((R^d)^{\max\{1-1/d, 3-\tau\}}).$$

Since $3 - \tau \ge 2 - \alpha$, this term agrees with $\tilde{\Theta}((R^d)^\nu)$. If $\alpha < \tau - 1$, then we may still ignore I_2^{strong}, but we need to sum $I_1 + I_2^{\text{weak}}$, and obtain

$$\begin{aligned} \mathbb{E}[|E(V', V \setminus V')|] &= \tilde{\Theta}(I_1 + I_2^{\text{weak}}) = \tilde{\Theta}((R^d)^{\max\{1-1/d, 3-\tau\}} + (R^d)^{\max\{1-1/d, 2-\alpha\}}) \\ &= \tilde{\Theta}((R^d)^{\max\{1-1/d, 3-\tau, 2-\alpha\}}), \end{aligned}$$

as required. □

Theorem 5.6 says that the number of edges going out of community V' is $\tilde{\Theta}((n')^\nu)$. Note that $\nu < 1$, so the number of edges is indeed much smaller (for large n') than the number of edges inside of V', which is $\Theta(n')$. The community structure is very rich: we find communities of all sizes, and communities can be heavily overlapping. This reflects the complex community structures that we find in real social networks.

Separators and algorithmic implications

An important special case of Theorem 5.6 is for $R = \Omega(n^{1/d})$. In this case, we split V into two parts of linear size, with only $\tilde{\Theta}(n^{\nu}) = o(n)$ edges between them. Moreover, we also split the *giant component* into two parts of linear size. Rephrased, we find a sublinear *separator* of the giant component. Recall that these do not exist for Erdős-Rényi graphs: for every partitioning of the giant component of an Erdős-Rényi graph into two linear parts, there are $\Omega(n)$ edges between those parts.

We have formulated Theorem 5.6 for a cube and its complement, but of course, any hyperplane that cuts $\mathcal{X}$ into two halfs yields a separator of size $\tilde{\Theta}(n^{\nu})$. (Or rather, *two* hyperplanes if we use the torus topology.) We can also iterate this process, i.e., if we have cut a GIRG into two parts of equal size, we can cut those two parts again by another hyperplane, and so on. This gives us a *separator hierarchy*. Separator hierarchies also exist for some other graph classes, for example for planar graphs. They can often be used to make branch-and-bound algorithms more efficient. For example, for every $\varepsilon > 0$ it is possible to find a $(1+\varepsilon)$-approximation for the INDEPENDENT SET problem on GIRGs (and on planar graphs, too) in polynomial time, while even an $n^{1-\varepsilon}$-approximation is NP-hard on general graphs. A similar technique allows to find approximate vertex covers. Also, it is possible to compute a maximum matching on GIRGs in time $O(n^{(5-\tau)/2})$, which is faster than the best known algorithm of runtime $O(\sqrt{n}m)$ for general graphs [BFK16].

5.3 Greedy routing

We have earlier encountered the Kleinberg model for navigatibility. That model was constructed in a way that makes navigation possible. As we will see, greedy routing works automatically in GIRGs. Let us briefly return to Milgram's small world experiment. The test subjects were told to send the message "to a personal acquaintance who is more likely than you to know the target person". This has a very natural analogon in GIRGs, as the connection probabilities p_{uv}. Note that, if a vertex v knows weight w_t and position x_t of the target t, and it knows the weight w_u and position x_u of all its neighbours, then it can compute the connection probability p_{ut} for all its neighbours. Hence, it can decide which neighbour maximizes

the connection probability and forward the message to that neighbour. In fact, in order to determine which neighbour u maximizes the connection probability p_{ut}, it is not even necessary to know the weight w_t of the target. Recall that $p_{ut} = \min\{1, w_u w_t / \|x_u - x_t\|^d\}^\alpha$. We may ignore the minimum with one, since we are finished in one more step when we reach $p_{ut} = 1$. Thus we must find the u which maximizes $(w_u w_t / \|x_u - x_t\|^d)^\alpha$. But this is the same u that maximizes $w_u w_t / \|x_u - x_t\|^d$, and the same u that maximizes $w_u / \|x_u - x_t\|^d$. Let us define this last expression as *potential function*:

$$\phi(u) := \phi_t(u) = \frac{w_u}{\|x_u - x_t\|^d} \tag{5.12}$$

Then we obtain a simple algorithm, Algorithm 2.

Algorithm 2: Greedy Routing

Assumption: Every vertex knows the position and weight of itself and of all its neighbours. Vertex s has a message which contains the position of target t.

Initialization: $v := s$

repeat

  $u := \operatorname{argmax}\{\phi_t(u') \mid u' \in V, u' \sim v\}$.

  if $\phi_t(u) \leq \phi_t(v)$ **then**

    return failure.

  else

    $v \leftarrow u$

until $v = s$;

Obviously, Algorithm 2 is not very smart. It just gives up when it reaches a local maximum. At least it avoids infinite loops, since it guarantees that p_{vt} increases in each step. Despite this naive handling of local optima, the next theorem, taken from [BKL+22] shows that the algorithm is stunningly successful.

Theorem 5.7. Let $G = (V, E)$ be a GIRG, and let $s, t \in V$ be uniformly at random from V. Then with probability $\Omega(1)$ the greedy routing algorithm, starting in s, finds t in at most $\frac{2+o(1)}{|\log(\tau-2)|} \log\log n$ steps.

Proof Sketch. We will only give the main calculation. Let v be the current vertex, and let $R(v)$ be the maximal length of an edge of v, i.e., $R(v) := \max\{\|x_u - x_v\| \mid u \sim v\}$. So all neighbours of v lie in a ball of radius $R(v)$ around v. Moreover, for brevity we write $D(v) := \|x_v - x_t\|$ for the distance to the target. One can show that the algorithm goes through two phases, where in the first phase $R(v) < D(v)$, and in the second phase $R(v) > D(v)$.

For random start points s and targets t, we typically have $w_s, w_t = O(1)$. In this case, all neighbours of s are in distance $O(1)$, and thus $R(v) = O(1)$. On the other hand, typically $D(v) = \Theta(n^{1/d})$, so we start in the first phase with $R(v) = o(D(v))$.

So assume that the algorithm is currently in some vertex v in the first phase. For the sake of this intuition, we even assume $R(v) = o(D(v))$, not just $R(v) < D(v)$. Which neighbour does v select in the first phase? By the triangle inequality, all neighbours u satisfy $D(u) = \|x_u - x_t\| \in [D(v) - R(v), D(v) + R(v)]$. Since $R(v) = o(D(v))$, this means that for any two neighbours u, u' we have $D(u) = (1 \pm o(1))D(v) = (1 \pm o(1))D(u')$. Therefore, all neighbours u of v have essentially the same denominator in the potential $\phi(u) = w_u/D(u)^d$, up to $(1 \pm o(1))$-factors. On the other hand, the weight of the heaviest neighbour of v is roughly $w_v^{1/(\tau-2)}$ by Corollary 5.3, so the factor w_u covers the whole range $[1, w_v^{1/(\tau-2)}]$. Thus in order to maximize the potential $\phi(u) = w_u/D(u)^d$, we must essentially maximize w_u, i.e, select the neighbour of v of largest weight. Therefore, in the first phase the algorithm iteratively goes to the (approximatively) heaviest neighbour. We already know that this procedure leads to doubly exponentially growing weights, to weight $w_s^{1/(\tau-2)^i}$ after i steps. It reaches the heavy core in $\frac{1\pm o(1)}{|\log(\tau-2)|} \log\log n$ steps. Moreover, for large w_v it is very unlikely that w_v does not have neighbours of larger weight. Thus the algorithm has a constant failure probability in the first few steps, when w_v is still small, but with growing w_v the failure probability quickly becomes negligibly small.

Recall that vertices in the heavy core form a single clique, regardless of their position. Therefore, a vertex v in the heavy core reaches vertices in all places of $\mathcal{X}$, and thus it is not hard to show that $R(v) > D(v)$. Hence, when the algorithm reaches the heavy core, it enters the second phase. (It would not harm the analysis if the algorithm enters the second phase earlier, or if we only have $R(v) \approx D(v)$ when the algorithm reaches the heavy core, but both scenarios are not likely.)

Which neighbour does a vertex v pick in the second phase of the al-

gorithm? This is a bit trickier. First of all, mind that $R(v) > D(v)$ does not mean that t is a neighbour of v. While v has *some* neighbours in distance range $D(v)$, it does not connect to *all* vertices in this distance range. In particular, it typically does not connect to t. However, v has some neighbours which are much closer to the target t, and one of them will be optimal. More precisely, we will show that the best neighbour u has weight $w_u \approx w_{opt} := \phi(v)^{-1}$ and distance $D(u) \approx D_{opt} := \phi(v)^{(1-\tau)/d}$ from t. Thus $\phi(u) \approx w_u/D(u)^d = \phi(v)^{-1}/\phi(v)^{1-\tau} = \phi(v)^{\tau-2} =: \phi_{opt}$. Hence, in the second phase we increase the potential by an exponent of $\tau-2$ in each step, see also Figure 5.1. (We have $\tau-2 < 1$, so taking the $(\tau-2)$-th power brings the potential closer to one. Since the potential is less than one, this corresponds to an increase.) We start the second phase (and in fact, also the first phase) with a potential $w_v/\|x_v - x_t\|^d \geq 1/n$, and if we raise the potential to power $\tau-2$ in each step, then an easy calculation shows that we need at most $\frac{1\pm o(1)}{|\log(\tau-2)|} \log\log n$ steps to reach potential $\Omega(1)$. Here, the $o(1)$ term swallows the approximations that we have swept under the rock. Once the potential is at $\Omega(1)$, we are finished, since then the algorithm has a probability of $\Omega(1)$ to hit t in the next step.

It remains to show that the best neighbour u indeed satisfies $w_u \approx w_{opt}$ and $D(u) \approx D_{opt}$. Let us first show that such a neighbour indeed exists. We will use without proof that $D_{opt} \leq D(v)$.[7] Every vertex u of weight w_{opt} and distance at most $D(v)$ from v is a neighbour of v because $w_{opt}w_v/D(v)^d = \phi(v)^{-1} \cdot w_v/D(v)^d = 1$. This is not quite the set of vertices we are looking for. Instead, we want to find a neighbour u of v which has distance D_{opt} *from the target*. But this does not change much: every such vertex has distance at most $D(v)+D_{opt} \leq 2D(v)$ from v, so all such vertices of weight w_{opt} connect to v with probability $\Omega(1)$.

On the other hand, there are vertices of weight $\approx w_{opt}$ and distance $\approx D_{opt}$ from the target: the expected number of such vertices is $\Theta(D_{opt}^d w_{opt}^{1-\tau}) = \Theta(\phi(v)^{1-\tau} \cdot \phi(v)^{\tau-1}) = \Theta(1)$. In a real proof, we would choose D_{opt} slightly larger to make sure that there are many such vertices. Hence, we have shown that v does have neighbours with weight w_{opt} and distance D_{opt}

[7] This step is actually not completely trivial. It involves showing that not all combination of $D(v)$ and w_v are possible, since whp there are no vertices with very high weight which are very close to t. Alternatively, one can show inductively that throughout the second phase (except for the very first step) the relation $D(v)^d \approx w_v^{\tau-1}$ holds, which follows the formulas for D_{opt} and w_{opt}.

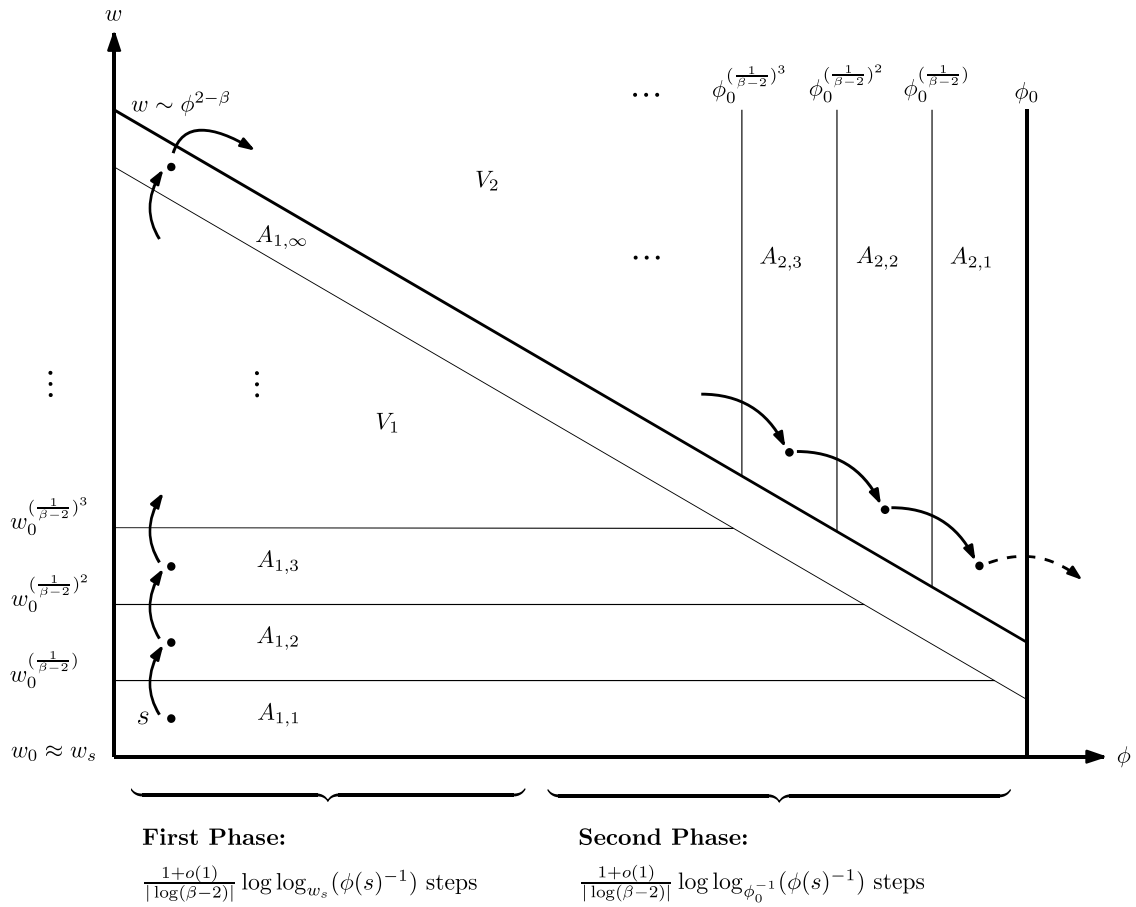

Figure 5.1: A typical trajectory of greedy routing. (Figure taken from [BKL+22], $\beta = \tau$.) In the first phase the weight is increased by an exponent of $1/(\tau-2)$ in each step. In the second phase, the potential is increased by an exponent $\tau-2$ in each step.

from t.

We still need to show that there are no neighbours with better potential. To this end, consider vertices of weight w in distance r from t. Such vertices exist if $w \leq (r^d)^{1/(\tau-1)}$, since this is the maximum weight among r^d vertices.

We write equivalently

$$r^{-d} w^{1-\tau} \leq 1. \tag{5.13}$$

Let us now compute the expected number of neighbours of v which have weight $\approx w$ and distance $\approx r$ from t. For simplicity, we will restrict ourselves to the case $r \leq D(v)/2$. Mind that those vertices still have distance $\approx D(v)$ *from* v, not distance r. Hence, the probability to form an edge with v is $w_v w / D(v)^d$, and the expected number of such neighbours is

$$\Theta\Big(r^d \cdot w^{1-\tau} \cdot \frac{w_v w}{D(v)^d}\Big) = \Theta\Big(r^d w^{2-\tau} \frac{w_v}{D(v)^d}\Big) = \Theta(r^d w^{2-\tau} \phi(v)). \tag{5.14}$$

In order for such neighbours to exist, we require that the above expectation is at least one. (In a full proof, we would allow some slack.) So, ignoring constant factors, we obtain the condition

$$r^{-d} w^{\tau-2} \leq \phi(v). \tag{5.15}$$

We want to maximize the potential w/r^d under the side constraints (5.13) and (5.15). This gives us a linear optimization problem, which can be solved by standard methods. Here we just give the solution. We take the $(3-\tau)$-th power of (5.13) and multiply it with the $(\tau-2)$-th power of (5.15), and obtain

$$w/r^d = (r^{-d} w^{1-\tau})^{3-\tau} \cdot (r^{-d} w^{\tau-2})^{\tau-2} \leq 1^{3-\tau} \cdot \phi(v)^{\tau-2} = \phi_{\text{opt}}. \tag{5.16}$$

Since the left hand side is the potential of the neighbour, we have shown that all neighbours of v have potential at most ϕ_{opt}.

This concludes the intuition that we give here. Of course, a full proof would be much more technical. For example, we would need to show that once the algorithm reaches the second phase, it stays in the second phase, and formally argue that in the second phase we really only need to investigate distances $r \leq D(v)$. □

Theorem 5.7 is remarkable in several aspects. Firstly, it is not clear a priori why such a stupid algorithm should have constant success probability. Recall that the Kleinberg model only succeeded because of the underlying grid; without the grid, routing in the Kleinberg model fails with high probability.[8] Note that we cannot expect the success probability for GIRGs to

[8]For GIRGs, a grid would not guarantee success of greedy routing, since a vertex v may have higher potential $\phi(v)$ than all its neighbours. This can not easily be fixed by removing the check $\phi(u) \leq \phi(v)$ from Algorithm 2. Then from a local maximum v, the algorithm might enter an infinite loop between v and its best neighbour.

be arbitrarily close to 1, since there is a constant probability that s or t are not in the giant component. Thus success probability $\Omega(1)$ is the best we can hope for.

Secondly, the number of steps is optimal. Typical distances in GIRGs are $\frac{2+o(1)}{|\log(\tau-2)|}\log\log n$ by Corollary 5.3, so shortest paths in G have the same length as the paths that greedy routing finds, up to a factor of $1+o(1)$. Indeed, it is even possible to make this connection for indidvidual pairs s, t: conditioned on the algorithm finding t, with high probability the greedy routing algorithm finds paths of stretch $1+o(1)$, i.e., which are only by a factor $1+o(1)$ longer than the shortest s-t-path in G. Moreover, even if we select s and t with given weights w_s and w_t and given distance $D(s)$, then the stretch is still $1+o(1)$, even if the typical shortest path for such conditions is shorter than $\frac{2\pm o(1)}{|\log(\tau-2)|}\log\log n$, as long as the path length is $\omega(1)$.

Unfortunately, it is usually not possible to use Algorithm 2 directly for real-world networks, because we do not know the underlying vertex positions x_v. (The weights can be approximatively deduced from the degrees). Recall that vertex positions do not just encode GPS coordinates (if these are available), but also intrinsic properties like age, profession and hobbies in social networks. However, there is active research on reconstructing the vertex positions from the abstract graph, which can then be used for greedy routing.[9]

Patching

We have already mentioned that Algorithm 2 has a pretty stupid policy (or rather, no policy) for handling local optima. It is possible to patch the algorithm with smarter policies which ensure that the target is always found if s and t are in the same component. This can be done in two ways: either the routing path is added to the message itself, similar as for emails. Or every vertex v remembers whether the message has already visited v, and remembers the last neighbour to which v has forwarded the message. With such patches, the algorithm can be modified such that it always finds t if it is in the same connected component as s. Importantly, these patches do

[9] Or rather, for finding vertex positions that are compatible with the abstract graph structure. The abstract graph does not contain enough information to reconstruct the original coordinates, but this is not necessary for such a task.

not destroy the performance guarantee: with high probability, the stretch of the greedy routing algorithm remains $1 + o(1)$ [BKL+22].

Geometric Routing

A natural variant of greedy routing is *geometric routing*. In this variant, we do not greedily optimize the connection probability p_{ut} (or equivalently, the potential $\phi(u)$), but rather we optimize the geometric distance from the target. So v sends the message to the neighbour u which minimizes the distance $\|x_u - x_t\|$ from the target. Otherwise, we follow the same scheme as in Algorithm 2.

Interestingly, geometric routing may also work in some situations. More precisely, it works if and only if $\alpha > \tau - 1$. Note that this is precisely the regime in which most neighbours in distance class $r > r_1(v)$ are strong neighbours or almost strong neighbours by Lemma 5.4. This is no coincidence, and we will now sketch the reason.

Geometric routing goes to the neighbour u which is closest to the target. Let us first consider the first phase. Then all neighbours u have similar distance from the target t, but we can still study which of them is closest to t. To make progress in direction of t, we need two properties:

(i) u and t should lie in the same direction from v, so the vector $\overrightarrow{x_v x_u}$ should have a similar direction as the vector $\overrightarrow{x_v x_t}$.

(ii) The distance $\|x_u - x_v\|$ should be as large as possible.

For condition (i), all neighbours of v have the same chance to lie in a good direction, regardless of their weight. But for condition (ii), it depends on α. If $\alpha > \tau - 1$, then for large r there are more strong vertices in distance $\approx r$ than weak neighours. By Lemma 5.4, the expected number of strong neighbours in distance $\approx r$ from v is $\Theta((r^d)^{2-\tau} w_v^{\tau-1})$. This expectation is $\Theta(1)$ for $r = r_{\max}$, where $r_{\max}^d := w_v^{(\tau-1)/(\tau-2)}$. Hence, the farthest neighbours have distance $\approx r_{\max}$ from v. Moreover, "most" of the neighbours in this distance are strong (or almost strong) neighbours, so they have weight $w \geq r_{\max}^d / w_v = w_v^{1/(\tau-2)}$. Since there are more strong than weak neighbours in distance $\approx r$, it is not hard to check that for $\alpha > \tau - 1$ there are no vertices of lower weight in distance $r_{\max}$. Therefore, geometric routing will pick among neighbours of weight $w \approx w_v^{1/(\tau-2)}$, and it will select among them whichever neighbour is aligned best with the direction $\overrightarrow{x_v x_t}$. But that

means that geometric routing *also* increases the weight in the first phase by an exponent of $1/(\tau-2)$ in each step, just like greedy routing. Hence, it will also reach the inner core in $\frac{1\pm o(1)}{|\log(\tau-2)|}\log\log n$ steps.

Note that this only works if $\alpha > \tau - 1$. If $\alpha < \tau - 1$, then a similar argument shows that the farthest neighbours of v are weak neighbours with weight $O(1)$. Hence, for $\alpha > \tau - 1$ the algorithm will stay within vertices of weight $O(1)$, and will only cover a distance of $r_{\max} = O(1)$ per step. Thus it would need $\Omega(n^{1/d})$ steps to overcome the distance $\Omega(n^{1/d})$ from s to the target t. It is even worse, because for vertices of weight $O(1)$ there is a constant failure probability per step, and with high probability the algorithm will get stuck quickly.

For the second phase, we can make a similar argument. Let us assume we have a vertex v which has distance $D(v)$ from the target. We may assume $D(v) > r_I(v)$, where $r_I(v)$ is the radius of the ball of influence of v. Since v connects to all vertices in the ball of influence, t would be adjacent to v if $D(v) \le r_I(v)$, and the algorithm would terminate. By defintion of the second phase, we may also assume $D(v) \le R(v)$, where $R(v)$ is the length of the longest edge incident to v. Hence, v has neighbours in distance $\approx D(v)$, but does not connect to *all* vertices in this distance. In particular, let us consider a ball B with radius $r < D(v)/2$ around t. Then the whole ball B is in distance $\approx D(v)$ from v. It depends on the volume $\Theta(r^d)$ of B whether v has a neighbour in the ball B. Let us choose r such that the expected number of neighbours of v in B is one. Which weight does a neighbour u of v in B have? Since $\alpha > \tau-1$, there are more strong (or almost strong) than weak neighbours in distance $\approx D(r)$. Therefore, with high probability the neighbour u is a strong neighbour and thus has large weight. Calculating r and w_u yields exactly the same values D_{opt} and w_{opt} as in the proof of Theorem 5.7.[10] Hence, geometric routing takes the same trajectory as greedy routing for $\alpha > \tau - 1$, also in the second phase.

5.4 Bidirectional search

Assume that we are given two vertices s and t, and we want to find a shortest path from s to t. One of the first algorithms that you have learned

[10] We did not show it explicitly in the proof of Theorem 5.7, but D_{opt} can be described as the smallest radius for which the ball B contains a strong neighbour of v.

is breadth-first search (BFS), which needs time $O(n)$. However, it turns out that this is not the fastest algorithm in practice. Instead, it is more efficient to use *bidirectional search*. In bidirectional search, we simultaneously start a BFS from both s and t. We call the two processes s-BFS and t-BFS. We alternate between the two processes, so whenever we have explored a layer of one BFS, we switch to the other. As soon as a vertex appears in both search trees, we have found a shortest path between s and t.

There are some variants. In practice, one should not just alternate, but one should always continue with the process which has so far the smaller BFS tree. This avoids the possibility that one BFS tree becomes much larger than the other. As an alternative, one could also swap the process after every explored vertex.

Speaking with practitioners, bidirectional search is generally faster than the classical unidirectional search of BFS. It depends on the network how much faster it is. As a rule of thumb, in networks with homogeneous degrees the speed-up is modest, typically a factor of 2, but it can be much larger in heterogeneous networks.

Traditional worst-case analysis does not show a difference between unidirectional and bidirectional search. Without further assumptions on the graph, both have the same worst-case performances. It is even possible to construct examples where bidirectional search performs worse than unidirectional search. So the differences in practice seem to come for the structure of the underlying networks. For unidirectional search, the performance does not depend on the underlying network. A BFS starting in s explores the set of vertices in some linear order. Since t is uniformly at random, it will appear in a random position in this ordering. So in expectation, we need to explore $n/2$ vertices to find t. (For a connected network of n vertices.) This is independent of the graph structure. So in order to understand the difference, we should study bidirectional search on different network models. In the following we will give heuristic arguments, but not full proofs. Analyses for Chung-Lu random graphs and hyperbolic random graphs (i.e., one-dimensional GIRGs with $\alpha = \infty$) can be found in [BN19] and [BFF$^+$22] respectively.

Since the speed-up depends on the degree distribution, our first attempt might be to analyze and compare Erdős-Rényi with Chung-Lu betworks. We will restrict ourselves to the giant component of both models, and always assume that the vertices s and t are drawn uniformly at random

form the giant component.

Erdős-Rényi graphs

In Erdős-Rényi graphs, we need to explore $\Theta(\sqrt{n})$ vertices from both sides. Then every vertex in the s-BFS has probability $\Theta(1/\sqrt{n})$ to appear in the t-BFS, so the expected number of vertices which appear both in the s-BFS and the t-BFS is $\Theta(1)$. This is know as the birthday paradox. By standard probabilistic arguments, we thus only need to explore $\Theta(\sqrt{n})$ vertices in expectation. This is much fewer than the $\Theta(n)$ vertices that we need to explore by unidirectional search.

Chung-Lu graphs

For Chung-Lu graphs with power-law exponents $\tau \in (2,3)$, we know that shortest paths run via the inner core. I.e., the two BFS will run until both have reached the inner core, and then they will find an overlap. The question is. How many vertices does a BFS explore before it finds the inner core? The answer will turn out to be surprisingly simple, but in order to understand the answer, we need to take one step back and return to the question of how many friends our friends have.

Consider a vertex v of weight w_v. The weights in the neighbourhood of v follow a power-law distribution with exponent $\tau - 1$. Let $v_{\max}$ be the neighbour of largest weight of v. Then the weight of $v_{\max}$ is roughly $w_{\max} := w_v^{1/(\tau-2)}$, and $v_{\max}$ has $\Theta(w_{\max})$ neighbours. But how many neighbours have all neighbours of v combined? Since each neighbour v' contributes $\Theta(w_{v'})$ neighbours, and $\Pr[w_{v'} = w] = \Theta(w^{1-\tau})$, we can compute this with an integral:

$$\begin{aligned}\#\{\text{nbs of nbs of } v \approx w_v \cdot \int_1^{w_{\max}} \Theta(w^{1-\tau} \cdot w)\}dw &= \Theta(w_v \cdot w_{\max}^{3-\tau}) \\ &= \Theta(w_v^{1+(3-\tau)/(\tau-2)}) = \Theta(w_v^{1/(\tau-2)}) = \Theta(w_{\max}).\end{aligned}$$

So, the neighbour $v_{\max}$ has about as many neighbours as *all other neighbours of v combined*. Actually, we shouldn't be surprised because we know from Section 3.2.2 that in the limit, our friends have infinitely many friends in expectation. This can only happen if the expectation is dominated by the friend(s) of largest weight, which means that they must contribute as much (or more) to the expectation than all other friends combined.

The same insight can also be transferred to sets of vertices. Assume that S is some layer of the BFS tree S of vertices, and assume that this set has total weight $w_S := \sum_{v \in S} w_v$. Consider the set S' that we explore in the next step. We have $|S'| = \Theta(w_S)$, and the weights of vertices in S' follow a power-law with exponent $\tau - 1$. Hence, the vertex $v_{\max}$ of largest weight in S' will have $w_S^{1/(\tau-2)}$ neighbours, about as many neighbours as all other vertices in S'. In particular, the size of the next layer in the BFS tree (and in fact, of the whole BFS tree) is in $w_S^{1/(\tau-2)}$.

Unfortunately, the above calculation comes with a restriction. The weights of the neighbours of a vertex v follow a power-law of exponent $\tau - 1$ *up to the cut-off point* n/w_v. Beyond this cut-off point, the above computation no longer holds. In particular, if the weight of v is too large then the heaviest neighbour does not have weight $w_v^{1/(\tau-2)}$. Fortunately, this restriction only applies for very large weights w_v. More precisely, it is only relevant for the last layer, when the BFS finds the inner core.

Assume that S is the last layer before the BFS finds the inner core. Let $v_{\max}$ be the heaviest vertex in S, and let $w_{\max}$ be its weight. Let S' be the neighbourhood of S, and let $v'_{\max}$ be the heaviest vertex in S', with weight $w'_{\max}$. Since we proceed layer by layer, there can be non-trivial fluctuations $w'_{\max}$. If we are lucky, then $w'_{\max} = n^{1/2+\varepsilon}$. In this case, it can be checked that $w'_{\max}$ is below the cut-off point for $w_{\max}$, so the above calculations still hold and the BFS only needs to explore $O(w'_{\max}) = O(n^{1/2+\varepsilon})$ vertices. If we are unlucky, then $w'_{\max} \approx n^{1/(\tau-1)}$, which is the maximal weight in the whole graph. In this case, we need to explore the neighbourhood of the whole inner core. The inner core contains $\Theta(n \cdot (n^{1/2})^{1-\tau}$ vertices, all of which have degree $\Omega(n^{1/2})$, so its neighbourhood has size $\Omega(n^{1+(1-\tau)/2+1/2}) = \Omega(n^{(4-\tau)/2})$. This bound is indeed tight, both for the size of the neighbourhood of S' and for the runtime of the BFS. Hence, the runtimes of bidirectional search varies in the interval $[n^{1/2}, n^{(4-\tau)/2}]$. Note that, for different values $\tau \in (2,3)$, the upper bound may take any value between $n^{1/2}$ and n. Thus bidirectional search is asymptotically faster than unidirectional search, but the speed-up is smaller than for Erdős-Rényi graphs.

Geometric networks

The previous analysis have shown different things: bidirectional search is indeed asymptotically faster than unidirectional search. This matches practical findings. However, the speed-up for Erdős-Rényi is *larger* than for Chung-Lu graphs. This is strange, since the practical experience is opposite: the speed-up of bidirectional search is stronger for inhomogeneous degrees, and modest for homogeneous graphs.

To understand this, let us go to geometric models. Applications are often routing and navgation problems, in which geometry plays an important role. As homogeneous network model, let us simply consider a grid. To find a connection between two vertices u and v, a unidirectional BFS will explore a ball around s of radius $\|x_s - x_t\|$, which has volume $\approx c \cdot \|x_s - x_t\|^d$, where c is the volume of the unit ball.[11] A bidirectional search needs to explore the two balls around s and t of radius $\|x_s - x_t\|/2$, at which point they touch. Thus the runtime is $2c \cdot (\|x_s - x_t\|/2)^d = 2^{1-d} c \|x_s - x_t\|^d$. Hence, in a grid graph the bidirectional search is faster by a factor of 2. This matches quite well the reported speed-up factor of 2, assuming that many applications have a two-dimensional underlying geometry. The same result also holds for more flexible models like Geometric Random Graphs, but is more difficult to prove there.

For GIRGs, the analysis of Chung-Lu graphs still applies. The main difference is that due to clustering and community structure, the BFS will quite often encounter vertices that it has found before. However, the size of the layers is determined by the weights that we obtain by picking iteratively the heaviest vertex, so it grows doubly exponential. Hence, the decelerating effects of clustering and communities are negligible, and the runtime varies again in the interval $[n^{1/2}, n^{(4-\tau)/2}]$. With these models, we find only a modest improvement for homogeneous graphs, and an asymptotic improvement for heterogeneous graphs. This indicates that we are on track for understanding bidirectional search in practice.[12]

[11] The unit ball in the right norm, for example one needs to take the $\|.\|_1$-norm for square grids. For our purpose it is just important that c is a universal constant that does not depend on s and t.

[12] Of course, this is so far only a hypothesis, and one should not stop at this point. But further research has indeed confirmed that these models are a good match. For example, the improvements predicted by these models match well the actual speed-ups on various real networks.

5.5 Non-Euclidean GIRGs

We have already discussed some minor variations of GIRGs in Section 5.1.1. In this section, we will see a more fundamental way of generalizing GIRGs. In GIRGs, we measured "closeness" of vertices by their Euclidean distance. However, we could equip the underlying space $\mathcal{X}$ with other distance functions as well. The idea of the following definition is that the distance between $x, y \in \mathbb{R}^d$ is measured by $\kappa(x-y)$ for some function $\kappa : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$.

Definition 5.8. Let $d \in \mathbb{N}$. Consider a measurable function $\kappa : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. For $x \in \mathbb{R}^d$ and $r, R \geq 0$, we call $B_r^\kappa(x) := \{y \in \mathbb{R}^d \mid \kappa(x-y) \leq r\}$ the κ-ball of radius r around x, and we call $B_{R/2}^\infty(x) := \{y \in \mathbb{R}^d \mid \|x-y\|_\infty \leq R/2\}$ the ∞-ball of radius $R/2$ and volume R^d around x. For $R, r > 0$, let $\mathrm{Vol}_R^\kappa(r) := \mathrm{Vol}(B_r^\kappa(0) \cap B_{R/2}^\infty(0))$.

We call κ a *feasible distance function* if it satisfies the following properties:

(i) *Symmetry:* $f(x) = f(-x)$ for all $x \in \mathbb{R}^d$.

(ii) *Boundedness:* There is $C > 0$ such that $\kappa(x) \leq C \cdot \|x\|_\infty$ for all $x \in \mathbb{R}^d$.

(iii) *Continuity of volume:* For all $R > 0$, the function $\mathrm{Vol}_R^\kappa(r) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$; $r \mapsto \mathrm{Vol}_R^\kappa(r)$ is surjective onto $[0, R^d]$.

When κ and R are clear from the context, we also write $\mathrm{Vol}(r)$ instead of $\mathrm{Vol}_R^\kappa(r)$.

For the last condition, note that the volume of $B_{R/2}^\infty(0)$ is R^d. Hence, we have $\mathrm{Vol}_R^\kappa(r) \leq R^d$ for all $r \geq 0$. Moreover, condition (ii) implies that for $r = CR/2$, we have that $B_R^\infty(0) \subseteq B_r^\kappa(0)$, and thus $\mathrm{Vol}_R^\kappa(CR/2) = R^d$. The function $\mathrm{Vol}_R(r)$ is increasing in r since the set $B_r^\kappa(0)$ is growing with r, and so the third condition requires that $\mathrm{Vol}_R^\kappa(0) = 0$ and that the volume increases *continuously* from 0 to R^d as r increases from 0 to $CR/2$.

For feasible distance functions, we can define a generalized GIRG model. As for ordinary GIRG, we draw the vertex locations from an axis-parallel cube $\mathcal{X}$ of volume n and radius $R = n^{1/d}$. To avoid boundary effects, we will not apply κ directly to $x_u - x_v$, but rather to $x_u - x_v \bmod R$, where

the "mod R" operator is applied componentwise for the d components of the vector $x_u - x_v$. Recall that for $y \in \mathbb{R}$ we define $y \bmod R := y'$ for the unique $y' \in [0, R)$ for which $(y - y')/R \in \mathbb{Z}$. In this way, the shape and volume of the "ball" $B_r^{\mathrm{mod}}(x) := \{y \in \mathcal{X} \mid \kappa(x - y \bmod R) \leq r\}$ around $x \in \mathcal{X}$ is independent of x.

Definition 5.9. Let $\alpha > 1$, $d \in \mathbb{N}$, let $\mathcal{D}$ be a power-law distribution on $[1, \infty)$ with exponent $\tau \in (2, 3)$ and let $\kappa : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be a feasible distance function. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the d-dimensional cube around zero of volume $n \in \mathbb{N}$, and let $R := n^{1/d}/2$ be its radius. A κ-*Geometric Inhomogeneous Random Graph (κ-GIRG)* $G = (V, E)$ on n vertices is obtained by the following three-step procedure.

(a) Every vertex v draws independently a weight w_v from distribution $\mathcal{D}$.

(b) Every vertex v draws indepdendently a uniformly random position $x_v \in \mathcal{X}$.

(c) Every two different vertices $u, v \in V$ are independently connected by an edge with probability

$$p_{uv} := \min\left\{1, \frac{w_u w_v}{\mathrm{Vol}_R^\kappa(r_{uv})}\right\}^\alpha, \tag{5.17}$$

where $r_{uv} := \kappa(x_u - x_v \bmod R)$.

The only difference to the ordinary GIRG definition is that we have replaced the factor $\|x_u - x_v\|^d$ in the definition of p_{uv} by the factor $\mathrm{Vol}_R^\kappa(r_{uv})$. The crucial insight is that the marginal probability stays the same as for GIRGs and Chung-Lu graphs. Note that we obtain the GIRG model as special case for $\kappa(x) := \|x\|_\infty$.[13]

Lemma 5.10. For any feasible distance function κ, consider a κ-GIRG $G = (V, E)$ and let $u, v \in V$. Assume we have drawn w_u, w_v and

[13] We are very slightly cheating here. We would need $\mathrm{Vol}_R^\kappa(r) = r^d$ for $\kappa = \|.\|_\infty$ to obtain the GIRG model. This is true for small r, but fails to be true if $r > R/2$ due to boundary effects. But the difference is negligible.

$x_u \in \mathcal{X}$, but that x_v is still random. Then the probability that u and v are connected is

$$\Pr[u \sim v \mid w_u, w_v, x_u] = \Theta\Big(\min\Big\{1, \frac{w_u w_v}{n}\Big\}\Big).$$

Proof. We will show that the marginal probability is the same as in the GIRG model. Let $0 \le y \le n$, and let us study $q_{\le y} := \Pr[\mathrm{Vol}_R^\kappa(r_{uv}) \le y \mid w_u, w_v, x_u]$. Since the function $\mathrm{Vol}(r) = \mathrm{Vol}_R^\kappa(r)$ is surjective by Definition 5.8, there exists r_y such that $\mathrm{Vol}(r_y) = y$. Let us momentarily assume that $\mathrm{Vol}(r)$ is even *bi*jective. Then we may compute

$$\begin{aligned}\Pr[\mathrm{Vol}_R^\kappa(r_{uv}) \le y] = \Pr[r_{uv} \le r_y] &= \Pr[x_v \in B_{r_y}^{\mathrm{mod}}(x_v)] \\ &= \frac{\mathrm{Vol}(B_{r_y}^{\mathrm{mod}}(x_v)}{\mathrm{Vol}(\mathcal{X})} = \frac{\mathrm{Vol}(B_{r_y}^{\mathrm{mod}}(0)}{\mathrm{Vol}(\mathcal{X})} = \frac{y}{n}.\end{aligned} \tag{5.18}$$

Note that this expression is independent of κ. Hence, for any fixed w_u, w_v, x_v, the denominator in (5.17) is a random variable whose cumulative distribution is independent of κ. Thus, the distribution of p_{uv} is independent of κ, and in particular $\Pr[u \sim v \mid w_u, w_v, x_u] = \mathbb{E}[p_{uv} \mid w_u, w_v, x_u]$ is independent of κ. Thus it must be identical to the GIRG model, which we obtain as the special case $\kappa(x) := \|x\|_\infty$.

We have used the assumption that $\mathrm{Vol}(r)$ is bijective. That was not necessary, but avoided technical difficulties. Otherwise, one would need to define $r_y := \sup\{r \mid \mathrm{Vol}(r) = y\}$, since otherwise the first step in 5.18 would fail. We omit the details. □

Lemma 5.10 has vast consequences. It implies that all arguments that are based on the marginal connection probabilities remain true. In particular, all result from Corollary 5.3 directly transfer to arbitrary κ-GIRGs, including $\mathbb{E}[\deg(v)] = \Theta(w_v)$, the degree distribution in the neighbourhood, and the existence of a giant component with typical distances. Moreover, Lemma 5.4, which counted the number of strong and weak neighbours in distance at least r (or in $[r, 2r]$) also remains true. In particular, most neighbours of v are in or almost in the ball of influence $I(v)$, and in distance $r > r_I(v)$, there are more weak neighbours for $\alpha < \tau - 1$, and more strong or almost strong neighbours for $\alpha > \tau - 1$. The result for bidirectional search also relies only on the marginal connection probabilities, and thus carries over as well.

5.5.1 The minimum component distance

One of the most intriguing examples of a distance function κ is the minimum component distance. (Not to be confused with the *maximum* component distance, which is just the good old ∞-norm we know from analysis.) For $d \geq 2$ and a vector $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, we define $\kappa(x) := \|x\|_{\min} := \min\{|x_i| \mid 1 \leq i \leq d\}$. Note that this is not a norm, and it does not even satisfy the triangle inequality. For the vectors $x := (0,0), y := (1,0)$, $z := (0,1)$, the vector x has distance zero from both y and z. But this does not imply that y and z also have distance zero from each other, or even "small" distance. There are also some other peculiarities. We have some non-zero vectors x with $\|x\|_{\min} = 0$. The scaling is also different than what we know from norms and metrics. The r-neighbourhood of 0 consists of the union of d "thickened" hyperplanes, each defined by $-\varepsilon \leq x_i \leq \varepsilon$ for a coordinate i. Since each of these thickened hyperplanes has volume $2\varepsilon \cdot R^{d-1}$, we generally have $\mathrm{Vol}_R^\kappa(r) = \Theta(R^{d-1} \cdot r)$. Despite all these quirks, it is easy to check that the minimum component distance is a feasible distance.

For a κ-GIRG, this means that two nodes are considered to be close if they agree in at least one coordinate. Intuitively, this makes sense for social networks: most of your acquaintances probably share *at least one* aspect with you: you may know colleagues taking the same lectures, comrades from your sport team or some other hobby, your family members, and your neighbours. But typically, your acquaintances do not share *all of your* aspects with you. Few (if any) of them will be a family member *and* study the same subject *and* play in the same sport team *and* live in the same house.

Clustering and the probabilistic triangle inequality

In GIRGs, the triangle inequality was the ultimate reason for the large clustering coefficient. Let us briefly recall the argument. A typical vertex u has weight $O(1)$. The typical random neigbhour of u has also weight $O(1)$ and is in distance $O(1)$ from u. Hence, if v_1 and v_2 are two random neighbours of u, they typically both have distance $O(1)$ from u, and then by the triangle inequality they also have distance $O(1)$ from each other. Thus they have connection probability $\Omega(1)$, which leads to a clustering coefficient of $\Omega(1)$.

Since the minimum component distance does not satisfy the triangle

inequality, it may seem it leads to low clustering coefficient. But that is not so. The minimum component distance still satisfies the following relaxed version of the triangle inequality.

Definition 5.11. Let $\kappa : \mathbb{R}^d \to \mathbb{R}_{\geq}$ be a feasible distance function. We say that κ satisfies a *stochastic triangle inequality* if there are $c, \varepsilon_0, R_0 > 0$ such that the following holds for all $0 < \varepsilon \leq \varepsilon_0$ and all $R \geq R_0$.

Let $\mathcal{X} := B^{\infty}_{R/2}(0)$, and let x, y be uniformly at random from $B^{\kappa}_{\varepsilon}(0) \cap \mathcal{X}$. Then $\Pr[\kappa(x - y) \leq 2\varepsilon] \geq c$.

For the usual triangle inequality, we would require $\kappa(x - y) \leq 2\varepsilon$ holds for *all* $x, y \in B^{\kappa}_{\varepsilon}(0)$. Here, we only require that this holds with probability $\Omega(1)$ when we draw x and y uniformly at random from $B^{\kappa}_{\varepsilon}(0)$. We need to restrict ourselves to the compact set $\mathcal{X}$, because in general we can not draw uniformly at random from unbounded sets. Let us show that the minimum component distance satisfies this condition. For $\kappa := \|.\|_{\min}$, if $x \in B^{\kappa}_{\varepsilon}(0)$ then we must have $|x_i| \leq \varepsilon$ for at least one coordinate $i \in [d]$. Any $y \in B^{\kappa}_{\varepsilon}(0)$, it must also must satisfy $|y_j| \leq \varepsilon$ for at least one coordinate $j \in [d]$. Since κ is symmetric with respect to permutation of the axes, for a random y we have $p_y := \Pr[|y_1| \leq \varepsilon] = \Pr[|y_2| \leq \varepsilon] = \ldots = \Pr[|y_n| \leq \varepsilon]$. By a union bound, we have $1 = \Pr[\exists j : |y_j| \leq \varepsilon] \leq d \cdot p_y$. Hence, $p_y \geq 1/d$. In particular, $\Pr[|y_i| \leq \varepsilon] = p_y \geq 1/d$. Hence, x and y are close to zero *in the same coordinate* i with probability at least $1/d$. If this happens, then $|x_i - y_i| \leq 2\varepsilon$ by the ordinary triangle inequality on $\mathbb{R}$, and hence $\kappa(x - y) \leq 2\varepsilon$. Thus $\kappa = \|.\|_{\min}$ satisfies condition (b) in Definition 5.11 with $c = 1/d$ (and arbitrary ε_0 and R_0).

The stochastic triangle inequality implies a large clustering coefficient, as the following theorem shows.

Theorem 5.12. Let κ be a feasible distance function that satisfies a stochastic triangle inequality. Then the clustering coefficient of κ-GIRG G satisfies $CC(G) = \Omega(1)$ in expectation and with high probability.

Proof. The proof is similar to the proof for GIRG in Corollary 5.5, so we only stress the main difference. It still suffices to show that $\mathbb{E}[CC(v)] =$

$\Omega(1)$ for vertices v with weight $w_v \in [1,2]$ and $\deg(v) \geq 2$. If we choose two random neighbours u_1, u_2 of v, then they have a constant probability to be both in the ball of influence $I(v)$ of v. Since every vertex in $I(v)$ connects to v, the position of u and v is uniformly at random in $I(v)$. By the weak stochastic triangle inequality, with probability $\Omega(1)$, u and v have distance at most $2r_I(v)$, which implies that they connect with probability $\Omega(1)$. This yields $\mathbb{E}[CC(v)] = \Omega(1)$, as required. □

So quite a lot of the properties of GIRGs carries over to κ-GIRG, especially if κ satisfies a stochastic triangle inequality. One thing that *does* change are the communities. It is still true that we can find relatively dense subgraphs. For example consider the thickened hyperplane $\mathcal{H}$ of width $2r$, defined by $|x_1| \leq r$. For small r, vertices in $\mathcal{H}$ are all relatively close to each other, so there are relatively many edge and cycle between them. If we take a connected component, then we find by a constant factor more edges than in a connected subgraph of Erdős-Rényi graphs, so we may still call it a community. However, every vertex in $\mathcal{H}$ has only about a $1/d$-fraction of its ball of influence in $\mathcal{H}$, and therefore the number of edges going out of the community is comparable or larger than the number of edges inside of the community. Recall that the situation for GIRGs was very different: in Theorem 5.6 we found communities which had asymptotically fewer outgoing edges than internal edges.

In fact, it can be shown that this does not just hold for the hyperplane $\mathcal{H}$, but that there are in general no small separators $\|.\|_{\min}$-GIRGs. For example, as for Erdős-Rényi graphs, whenever we partition the giant component of a $\|.\|_{\min}$-GIRG into two sets of size at least εn each, then there are at least δn edges between the partite sets, for some δ that depends on ε [LT17]. Again, this is different for GIRGs by Theorem 5.6.

Let us conclude this chapter with a final qualitative difference between GIRGs and $\|.\|_{\min}$-GIRGs. Consider a vertex v in a $\|.\|_{\min}$-GIRG, and let us pick two different neighbours u, u' of v. We have learned that with constant probability u and u' are close to v in the same coordinate, and thus also close to each other. Let us now focus on a different case, that u and u' are close to v (say, in its ball of influence $I(v)$), but in *different* coordinates. For simplicity, let us assume that those are the first and second coordinate, i.e., $|v_1 - u_1| \leq r_I(v)$ and $|v_2 - u'_2| \leq r_I(v)$. Then it is not hard to see that the other coordinates $u_2, \dots, u_d$ of u are essentially distributed uniformly

at random in the possible range $[-R/2, R/2]$ for $R := n^{1/d}$.[14] Likewise, the coordinates $u'_1, u'_3, u'_4, \dots, u'_d$ of u' are uniformly at random. Therefore, the vector $u - u'$ mod R is just a uniformly random vector in $[0, R]^d$. Now consider the shortest path between u and u' in $G[V \setminus \{v\}]$, i.e. after removing v from the graph. Since $u - u'$ mod R is random, we are simply left with two vertices u, u' with random distance vector from each other, and with high probability the shortest path between them has length $\frac{2 \pm o(1)}{|\tau - 2|} \log \log n$ as for every other random pair of vertices in the graph. Thus in the $\|.\|_{\min}$-GIRG model, if we pick two random neighbours u, u' of a vertex v, then they have probability $\Omega(1)$ of having a large graph-distance in $G[V \setminus \{v\}]$.

Note that this is very different from GIRGs. A typical vertex v may have a few weak outgoing edges, but even those tend to be short. It is very unlikely that v has a weak neighbour on the other side of $\mathcal{X}$. Hence, in GIRGs typically any two neighbours u and u' of v are geometrically rather close to each other. They may not be direct neighbours, but the shortest path between them is typically much shorter than $\frac{2 \pm o(1)}{|\tau - 2|} \log \log n$.

How does this compare to real social networks? We do not really know the answer, but it seems to make sense to some degree. Think of a fellow student v who has come to ETH from abroad. If you pick a family member u of v, do you expect to find an untypically short path from yourself to u in the friendship network without going through v? It seems plausible that the answer is No. (Of course, all paths are pretty short in social networks. We want paths that are shorter than the typical distance.) One interpretation is that any person belongs to several social circles or *communities*, and there is not necessarily any other connection between these communities. This corresponds nicely the $\|.\|_{\min}$-GIRGs, where each of the d dimensions of a vertex v define communities which have nothing in common except d. On the other hand, $\|.\|_{\min}$-GIRGs also don't seem to capture the whole truth. Your neighbours and your fellow students probably belong to different social circles who have little to do with each other (unless you live in a student hostel). But they all live in Zürich. So they are maybe as far from each other as random Zürich citizens, but probably less far than random people on earth. Relatedly, from a global perspective, there are certainly small separators. The number of acquaintanships within Europe is certainly much higher than the number of acquaintanships between Eu-

[14] It is not perfectly uniform because there is a non-empty overlap between the d hyperplanes whose union is $I(v)$. But the overlap is negligibly small if w_v is small.

ropeans and the rest of the world. So geometry seems to play a stronger role than in $||.||_{\min}$-GIRGs, but a weaker role than in GIRGs. There is still a lot to be learned about real-world networks, and a lot to improve about our random network models.

Bibliography

[AS16] Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2016.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[BB03] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288(5):60–69, 2003.

[BFF+22] Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Maximilian Katzmann, Felix Montenegro-Retana, and Marianne Thieffry. Efficient shortest paths in scale-free networks with underlying hyperbolic geometry. *ACM Transactions on Algorithms (TALG)*, 18(2):1–32, 2022.

[BFK16] Thomas Bläsius, Tobias Friedrich, and Anton Krohmer. Hyperbolic random graphs: Separators and treewidth. In *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[BKL19] Karl Bringmann, Ralph Keusch, and Johannes Lengler. Geometric inhomogeneous random graphs. *Theoretical Computer Science*, 760:35–54, 2019.

[BKL+22] Karl Bringmann, Ralph Keusch, Johannes Lengler, Yannic Maus, and Anisur R Molla. Greedy routing and the algorithmic small-world phenomenon. *Journal of Computer and System Sciences*, 125:59–105, 2022.

[BN19] Michele Borassi and Emanuele Natale. Kadabra is an adaptive algorithm for betweenness via random approximation.

Journal of Experimental Algorithmics (JEA), 24:1–35, 2019.

[Cen10] Damon Centola. The spread of behavior in an online social network experiment. *science*, 329(5996):1194–1197, 2010.

[CL02] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[Dur07] Richard Durrett. *Random graph dynamics*, volume 200. Citeseer, 2007.

[EDF+16] Sergey Edunov, Carlos Diuk, Ismail Onur Filiz, Smriti Bhagat, and Moira Burke. Three and a half degrees of separation. *Research at Facebook*, 694, 2016.

[Fel91] Scott L Feld. Why your friends have more friends than you do. *American journal of sociology*, 96(6):1464–1477, 1991.

[Gra73] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, 78(6):1360–1380, 1973.

[Kle00] Jon M Kleinberg. Navigation in a small world. *Nature*, 406(6798):845–845, 2000.

[KPK+10] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[LEA+01] Fredrik Liljeros, Christofer R Edling, Luis A Nunes Amaral, H Eugene Stanley, and Yvonne Åberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, 2001.

[LM01] Malwina J Luczak and Colin McDiarmid. Bisecting sparse random graphs. *Random Structures & Algorithms*, 18(1):31–38, 2001.

[LNR17] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex networks: principles, methods and applications*. Cambridge University Press, 2017.

[LT17] Johannes Lengler and Lazar Todorovic. Existence of small separators depends on geometry for geometric inhomogeneous random graphs. *arXiv preprint arXiv:1711.03814*, 2017.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge university press, 1995.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[New03] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[UKBM11] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[VCP+11] Lav R Varshney, Beth L Chen, Eric Paniagua, David H Hall, and Dmitri B Chklovskii. Structural properties of the caenorhabditis elegans neuronal network. *PLoS computational biology*, 7(2):e1001066, 2011.

[VDH09] Remco Van Der Hofstad. Random graphs and complex networks. *Available on http://www. win. tue. nl/rhofstad/NotesRGCN. pdf*, 11:60, 2009.

[vdHHVM05] Remco van der Hofstad, Gerard Hooghiemstra, and Piet Van Mieghem. Distances in random graphs with finite variance degrees. *Random Structures & Algorithms*, 27(1):76–123, 2005.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of â€˜small-worldâ€™networks. *nature*, 393(6684):440–442, 1998.