University of Potsdam
Hasso Plattner Institute
Algorithm Engineering Group

Structures & Algorithms in Hyperbolic Random Graphs

Anton Krohmer

Dissertation for obtaining the degree of

Dr. rer. nat.

submitted to the Faculty of Science,
University of Potsdam

May 16, 2017

Reviewers

Prof. Dr. Tobias Friedrich
University of Potsdam, Hasso Plattner Institute

Prof. Dr. Konstantinos Panagiotou
Ludwig Maximilian University of Munich

Dr. Thomas Sauerwald
Cambridge University, United Kingdom

Published online at the
Institutional Repository of the University of Potsdam:
URN urn:nbn:de:kobv:517-opus4-395974
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-395974

Abstract

Complex networks are ubiquitous in nature and society. They appear in vastly different domains, for instance as social networks, biological interactions or communication networks. Yet in spite of their different origins, these networks share many structural characteristics. For instance, their degree distribution typically follows a power law. This means that the fraction of vertices of degree k is proportional to $k^{-\beta}$ for some constant β; making these networks highly inhomogeneous. Furthermore, they also typically have high clustering, meaning that links between two nodes are more likely to appear if they have a neighbor in common.

To mathematically study the behavior of such networks, they are often modeled as random graphs. Many of the popular models like inhomogeneous random graphs or Preferential Attachment excel at producing a power law degree distribution. Clustering, on the other hand, is in these models either not present or artificially enforced.

Hyperbolic random graphs bridge this gap by assuming an underlying geometry to the graph: Each vertex is assigned coordinates in the hyperbolic plane, and two vertices are connected if they are nearby. Clustering then emerges as a natural consequence: Two nodes joined by an edge are close by and therefore have many neighbors in common. On the other hand, the exponential expansion of space in the hyperbolic plane naturally produces a power law degree sequence. Due to the hyperbolic geometry, however, rigorous mathematical treatment of this model can quickly become mathematically challenging.

In this thesis, we improve upon the understanding of hyperbolic random graphs by studying its structural and algorithmical properties. Our main contribution is threefold. First, we analyze the emergence of cliques in this model. We find that whenever the power law exponent β is $2 < \beta < 3$, there exists a clique of polynomial size in n. On the other hand, for $\beta \geqslant 3$, the size of the largest clique is logarithmic; which severely contrasts previous models with a constant size clique in this case. We also provide efficient algorithms for finding cliques if the hyperbolic node coordinates are known. Second, we analyze the diameter, i. e., the longest shortest path in the graph. We find that it is of order $O(\log^{\frac{2}{3-\beta}} n)$ if $2 < \beta < 3$ and $O(\log n)$ if $\beta > 3$. To complement these findings, we also show that the diameter is of order at least $\Omega(\log n)$. Third, we provide an algorithm for embedding a real-world graph into the hyperbolic plane using only its graph structure. To ensure good quality of the embedding, we perform extensive computational experiments on generated hyperbolic random graphs. Further, as a proof of concept, we embed the Amazon product recommendation network and observe that products from the same category are mapped close together.

Zusammenfassung

Komplexe Netzwerke sind in Natur und Gesellschaft allgegenwärtig. Sie tauchen in unterschiedlichsten Domänen auf, wie zum Beispiel als soziale Netzwerke, biologische Interaktionen oder Kommunikationsnetzwerke. Trotz ihrer verschiedenen Ursprünge haben diese Netzwerke jedoch viele strukturelle Gemeinsamkeiten. So sind die Grade der Knoten typischerweise Pareto-verteilt. Das heißt, der Anteil an Knoten mit k Nachbarn ist proportional zu $k^{-\beta}$, wobei β eine beliebige Konstante ist. Weiterhin haben solche Netzwerke einen hohen Clusterkoeffizienten, was bedeutet, dass zwei benachbarte Knoten viele gemeinsame Nachbarn haben.

Um das Verhalten solcher Netzwerke mathematisch zu studieren, werden sie häufig als Zufallsgraphen modelliert. Klassische Modelle wie inhomogene Zufallsgraphen oder das Preferential-Attachment-Modell erzeugen Graphen mit Pareto-verteilten Knotengraden. Cluster sind darin jedoch häufig nicht vorhanden, oder werden durch das Hinzufügen unnatürlicher Strukturen künstlich erzeugt.

Hyperbolische Zufallsgraphen lösen dieses Problem, indem sie dem Graphen eine Geometrie zugrunde legen. Jeder Knoten erhält hyperbolische Koordinaten, und zwei Knoten sind verbunden, wenn ihre hyperbolische Distanz klein ist. Cluster entstehen dann natürlich, da benachbarte Knoten samt ihrer Nachbarschaften in der Geometrie nah beieinander liegen, und die Pareto-Verteilung der Knotengrade folgt aus der exponentiellen Expansion des hyperbolischen Raumes. Durch die hyperbolische Geometrie wird jedoch auch die mathematische Analyse des Modells schnell kompliziert.

In dieser Arbeit studieren wir die strukturellen und algorithmischen Eigenschaften von hyperbolischen Zufallsgraphen. Wir beginnen mit der Analyse von Cliquen. Wir beobachten, dass wenn der Pareto-Exponent β zwischen 2 und 3 liegt, es Cliquen von polynomieller Größe in n gibt. Mit $\beta \geqslant 3$ ist die größte Clique noch logarithmisch groß, was früheren Modellen mit konstanter Cliquengröße stark widerspricht. Wir geben auch einen effizienten Algorithmus zur Cliquenfindung an, wenn die Koordinaten der Knoten bekannt sind. Als Zweites analysieren wir den Durchmesser, also den längsten kürzesten Pfad in hyperbolischen Zufallsgraphen. Wir beweisen, dass er $O(\log^{\frac{2}{3-\beta}} n)$ lang ist, wenn $2 < \beta < 3$, und $O(\log n)$ falls $\beta > 3$. Komplementär dazu zeigen wir, dass der Durchmesser mindestens $\Omega(\log n)$ beträgt. Als Drittes entwickeln wir einen Algorithmus, der reale Netzwerke in die hyperbolische Ebene einbettet. Um eine gute Qualität zu gewährleisten, evaluieren wir den Algorithmus auf über 6000 zufällig generierten hyperbolischen Graphen. Weiterhin betten wir exemplarisch den Produktempfehlungsgraphen von Amazon ein und beobachten, dass Produkte aus gleichen Kategorien in der Einbettung nah beieinander liegen.

Acknowledgments

First and foremost, I would like to thank my advisor Tobias Friedrich for giving me the chance of working on a topic of my own choosing. I understand that this is by no means usual for PhD students, which is all the more why I want to offer my thanks for his trust in me, and for tackling the—at that time—foreign topic of hyperbolic random graphs together. His advice was invaluable at all times during my studies, yet it always felt to me that I can discuss my research with him as equals. Finally, since this is not to be taken for granted, I want to thank him for offering me a good contract that made it possible for me to live financially secure and plan for the future.

Second, I would like to thank Timo Kötzing and Thomas Bläsius, who felt like second advisors to me, and who I bugged for a long time with all kinds of scientific and non-scientific questions. Their attitude and insights into research, science and teaching often helped me push further and gave me the strength to continue my studies.

I am also thankful to my colleagues and coauthors Thomas Bläsius, Karl Bringmann, Maximilian Katzmann, Sören Laue, Ralf Rothenberger, Thomas Sauerwald and Andrew M. Sutton for teaching me new techniques, and for the opportunity to work with you together.

I am further grateful to the whole Algorithm Engineering Chair at the Hasso Plattner Institute for our lunch together, our countless board game sessions, for making me feel at home, and for being great people in general. I would also like to extend a special thanks for all our scientific and non-scientific discussions, and for reading and correcting my whole thesis to Thomas Bläsius and Pengming Wang. Furthermore, I thank Annemarie Kunze and Katrin Heinrich for their invaluable help with all kinds of non-scientific problems.

To close, I would like to thank all my personal friends and family, without whose support I would not stand where I am today. In particular, I would like to thank Pengming Wang, Ole Rehmsen, Hendrik Molter, Dominik Laur, David Neunemann, Anna Schwind, and Rumjana Slodička for our great times; my whole hiking group for our trips together; and my family Swetlana, Eugen, Xenia and Nikita Krohmer for their unwavering support. Finally and most importantly, I want to thank my girlfriend Asuka Wakabayashi, who stood by me and believed in me for the entire four years, which seem to me both like a breeze and an eternity now. For her and all others' help I am ever grateful.

Contents

1 Introduction

Most relational data can be expressed as a network. Cities are connected by roads in road networks; users form friendships in social networks; authors collaborate in co-author networks; and autonomous systems communicate in the Internet network. Understanding the structure and behavior of such systems has therefore become a crucial task in industry and applied sciences.

Such networks are typically modeled as *graphs*. A graph G is a fundamental combinatorial object, consisting of a set of nodes V and a set of edges $E \subseteq \binom{V}{2}$, where each edge $\{u, v\} \in E$ joins two nodes $u, v \in V$. Due to their importance, algorithmic problems on these structures have been intensely studied in the last decades and their behavior is theoretically well-understood.

Nevertheless, considering general graphs for algorithm analysis poses several drawbacks. For instance, many fundamental algorithmic problems like Clique, Independent Set or Traveling Salesman are known to be NP-hard on general graphs. For such problems, it is therefore highly unlikely that an efficient worst-case algorithm will be developed any time in the future. The desire to solve them efficiently in practice, however, remains. This gave rise to a myriad of heuristic approaches that work well for many problems despite their hard worst-case complexity. To convincingly explain why this is the case, the theoretical computer science community devised several more fine-grained approaches that go beyond NP-hardness.

Instead of finding the optimum, *approximation algorithms* try to find solutions that are provably close to the optimum in quality. Many notoriously hard problems like Vertex Cover or Metric Traveling Salesman can be approximated within a factor of 2 in polynomial time using this approach. Another possibility is to refine the worst-case analysis of an algorithm. Typically, the runtime of an algorithm is measured in the size of the input n. The more fine-grained *parameterized complexity*, however, attempts to find underlying structures in the input that make an algorithmic problem hard. These structures are then quantified in an extra parameter k. Then, one may express the runtime of an algorithm in the size of the input and the parameter k. For some problems like k-VertexCover, this makes it possible to achieve runtimes polynomial in n but exponential (or worse) in k.

Another approach is to assume that real-world problem instances are drawn from some probability distribution. Instead of analyzing worst-case complexity, one may then look at average-case behavior of these structures. Random graphs, for example, were modeled for a long time as so-called Erdős-Rényi random graphs $G(n, p)$. To sample a graph of size n, every edge $\{u, v\} \subseteq \binom{V}{2}$ is independently added with probability p. This uniform model quickly became popular since it captures "almost all graphs": If a property P holds with probability x for the $G(n, p)$ model, then P holds for a fraction of x of all $2^{\binom{n}{2}}$ graphs of size n. Therefore, if one can show that P holds with a probability approaching 1 as $n \to \infty$ in the $G(n, p)$ model, this means that P holds for essentially all graphs for large n.

It is easy to see that in this model, every node has $p \cdot (n-1)$ neighbors in expectation. The *degree sequence*—i. e., the fraction of nodes $f(k)$ having k neighbors—is thus strongly concentrated around a single value $p \cdot (n-1)$ in the $G(n, p)$. Recent research suggests, however, that for many real-world networks this is not the case. Rather, these networks seem to have a degree distribution following a *power law* sequence, i. e., $f(k) \sim k^{-\beta}$ for some constant β depending on the network [New03]. For most of these networks, β lies between 2 and 3, but networks with larger power law exponents exist.

A power law degree distribution implies that these networks are inhomogeneous: Most nodes have a small (constant size) neighborhood; but there exist nodes in the graph with an extremely large (polynomial size) neighborhood. These findings suggest that the $G(n, p)$ does not capture all properties of real-world complex networks; and other models should be considered.

Scale-Free Networks. For many real-world networks stemming from vastly different fields, it has been observed that along with a power law degree distribution, they share more structural characteristics. They have hub nodes (nodes that interconnect the graph), community structures (subgraphs with high edge density), very low diameter (longest shortest path) and a giant component (a connected component containing a constant fraction of all vertices). Such networks are also often called *scale-free* even though there is no universally agreed-upon definition of this term [Li+05].

Over the course of the last decade, research has been striving to produce generative models for these types of networks that are able to accurately predict properties of real-world graphs. Popular models include the preferential attachment graphs [BA99; Bol+01] and variants of inhomogeneous random graphs [Hof16; Söd02; BJR07]. The latter generalizes the Erdős-Rényi random graphs $G_{n,p}$ by using non-uniform—but independent—edge probabilities. These models excel at producing a power-law degree distribution; and they have a giant component, hub nodes and low diameter. Due to their independent edge probabilities, they are also accessible to rigorous studies. Independent edge probabilities also imply, however, that the graphs have low clustering, meaning that there exist no community structures.

In contrast, most real-world graphs do have high clustering, see e. g. [LK14]. In the case of social networks this is easy to envision: Two people are more likely to be connected if they already have a friend in common. A number of fixes to the above models have been proposed to incorporate that intuition [Váz03; New01; LK07] (e.g.

first construct a random graph, and then replace all nodes with k-cliques). Often times, however, these fixes seem artificial and introduce structural artifacts that are unlikely to appear as such in nature.

Hyperbolic Random Graphs. Boguñá, Claffy, Krioukov and Serrano [BKC09; SKB08] took a different approach by assuming an underlying geometric space to the graph. Each node is equipped with a weight and a set of coordinates; and two nodes are connected if they are either close by, or have large enough weights. One may interpret this as comparing similarity and popularity of two individuals: Two individuals are connected if they are either similar or popular enough.

While this approach more satisfyingly explained the occurrence of clustering, the conjunction of geometry and weights still seemed artificial. This was resolved, however, by considering the *right* underlying geometry. Krioukov, Papadopoulos, Kitsak, Vahdat, and Boguñá [Kri+10] introduced *hyperbolic random graphs*, where nodes are sampled in the hyperbolic plane and connected if they are close by. High clustering and a power law degree distribution emerge in this model as a natural reflection of the underlying hyperbolic plane. Since space expands exponentially in the hyperbolic plane (as opposed to polynomially in euclidean spaces), sampling nodes uniformly in a hyperbolic circle places most of them close to the boundary of said circle. On the other hand, again due to the exponential space expansion, these nodes have very few neighbors. Figure 1.1 shows an example of a hyperbolic random graph.

The result is a natural model that explains many characteristics observed in real-world networks. Further, the geometric representation of these graphs in the hyperbolic plane can be used for visualization and can even be algorithmically exploited. For instance, Boguñá, Papadopoulos, and Krioukov [BPK10] use a hyperbolic embedding of the Internet graph for greedy routing. There, nodes forward packets to the neighbor that is geographically closest in the hyperbolic plane to the packet's destination. They show that using their embedding, this method is near optimal. Without backtracking, 97% of packets reach their destination; and the average path length is just 10% longer than optimal. Due to this exceptional performance, a proposed replacement for the IP protocol, the Named Data Network [Zha+14], adopted its routing technique to use greedy routing in the hyperbolic plane. Following this example, metabolic networks [SBS12] and trade networks [Gar+16] have been successfully embedded into the hyperbolic plane as well. Taken together, these results suggest that multiple real-world networks seem to have an underlying hyperbolic geometry.

1.1 Related Work

Even though the topic of sale-free networks is relatively new, a vast amount of research already exists upon which we can build. Since the focus of this work lies on hyperbolic

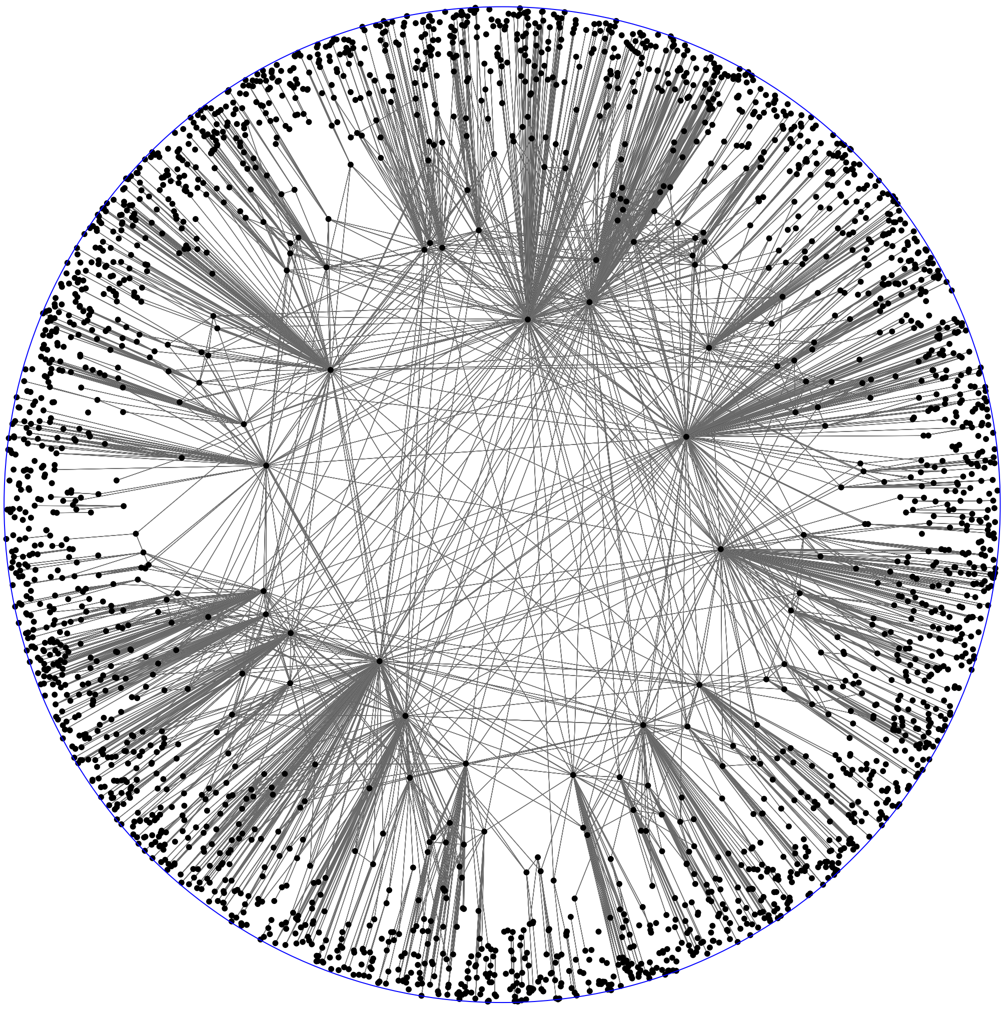


Figure 1.1: An example of a hyperbolic random graph, sampled with parameters $n = 2\,500$, $\alpha = 0.6$, $R = 19$, $T = 0.3$. See Chapter 3 for a thorough explanation of these parameters. Due to the exponential expansion of space, most nodes are sampled close to the boundary of the disk and have few neighbors. Nodes close to the center are rare, but due to their centrality they are connected to a significantly larger amount of vertices. This results in a power law degree distribution. The center of the graph (often called the *core*) is usually tightly connected. Due to the geometric closeness of joined nodes, their neighborhoods have a large overlap. This yields a high clustering coefficient as observed in real-world networks.

random graphs, we refer the interested reader to [Bar16] for a a general introduction to complex networks. Alternatively, the lecture notes of van der Hofstad [Hof16] are more mathematical in nature but provide an excellent overview over existing theoretical results.

Model. Hyperbolic random graphs were first suggested by Krioukov, Papadopoulos, Kitsak, Vahdat, and Boguñá [Kri+10]. Their work builds upon earlier research on the underlying geometry of complex networks [SKB08; BKC09; Pap+10]. Gugelmann, Panagiotou, and Peter [GPP12] examine the hyperbolic random graph model more closely and rigorously prove that it has a power law degree distribution, and a constant clustering coefficient. Fountoulakis [Fou15] bridges this model with the well-known Chung-Lu random graphs [ACL01; CL02a; CL02b] by showing that one can relate the radius of a vertex in hyperbolic random graphs to the weight of a vertex in Chung-Lu random graphs. Conditioned on knowing *only* the radius or weight of a vertex, respectively, the connection probabilities in both models are asymptotically equal. This, however, does not hold once (parts of) the neighborhoods of u and v are revealed, since hyperbolic random graphs have constant clustering whereas in Chung-Lu random graphs all edges are sampled independently. Bringmann, Keusch, and Lengler [BKL15] recently used this to generalize hyperbolic random graphs to geometric inhomogeneous random graphs (GIRGs) that also capture other kinds of underlying geometric spaces. A modified version of hyperbolic random graphs was also proposed by Papadopoulos, Kitsak, Serrano, Boguñá, and Krioukov [Pap+12]. Similar to preferential attachment, they consider an individual node arriving to an existing network. The node is equipped with hyperbolic coordinates, and it then connects preferably either to nodes with high degree (as in the preferential attachment), or to nodes that are close by (as in the hyperbolic random graph).

As of today, most fundamental structural properties of hyperbolic random graphs are well understood. Whenever the power law exponent β is between 2 and 3—which is typically the case in real-world graphs [New03]—these graphs have a giant component of size $\Omega(n)$ [BFM13], whereas all other components have at most polylogarithmic size [KM15]. If $\beta > 3$, the largest component is of polynomial size $O(n^c)$ with $c < 1$; and for $\beta = 2$ the graph is connected [BFM14]. For $\beta = 3$, there is a phase transition, i. e., the component structure depends on the average degree of the graph. The average distance between two nodes in hyperbolic random graphs is of order $\Theta(\log \log n)$ [BKL15; ABF15], and all shortest paths are at most of polylogarithmic length [KM15; BKL15]. For the case $2 < \beta < 3$, Kiwi and Mitsche [KM16] also show that the conductance and spectral gap of the giant component of hyperbolic random graphs is $\Theta(n^{2-\beta})$ up to a polylogarithmic factor. All of the aforementioned results hold with probability at least $1 - o(1)$.

Applications. Hyperbolic random graphs have small balanced separators and therefore a sublinear treewidth [BFK16]. This strongly differs from previous models. For instance, inhomogeneous random graphs have balanced separators of at least linear size, due to their independent edge probabilities [BJR07]. Classical hard problems like Independent Set in hyperbolic random graphs can thus be solved in subexponential time and admit a PTAS, if the geometry is known.

The naive generation of a hyperbolic random graph takes $\Theta(n^2)$ time [AOK15]. Using a polar quadtree adapted to hyperbolic space, von Looz et al. [LMP15] achieved a time complexity of $O((n^{3/2} + m) \log n)$; and by a more sophisticated partitioning of the space, Bringmann et al. [BKL15] obtained an optimal expected linear runtime for generation, which is crucial for large-scale experiments. Furthermore, they showed that such graphs can be stored using a linear number of bits instead of the $O(n \log n)$ bits needed for general graphs. Candellero and Fountoulakis [CF16] also analyze bootstrap percolation in this model and show that already a sublinear number of infected nodes suffices for the infection to spread to $\Omega(n)$ nodes.

Finally, as already pointed out in the introduction, there is an ongoing interest to find a good embedding algorithm that achieves the following. Given a graph G, one seeks to find hyperbolic coordinates for the vertices such that neighboring nodes are placed nearby and disconnected nodes are placed far apart. Such embeddings have been found for the Internet graph [BPK10], metabolic networks [SBS12] and trade networks [Gar+16]. All present embedding algorithms are heuristics with runtimes of order $\Omega(n^2)$ [PPK15; PAK15; Wan+16a; AMA16]. Such embeddings have been used for visualization, greedy routing and ad allocation in social networks [GMB14].

1.2 Contribution & Outline

The thesis is structured as follows.

- In Chapter 2, we introduce basic notations and general techniques that will often be used throughout the thesis. To demonstrate the usefulness of the presented methods, we also prove a novel tail bound on the sum of power-law distributed random variables.

- In Chapter 3, we formally define the hyperbolic random graph model. We also restate some known results about the model which will often be used in this thesis. For the sake of completeness, we also reprove them. Many of our proofs are much shorter and simpler than in the available literature.

- In Chapter 4, we analyze the emergence of cliques in the hyperbolic random graph model. We prove that when $\beta \geqslant 3$, the largest clique is of order $\Theta(\frac{\log n}{\log \log n})$; and for $2 < \beta < 3$ the largest clique is of order $\Theta(n^{(3-\beta)/2})$. For a given k, we also compute the expected number of k-cliques. Finally, we present an algorithm for finding these cliques in polynomial time when the geometry of the graph is known. The results in this section appeared in part in [FK15a].

- In Chapter 5, we analyze the diameter of hyperbolic random graphs. We show that the giant component has a diameter of order $O(\log^{2/(3-\beta)} n)$ when $2 < \beta < 3$. When $\beta > 3$, a giant component does not exist and we show that all components have diameter $O(\log n)$. We complement this result by showing a lower bound of $\Omega(\log n)$ for both cases. The results in this section appeared in part in [FK15b].
- In Chapter 6, we present a new algorithm for embedding graphs into the hyperbolic plane. Our algorithm uses only the graph structure (i. e., an edge list) and no further meta-information. It is the first to run in quasilinear time and thus manages to embed even large-scale real-world graphs like the Amazon product network ($n = 300\,000$) in under one hour on commodity hardware. To evaluate our algorithm, we perform large-scale experiments on over 6 000 generated hyperbolic random graphs. This makes it possible to make statistically valid statements on the quality and time performance of our algorithm compared to previous results that were only able to examine very few graphs due to time constraints. We find that our algorithm performs well on all inputs; and its quality increases with the number of nodes and the average degree of the graph. The results in this section appeared in part in [Blä+16].
- In Chapter 7, we conclude and give an outlook on open problems and future directions for research on this topic.

2 Preliminaries

In this thesis, we mostly work with probability distributions on graphs. In this chapter, we give a short overview over these topics to introduce notation and often used methods. We assume, however, that the reader is already familiar with the basics of stochastics and graph theory. For a more in-depth introduction, we refer the reader to standard textbooks [Die12; MU05]. That said, we point out that our approach to probability theory does have a slightly different paradigm than presented in most of these books.

In fact, over the past decades, the analysis of stochastic processes has been refined to yield numerous powerful tools. Entire books have been written on proof techniques like tail bounds [DA09], inequalities [Bul15], Markov chains [LPW08], the probabilistic method [AS15a] and many more. Obtaining an overview over all of these methods can seem to be a daunting task. Many of these tools are also very sharp. They require an exact description of the underlying probability space and yield results with tight constant factors. Our setting, however, is more relaxed. We usually assume a distribution that follows *roughly* a power-law (we define what this means later on), and we are content with asymptotic results that hold in the limit.

Assume, for example, that X is a positive discrete random variable that is distributed as $\Pr[X \geqslant i] = O(i^{1-\beta})$, where β is some constant larger than 2. Note that the asymptotic term $O(\cdot)$ hides the exact distribution function and we therefore are unable to access the individual probabilities $\Pr[X = i]$. Thus, it becomes difficult to compute the expectation of X using the common approach

$$\mathbb{E}\left[X\right] = \sum_i i \cdot \Pr[X = i].$$

To resolve this, one might be tempted to reconstruct the individual probabilities $\Pr[X = i]$ by considering $\Pr[X = i] = \Pr[X \geqslant i] - \Pr[X \geqslant i+1]$. Note that this method is, however, too exact: We have no information on the leading constants of $\Pr[X \geqslant i]$; and thereby, we cannot solve above difference. It may even be equal to 0: Consider the probability distribution

$$\Pr[X = i] = \begin{cases} \delta i^{-\beta} & : \text{ if } i \text{ is even} \\ 0 & : \text{ if } i \text{ is odd.} \end{cases} \tag{2.1}$$

For an appropriately chosen constant δ, this probability distribution sums to 1 and fulfills the above property of $\Pr[X \geqslant i] = O(i^{1-\beta})$. Yet, half of the image of X has probability 0. While this may seem like an artificial example, similar distributions may occur in the real world. For example, one may require that a generated random graph has a Euler tour which is only possible if all degrees are even.

While precise methods certainly have many applications, they are sometimes unsuitable for trying to understand the asymptotic behavior of random structures and algorithms. They require too many assumptions on the distribution and yield results with a precision we are not interested in. It thus is helpful to have a collection of tools that are applicable even when the underlying probability distribution is only "roughly known". One such counterpart for the example above is for instance the formula

$$\mathbb{E}\left[X\right] = \sum_i \Pr[X \geqslant i] = \sum_i O(i^{1-\beta}) = O(1), \tag{2.2}$$

if $\beta > 2$. The first equation $\mathbb{E}\left[X\right] = \sum_i \Pr[X \geqslant i]$ is a classical result and can be proven by reordering elements of the sum. The last equation can be computed by approximating sums via integrals, which we explain in Section 2.5. In light of the asymptotic analysis, we have thus gained the intuition that to compute the (asymptotic) expectation we do not need to know the exact probability distribution of the random variable X.

To be more precise, our setting is often of the kind $\Pr[X \geqslant i] \leqslant O(f(i))$, where f is some function. In this chapter we present techniques that are still applicable in this setting. We also give a few preliminary examples on how to use these methods, along with a more involved derivation of a tail bound on power-law distributed random variables.

2.1 Graph Theory

Before we introduce the relevant probabilistic notions, let us briefly revisit key concepts of graph theory. We note that in order to keep this section brief, we refer the interested reader for a more thorough introduction to standard textbooks, e. g. [Die12].

An (undirected) graph $G = (V, E)$ consists of a finite set of nodes (also called vertices) V and a set of edges $E \subseteq \binom{V}{2}$. Typically, a graph represents a network like a social or computer network, where nodes represent individuals and edges represent some kind of relation between those individuals (e. g. friendship or connection). We typically denote $n = |V|$ and $m = |E|$. To avoid confusion, we require that $E \cap V = \emptyset$, i. e., there exists no node that is also an edge.

Whenever $\{x, y\} \in E$, we say that x is *adjacent* to y. Equivalently, we say that x and y are *neighbors*. The set of all neighbors of x is denoted by $\Gamma(x)$, and the *degree* of x is $\deg(x) := |\Gamma(x)|$. A *clique* is a set $C \subseteq V$ of pairwise adjacent vertices. We call $\pi = [v_1, \dots, v_k]$ a *path* from v_1 to v_k of length $|\pi| = k - 1$, if for all $1 \leqslant i < k$, v_i is adjacent to v_{i+1}. The *distance* between two nodes u, v is defined as $\mathrm{dist}(u, v) := \min\{|\pi| \mid$

π is a path from u to $v\}$. If there is no such path, we set $\mathrm{dist}(u, v) := \infty$. The *diameter* of a graph G is then defined as $\mathrm{diam}(G) := \max\{\mathrm{dist}(u, v) \mid u, v \in V, \mathrm{dist}(u, v) < \infty\}$. We say that the graph G is *connected*, if for every pair of vertices $u, v \in V$ there exists a path from u to v.

The considered graphs in this thesis are usually disconnected. Following text book notation, we define the induced subgraph $G[S]$ on a set of nodes $S \subseteq V$ as $G[S] = (S, \{e \in E \mid e \subseteq S\})$. Then, we say that $S \subseteq V$ is a connected component if the induced subgraph $G[S]$ is connected and S is maximal. Finally, we say that a sequence of graphs $(G_n)_{n=1}^{\infty}$ has a *giant component*, if the graphs have a connected component of size $\Omega(n)$. If all other components have sublinear size $o(n)$, we say that it has a *unique* giant component.

Finally, we define the *clustering coefficient* $\mathrm{cc}(G)$ of a graph G as follows. For a single node v, its clustering $\mathrm{cc}(v)$ is given by the fraction of closed triangles that are formed by its neighbors. Formally,

$$\mathrm{cc}(v) := \frac{|\{e \in E \mid e \subseteq \Gamma(v)\}|}{\binom{\deg(v)}{2}}.$$

For the whole graph, we average over these values with $\mathrm{cc}(G) := \frac{1}{n} \sum_{v \in V} \mathrm{cc}(v)$.

2.2 Probability Theory

In this section, we introduce the basic concepts of probability theory that are often (implicitly) used throughout the thesis. For a thorough introduction, we again refer to standard textbooks [Bol98; AS15a; MU05].

A *probability space* is a triple $(\Omega, \mathcal{F}, P)$, where Ω describes the set of possible outcomes $\omega \in \Omega$; $\mathcal{F}$ is a σ-Algebra on Ω describing the set of events; and $P\colon \mathcal{F} \to [0, 1]$ is a probability measure that describes the probability of an event to occur. Throughout this thesis, we are mostly concerned with the probability space of graphs of size n.[1] Since the number of such graphs is finite, we simply obtain that the set of outcomes Ω_n is the set of all graphs of size n. An event then encodes whether a certain property holds for a given outcome (e. g., the graph is a tree). This is achieved by setting $\mathcal{F}_n = 2^{\Omega}$. The probability that an event occurs is then simply the sum of all probabilities of the outcomes in the event, i. e. for $E \in \mathcal{F}_n$, $P_n(E) = \sum_{\omega \in E} P_n(\omega)$.

These properties hold for all random graph models in this thesis. They differ only in the definition of $P_n(\omega)$, which assigns sampling probabilities to different graphs. In an effort to keep notation terse, we omit the probability space and the dependence on n whenever it is clear from context. We note that the asymptotic notation throughout the thesis is in the parameter n, i. e., we may assume that n is large. To this end, we

[1] One can also interpret this dependence on n as a sequence of probability spaces.

say that an event E holds *with high probability (w. h. p.)*, if there exists a constant c (independent of n) such that $\Pr[E] \geqslant 1 - O(n^{-c})$.

A function $X\colon \Omega \to \mathbb{R}$ is called a *random variable*. As we are usually interested in the value of some random variable, we define

$$\Pr[X = x] := \sum_{\substack{\omega \in \Omega \\ X(\omega) = x}} P(\omega).$$

Following standard notation, we denote by $\mathbb{E}[X]$ the *expectation* of X, i.e.

$$\mathbb{E}[X] := \sum_{\omega \in \Omega} P(\omega) \cdot X(\omega) = \sum_{x} x \cdot \Pr[X = x],$$

and by $\mathbf{Var}[X]$ the *variance* of X, i.e.

$$\mathbf{Var}[X] := \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2.$$

We are often concerned with the *distribution* of a random variable X (e. g. the degree distribution of a random graph). To this end, we define the *complementary cumulative distribution function* (CCDF) as

$$F_X(x) := \Pr[X \geqslant x].$$

We omit the subscript X whenever the random variable X is clear from the context. Note that this does *not* follow standard textbook notation, as $F(x)$ is classically defined as the cumulative distribution function (CDF) $\Pr[X < x]$. We deviate from this notion since we almost exclusively need $\Pr[X \geqslant x]$ and writing $[1 - F](x)$ throughout the whole thesis becomes rather verbose.

Finally, we define the notion of a power law degree distribution. Since the formal definition has to deal with some technicalities, let us first describe the naive approach that explains why these technicalities occur.

Intuitively, we would like to write that the fraction of nodes of degree exactly k is roughly $k^{-\beta}$. By counting the exact number of occurrences of degrees, however, this forbids degree distributions with “gaps” as in Equation (2.1). To make the definition more general and robust, it is therefore usual to consider the CCDF and require that the number of nodes with degree *larger* than k is roughly $k^{1-\beta}$. To make this formal, we would like to write

$$\alpha_1 k^{1-\beta} \leqslant \tfrac{1}{n} |\{v \mid \deg(v) \geqslant k\}| \leqslant \alpha_2 k^{1-\beta},$$

where α_1, α_2 are some constants depending on the network. This requires us, however, to keep track of these constants and parameterize the definition not only by β, but also by α_1, α_2. We circumvent this issue by giving the definition for a sequence of graphs, as we typically are interested in the asymptotic behavior, i. e., when $n = |V| \to \infty$. Then, we may simply require that there *exist* constants α_1, α_2.

Another issue is that the lower bound on the degree distribution $\alpha_1 k^{1-\beta}$ is larger than 0 for all k. Finite graphs can not fulfill such a definition. We therefore introduce a cutoff for the lower bound on the power law degree distribution. The choice of the cutoff is somewhat arbitrary (other than that it should increase with n), which is why we set it as large as possible such that hyperbolic random graphs still fulfill this property. This leads us to the following formal definition.

Definition 1. Let $(G_n)_{n=1}^{\infty}$ be a sequence of graphs. We say that these graphs have a *power law degree distribution with exponent* β, if there exist constants $\alpha_1, \alpha_2 > 0$ such that for all n, the following holds.

1. $\frac{1}{n}|\{v \mid \deg(v) \geqslant k\}| \geqslant \alpha_1 k^{1-\beta}$ for $k \leqslant n^{\frac{1}{\beta-1}} / \log n$, and
2. $\frac{1}{n}|\{v \mid \deg(v) \geqslant k\}| \leqslant \alpha_2 k^{1-\beta}$ for $k > 1$.

The cutoff for first condition stems from the fact that the largest degree in such graphs is of order $\Theta(n^{\frac{1}{\beta-1}})$. We note that in practice, however, the second condition is used much more often as it upper bounds the number of nodes in the graph with too large degree.

2.3 Taylor Approximation

Almost all inequalities that are often applied in probabilistic analysis are consequences of the Taylor approximation. This is a useful tool for transforming convoluted expressions into simpler ones that are susceptible to further simplifications. We first state the method and then give a few examples.

Definition 2 (Taylor Expansion)**.** Let $f\colon \mathbb{R} \to \mathbb{R}$ be $k > 1$ times differentiable at $a \in \mathbb{R}$. Then, we call $T_f(x)$ the degree k *Taylor expansion* of f around point a with

$$T_f(x) = \sum_{n=0}^{k} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a) + \tfrac{1}{1!}f'(a) \cdot (x-a) + \ldots + \tfrac{1}{k!}f^{(k)}(a) \cdot (x-a)^k.$$

The Taylor expansion is useful since it is very accurate around point a. The error is quantified by the remainder $f(x) - T_f(x)$. To compute tight approximation for most functions, we use the following estimate of the remainder.

Theorem 2.1 ([Kli98])**.** *Let $f\colon \mathbb{R} \to \mathbb{R}$ be $k+1$ times differentiable on the open interval between a and x; and $f^{(k)}$ be continuous on the closed interval between a and x. Consider* *the degree k Taylor expansion T_f around point a. Then, there exists a ξ between a and x such that*

$$r(x) := f(x) - T_f(x) = \frac{f^{(k+1)}(\xi)}{(k+1)!}(x-a)^{k+1}.$$

Let us now review a few popular inequalities that are often used.

Lemma 2.1. *Let $x \in \mathbb{R}$. Then, $1 + x \leqslant e^x$.*

Proof. Let $f(x) = e^x$ and consider the degree-1 Taylor expansion around $a = 0$ with $T_f(x) = 1 + x$. By Theorem 2.1 there exists a constant ξ between 0 and x such that the remainder is

$$r(x) = \tfrac{1}{2}e^{\xi}x^2 \geqslant 0.$$

Thus, we can write $e^x = 1 + x + \frac{1}{2}e^{\xi}x^2 \geqslant 1 + x$. □

Sometimes, it is useful to have a bound in the other direction on $1 - x$, when $x \to 0$. This is given by the following lemma.

Lemma 2.2. *Let $0 < x < 1$ and let ε be such that $1 - x = e^{-\varepsilon}$. Then, $1 - x \geqslant e^{-(1+\varepsilon)x}$.*

Proof. Since $1-x = e^{-\varepsilon}$, it suffices to show that $e^{-\varepsilon} \geqslant e^{-(1+\varepsilon)x}$. To see this, we compute

$$\begin{aligned} e^{-\varepsilon} &\geqslant e^{-(1+\varepsilon)x} \\ \Leftrightarrow \varepsilon &\leqslant (1+\varepsilon)x \\ \Leftrightarrow (1-x) &\leqslant \tfrac{1}{1+\varepsilon}. \end{aligned}$$

Thus, by estimating $1 + \varepsilon \leqslant e^{\varepsilon}$ using Lemma 2.1, the statement holds if $1 - x \leqslant e^{-\varepsilon}$. But this is true by assumption. □

Notice that when x is very small, i. e., $x = o(1)$, it must hold that $\varepsilon = o(1)$. Thus, we have that $1 - x \geqslant e^{-(1+o(1))x}$, i. e., the estimation $1 - x \approx e^{-x}$ is accurate in that case.

Lemma 2.3. *Let $x \in \mathbb{R}$ with $x = \pm o(1)$. Then, $\frac{1}{1+x} = 1 - x + \Theta(x^2) = 1 - \Theta(x)$.*

Proof. Let $f(x) = \frac{1}{1+x}$ and consider the degree-1 Taylor expansion around $a = 0$. We have $f'(x) = -(\frac{1}{1+x})^2$ and thus $T_f(x) = 1 - x$. The remainder is then

$$r(x) = f(x) - T_f(x) = \tfrac{1}{1+x} - 1 + x = \tfrac{1+x^2-1}{1+x} = \tfrac{x^2}{1+x}.$$

For $x = \pm o(1)$ this becomes $r(x) = \Theta(x^2)$. □

Lemma 2.4. *Let $n \geqslant 0$. Then, $n^{1+\frac{1}{n}} - n = (1 + o(1)) \log n$.*

Proof. We consider the degree-1 Taylor expansion at infinity. To this end, we substitute $x = \frac{\log n}{n}$ and write $g(x) := e^x - 1$. Thus, we have $f(n) = ng(\frac{\log n}{n})$. We consider the degree 1 expansion T_g at $a = 0$, which is

$$T_g(x) = 0 + x.$$

Again, we have that $\exists \xi \in [0, x]$ such that $r(x) = \frac{1}{2}e^{\xi}x^2 \leqslant e^x x^2$. Thus, $g(x) \leqslant x + e^x x^2$. By resubstituting $x = \frac{\log n}{n}$ we obtain

$$f(n) \leqslant \log n + O(\tfrac{\log^2 n}{n}). \qquad \square$$

2.4 Chernoff & Union Bound

The Chernoff and the union bound number among the most important tools in probabilistic analysis. The former states that a sum of random variables in $[0, 1]$ is strongly concentrated around its expectation. It comes in several variations, we only need the multiplicative form, which is as follows.

Theorem 2.2 (Chernoff bound)**.** *Let* $X = \sum_{i \in [n]} X_i$*, where* X_i*,* $i \in [n]$ *are independently distributed in* $[0, 1]$*. Then,*

$$\begin{aligned} &\Pr[X \geqslant (1+\varepsilon)\mathbb{E}[X]] \leqslant \exp(-\tfrac{\varepsilon^2}{2+\varepsilon}\mathbb{E}[X]) && \text{for } \varepsilon > 0 \\ &\Pr[X \leqslant (1-\varepsilon)\mathbb{E}[X]] \leqslant \exp(-\tfrac{1}{2}\varepsilon^2\mathbb{E}[X]) && \text{for } 0 < \varepsilon < 1. \end{aligned}$$

The proof is folklore and can be found e. g. in [DA09]. Note that the Chernoff bound requires no precise knowledge of the probability distribution of X; apart from it being a sum of independent variables. It can be even further relaxed to only require submartingales. Note further that this is not the strongest version of the Chernoff bound. As stated above, however, we are not interested in constants but rather in the asymptotic behavior. To this end, this bound is tight, see e. g. [KY15, Lemma 4]. As a result of the Chernoff bound, and often used implicitly in this thesis, we obtain that if $\mathbb{E}[X] = \Theta(\log n)$, then X is with high probability of order $\Theta(\log n)$; and if $\mathbb{E}[X] = \omega(\log n)$, then X is with high probability of order $(1 \pm o(1)) \cdot \mathbb{E}[X]$.

The union bound is used to bound the probability that given a set of events at least one of them happens. The proof can be found for example in [MU05].

Theorem 2.3 (Union Bound)**.** *Let* $\mathcal{E}$ *be a finite or countable set of events. Then,*

$$\begin{aligned} \Pr[\exists E \in \mathcal{E}] &\leqslant \sum_{E \in \mathcal{E}} \Pr[E], \\ \Pr[\exists E \in \mathcal{E}] &\geqslant \sum_{E \in \mathcal{E}} \Pr[E] - \sum_{E, F \in \mathcal{E}} \Pr[E \wedge F]. \end{aligned}$$

2.5 Replacing Sums by Integrals

In this section, we mention a useful technique to approximate sums via integrals. Even though we do not need this method explicitly within the main chapters of this thesis, we still believe that this is a crucial method for analyzing random graphs. We demonstrate the usefulness of this technique in Section 2.6, where we prove a tail bound on the sum of power law distributed random variables.

Since graphs are discrete structures, most derived values like the expectation appear in the form of sums, see e. g. Equation (2.2). Unfortunately, these sums can analytically not be computed when the summands follow a power law as in the above example. Consider for instance

$$\sum_{i=x}^{n} i^{-1.5}.$$

Computing an exact representation requires the use of the Riemann Zeta function. Such an expression is undesirable since it is hard to further digest. We can, however, approximate the sum by an integral. To this end, we observe that $i^{-1.5}$ is a monotonically decreasing function in i; and thus

$$\sum_{i=x}^{n} i^{-1.5} \leqslant \int_{x-1}^{n} i^{-1.5}\,\mathrm{d}i = \left[-2i^{-\frac{1}{2}}\right]_{x-1}^{n} = \Theta(x^{-\frac{1}{2}}).$$

Bringmann, Keusch, and Lengler [BKL15] recently have shown a tight representation of sums by integrals, which we restate in the following.

Theorem 2.4 ([BKL15])**.** *Let $\vec{w} = (w_1, \dots, w_n)$ and define $F(w) := \frac{1}{n}|\{i \mid w_i \geqslant w\}|$ and $F^{>}(w) := \frac{1}{n}|\{i \mid w_i > w\}|$. Let further $f\colon \mathbb{R} \to \mathbb{R}$ be a continuously differentiable function. Then, for any $0 \leqslant \underline{w} \leqslant \overline{w}$,*

$$\sum_{i\in[n],\, \underline{w}\leqslant w_i\leqslant\overline{w}} \tfrac{1}{n} f(w_i) = f(\underline{w}) \cdot F(\underline{w}) - f(\overline{w}) \cdot F^{>}(\overline{w}) + \int_{\underline{w}}^{\overline{w}} f'(w) \cdot F(w)\,\mathrm{d}w.$$

Applied to our example from above, we set $\vec{w} := (1^{-1.5}, 2^{-1.5}, \dots, n^{-1.5})$; and define $f(w) := w$. Obviously, f is continuously differentiable. For $F(w)$, we obtain

$$F(w) = \tfrac{1}{n}|\{i \mid i^{-1.5} \geqslant w\}| = \tfrac{1}{n}|\{i \mid i \leqslant w^{-\frac{2}{3}}\}| = \Theta(\tfrac{1}{n} w^{-\frac{2}{3}}).$$

The same can be shown for $F^{>}(w)$. Thus, we have

$$\sum_{i=x}^{n} i^{-1.5} = n \sum_{w \leqslant x^{-1.5}} \tfrac{1}{n} f(w) = \Theta(n) \cdot \left(n^{-1.5} - \tfrac{1}{n} x^{-\frac{1}{2}} + \int_{n^{-1.5}}^{x^{-1.5}} \tfrac{1}{n} w^{-\frac{2}{3}}\,\mathrm{d}w \right) = \Theta(x^{-\frac{1}{2}}).$$

This technique comes in particularly handy when we deal with weight distributions $\vec{w}$ of which we only know the complementary cumulative distribution function $F(w)$.

2.6 Tail Bounds for Sums of Power-Law Variables

In this section, we derive new tail bounds for sums of power-law distributed random variables using the methods introduced above. If the variables are independent, we further show that these bounds are tight up to an arbitrary small ε in the exponent. To the best of our knowledge, such results do not exist to date. We apply our tail bounds to analyze the size of a random Galton-Watson tree, where the number of children per node is distributed as a power law.

Theorem 2.5. *Let $\beta > 2$ and the random variables $X_1, \ldots, X_n$ be such that for all x*

$$\min\{\alpha_1 x^{1-\beta}, 1\} \leqslant \Pr[X_i \geqslant x \mid X_1, \ldots, X_{i-1}] \leqslant \alpha_2 x^{1-\beta},$$

for some constants α_1, α_2. Let $X := \sum_{i=1}^n X_i$ and $s = (1+\delta)\mathbb{E}[X]$ for some constant $\delta > 0$. Then, if s is at least a large enough constant,

$$\Pr[X \geqslant s] = O(ns^{1-\beta} \log^{\beta-1} s).$$

If the X_i's are independent and $n \cdot \Pr[X_i \geqslant s] < 1 - \varepsilon$, for some $\varepsilon > 0$ we further have

$$\Pr[X \geqslant s] = \Omega(ns^{1-\beta}).$$

Proof. We first prove the second, more obvious statement. Let x_0 be defined such that $\alpha_1 x_0^{1-\beta} = 1$. Since the X_i's are independent, we then have $\Pr[X_i \geqslant x] \geqslant \alpha_1 x^{1-\beta}$ for all i and $x \geqslant x_0$. By Theorem 2.3,

$$\begin{aligned}\Pr[X \geqslant s] \geqslant \Pr[\exists i, X_i \geqslant s] &\geqslant \sum_i \Pr[X_i \geqslant s] - \sum_{i,j} \Pr[X_i, X_j \geqslant s] \\ &\geqslant \sum_i \Pr[X_i \geqslant s] \cdot \Big(1 - \sum_j \Pr[X_j \geqslant s]\Big).\end{aligned}$$

By using the assumption $n \cdot \Pr[X_i \geqslant s] < 1 - \varepsilon$, we can upper bound the second sum with $1 - \varepsilon$. Furthermore, because $s = (1+\delta)\mathbb{E}[X] > \mathbb{E}[X]$ and therefore $s > x_0$, we obtain

$$\Pr[X \geqslant s] \geqslant \varepsilon \alpha_1 n s^{1-\beta},$$

proving the second statement. To show the first statement, consider some value t (which we choose appropriately later) that is at least some large enough constant. Then,

$$\Pr[X \geqslant s] \leqslant \Pr[\exists i\colon X_i \geqslant t] + \Pr[X \geqslant s \mid \forall i\colon X_i < t]. \tag{2.3}$$

For the first part of the sum, we can immediately follow by the union bound $\Pr[\exists i\colon X_i \geqslant t] \leqslant \alpha_2 \cdot nt^{1-\beta}$. It thus remains to upper bound the second part of the sum.

To this end, we first compute the conditional moments of the X_i's.

$$\begin{aligned}
\mathbb{E}\left[X_i^k \mid X_i < t, X_1, \ldots, X_{i-1}\right] &= x_0^k + \int_{x_0}^{t} kx^{k-1} \cdot \Pr[X_i \geqslant x \mid X_i < t, X_1, \ldots, X_{i-1}] \,\mathrm{d}x \\
&\leqslant x_0^k + \int_{x_0}^{t} \frac{\Pr[X_i \geqslant x \mid X_1, \ldots, X_{i-1}]}{\Pr[X_i < t \mid X_1, \ldots, X_{i-1}]} \cdot kx^{k-1} \,\mathrm{d}x \\
&\leqslant x_0^k + 2\alpha_2 k \int_{x_0}^{t} x^{k-\beta} \,\mathrm{d}x \\
&\leqslant x_0^k + \begin{cases} \frac{2\alpha_2 k}{k-\beta+1} \cdot t^{k-\beta+1} & \text{if } \beta < k+1, \\ 2\alpha_2 k \log t & \text{if } \beta = k+1, \\ \frac{2\alpha_2 k}{\beta-k-1} \cdot x_0^{\beta-k-1} & \text{if } \beta > k+1. \end{cases}
\end{aligned}$$

In the following, we only need $k \leqslant 2$. For $k = 1$ we obtain regardless of t that $\mathbb{E}[X_i] = \Theta(1)$ holds for all i. Thus, for the sum we have $\mathbb{E}[X] = \Theta(n)$. For $k = 2$, we obtain a bound on the conditional variances that is of order $o(t)$. The sum of conditional variances $\hat{v} := \sum_{i=1}^{n} \mathbf{Var}[X_i \mid X_i < t, X_1, \ldots, X_{i-1}]$ is then of order $o(nt)$.

We are then ready to prove the main result. To this end, we apply a variant of Bernstein's inequality for Martingales [McD98, Theorem 3.15]. Recall that $s = (1+\delta)\mathbb{E}[X]$ for some $\delta > 0$. Observe further that due to the condition, $\sup X_i \leqslant t$.

$$\begin{aligned}
\Pr[X \geqslant s \mid \forall X_i\colon X_i < t] &= \Pr[X - \mathbb{E}[X] \geqslant \delta\mathbb{E}[X] \mid \forall X_i\colon X_i < t] \\
&\leqslant \exp\left(-\frac{\delta^2 \mathbb{E}[X]^2}{2\hat{v} + \frac{2}{3}\delta t \mathbb{E}[X]}\right).
\end{aligned}$$

Since $\hat{v} = o(nt)$, we have $2\hat{v} \leqslant \frac{1}{3}\delta t\mathbb{E}[X]$ if t is a large enough constant.

$$\begin{aligned}
\Pr[X \geqslant s \mid \forall X_i\colon X_i < t] &\leqslant \exp\left(-\frac{\delta^2 \mathbb{E}[X]^2}{\delta t \mathbb{E}[X]}\right) \\
&\leqslant \exp\left(-\tfrac{1}{t}\delta\mathbb{E}[X]\right).
\end{aligned}$$

It remains to choose an appropriate value for t. By setting $t := \frac{\delta\mathbb{E}[X]}{(\beta-1)\log s}$, we have that if s is a large enough constant, then so is t. Therefore, the above steps were correct and we write

$$\begin{aligned}
\Pr[X \geqslant s \mid \forall X_i\colon X_i < t] &\leqslant \exp\left((1-\beta) \cdot \frac{\log s}{\delta\mathbb{E}[X]} \delta\mathbb{E}[X]\right) \\
&\leqslant s^{1-\beta}.
\end{aligned}$$

By combining this with Equation (2.3), we obtain that

$$\Pr[X \geqslant s] \leqslant \alpha_2 n \left(\frac{\delta \mathbb{E}\,[X]}{(\beta-1)\log s}\right)^{1-\beta} + s^{1-\beta} = O(ns^{1-\beta}\log^{\beta-1} s). \qquad \square$$

Using this tail bound, we analyze the size of a random Galton-Watson tree T. Such a tree is obtained as follows: We begin with a single node as root and let T_i denote the number of nodes in generation i. This means that in generation 0 we have $T_0 = 1$. Then, for every subsequent generation, we compute

$$T_{i+1} := \sum_{j=1}^{T_i} C_j,$$

where all C_j's are independent copies of the same random variable C.

This process has been throughly analyzed for various offspring distributions C [Lin76; Ald91a; Ald91b; Ald93]. Most notable is the result on the extinction probability $\Pr[\exists i\colon T_i = 0] = 1 \Leftrightarrow \mathbb{E}\,[C] \leqslant 1$ (if one excludes the trivial case $\Pr[C = 1] = 1$). Many other results have been derived, but most of them rely on the probability generating function of C. Unfortunately, if C is power-law distributed, the probability generating function maps to infinity and is therefore of limited use. In the following, we bound the size of a Galton-Watson tree when $\mathbb{E}\,[C] \leqslant 1$, but C is distributed as a power-law.

Corollary 2.1. *Let $\beta > 2$ and let T be a Galton-Watson random tree with offspring distribution $F(c) = \Theta(c^{1-\beta})$ for $c \geqslant c_0$ where $c_0 = \Theta(1)$. If the expected number of children is bounded away from 1, i. e., if $1 - \mathbb{E}\,[C] = \Omega(1)$, then for all $\varepsilon > 0$*

$$\Pr[|T| \geqslant t] = O(t^{2-\beta+\varepsilon}).$$

Proof. Observe that in a finished Galton-Watson tree T of size t, the t nodes produced $t-1$ offspring in total. Thus, the probability that T has more than t nodes is upper bounded by the probability that t nodes produce $> t-1$ offspring. By definition, all number of children C_i are power-law distributed and independent. Further, we have that $(t-1) - t \cdot \mathbb{E}\,[C] = \Omega(t)$. Thus, all conditions of Theorem 2.5 are fulfilled and we obtain

$$\Pr[|T| > t] \leqslant \Pr[\sum_{i=1}^{t} C_i \geqslant t-1] \leqslant O(t^{2-\beta+\varepsilon}). \qquad \square$$

Note that since the extinction probability is 1 when $\mathbb{E}\,[X] < 1$, the expected size of such a tree is constant. By applying a Markov bound one obtains $\Pr[|T| > t] \leqslant O(\frac{1}{t})$. The above bound thus only becomes useful whenever $\beta > 3$.

3 Hyperbolic Random Graphs

In this chapter, we present the definition of the hyperbolic random graphs, give an overview over existing results in the literature and show several important theorems that will often be referred to in this thesis.

3.1 Model Definition

As pointed out in the introduction, the basic premise of hyperbolic random graphs is simple. To create a graph, sample nodes randomly in the hyperbolic plane, and connect them by edges if they are nearby. In this section, we describe the graph generation in more detail.

For convenience, we use the *native* representation of the hyperbolic plane $\mathbb{H}_2$. Here, a point x is identified by a radial and an angular coordinate (r_x, φ_x), where the radial coordinate denotes the hyperbolic distance from the coordinate origin. The hyperbolic space is also typically equipped with some negative curvature. In our case, however, it has been shown that there exists a coupling between random hyperbolic graphs on different curvatures [BFM14]. Therefore, using different curvatures is equivalent to rescaling other model parameters which is why we implicitly set the curvature to -1 in all subsequent considerations.

Then, the hyperbolic distance $\operatorname{dist}(x, y)$ between two points $x, y \in \mathbb{H}_2$ is given by

$$\cosh(\operatorname{dist}(x, y)) := \cosh(r_x)\cosh(r_y) - \sinh(r_x)\sinh(r_y)\cos(\Delta\varphi_{x,y}), \tag{3.1}$$

where $\Delta\varphi_{x,y} := \pi - |\pi - |\varphi_x - \varphi_y||$ is the angular distance between two nodes x, y. Furthermore, the area of a hyperbolic circle with radius r is

$$A(r) = 2\pi(\cosh(R) - 1).$$

The hyperbolic sine and cosine which naturally appear often in this model are

$$\sinh(x) = \frac{e^x - e^{-x}}{2}, \quad \cosh(x) = \frac{e^x + e^{-x}}{2}. \tag{3.2}$$

Notice that both functions grow exponentially and are equal to $(1 \pm e^{-2x})e^x/2$. Thus, for large x, these functions essentially behave as $e^x/2$.

Using these notions, we may define the hyperbolic random graph model. Informally, the positions of n nodes are sampled with uniform density in a hyperbolic circle with

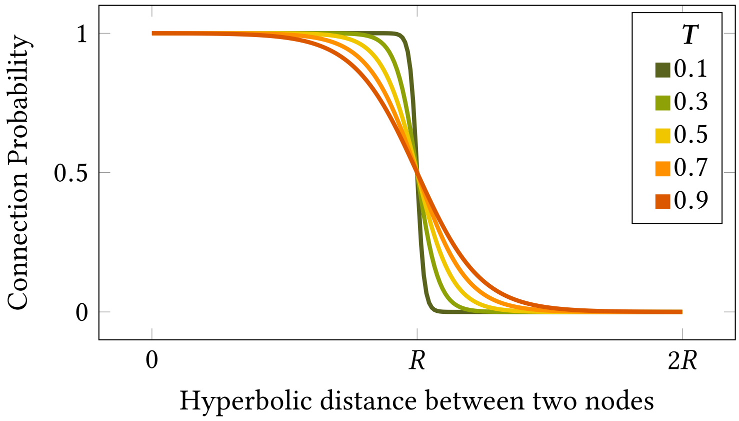


Figure 3.1: Connection probability of two nodes in the binomial model depending on their hyperbolic distance. The parameter T adjusts the number of long-range edges. Typically, T is chosen between 0 and 1. When $T \to 0$, this corresponds to the step model and edges appear if and only if two nodes have distance $\leqslant R$.

radius $R = 2\log n + C$ for some constant C. Two nodes are then connected, if their distance is at most R. This approach yields a hyperbolic random graph with power law exponent $\beta = 3$. To allow for different power law exponents, one may distort the node density in the circle. This corresponds to sampling nodes uniformly in the circle for a hyperbolic plane with different curvature, and then taking the distance of nodes in the original curvature. For convenience, we stick to the curvature -1, and define hyperbolic random graphs as follows.

Definition 3 (Hyperbolic Random Graphs)**.** Let $n \in \mathbb{N}$ and $\alpha, C \in \mathbb{R}$ with $\alpha > \frac{1}{2}$ be given. Consider the disc D_R with radius $R = 2\log n + C$ centered in $\vec{0}$. For each $i \in [n]$, sample a point $(r_i, \varphi_i) \in D_R$ using the distribution function

$$f(r,\varphi) := f(r) := \frac{1}{2\pi} \cdot \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1}. \tag{3.3}$$

Then, $G = ([n], E)$ is a hyperbolic random graph drawn from the

1. *step model*, if $E = \{\{u,v\} \mid \operatorname{dist}(u,v) \leqslant R\}$, and
2. *binomial model*, if for some given $T > 0$ every edge $\{u,v\}$ is independently added to E with probability

$$p_{uv} := p(\operatorname{dist}(u,v)) := (1 + \exp(\tfrac{1}{2T}(\operatorname{dist}(u,v) - R)))^{-1}.$$

Let us briefly describe the intuition behind each of the parameters. The value R is the radius of the hyperbolic disc and is always of order $2 \cdot \log n$. The deviation of R

from $2 \log n$, which is parameterized by C, adjusts the average degree δ of the resulting hyperbolic random graph:

$$\delta = (1 + o(1)) \frac{2\alpha^2 e^{-C/2}}{(\alpha - 1/2)^2} \left(\lim_{t \to T} \frac{t}{\sin(\pi t)} \right).$$

The value α adjusts the resulting power-law degree exponent $\beta = 2\alpha + 1$ [GPP12; Pet14; BKL15] (c. f. Definition 1). Similarly to R, the radial node coordinate r_i prescribes the expected degree of node i, whereas φ_i determines the neighborhood of node i. Note that for brevity, we often identify a node u with its coordinates (r_u, φ_u). Finally, T adjusts the noise in the binomial model: For large values of T, it becomes more likely that close nodes are disconnected and far apart nodes still have an edge between them. Figure 3.1 contains an illustration of the connection probability depending on T.

Bode, Fountoulakis, and Müller [BFM13] show that for $\frac{1}{2} < \alpha < 1$ there exists a unique giant component in the graph. At $\alpha = 1$ there is a phase transition where the size of the largest component depends on the average degree, and for $\alpha > 1$ the largest component is of polynomial size.

3.2 Auxiliary Lemmas

We now present a handful of Lemmas useful for analyzing the hyperbolic random graph. Most of them can be found in [GPP12], but we present shorter and simpler proofs in this thesis. We first show a close approximation for $f(r)$ which allows us to calculate with a simple exponential function instead of hyperbolic sines and cosines. We will use this extensively.

Lemma 3.1. *The density function $f(r)$ in Equation* (3.3) *is approximated by*

$$f(r) = \tfrac{\alpha}{2\pi} e^{\alpha(r-R)} \cdot (1 + \Theta(e^{-\alpha R} - e^{-2\alpha r})).$$

Proof. By applying Lemma 2.3, we obtain

$$\begin{aligned} f(r) &= \frac{1}{2\pi} \cdot \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1} \\ &= \tfrac{\alpha}{2\pi} \cdot \frac{e^{\alpha r} - e^{-\alpha r}}{e^{\alpha R} + e^{-\alpha R} - 2} \\ &= \tfrac{\alpha}{2\pi} \cdot \frac{e^{\alpha r}(1 - e^{-2\alpha r})}{e^{\alpha R}(1 - \Theta(e^{-\alpha R}))} \\ &= \tfrac{\alpha}{2\pi} e^{\alpha(r-R)}(1 + \Theta(e^{-\alpha R} - e^{-2\alpha r})). \end{aligned}$$ □

We next compute an upper bound for the angular distance between two connected nodes. Consider two nodes with radial coordinates r, y. Denote by $\theta(r, y)$ the maximal angular distance such that these two nodes are connected in the step model. By Equation (3.1),

$$\theta(r,y) = \max_{\varphi}\{\mathrm{dist}((r,0),(y,\varphi)) \leqslant R\} = \arccos\left(\frac{\cosh(y)\cosh(r) - \cosh(R)}{\sinh(y)\sinh(r)}\right). \quad (3.4)$$

This complicated expression is closely approximated by the following Lemma. Notice that the second condition in the statement is required as otherwise $r + y < R$ and the two corresponding nodes are always connected by the triangle inequality.

Lemma 3.2. *Let $0 \leqslant r, y \leqslant R$ and $y + r \geqslant R$. Then,*

$$\theta(r,y) = \theta(y,r) = 2e^{\frac{R-r-y}{2}}(1 \pm \Theta(e^{R-r-y})).$$

Proof. By definition of cosh and sinh we have

$$\begin{aligned}\theta(r,y) &= \arccos\left(\frac{\cosh(y)\cosh(r) - \cosh(R)}{\sinh(y)\sinh(r)}\right)\\ &= \arccos\left(\frac{e^{y+r}(1 + O(e^{-2y} + e^{-2r})) - 2e^{R}(1 + O(e^{-2R}))}{e^{r+y}(1 - O(e^{-2r} + e^{-2y}))}\right)\\ &= \arccos\left(1 - 2e^{R-r-y} + O(e^{-2r} + e^{-2y})\right). \quad (3.5)\end{aligned}$$

Observe that by assumption, $r + y \geqslant R$ and therefore the inner term is close to 1. We therefore use the series expansion for arccos around point 1, see e. g. [Olv10, p. 121]:

$$\arccos(1-x) = \sqrt{2x} + \frac{\sqrt{2}}{12}x^{\frac{3}{2}} + \frac{3\sqrt{2}}{160}x^{\frac{5}{2}} + O(x^{\frac{7}{2}}).$$

Applying this to Equation (3.5), we get

$$\begin{aligned}\theta(r,y) &= \sqrt{2}\sqrt{2e^{R-r-y} - O(e^{-2r} + e^{-2y})} + \Theta(e^{\frac{3}{2}(R-r-y)})\\ &= 2e^{\frac{R-r-y}{2}} \cdot \left(\sqrt{1 - O(e^{-r+y-R} + e^{-y+r-R})} + \Theta(e^{R-r-y})\right).\end{aligned}$$

Using that $\sqrt{1-x} = 1 - \frac{x}{2} - O(x^2)$, which again follows from a series expansion, we conclude

$$\theta(r,y) = 2e^{\frac{R-r-y}{2}}(1 \pm \Theta(e^{R-r-y})). \qquad \square$$

Note that the error term is $(1 + \Theta(e^{R-r-y}))$ (i. e., we can drop the minus) if either $R - r = \omega(1)$ or $R - y = \omega(1)$.

For most computations on hyperbolic random graphs, we need expressions for the probability that a sampled point falls into a certain area. To this end, we define the *probability measure* of a set $S \subseteq D_R$ as

$$\mu(S) := \int_S f(x)\,\mathrm{d}x,$$

where f is the probability distribution function defined in Definition 3. We further define the *ball* with radius x around a point (r, φ) as

$$B_{r,\varphi}(x) := \{(r', \varphi') \mid \mathrm{dist}((r', \varphi'), (r, \varphi)) \leqslant x\}.$$

We shortly write $B_r(x)$ for $B_{r,0}(x)$. Note that $D_R = B_0(R)$.

Using these definitions, we can formulate the following Lemma.

Lemma 3.3. *For any* $0 \leqslant r, m \leqslant R$ *we have*

$$\mu(B_0(r)) = e^{-\alpha(R-r)}(1 - \Theta(e^{-\alpha r})) \tag{3.6}$$

$$\mu(B_r(R) \cap B_0(R-m)) = \begin{cases} \mu(B_0(R-m)) & \text{if } r \leqslant m, \\ \frac{4\alpha}{\pi(2\alpha-1)} e^{\frac{m-r}{2} - \alpha m} \cdot \mathcal{E} & \text{if } r > m, \end{cases} \tag{3.7}$$

with error term $\mathcal{E} = 1 \pm O(e^{(m-r)(\alpha - \frac{1}{2})})$ *if* $\alpha \neq \frac{3}{2}$ *and* $\mathcal{E} = 1 \pm O(e^{m-r}(r-m))$ *otherwise.*

Proof. For the proof of Equation (3.6), we compute by Lemma 3.1

$$\begin{aligned} \mu(B_0(r)) &= \int_0^r \int_0^{2\pi} f(y)\,\mathrm{d}\varphi\,\mathrm{d}y \\ &= \int_0^r \alpha e^{\alpha(r-R)}(1 + \Theta(e^{-\alpha R} - e^{-2\alpha r}))\,\mathrm{d}r \\ &= \left[e^{\alpha(r-R)}(1 + \Theta(e^{-\alpha R} + e^{-2\alpha r}))\right]_0^r \\ &= e^{\alpha(r-R)}(1 + \Theta(e^{-\alpha R} + e^{-2\alpha r})) - \Theta(e^{-\alpha R}) \\ &= e^{\alpha(r-R)} \cdot (1 - \Theta(e^{-\alpha r})). \end{aligned}$$

The proof of Equation (3.7) is as follows. Consider first the case $r \leqslant m$, for which Figure 3.2a contains an illustration. In that case, the ball $B_r(R)$ fully encloses $B_0(R-m)$, as all points in $B_0(R-m)$ have at most distance $R-m$ to 0; and by the triangle inequality at most distance $R - m + r \leqslant R$ to the center of $B_r(R)$. Therefore, the intersection of those two areas has probability measure exactly $\mu(B_0(R-m)) = e^{-\alpha m}(1 \pm o(1))$, proving the first case.

We now assume $r \geqslant m$. Figure 3.2b contains an illustration for this case. Then, we can write

$$\mu(B_r(R) \cap B_0(R-m)) = \mu(B_0(R-r)) + 2\int_{R-r}^{R-m} \int_0^{\theta(r,y)} f(y)\,\mathrm{d}\theta\,\mathrm{d}y,$$

where $\theta(r,y) = \arccos\left(\frac{\cosh(r)\cosh(y)-\cosh(R)}{\sinh(r)\sinh(y)}\right)$ is given by the definition of the distance function, see Equation (3.4). The first part of the sum vanishes in the error term $\mathcal{E}$, since it simplifies to $(1 \pm o(1))e^{-\alpha r} = e^{\frac{m-r}{2}-\alpha m} \cdot O(e^{(m-r)(\alpha-\frac{1}{2})})$.

For the second part of the sum, have

$$2\int_{R-r}^{R-m}\int_{0}^{\theta(r,y)} f(y)\,\mathrm{d}\theta\,\mathrm{d}y = 2\int_{R-r}^{R-m} \theta(r,y)\cdot f(y)\,\mathrm{d}y$$

By simplifying $\theta(r,y)$ using Lemma 3.2 and $f(y)$ using Lemma 3.1, this term can be transformed to obtain

$$\frac{2\alpha}{\pi}(1+O(e^{-\alpha R}))\int_{R-r}^{R-m} e^{\frac{R-r-y}{2}+\alpha y-\alpha R}(1+O(\pm e^{R-r-y}-e^{-2\alpha y}))\,\mathrm{d}y. \tag{3.8}$$

Observe that the dominant error term is $O(\pm e^{R-r-y})$. This holds since $-2\alpha y < R-r-y$ follows from $(1-2\alpha)y < 0 < R-r$ and thereby $O(\pm e^{R-r-y} - e^{-2\alpha y}) = O(\pm e^{R-r-y})$.

We now first compute the integral without the error term and later add the error term. We obtain

$$\begin{aligned}
&\frac{2\alpha}{\pi}(1+O(e^{-\alpha R}))\int_{R-r}^{R-m} e^{\frac{R-r-y}{2}+\alpha y-\alpha R}\,\mathrm{d}y\\
&= \frac{4\alpha}{\pi(2\alpha-1)}(1+O(e^{-\alpha R}))\left[e^{\frac{R-r-y}{2}+\alpha y-\alpha R}\right]_{R-r}^{R-m}\\
&= \frac{4\alpha}{\pi(2\alpha-1)}(1+O(e^{-\alpha R}))\left(e^{\frac{m-r}{2}-\alpha m}-e^{-\alpha r}\right)\\
&= \frac{4\alpha}{\pi(2\alpha-1)}e^{\frac{m-r}{2}-\alpha m}(1+O(e^{-\alpha R}-e^{(m-r)(\alpha-\frac{1}{2})}))\\
&= \frac{4\alpha}{\pi(2\alpha-1)}e^{\frac{m-r}{2}-\alpha m}(1+O(e^{(m-r)(\alpha-\frac{1}{2})})),
\end{aligned}$$

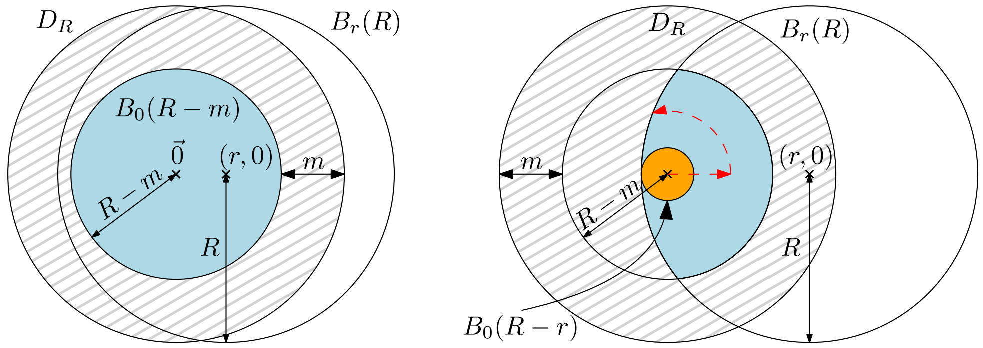
since again the dominating error term is $e^{(m-r)(\alpha-\frac{1}{2})} > e^{-R(\alpha-\frac{1}{2})} > e^{-\alpha R}$.

It is left to bound the error term in Equation (3.8). To this end, we compute

$$\begin{aligned}
&\int_{R-r}^{R-m} O(e^{\frac{3}{2}(R-r-y)+\alpha y-\alpha R})\,\mathrm{d}y\\
&= e^{\frac{m-r}{2}-\alpha m}\cdot\begin{cases} O(e^{m-r}) \leqslant O(e^{(m-r)(\alpha-\frac{1}{2})}), & \text{if } \alpha > \frac{3}{2},\\ O(e^{m-r}(r-m)), & \text{if } \alpha = \frac{3}{2},\\ O(e^{(m-r)(\alpha-\frac{1}{2})}), & \text{if } \alpha < \frac{3}{2}.\end{cases}
\end{aligned}$$

Plugging everything together, we obtain the solution as stated in Equation (3.7). □

Let us also restate a useful result from [BFM14]. Consider two vertices u, v in the



(a) First case in Equation (3.7). Here, the disk $B_r(R)$ fully encloses $B_0(R-m)$, so their intersection is determined by the smaller disk $B_0(R-m)$.

(b) Second case in Equation (3.7). The two disks $B_r(R)$ and $B_0(R-m)$ intersect. The orange circle $B_0(R-r)$ is fully enclosed in the intersection. For the remainder, we integrate over the area as indicated by the red arrows.

Figure 3.2: Proof illustration for Lemma 3.3. The gray area denotes the disc D_R. Note that these illustrations should only be understood schematically, as the behavior of circles in the hyperbolic plane is different to the classical euclidean geometry.

hyperbolic random graph. Moving one vertex closer to 0—i. e., decreasing r_u—typically does not result in a monotone behavior of dist(u, v). In particular, u can first move closer to v; and then farther away again. However, if u, v had distance at most x to each other and to the origin 0, this fact remains true even when u is moved closer to the center. In this sense, a node's neighborhood is monotone in its radial coordinate: $B_u(R) \cap D_R \subset B_{u'}(R) \cap D_R$ for $u' < u$. The next lemma formalizes this intuition.

Lemma 3.4 ([BFM14])**.** *Consider two nodes $u = (r_u, \varphi_u), v = (r_v, \varphi_v)$ in the hyperbolic random graph. If* dist$(u, v) \leqslant x$ *and* $r_u, r_v \leqslant x$*, then it holds*

$$\operatorname{dist}(u', v') \leqslant x,$$

where $u' = (r'_u, \varphi_u), v' = (r'_v, \varphi_v)$ with $r'_u \leqslant r_u$ and $r'_v \leqslant r_v$.

Proof. We prove the case for $r'_u < r_u$, $r'_v = r_v$. The statement follows by another application of the same proof.

Consider $B_x(v)$. By assumption, it contains both 0 and u. Since circles in the hyperbolic plane are convex, the geodesic between 0 and u is also fully contained in $B_x(v)$. In the native representation of the hyperbolic plane, this geodesic consists of all points (r, φ_u) with $0 \leqslant r \leqslant r_u$. The Lemma thus follows. □

4 Cliques in Hyperbolic Random Graphs

This chapter is based on joint work with Tobias Friedrich [FK15a]. It contains an in-depth analysis of algorithms for finding cliques; and several minor fixes over the conference version.

4.1 Introduction

In this chapter, we study the emergence of cliques in hyperbolic random graphs. Cliques were introduced as fully connected subgraphs in the field of psychology by Luce and Perry [LP49]. Originally, they served as a concept to study communities in social networks. Due to the simplicity of this structure, however, they have quickly been adopted and well-studied in mathematics and computer science.

Finding a (large enough) clique in a given graph is a notoriously difficult problem. It belongs to one of the most prominent examples for NP-hard and $W[1]$-hard problems, and efficient algorithms are therefore unlikely to exist. This presents an interesting contrast: Even though the structure is very intuitive, it remains elusive to be handled algorithmically.

In spite of their algorithmic complexity, they play a tremendous role in network science. Due to their inherent interconnectedness, they are closely related to clustering and community structures. Most work on extraction of communities from a large given network usually begins with a definition of communities that is strongly reminiscent of cliques—yet still different enough to offer an efficient algorithm [GKT05; Che+11; Tso+13]. In bioinformatics, many problems like inferring evolutionary trees or protein structure prediction are often modeled as finding maximum cliques in a biological network [DS86; SM98].

In the field of random graphs, nearly all research focuses on the uniform $G(n, p)$ model by Erdős and Rényi [JŁR11], see e. g. [Bol98; Ros08; Ros10a; Ros10b]. This model, however, shares only few properties with real-world networks, such as those arising from social interactions. With the recent proliferation of generative social network models, it is thus only a natural consequence to study cliques in the context in which they originally appeared. To this end, Janson, Łuczak, and Norros [JŁN10] and Bianconi and Marsili [BM06] provide asymptotic bounds on the size of the cliques in inhomogeneous random graphs [Hof16]; and methods to approximate them efficiently. Similarly, Friedrich and Krohmer [FK15c] investigate the computational complexity of finding the largest clique in these networks.

		Hyperbolic Random Graphs	
		$\frac{1}{2} < \alpha < 1$	$\alpha \geqslant 1$
$\mathbb{E}\left[K_k\right]$	$\leqslant$	$\frac{n^{(1-\alpha)k}}{k^k \exp(k(\alpha\frac{C}{2}-1))}\left(\frac{\alpha k c_1^{k-1}}{(1-\alpha)k+1}+1\right)$	$nk^{-k} \cdot \frac{\alpha k e(c_1 e^{-C/2+1})^{k-1}}{(\alpha-1)k+1} \cdot (1+o(1))$
	$\geqslant$	$\left(\frac{e^{-\alpha C/2} n^{1-\alpha}(1-o(1))}{k}\right)^k$	$nk^{-k}\left(\frac{e^{-C/2}}{\pi}\right)^{k-1} \cdot (1+o(1))$
	$=$	$n^{(1-\alpha)k}\Theta(k)^{-k}$	$n \cdot \Theta(k)^{-k}$
$\omega(G)$	$\leqslant$	$c_1 e^{-\alpha\frac{C}{2}+1} n^{1-\alpha}(1+o(1))$	$\frac{\log n}{\log\log n}(1+o(1))$
	$\geqslant$	$e^{-\alpha\frac{C}{2}} n^{1-\alpha}(1-o(1))$	$\frac{\log n}{\log\log n}(1-o(1))$
	$=$	$\Theta(n^{1-\alpha})$	$\frac{\log n}{\log\log n}(1 \pm o(1))$

Table 4.1: New results on the expected number of k-cliques $\mathbb{E}\left[K_k\right]$ and the size of the largest clique $\omega(G)$ in hyperbolic random graphs drawn from the step model. Sections 4.3 and 4.4 prove the upper and lower bounds on $\mathbb{E}\left[K_k\right]$. Section 4.5 proves the bounds on $\omega(G)$.

4.2 Our Contribution

In this chapter, we extend this work to hyperbolic random graphs sampled from the step model. In particular, we present bounds on the expected number of k-cliques and the size of the largest clique. The results are summarized in Table 4.1. We observe a phase transition at power-law exponent $\beta = 2\alpha + 1 = 3$, with smaller exponents yielding polynomial-size cliques and larger exponents yielding logarithmic-size cliques. While clique is NP- and W[1]-complete for general graphs, we show that the largest clique of hyperbolic random graphs (in the step model) can be found in polynomial time, if the geometry is known. This stands in contrast to previous results on similar models like Chung-Lu [FK12], which need exponential time for a power-law of $2 < \beta < 3$.

Comparison with other scale-free models. Using the results of Janson, Łuczak, and Norros [JŁN10], we compare the *clique numbers*, i.e., the size of the largest clique, of some popular scale-free network models to hyperbolic random graphs in Table 4.2.

We notice that the (asymptotic) clique number is nearly the same for Chung-Lu [ACL00], Norros-Reittu [NR06] and hyperbolic random graphs in the case where the power law exponent is $2 < \beta < 3$. An intuitive explanation for this phenomenon is that all these models have a tightly connected *core*: A subgraph of polynomial size in which the edge probability is $1 - o(1)$ or even 1. Large cliques emerge as a consequence of this core.

But even when such a core does not exist in the graph (which is the case for $\beta \geqslant 3$), one would expect to have small communities and therefore cliques in the graph. In particular, due to the large clustering coefficient it is likely that a node's neighbors

	Power-Law Exponent		
Random Graph Model	$2 < \beta < 3$	$\beta = 3$	$\beta > 3$
Hyperbolic (new results)	$\Theta(n^{(3-\beta)/2})$	$\Theta\left(\frac{\log n}{\log\log n}\right)$	$\Theta\left(\frac{\log n}{\log\log n}\right)$
Chung-Lu	$\Theta(n^{(3-\beta)/2})$	$\Theta(1)$	3
Norros-Reittu	$\Theta(n^{(3-\beta)/2}\log^{-(\beta-1)/2} n)$	$\Theta(1)$	3
Generalized RG	$\Omega(n^{\frac{3-\beta}{1+\beta}}), O(n^{\frac{3-\beta}{1+\beta}}\log^{\frac{\beta-1}{\beta+1}} n)$	$\Theta(1)$	3
Pref. Attachment	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$

Table 4.2: Comparison of our new results on the clique number $\omega(G)$ of hyperbolic random graphs to known results by Janson, Łuczak, and Norros [JŁN10] for other scale-free random graph models. All bounds hold with high probability.

(or a subset of the neighbors) form a clique. Consequently, the hyperbolic random graph has in this case a largest clique of size $\Theta(\frac{\log n}{\log\log n})$. Previous scale-free network models with independent edge probabilities predict in this case a largest clique of size $\leqslant 3$ almost asymptotically surely (i.e., with probability $\geqslant 1 - o(1)$), which seems counter-intuitive.

Organization. The remainder of this chapter is organized as follows. Section 4.3 shows the upper bounds on $\mathbb{E}[K_k]$ in Table 4.1, while Section 4.4 shows the lower bounds on $\mathbb{E}[K_k]$. By applying these results, we obtain in Section 4.5 bounds on the size of the largest clique $\omega(G)$ in the hyperbolic random graph. Finally, we show in Section 4.6, given the geometric representation, how to retrieve the largest clique in the graph in worst-case polynomial time. Section 4.7 concludes.

4.3 Proof of the Upper Bound

The goal of this section is to show the upper bounds for $\mathbb{E}[K_k]$ stated in Table 4.1. The following theorem summarizes these results in their asymptotic form.

Theorem 4.1. *In a hyperbolic random graph, the expected number of k-cliques is at most $n^{(1-\alpha)k}\Theta(k)^{-k}$ and $n \cdot \Theta(k)^{-k}$ if $\frac{1}{2} < \alpha < 1$ and $\alpha \geqslant 1$, respectively.*

In a clique, each pair of nodes is connected. To compute an upper bound on the probability that k nodes form a clique, we examine a relaxed condition; namely that all nodes connect to one specific node v.

For a set U of k independently sampled points, let $v \in U$ be the node with the largest radial coordinate, i. e. $r_v = \max_{u \in U}\{r_u\}$. We begin by computing the probability density function of r_v which we call $\rho_v(r)$. By the definition of the cumulative distribution function, we have

$$\Pr[r_v \leqslant x] = \Pr[\forall u \in U\colon r_u \leqslant x] = \left(\int_0^x \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1}\,\mathrm{d}r\right)^k = \left(\frac{\cosh(\alpha x) - 1}{\cosh(\alpha R) - 1}\right)^k.$$

The resulting probability density function is given by

$$\begin{aligned}
\rho_v(r) &= \frac{\partial}{\partial r}\left(\frac{\cosh(\alpha r) - 1}{\cosh(\alpha R) - 1}\right)^k \\
&= \alpha k \sinh(\alpha r)\frac{(\cosh(\alpha r) - 1)^{k-1}}{(\cosh(\alpha R) - 1)^k} \\
&= \alpha k e^{\alpha k(r-R)}(1 - e^{-2\alpha r}) \cdot \frac{(1 + e^{-2\alpha r} - 2e^{-\alpha r})^{k-1}}{(1 + e^{-2\alpha R} - 2e^{-\alpha R})^k} \\
&= \alpha k e^{\alpha k(r-R)}(1 - O(e^{-\alpha r}))^k \\
&\leqslant \alpha k e^{\alpha k(r-R)},
\end{aligned}$$

where we used Lemma 2.3 for bounding the error term. Following the explanation above, the probability that a set U of k independently sampled nodes forms a clique is upper bounded by the probability that all nodes are connected to v. Formally,

$$\begin{aligned}
&\Pr[U \text{ is clique}] \leqslant \Pr[\forall u \in U\colon\ \operatorname{dist}(u, v) \leqslant R] \\
&= \int_0^R \rho_v(r) \cdot \Pr[\forall u \in U\colon u \in B_r(R) \mid r_v = r] \\
&= \int_0^R \rho_v(r) \cdot \left(\frac{\mu(B_r(R) \cap B_0(r))}{\mu(B_0(r))}\right)^{k-1} \mathrm{d}r
\end{aligned}$$

For the last equality, observe that we condition on the fact that the largest radial coordinate among the nodes in U is r, i. e., all other radial coordinates are $\leqslant r$. Hence, the probability that a node u is connected to v is the probability that $u \in B_r(R) \cap B_0(R)$, conditioned on the fact that $r_u \leqslant r$, i.e. $u \in B_0(r)$.

We split the integral in two parts. If $r < R/2$, then by triangle inequality it follows that all k nodes are connected. This agrees with Lemma 3.3, and we obtain

$$\begin{aligned}
&\int_0^{R/2} \rho_v(r) \cdot \left(\frac{\mu(B_r(R) \cap B_0(r))}{\mu(B_0(r))}\right)^{k-1} \mathrm{d}r \\
&= \int_0^{R/2} \rho_v(r)\,\mathrm{d}r \leqslant \left(\frac{\cosh(\alpha R/2) - 1}{\cosh(\alpha R) - 1}\right)^k \leqslant e^{-\alpha k \frac{R}{2}}. \qquad (4.1)
\end{aligned}$$

When $r \geqslant \frac{R}{2}$, we estimate again using Lemma 3.3

$$\begin{aligned}\frac{\mu(B_r(R) \cap B_0(r))}{\mu(B_0(r))} &= \tfrac{4\alpha}{\pi(2\alpha-1)} e^{\frac{R}{2}-r-\alpha(R-r)+\alpha(R-r)} \cdot \mathcal{E} \\ &= \tfrac{4\alpha}{\pi(2\alpha-1)} \cdot e^{\frac{R}{2}-r} \cdot \mathcal{E},\end{aligned}$$

where $\mathcal{E} = (1 \pm O(e^{(R-2r)(\alpha-\frac{1}{2})} + e^{-\alpha r}))$, if $\alpha \neq \frac{3}{2}$ and $\mathcal{E} = (1 \pm O(e^{(R-2r)}(2r-R) + e^{-\alpha r}))$ otherwise. Observe that in both cases, since $r \geqslant \frac{R}{2}$, the error term is upper bounded by a constant. Thus, we write

$$\frac{\mu(B_r(R) \cap B_0(r))}{\mu(B_0(r))} \leqslant c_1 e^{\frac{R}{2}-r} \tag{4.2}$$

for some large enough constant $c_1 > 1$. Then, we compute for the second part of the integration

$$\begin{aligned}&\int_{R/2}^{R} \rho_v(r) \cdot \left(\frac{\mu(B_r(R) \cap B_0(r))}{\mu(B_0(r))}\right)^{k-1} \mathrm{d}r \\ \leqslant &\int_{R/2}^{R} \alpha k e^{\alpha k(r-R)} \left(c_1 e^{R/2-r}\right)^{k-1} \mathrm{d}r && (4.3)\\ = &\frac{\alpha k c_1^{k-1}}{(\alpha-1)k+1} \left[e^{\alpha k(r-R)+(k-1)(\frac{R}{2}-r)}\right]_{R/2}^{R} && (4.4)\\ = &\frac{\alpha k c_1^{k-1}}{(\alpha-1)k+1} \left(e^{-\frac{R}{2}(k-1)} - e^{-\frac{R}{2}\alpha k}\right), && (4.5)\end{aligned}$$

where Equations (4.4) and (4.5) hold if $\alpha \neq 1$ and $k \neq 1/(1-\alpha)$. In the following, we consider all possible combinations of α and k. Whenever possible, we continue computing with Equation (4.5), otherwise we use Equation (4.3). We distinguish the following cases:

(a) $\alpha = 1$. In this case, Equation (4.3) evaluates to

$$(4.3) = kc_1^{k-1}[e^{-\frac{R}{2}(k+1)+r}]_{R/2}^{R} \leqslant kc_1^{k-1}e^{-\frac{R}{2}(k-1)}.$$

(b) $\alpha > 1$. Then, $0 > \frac{1}{1-\alpha} \neq k$ and thus we may use Equation (4.5):

$$(4.5) \leqslant \frac{\alpha k c_1^{k-1}}{(\alpha-1)k+1} e^{-\frac{R}{2}(k-1)}.$$

(c) $\frac{1}{2} \leqslant \alpha < 1$. In this case, the sign in front of the antiderivative depends on k:

(c.i) $k < \frac{1}{1-\alpha}$. In that case, $(\alpha-1)k > -1$, and Equation (4.5) is again upper bounded by

$$(4.5) \leqslant \frac{\alpha k c_1^{k-1}}{(\alpha-1)k+1} e^{-\frac{R}{2}(k-1)}.$$

(c.ii) $k = \frac{1}{1-\alpha}$. Then, we substitute $\alpha = \frac{k-1}{k}$ in Equation (4.3):

$$(4.3) = \int_{R/2}^{R} \alpha k c_1^{k-1} e^{-\frac{R}{2}(k-1)} \,\mathrm{dr} = \alpha k c_1^{k-1} \frac{R}{2} e^{-\frac{R}{2}(k-1)}$$

(c.iii) $k > \frac{1}{1-\alpha}$. Here, the sign of the antiderivative is negative, and we obtain

$$(4.5) \leqslant \frac{\alpha k c_1^{k-1}}{(1-\alpha)k+1} e^{-\alpha k \frac{R}{2}}.$$

Recall that we split the integral into two parts and thus have to add $e^{-\alpha k \frac{R}{2}}$ to the result, c. f. Equation (4.1). Cases (a)–(c.ii) only change by a factor of $(1+o(1))$, and in the case of (c.iii) we obtain that $\Pr[U \text{ is a clique}] \leqslant (1 + \frac{\alpha k c_1^{k-1}}{(1-\alpha)k+1})e^{-\alpha k R/2}$. When $\alpha > 1$ (i.e. when the graph has a power law exponent $\beta > 3$), the number of cliques is therefore bounded by

$$\begin{aligned}
\mathbb{E}\left[K_k\right] &= \binom{n}{k} \Pr[U \text{ is clique}] \\
&\leqslant \left(\frac{ne}{k}\right)^k \frac{\alpha k c_1^{k-1}}{(\alpha-1)k+1} e^{-\frac{R}{2}(k-1)}(1+o(1)) \\
&= nk^{-k} \cdot \frac{\alpha k e (c_1 e^{-C/2+1})^{k-1}}{(\alpha-1)k+1}(1+o(1)) \\
&= n \cdot \Theta(k)^{-k},
\end{aligned}$$

since $n = e^{\frac{R-C}{2}}$. For $\alpha = 1$ we obtain a similar bound $\mathbb{E}\left[K_k\right] \leqslant n \cdot \Theta(k)^{-k+1} = n \cdot \Theta(k)^{-k}$.

For networks with a dense core ($\frac{1}{2} \leqslant \alpha < 1$), we obtain

$$\begin{aligned}
\mathbb{E}\left[K_k\right] &\leqslant \left(\frac{ne}{k}\right)^k \left(1 + \frac{\alpha k c_1^{k-1}}{(1-\alpha)k+1}\right) e^{-\alpha k \frac{R}{2}} \\
&= n^{(1-\alpha)k} k^{-k} \left(1 + \frac{\alpha k c_1^{k-1}}{(1-\alpha)k+1}\right) e^{(1-\alpha\frac{C}{2})k} \\
&= n^{(1-\alpha)k} \Theta(k)^{-k},
\end{aligned}$$

if $k > \frac{1}{1-\alpha}$. Table 4.1 contains the detailed results for these cases. In the case where

$k \leqslant 1/(1-\alpha)$, which is not shown in the table, our result states that there is at most a linear number of k-cliques. This agrees, for instance, with the known fact that for $k = 2 \leqslant \frac{1}{1-\alpha}$ there are $\Theta(n)$ many edges in G.

4.4 Proof of the Lower Bound

In this section, we show the lower bounds for $\mathbb{E}[K_k]$ stated in Table 4.1, which asymptotically match the upper bounds we proved in the previous section.

Theorem 4.2. *In a hyperbolic random graph, the expected number of k-cliques is at least $n^{(1-\alpha)k}\Theta(k)^{-k}$ and $n \cdot \Theta(k)^{-k}$ if $\frac{1}{2} < \alpha < 1$ and $\alpha \geqslant 1$, respectively.*

To obtain these matching lower bounds, we consider two cases. In the case when $\frac{1}{2} < \alpha < 1$, hyperbolic random graphs are known to exhibit a tightly connected core. We show that, in fact, the high-degree nodes form a clique of polynomial size. The number of k-cliques in G is then simply dominated by the number of distinct k-subsets of nodes in the core.

To be more precise, consider the ball $B_0(R/2)$. All nodes in this area have distance at most R from each other by the triangle inequality. It is therefore left to bound the number of nodes in $B_0(R/2)$. By Lemma 3.3 we know that

$$\mu(B_0(x)) = e^{-\alpha(R-x)}(1 - O(e^{-\alpha x})),$$

i.e. the probability that a sampled point has at most distance x from the center of D_R is $e^{-\alpha(R-x)}(1 + o(1))$. Consequently, we expect $ne^{-\alpha R/2}(1 - o(1))$ nodes in $B_0(R/2)$. Observe that for $\frac{1}{2} < \alpha < 1$ and $R = 2\ln n + C$ this amounts to $e^{-\alpha C/2}n^{1-\alpha}(1 - o(1))$, which is polynomial. In Section 4.5, we will also see that this number is close to the size of the maximum clique.

Let $K_k(G)$ be the number of k-cliques in G. Clearly, if $G' \subseteq G$, then we have that $K_k(G') \leqslant K_k(G)$. Consider for G the hyperbolic random graph and for G' the graph induced on G by only taking vertices v with $r_v \leqslant R/2$. Then, we get

$$\mathbb{E}[K_k] = \mathbb{E}[K_k(G)] \geqslant \mathbb{E}[K_k(G')] = \mathbb{E}\left[\binom{X}{k}\right],$$

where X is the random variable describing the number of nodes that drop in $B_0(R/2)$. To show the lower bound, we use the following well-known lemma, which can e.g. be found in [WY05, Ex. 1].

Lemma 4.1. *The function $f(x) = \binom{x}{k}$ is convex on $x \geqslant k$.*

Therefore, using Jensen's inequality [Jen06] which says $f(\mathbb{E}[X]) \leqslant \mathbb{E}[f(X)]$ for convex functions f, we obtain

$$\begin{aligned}\mathbb{E}\left[\binom{X}{k}\right] &\geqslant \binom{\mathbb{E}[X]}{k} = \binom{e^{-\alpha C/2}n^{1-\alpha}(1-O(e^{-\alpha R/2}))}{k} \\ &\geqslant \left(\frac{e^{-\alpha C/2}n^{1-\alpha}(1-o(1))}{k}\right)^k .\end{aligned}$$

Thus, we have that $\mathbb{E}[K_k] \geqslant n^{(1-\alpha)k} \cdot \Theta(k)^{-k}$, which proves the lower bound for the dense case.

4.4.1 Small Cliques Outside of the Core

So far, we have seen that for $\frac{1}{2} < \alpha < 1$, hyperbolic random graphs contain many cliques in the core. When $\alpha \geqslant 1$, however, the number of nodes in $B_0(R/2)$ is of order $O(1)$. We now show that due to the underlying geometry, cliques still emerge outside of the core.

To this end, we investigate a circular sector of the disk D_R with angle $\theta = a/n$, for some constant a which we choose later. Clearly, there are $\frac{2\pi n}{a}$ non-overlapping sectors. As we show in the following, such a circular sector has a (geometric) diameter of $\leqslant R$, if a is chosen as an appropriate constant. This means that all points in the sector have pairwise distance at most R and therefore form a complete subgraph.

Since the angular coordinates of nodes are sampled uniformly, the probability that we sample a node inside one specific circular sector of angle a/n is exactly $\frac{a}{2\pi n}$. Therefore, the probability that a set of k independently sampled points U is contained in one sector is

$$\Pr[U \text{ is clique}] \geqslant \frac{2\pi n}{a} \cdot \left(\frac{a}{2\pi n}\right)^k = \left(\frac{a}{2\pi n}\right)^{k-1} .$$

This probability is maximized by choosing a as large as possible, i.e. such that for any larger a' the diameter is $> R$. It remains to derive a suitable value for a.

Lemma 4.2. *Let S be a circular sector of D_R of angle $\frac{a}{n} = 2e^{-C/2}(1 - O(n^{-2}))\frac{1}{n}$. Then, S has a (geometric) diameter of at most R.*

Proof. Let u, v be two points inside S with maximal distance. Observe that these points have to lie on the boundary of S: Otherwise, consider the geodesic that goes through u, v and intersects S at u', v'. Clearly, $\operatorname{dist}(u', v') > \operatorname{dist}(u, v)$, a contradiction. Observe further that

$$\cosh(\operatorname{dist}(u, v)) := \cosh(r_u)\cosh(r_v) - \sinh(r_u)\sinh(r_v)\cos(\Delta\varphi)$$

is monotonously increasing for $0 \leqslant \Delta\varphi \leqslant \pi$. Since S has an angle of $\frac{a}{n} \ll \pi$, we thus may assume that u, v have a relative angle of $\Delta\varphi = \frac{a}{n}$.

We now show that if $r_u = r_v = R$, $\operatorname{dist}(u, v) \leqslant R$. By Lemma 3.4 it follows that all other pairs of points with smaller radial coordinates also have distance at most R.

By Lemma 3.2, the maximum angle between u, v such that their distance is at most R, is

$$\theta(R, R) = 2e^{-\frac{R}{2}}(1 \pm O(e^{-R})) = 2e^{-\frac{R-C}{2}-\frac{C}{2}}(1 \pm O(e^{-R})) = 2e^{-C/2}\frac{1}{n}(1 \pm O(n^{-2})).$$

Thus, we set $a = 2e^{-C/2}(1 \pm O(n^{-2}))$. □

Finally, the probability that a set U of k nodes is a clique is

$$\begin{aligned}\Pr[U \text{ is clique}] &\geqslant \left(\frac{a}{2\pi n}\right)^{k-1}\\ &\geqslant \left(\frac{e^{-C/2}}{\pi n}(1 - O(n^{-2}))\right)^{k-1}\\ &= \left(\frac{e^{-C/2}}{\pi n}\right)^{k-1}(1 \pm O(n^{-1})),\end{aligned}$$

since $(1 - O(n^{-2}))^{k-1} \geqslant (1 - O(n^{-2}))^n = (1 - O(n^{-1}))$ by Lemmas 2.1 and 2.2. For the expected number of k-cliques, this implies that

$$\begin{aligned}\mathbb{E}[K_k] &= \binom{n}{k}\Pr[U \text{ is } k\text{-clique}]\\ &\geqslant \left(\frac{n}{k}\right)^k\left(\frac{e^{-C/2}}{\pi n}\right)^{k-1}(1 \pm O(n^{-1}))\\ &= n \cdot \Theta(k)^{-k}.\end{aligned}$$

Taken together with the result from Section 4.4, we conclude

$$\mathbb{E}[K_k] \geqslant \max\{n, n^{(1-\alpha)k}\} \cdot \Theta(k)^{-k}.$$

4.5 Largest Clique

In this section, we present the bounds on the clique number $\omega(G)$, i.e., the size of the largest clique in G, as stated in Table 4.1. The asymptotic bounds are summarized in the following theorem.

Theorem 4.3. *The clique number of a hyperbolic random graph is with high probability $\Theta(n^{1-\alpha})$ if $\frac{1}{2} < \alpha < 1$, and asymptotically almost surely $\frac{\log n}{\log\log n}(1 \pm o(1))$ if $\alpha \geqslant 1$.*

We use the upper bounds on the number of k-cliques from Theorem 4.1 to obtain upper bounds for $\omega(G)$ by applying the Markov inequality

$$\Pr[K_k > 1] \leqslant \mathbb{E}\left[K_k\right].$$

Let therefore ε be an arbitrarily small constant and solve $\mathbb{E}\left[K_k\right] \leqslant n^{-\varepsilon}$ for k. If ε is constant, we obtain an upper bound on the clique number that holds with high probability.

4.5.1 Dense Core

Let us first consider the case when $\frac{1}{2} < \alpha < 1$ and there exists a dense core in the center of D_R. Due to Theorem 4.1, there exists some constant c such that

$$\Pr[K_k > 1] \leqslant \mathbb{E}\left[K_k\right] \leqslant n^{(1-\alpha)k} \cdot (ck)^{-k}.$$

We set $k = \frac{2}{c} n^{1-\alpha}$ to obtain

$$\begin{aligned}\mathbb{E}\left[K_k\right] &\leqslant n^{(1-\alpha)k} \cdot (ck)^{-k} \\ &= n^{(1-\alpha)\frac{2}{c}n^{1-\alpha}} \cdot (2n^{1-\alpha})^{-\frac{2}{c}n^{1-\alpha}} \\ &= 2^{-\frac{2}{c}n^{1-\alpha}}.\end{aligned}$$

This term is asymptotically smaller than $n^{-\varepsilon}$ for any constant ε, since

$$2^{-\frac{2}{c}n^{1-\alpha}} \leqslant n^{-\varepsilon} \Leftrightarrow \frac{2}{c}n^{1-\alpha} \geqslant \varepsilon \log_2 n$$

for large enough n. Therefore, we know that $\omega(G) \leqslant \Theta(n^{1-\alpha})$ in this case. The precise leading constant for this approach depends on c_1, see Equation (4.2). Since $c_1 > 1$, we have for $k = \omega(1)$

$$\mathbb{E}\left[K_k\right] \leqslant n^{(1-\alpha)k} (\tfrac{1}{c_1} e^{\alpha \frac{C}{2} - 1} (1 + o(1)))^{-k}.$$

By a similar approach as above, we can compute that

$$\omega(G) \leqslant c_1 e^{-\alpha \frac{C}{2} + 1} n^{1-\alpha} (1 + o(1)) = O(n^{1-\alpha})$$

holds with high probability.

To compute a matching lower bound, recall that Section 4.4 states that $B_0(R/2)$ contains $e^{-\alpha C/2} n^{1-\alpha}(1 - o(1))$ nodes in expectation. Let X be the number of nodes in $B_0(R/2)$. Since each node is sampled independently from all others, we may apply

Theorem 2.2 to obtain that

$$\Pr[X \leqslant (1 - \tfrac{1}{\log n})\mathbb{E}\,[X]] \leqslant \exp(-\Theta(1) \cdot \log^{-2} n e^{-\alpha C/2} n^{1-\alpha}).$$

As this tail probability decreases faster than any polynomial, we have that with high probability, the largest clique is of size

$$\omega(G) \geqslant e^{-\alpha C/2} n^{1-\alpha}(1 - o(1)) = \Omega(n^{1-\alpha}).$$

4.5.2 Sparse Core

For $\alpha \geqslant 1$, when a dense core is not present, we have proven that $\mathbb{E}\,[K_k] = n \cdot \Theta(k)^{-k}$. Thus, there exists a constant c such that $\mathbb{E}\,[K_k] \leqslant n \cdot (ck)^{-k}$. Again, we apply a Markov bound to upper bound the probability a large clique occurs. Thus, we need to choose k such that

$$\Pr[K_k > 1] \leqslant \mathbb{E}\,[K_k] \leqslant n \cdot (ck)^{-k} \overset{!}{\leqslant} n^{-\varepsilon},$$

which is equivalent to $(ck)^{-k} \leqslant n^{-1-\varepsilon}$. By taking $k := (1+\varepsilon)\frac{\log n}{\log\log n}$, we obtain for large enough n

$$\begin{aligned}
(ck)^{-k} &= \left((1+\varepsilon)\frac{c\log n}{\log\log n}\right)^{-(1+\varepsilon)\frac{\log n}{\log\log n}} \overset{!}{\leqslant} n^{-1-\varepsilon} \\
&\Leftarrow \log\left((1+\varepsilon)\frac{c\log n}{\log\log n}\right) \cdot \left((-1-\varepsilon)\frac{\log n}{\log\log n}\right) \leqslant (-1-\varepsilon)\log n \\
&\Leftarrow \log\log n \cdot (1 - o(1)) \cdot \left((-1-\varepsilon)\frac{\log n}{\log\log n}\right) \leqslant (-1-\varepsilon)\log n \\
&\Leftarrow (-1-\varepsilon)\log n \leqslant (-1-\varepsilon)\log n.
\end{aligned}$$

Therefore, there is no larger clique than $(1+\varepsilon)\frac{\log n}{\log\log n}$ with probability $1 - n^{-\varepsilon}$. Setting $\varepsilon > 0$ to any constant yields a result with high probability. We may, however, obtain an even tighter result by choosing e. g. $\varepsilon = \frac{1}{\log\log n}$. Then, $n^{-\varepsilon} = o(1)$ and therefore the largest clique is of size at most $\frac{\log n}{\log\log n}(1 + o(1))$ almost asymptotically surely, i. e., with probability $1 - o(1)$.

To obtain a matching lower bound, observe that the analysis in Section 4.4 corresponds to a balls-into-bins experiment: There are $\frac{2\pi n}{a}$ circular sectors (bins), and each node (ball) is uniformly sampled in one of those. Since there are n balls and $\Theta(n)$ bins, an application of [RS98, Theorem 1] yields the desired result. For reasons of completeness, we restate the relevant part of the theorem:

Theorem 4.4 ([RS98]). *Let M be the random variable that counts the maximum number of balls in any bin, if we throw n balls independently and uniformly at random into $m = \Theta(n)$ bins. Then,*

$$\Pr\left[M \geqslant \frac{\log m}{\log \frac{m \log m}{n}}\left(1 + 0.99 \cdot \frac{\log \log \frac{m \log m}{n}}{\log \frac{m \log m}{n}}\right)\right] = 1 - o(1).$$

Observe that since $m = \Theta(n)$, we have $\log m = (1 \pm o(1)) \log n$. Furthermore,

$$\log \frac{m \log m}{n} = \log(\Theta(\log m)) = (1 \pm o(1)) \log \log n.$$

Plugging this into the theorem, we obtain that with probability $1 - o(1)$, there is a clique of size at least

$$\frac{\log n}{\log \log n}(1 \pm o(1)) \left(1 + 0.99 \frac{\log \log \log n}{\log \log n}(1 \pm o(1))\right) \geqslant \frac{\log n}{\log \log n}(1 - o(1)).$$

This proves the lower bound for the maximum clique in Table 4.1.

4.6 Algorithms for Finding Cliques

So far, we showed bounds on the size of cliques in hyperbolic random graphs, but did not yet investigate on how to find them algorithmically. For the case $\alpha \geqslant 1$ we showed that there are only few cliques in the graph; and therefore, a simple enumeration algorithm finds the largest clique in polynomial time. In fact, it is even possible to find the largest clique in linear time, as shown by the following Theorem.

Theorem 4.5. *The largest clique of a hyperbolic random graph with power-law exponent $\beta = 2\alpha + 1 \geqslant 3$ can be found in expected time $O(n)$.*

Proof. Let X be the number of neighbors of a node v with radial coordinate smaller than r_v. By Lemma 3.3, this amounts in expectation to

$$\begin{aligned}
\mathbb{E}[X] &= n \cdot \mu(B_0(r_v) \cap B_{r_v}(R)) \\
&= \Theta(1) \cdot \exp\left(\frac{R}{2} + \frac{R - r_v - r_v}{2} - \alpha(R - r_v)\right) \\
&= \Theta(1) \cdot \exp\left((\alpha - 1)r_v - (\alpha - 1)R\right) \\
&= O(1),
\end{aligned}$$

if $r_v \geqslant \frac{R}{2}$, and

$$\mathbb{E}[X] = \Theta(1) \cdot e^{\frac{R}{2} - \alpha(R - r_v)} \leqslant \Theta(1) \cdot e^{\frac{R}{2} - \alpha(R - \frac{R}{2})} = O(1)$$

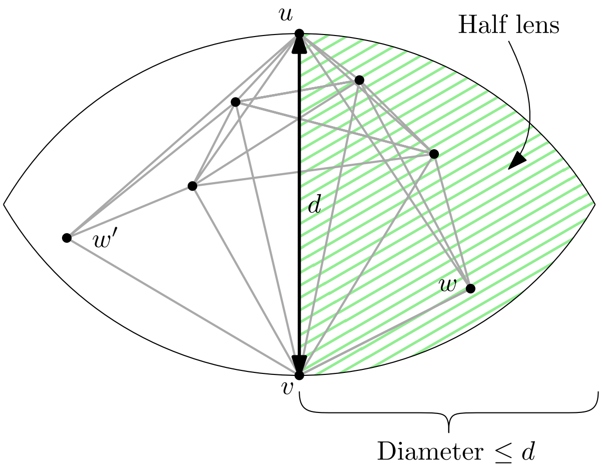


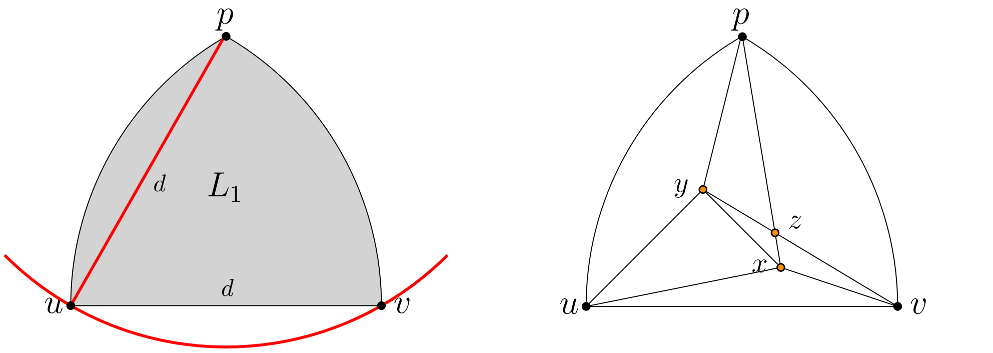
Figure 4.1: Illustration of the clique algorithm. Every two nodes u, v with distance $\operatorname{dist}(u, v) = d \leqslant R$ define a lens $B_u(d) \cap B_v(d)$. A half lens has geometric diameter $d \leqslant R$ and the nodes within thus form a clique. The union of the two half lenses is not necessarily a clique, since some nodes w, w' might have a distance $> d$.

otherwise. Thus, every node only has (in expectation) a constant number of neighbors with larger degree. Thus, the largest clique can be found by exhaustively searching all node neighborhoods as follows. In each step, pick the node v of smallest degree in the graph, and find the largest clique that v is a part of. Then, delete v and recurse. The technical analysis of this process is the same as in [FK12, Theorem 1], which reveals that the largest clique can be determined in $O(n)$ expected time. □

This algorithm is the same as in the Chung-Lu model with $\beta \geqslant 3$ [FK12]. In this model, however, no algorithm is known for finding the largest clique in polynomial time when $2 < \beta < 3$. In contrast to this, we now show that due to the underlying geometry in hyperbolic random graphs sampled from the step model, a polynomial runtime is also achievable for the case $2 \leqslant \beta < 3$, if the geometric representation of the graph is given. The proof is similar to [CCJ91, Section 3] and works roughly as follows.

Consider two connected nodes u, v with distance $\operatorname{dist}(u, v) = d \leqslant R$. We denote by $S^{u,v}$ the set of all nodes that have distance at most d to both nodes u, v. By definition, it holds $S^{u,v} \subset B_u(d) \cap B_v(d)$. Consider now the largest clique C in the graph, and let $x, y \in C$ be the two nodes with maximal distance in C. It is then easy to see that $C \subseteq S^{x,y}$. Thus, it suffices to find the largest clique in $S^{u,v}$ for all connected node pairs $u, v \in V$.

In the following, we prove that the graph induced by the nodes $S^{u,v}$ is complement to a bipartite graph. Finding the largest clique then boils down to finding the largest independent set in a bipartite graph, which is possible in polynomial time.



(a) The point p has distance at most d to all points in L_1. In particular, the arcs from p to u, v are fully contained in $B_p(d)$.

(b) Two arbitrary points $x, y \in L_1$ have distance at most d, which can be deduced from this construction and the triangle inequality.

Figure 4.2: Proof illustration for Lemma 4.3.

To show that $S^{u,v}$ is a complement of a bipartite graph, we partition the lens $B_u(d) \cap B_v(d)$ into two symmetrical areas (half lenses) and show that each half lens has geometric diameter at most d. Thus, both half lenses form complete subgraphs; while edges crossing the two half lenses may or may not be present. Figure 4.2 contains an illustration. Without loss of generality, we assume that $u = (0, 0)$ and $v = (d, 0)$. The statement generalizes to arbitrary positions by a simple coordinate transformation.

Lemma 4.3. *Consider a lens of the form $L = B_0(d) \cap B_d(d)$ in the hyperbolic plane. Then, the half lens $L_1 = \{(r, \varphi) \in L \mid 0 \leqslant \varphi < \pi\}$ has geometric diameter at most d.*

Proof. Let us denote with p the point where the two discs of radius d intersect. We first show that p has distance at most d to all points in the half lens. Figure 4.2a contains an illustration of this statement.

Consider a circle of radius d around p, i.e. $B_p(d)$. Since p has distance d to both u, v, they lie on this circle. Since circles in hyperbolic space are convex, the geodesic between u, v lies inside the circle as well. It remains to show that the two circular arcs from p to u, v also lie within $B_p(d)$. To this end, we use the basic fact that distinct circles in the hyperbolic plane meet at most twice. Due to symmetry, it suffices to show that the arc from p to u is contained in $B_p(d)$. Since u is on the boundary of $B_p(d)$, this leaves at most one more intersection. By symmetry of the lens, every such intersection in L_1 must also occur in $L_2 = \{(r, \varphi) \in L \mid \pi \leqslant \varphi < 2\pi\}$. Thus, the circular arc from p to u can not intersect $B_p(d)$, as otherwise there would be at least three intersections.

Therefore, we know that u, v and p have at most distance d to all nodes in the half lens L_1. Consider now two arbitrary points $x, y \in L_1$. We consider the three triangles obtained by using $\overline{xy}$ as base, and u, v or p as the third point, see Figure 4.2b. Since these three triangles use the same base, at least two of them intersect. W. l. o. g., we assume that $\overline{px}$ intersects $\overline{vy}$, the other cases are analog. We call the intersection point z. We observe now the following:

$$\overline{vy} = \overline{vz} + \overline{yz} \leqslant d \quad \text{since } v \text{ has distance at most } d \text{ to all points in } L_1\,, \tag{4.6}$$
$$\overline{px} = \overline{pz} + \overline{xz} \leqslant d \quad \text{since } p \text{ has distance at most } d \text{ to all points in } L_1, \tag{4.7}$$
$$\overline{vz} + \overline{pz} \geqslant d \quad \text{by triangle inequality, since } \overline{vp} = d. \tag{4.8}$$

Adding Equations (4.6) and (4.7) and subtracting Equation (4.8) yields

$$\overline{yz} + \overline{xz} \leqslant d.$$

Thus, by triangle inequality, we have $\overline{xy} \leqslant \overline{yz} + \overline{xz} \leqslant d$. □

Using this result, we may show that there exists a polynomial time algorithm for finding the maximum clique in a hyperbolic random graph drawn from the step model. Note that similarly as in the euclidean case, this result holds with probability 1, i. e. the proof is fully deterministic and does not use the distribution of nodes. Using Lemma 4.3, the proof is analogous to the euclidean case [CCJ91]. We reprove it here for completeness.

Theorem 4.6. *Let G be a graph sampled from the hyperbolic random graph in the step model. Given the geographic position of the nodes, the clique number $\omega(G)$ can be computed in worst-case $O(m \cdot n^{2.5})$ time.*

Proof. Let C be the largest clique in G. Then, there must exist two nodes $u, v \in C$ such that u, v have maximal geometric distance among all node pairs in C. Let $d := \text{dist}(u, v)$. Observe that $d \leqslant R$, as otherwise u, v are not connected.

Consider now the induced subgraph $G[S^{u,v}]$ on all nodes $S^{u,v}$ that lie within the lens $B_u(d) \cap B_v(d)$. This subgraph can be found using the geometric representation, and, as shown in Lemma 4.3, $S^{u,v}$ may be partitioned in two sets S_1, S_2, such that both sets form a clique. Finding the largest clique in $G[S^{u,v}]$ is then equivalent to finding the largest independent set in the complement graph $\overline{G}[S^{u,v}]$. Since S_1, S_2 both form a clique in $G[S^{u,v}]$, they are independent in $\overline{G}[S^{u,v}]$ and therefore, $\overline{G}[S^{u,v}]$ is a bipartite graph.

Finding a maximum independent set is again equivalent to finding a minimum vertex cover. By Kőnig's Theorem (see e. g. [Die12]), the size of the maximum matching in a bipartite graph is equal to the size of the minimum vertex cover. Thus, it suffices to compute the size k of a maximum matching in $\overline{G}[S^{u,v}]$ and return $|S^{u,v}| - k$. Using e. g. the Hopcroft-Karp algorithm [HK73], this may be done in time $O(|S^{u,v}|^{2.5})$.

Thus, an algorithm needs to simply check for each connected pair of nodes u, v for the largest clique in $S^{u,v}$, which takes time at most $O(m \cdot n^{2.5})$. □

4.7 Conclusion

We present an analysis of the emergence of cliques in the hyperbolic random graphs and suggest how to find them algorithmically. We found that the large clustering coefficient of these graphs strongly affects the clique number when $\beta > 3$. Previous models with independent edge probabilities predicted a clique number of 3 in this case, whereas the hyperbolic random graph contains a $\frac{\log n}{\log \log n}$ size clique.

Further, we show two algorithms for computing the largest clique in a hyperbolic random graph drawn form the step model. For graphs with power-law exponent $\beta \geqslant 3$, the largest clique can be found in expected linear time. On the other hand, if the node coordinates are known, the largest clique in any hyperbolic random graph may be found in time $O(n^{4.5})$ with probability 1. It is, however, an open problem to find the largest clique when given only the graph structure, but not the geometric locations of the nodes.

5 Diameter of Hyperbolic Random Graphs

This chapter is based on joint work with Tobias Friedrich [FK15b]. It contains improvements on several proofs and a corrected version of Theorem 5.2 over the published conference version.

5.1 Introduction

The *diameter* of a graph G is the longest shortest path between any two nodes in (the giant component of) G. It is a fundamental structural property of a random graph, as it sets a worst-case lower bound on the number of steps required for all communication processes. Assume for instance a simple broadcast protocol in which each activated node activates all neighbors. Starting with one active node, it takes $\Omega(D)$ iterations of this process to activate all nodes in the giant component, where D is the diameter.

In contrast to the average distance, the diameter is determined by a single—atypical—long path. Due to this sensitivity to small changes, it is notoriously hard to analyze. Even subtle changes to the graph model can make an exponential difference in the diameter, as can be seen when comparing Chung-Lu (CL) random graphs [CL02b] and Preferential Attachment (PA) graphs [BA99] in the range of the power law exponent $2 < \beta < 3$: On the one hand, it has been shown that a CL graph can be embedded in a PA graph and they behave effectively the same [FPS12]. On the other hand, the diameter of CL graphs is $\Theta(\log n)$ [CL02b] while for PA graphs it is $\Theta(\log \log n)$ [DHH10]. Table 5.1 provides an overview over existing results in other random graph models. It was open so far how the diameter of hyperbolic random graphs compares to the aforementioned bounds for other scale-free graph models. The only known upper bounds on their diameter are $O((\log n)^{\frac{32}{(3-\beta)(5-\beta)}+1})$ by Kiwi and Mitsche [KM15], and a polylogarithm with no explicit constant by Bringmann, Keusch, and Lengler [BKL15].

5.2 Our Contribution

We improve upon the previous results as follows. First, we present a much simpler proof which also shows a polylogarithmic upper bound for the diameter, but with a better (i.e. smaller) exponent.

Theorem 5.1. *Let $2 < \beta < 3$. The diameter of the giant component in the hyperbolic random graph is $O((\log n)^{\frac{2}{3-\beta}})$ with probability $1 - O(n^{-2})$.*

Random Graph Model	Diameter	
Sparse Erdős-Rényi [Bol98]	$\Theta(\log n)$ [RW10]	
d-dim. Euclidean [Pen03]	$\Theta(n^{1/d})$ [FSS13]	
Watts-Strogatz [WS98]	$\Theta(\log n)$ [BC88]	
Kleinberg [Kle00]	$\Theta(\log n)$ [MN04]	
Chung-Lu [CL02b]	$\Theta(\log n)$ [CL02b]	power-law graphs
Pref. Attachment [BA99]	$\Theta(\log \log n)$ [DHH10]	
Hyperbolic [Kri+10]	$O((\log n)^{\frac{32}{(3-\beta)(5-\beta)}+1})$ [KM15]	

Table 5.1: Known diameter bounds for various random graphs. In all cases the diameter depends on the choice of the model parameters. Here we consider a constant average degree. For scale-free networks, we also assume a power law exponent $2 < \beta < 3$.[1]

The proof of Theorem 5.1 is presented in Section 5.4. It serves as an introduction to the proof of a logarithmic upper bound for the diameter presented in Section 5.5. There we show with more advanced techniques that for large power-law exponents the following theorem holds.

Theorem 5.2. *Let $\beta > 3$. Then, the diameter of the giant component in the hyperbolic random graph is $O(\log n)$ with probability $1 - O(n^{-2})$.*

The logarithmic upper bound is best possible. In particular, we show that Theorem 5.2 is tight by presenting the following matching lower bound.

Theorem 5.3. *Let $\beta > 2$. Then, there exists a component in the hyperbolic random graph with diameter $\Omega(\log n)$ with probability $1 - O(n^{1-\frac{\beta}{2}})$. If $\beta < 3$, this is the giant component.*

Let us briefly discuss these results. First, even though we prove all diameter bounds on the giant component for the case $2 < \beta < 3$, our proofs will make apparent that the giant component is in fact the component with the largest diameter in the graph. Second, the statements in Theorems 5.1 and 5.2 hold with probability $1 - O(n^{-2})$. It is, however, straightforward to modify our proofs to show that these statements hold with probability $1 - O(n^{-c})$ for *any* constant c. Note that this does not hold for Theorem 5.3. Last, we note that all results in this chapter are for the *step* model of the hyperbolic random graph, c. f. Definition 3.

It is an open problem to close the gap between the lower bound $\Omega(\log n)$ and the upper bound $O(\log^{\frac{2}{3-\beta}} n)$ on the diameter in the case $2 < \beta < 3$. We conjecture that

[1] Note that the table therefore refers to a non-standard Preferential Attachment version with adjustable power law exponent $2 < \beta < 3$ (normally, $\beta = 3$).

the diameter in this case is $\Theta(\log n)$ as well. A major indicator is that the bound $O(\log^{\frac{2}{3-\beta}} n)$ becomes worse as $\beta \to 3$, whereas for $\beta > 3$ we have a tight result. On the other hand, similar to most scale-free random graph models, hyperbolic random graphs have a distinct phase transition at $\beta = 3$. This makes such a behavior unlikely, but not impossible. We discuss the difficulties in proving a tight bound for the case $2 < \beta \leqslant 3$ in more detail at the end of Section 5.5.

Used techniques. Our formal analysis of the diameter has to deal with a number of technical challenges. First, in contrast to proving a bound on the average distance, it is not possible to average over all path lengths. In fact, it is not even sufficient to exclude a certain kind of path with probability $1 - O(n^{-c})$; as this has to hold for all possible $\Omega(n!)$ paths. This makes a union bound inapplicable.

A second major challenge is the fact that a probabilistic analysis of shortest paths typically uncovers the probability space in a consecutive fashion. Successively revealing the positions of nodes on the path introduces strong stochastic dependencies that are difficult to handle with probabilistic tail bounds [DA09]. Instead of studying the stochastic dependence structure in detail, we use the geometry and model the hyperbolic random graph as a Poisson point process. This allows us to analyze different areas in the graph independently, which in turn supports our stochastic analysis of shortest paths.

We then bound the length of a shortest path by a multiplicative drift argument known from evolutionary computation [LW13]; and show that the length of $O(\log n)$ shortest paths follows an Erlang distribution and is thereby still $O(\log n)$. This result may be of independent interest, as it relaxes some of the conditions that are usually required to apply the drift theorem.

Notation. We use the notation and results introduced in Chapters 2 and 3. In particular, we often use $\beta = 2\alpha + 1$ interchangeably.

5.3 The Poisson Point Process

We often want to argue about the probability that an area $S \subseteq D_R$ contains one or more nodes. To this end, we usually apply the simple formula[2]

$$\Pr[\exists v \in S] = 1 - (1 - \mu(S))^n \geqslant 1 - \exp(-n \cdot \mu(S)). \tag{5.1}$$

Unfortunately, this formula significantly complicates once the positions of some nodes are already known. This introduces conditions on $\Pr[\exists v \in S]$ which can be hard to

[2] Note that we write $\exists v \in S$ informally to mean whether $S \subseteq D_R$ contains a vertex $v \in V$. To be formally precise, we would have to write $V \cap S \neq \emptyset$. Since it is usually clear from the context that v refers to a node, we chose to keep notation concise.

grasp analytically. For instance, assume we condition on the event that all nodes are in some area $S \subset D_R$. Then, the probability that a node is sampled in $D_R \setminus S$ is always 0.

To circumvent this technical problem, we use a Poisson point process (PPP) [Pen03] which describes a different way of distributing nodes inside D_R. Let the random variable $\mathcal{P}_n = \{(r_1, \varphi_1), (r_2, \varphi_2), \ldots, (r_N, \varphi_N)\}$ denote the set of nodes produced by the PPP. Then, $\mathcal{P}_n$ is fully characterized by the following two properties:

- If two areas S, S' are disjoint, then the number of nodes in $\mathcal{P}_n$ that fall within S and S' are independent random variables.
- The expected number of points in $\mathcal{P}_n$ that fall within S is $\int_S n\mu(S)$.

One can show that the above properties imply that the number of nodes inside S follows a Poisson distribution with mean $n\mu(S)$. In particular, we obtain that the number of nodes $N = |\mathcal{P}_n|$ inside D_R is distributed as $\mathrm{Po}(n)$, i.e. $\mathbb{E}[N] = n$ and

$$\Pr(N = n) = \frac{e^{-n} n^n}{n!} = \Theta\left(n^{-\frac{1}{2}}\right).$$

Moreover, by conditioning on $N = n$, we recover the original distribution of nodes in D_R. Thus, let P be any property that holds with probability at most $O(n^{-c})$ on a hyperbolic random graph whose node set was sampled using $\mathcal{P}_n$. Then, P also holds with probability at most $O(n^{\frac{1}{2}-c})$ in hyperbolic random graphs. This makes the PPP an extremely useful tool as any result that holds with a high enough polynomial probability directly translates to hyperbolic random graphs with an error term of $n^{\frac{1}{2}}$.

We explicitly state whenever we use the PPP instead of the normal hyperbolic random graph. A useful side effect of this model is that Equation (5.1) changes to an equality, i. e., it holds $\Pr[\exists v \in S] = 1 - \exp(-n \cdot \mu(S))$.

5.4 Polylogarithmic Upper Bound

In this section, we show a polylogarithmic upper bound on the diameter of the hyperbolic random graph. The proof proceeds in two steps: First, we show that nodes close to the center form a connected component of diameter $O(\log\log n)$. This covers all nodes that are at least b_O away from the boundary of D_R. We call this area $B_I := B_0(R - b_O)$ the *inner band*, where $b_O = \Theta(\log R)$ will be chosen suitably later. See Figure 5.1 for an illustration. Afterwards, we prove that all remaining nodes in the *outer band* $B_O := D_R \setminus B_I$ form components of at most polylogarithmic diameter.

During the proof, it will sometimes be useful to use a discretization of the radial coordinates. To this end, we partition D_R into R layers of constant thickness 1, where the first layer contains all nodes furthest away from the origin. Thus, all nodes with radial coordinates in $(R - i, R - i + 1]$ are in layer i.[3] We denote the layer i by

[3] Though we never need it explicitly, we remark that the last layer only covers $[0, R - \lfloor R \rfloor]$.

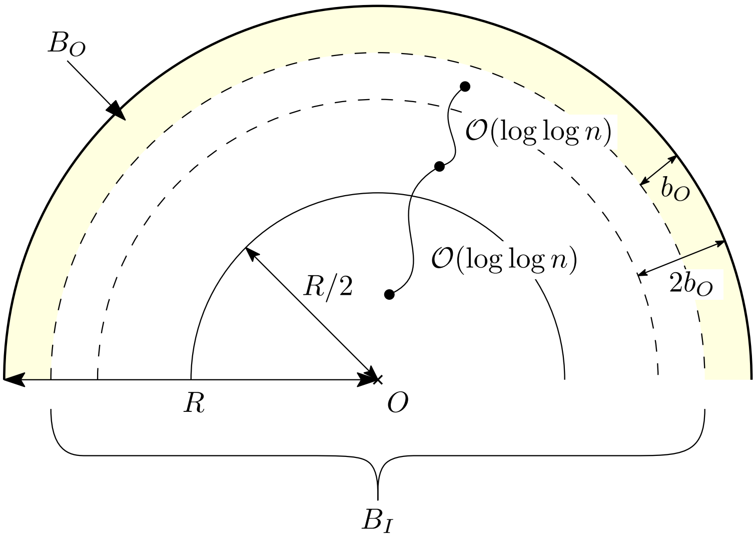


Figure 5.1: The disk D_R is separated into an inner band $B_I = B_0(R - b_O)$ and an outer band $B_O = D_R \setminus B_I$ of thickness b_O. All nodes closer than $R/2$ to the center form a clique and thus have diameter 1. All nodes closer than $R - 2b_O$ to the center have a path of length $O(\log\log n)$ to a node in $B_0(R/2)$. All nodes closer than $R - b_O$ have a path of length $O(\log\log n)$ to a node in $B_0(R - 2b_O)$. Thus, all nodes in B_I are connected, and the diameter of the induced graph is $O(\log\log n)$.

$L_i := B_0(R - i + 1) \setminus B_0(R - i)$. The next Lemma gives a bound on the maximal angle that two nodes in layers i, j may have while still being connected. Recall that for two nodes u, v with fixed radius r_u, r_v, the term $\theta(r_u, r_v)$ describes the maximum angle $\Delta\varphi_{u,v}$ such that u, v are still connected, see Equation (3.4).

Lemma 5.1. *Let* $1 \leqslant i, j \leqslant R/2$*, and consider two nodes* $u \in L_i, v \in L_j$*. Then,*

$$\frac{2}{e} e^{\frac{i+j-R}{2}}(1 + \Theta(e^{i+j-R})) \leqslant \theta(r_u, r_v) \leqslant 2e^{\frac{i+j-R}{2}}(1 + \Theta(e^{i+j-R})),$$

Proof. By Lemma 3.2, $\theta(r_v, r_w) = 2e^{\frac{R-r_v-r_w}{2}}(1 + O(e^{R-r_v-r_w}))$. Since $v \in L_i$, we have that $R - i \leqslant r_v \leqslant R - i + 1$; and similarly $R - j \leqslant r_w \leqslant R - j + 1$. Thus, we obtain

$$2e^{\frac{i+j-2-R}{2}}(1 + \Theta(e^{i+j-R})) \leqslant \theta(r_u, r_v) \leqslant 2e^{\frac{i+j-R}{2}}(1 + \Theta(e^{i+j-R})). \qquad \square$$

Furthermore, we require an estimate for the probability that a node u in layer L_i has a neighbor in layer L_j. To this end, the next lemma computes the probability mass of the area $B_u(R) \cap L_j$.

Lemma 5.2. *Let $1 \leqslant i, j \leqslant R/2$, and consider a node $u \in L_i$. Then,*

$$\mu(L_j \cap B_u(R)) = \Theta(e^{-\alpha j + \frac{i+j-R}{2}}).$$

If further $(i+j)/R < 1 - \varepsilon$ for some constant $\varepsilon > 0$ holds, we obtain the explicit bounds for large n

$$\frac{\alpha}{e\pi} e^{-\alpha j + \frac{i+j-R}{2}} \leqslant \mu(L_j \cap B_v(R)) \leqslant e^{\alpha - 1} \cdot \alpha e^{-\alpha j + \frac{i+j-R}{2}}.$$

Proof. We have by Lemma 3.2 and Lemma 3.1

$$\begin{aligned}
\mu(L_j \cap B_v(R)) &\leqslant \tfrac{1}{2\pi} \mu(L_j) \theta(R-i, R-j) \\
&\leqslant f(R-j+1) \cdot 2e^{\frac{i+j-R}{2}}(1 + \Theta(e^{i+j-R})) \\
&\leqslant \tfrac{\alpha}{\pi} e^{\frac{i+j-R}{2} + \alpha(R-j+1) - \alpha R}(1 + \Theta(e^{-\alpha R} - e^{-2\alpha(R-i)} + e^{i+j-R})) \\
&\leqslant \tfrac{e^\alpha \alpha}{\pi} e^{\frac{i+j-R}{2} - \alpha j}(1 + \Theta(e^{-\alpha R} + e^{i+j-R}))
\end{aligned}$$

For the other direction, a similar computation yields

$$\begin{aligned}
\mu(L_j \cap B_v(R)) &\geqslant \mu(L_j) \theta(R-i+1, R-j+1) \\
&\geqslant f(R-j) \cdot \tfrac{2}{e} e^{\frac{i+j-R}{2}}(1 + \Theta(e^{i+j-R})) \\
&\geqslant \tfrac{\alpha}{e\pi} e^{\frac{i+j-R}{2} + \alpha(R-j) - \alpha R}(1 + \Theta(e^{-\alpha R} - e^{-2\alpha(R-i)} + e^{i+j-R})) \\
&\geqslant \tfrac{\alpha}{e\pi} e^{\frac{i+j-R}{2} - \alpha j}(1 + \Theta(e^{-\alpha R} + e^{i+j-R})).
\end{aligned}$$

□

Using Lemmas 5.1 and 5.2, we can now prove that every node $v \in B_I$ has a path of length $O(\log \log n)$ that leads to a node in $B_0(R/2)$. Recall that the inner band was defined as $B_I := B_0(R - b_O)$.

Lemma 5.3. *Consider a node $v \in L_i \subset B_I$. If $\alpha < 1$, it holds with probability $1 - O(n^{-3})$*

1. *if $i \in [b_O, 2b_O]$, then v has a neighbor in layer L_{i+1}, and*
2. *if $i \in [2b_O, R/2]$, then v has a neighbor in layer L_j for $j = \frac{\alpha}{2\alpha - 1} i$.*

Proof. We begin by proving the first claim. By combining Equation (5.1) and Lemma 5.2, the probability that node $v \in L_i$ does not contain a neighbor in $L_j = L_{i+1}$ is at most

$$\begin{align*}
\exp(-n\mu(L_j \cap B_v(R))) &\leqslant \exp(-\Theta(1) \cdot e^{R/2} \cdot e^{-\alpha j + \frac{i+j-R}{2}}) \\
&= \exp(-\Theta(1) \cdot e^{-\alpha j + \frac{i+j}{2}}) \tag{5.2} \\
&= \exp(-\Theta(1) \cdot e^{-\alpha(i+1) + i + \frac{1}{2}}) \\
&= \exp(-\Theta(1) \cdot e^{(1-\alpha)i}).
\end{align*}$$

We now choose b_O appropriately. Since in the first case we have $i \geqslant b_O$, our goal is to set b_O to a value such that above term is at most $O(n^{-3})$. This is achieved by $b_O := \frac{\log R}{1-\alpha} + c$ for some large enough constant c. Then, we have

$$\exp(-n\mu(L_j \cap B_v(R))) \leqslant \exp(-\Theta(1) \cdot e^{\log R + (1-\alpha)c}) \leqslant \exp(-3 \log n).$$

This proves part (1) of the claim. For part (2), we set $j = \frac{\alpha}{2\alpha-1} i$ and $i \geqslant 2b_O$ in Equation (5.2). It is then upper bounded by

$$\begin{aligned} \exp(-\Theta(1) \cdot e^{-\frac{\alpha^2}{2\alpha-1} i + \frac{i}{2} + \frac{\alpha}{4\alpha-2} i}) &= \exp(-\Theta(1) \cdot e^{(3\alpha-1-2\alpha^2)\frac{i}{4\alpha-2}}) \\ &= \exp(-\Theta(1) \cdot e^{(1-\alpha)\frac{i}{2}}) \\ &\leqslant \exp(-\Theta(1) \cdot e^{\log R + \frac{1-\alpha}{2} c}) \\ &\leqslant \exp(-3 \log n), \end{aligned}$$

which again holds if the constant c in $b_O = \frac{\log R}{1-\alpha} + c$ was chosen large enough. □

Plugging everything together, we obtain that the diameter of the inner band is at most $O(\log \log n)$.

Corollary 5.1. *Let $\frac{1}{2} < \alpha < 1$. With probability $1 - O(n^{-2})$, each pair of nodes $u, v \in B_I$ in the hyperbolic random graph is connected by a path of length $O(\log \log n)$.*

Proof. By a union bound over at most n nodes in B_I, the statement in Lemma 5.3 holds for every node in B_I with probability $1 - O(n^{-2})$. Consider thus a node in a layer $[b_O, 2b_O]$. Since by Lemma 5.3, every such node has a neighbor in the subsequent layer, we need at most $O(\log \log n)$ hops to reach a node in layer $i \in [2b_O, \frac{R}{2}]$. Similarly, every such node has a neighbor in layer $j = \frac{\alpha}{2\alpha-1} i = (1+\varepsilon)i$ for some constant $\varepsilon > 0$. Thus, we need at most $O(\log R) = O(\log \log n)$ hops to reach some node in $B_0(R/2)$. Since all nodes in $B_0(R/2)$ form a clique by the triangle inequality, we therefore obtain that all nodes in B_I form a connected component with diameter $O(\log \log n)$. □

5.4.1 Outer Band

By Corollary 5.1, we obtain that the diameter of the graph induced by nodes in the inner band B_I is at most $O(\log \log n)$. In particular, since all nodes in $B_0(R/2)$ belong to the giant component [BFM14], the nodes in the inner band all belong to the giant component as well. In this section, we prove that each component in the outer band B_O has a polylogarithmic diameter. Then, one can conclude that the overall diameter of the giant component is at most polylogarithmic, since the diameter is then dominated by the components in the outer band.

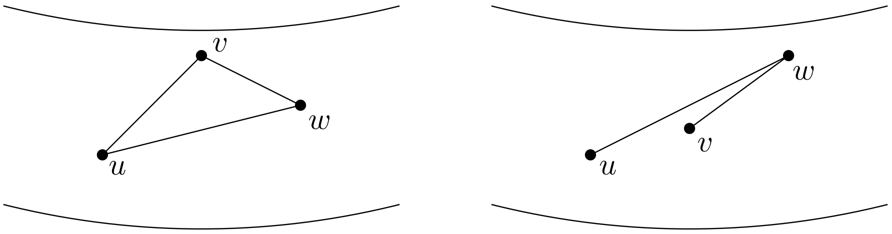


Figure 5.2: Illustration of the statement in Lemma 5.4. By definition, the edge $\{u, w\}$ passes *under* v in both cases.

To argue over sequences of nodes on a path, we introduce the concept of *betweenness*: We say that a node v is *between* two nodes u, w, if $\Delta\varphi_{u,v} + \Delta\varphi_{v,w} = \Delta\varphi_{u,w}$. As an example, consider the nodes $u = (r_1, 0), v = (r_2, \frac{\pi}{2})$ and $w = (r_3, \pi)$. Then, v lies between u and w, but w does not lie between u and v as $\Delta\varphi_{u,v} = \pi/2$ but $\Delta\varphi_{u,w} + \Delta\varphi_{w,v} = \frac{3}{4}\pi$.

If a node v is between two connected nodes u, w and has a small radial coordinate, it is also connected to u, w as shown by the following lemma. Figure 5.2 contains an illustration.

Lemma 5.4. *Let $u, v, w \in V$ be nodes such that v lies between u and w, and let $\{u, w\} \in E$. If $r_v \leqslant r_u$, then v is connected to w.*

Proof. By Lemma 3.4, we know that if two nodes $(r_1, \varphi_1), (r_2, \varphi_2)$ are connected, then so are $(r_1', \varphi_1), (r_2', \varphi_2)$ where $r_1' \leqslant r_1$ and $r_2' \leqslant r_2$. Observe that v may only lie between u, w if $\Delta\varphi_{u,w} \leqslant \pi$. Since the distance between nodes increases on $0 \leqslant \Delta\varphi \leqslant \pi$, this proves the claim. □

Note that by symmetry, if $r_v \leqslant r_u$ and $r_v \leqslant r_w$ both hold, then v is connected to both u, w.

We say that an edge $\{u, w\}$ *passes under* v if the requirement of Lemma 5.4 is fulfilled. Using this, we are ready to show Theorem 5.1. In this argument, we investigate the angular distance a path can at most traverse until it passes under a node in B_I. By Lemma 5.4, we then have with high probability a short path to the center $B_0(R/2)$ of the graph.

Theorem 5.1. *Let $2 < \beta < 3$. The diameter of the giant component in the hyperbolic random graph is $O((\log n)^{\frac{2}{3-\beta}})$ with probability $1 - O(n^{-2})$.*

Proof. Partition the hyperbolic disc into n disjoint sectors of equal angle $\Theta(1/n)$. Recall that $b_O = \frac{\log R}{1-\alpha} + c$ for a large enough constant c. By Equations (3.6) and (5.1), the probability that k consecutive sectors contain no node in B_I is

$$\begin{aligned}(1 - \Theta(k/n) \cdot \mu(B_0(R - b_O)))^n &\leqslant \exp(-\Theta(1) \cdot k \cdot e^{-\alpha \log R/(1-\alpha)})\\ &= \exp(-\Theta(1) \cdot k \cdot (\log n)^{-\frac{\alpha}{1-\alpha}}).\end{aligned}$$

By choosing $k := \Theta((\log n)^{\frac{1}{1-\alpha}})$ large enough, we obtain that with probability $1 - O(n^{-3})$, there are no k such consecutive sectors. By a Chernoff bound, the number of nodes in k such consecutive sectors is $\Theta((\log n)^{\frac{1}{1-\alpha}})$ with probability $1 - O(n^{-3})$. Applying a union bound, we get that with probability $1 - O(n^{-2})$, every sequence of k consecutive sectors contains at least one node in B_I and at most $\Theta(k)$ nodes in total. Consider now a node $v \in B_O$ that belongs to the giant component. Any path (without loops) from v that is longer than $\Theta(k)$ thus must span more than k sectors. In particular, this path either uses a node in B_I; or it passes under a node in B_I. By Lemma 5.4, there thus must exist a path from v to some node $u \in B_I$ of length at most $O(k)$. From u, there is a path of length $O(\log\log n)$ to the center $B_0(R/2)$ of the hyperbolic disc by Corollary 5.1. Since this holds for all nodes and the center forms a clique, the diameter is therefore $O((\log n)^{\frac{1}{1-\alpha}}) = O((\log n)^{\frac{2}{3-\beta}})$. □

From the proof it follows that every component inhabiting $\Omega((\log n)^{\frac{2}{3-\beta}})$ sectors is connected to the center. We derive the following Corollary.

Corollary 5.2. *Let $2 < \beta < 3$. The second largest component of the hyperbolic random graph is of size at most $O((\log n)^{\frac{2}{3-\beta}})$ with probability $1 - O(n^{-3/2})$.*

Proof. The second largest component may not be connected to a node in B_I. Otherwise, as shown above, it belongs (w. h. p.) to the giant component. By the same argument as in Theorem 5.1, the largest such component can contain at most $O(\log^{\frac{2}{3-\beta}} n)$ nodes. □

These bounds improve upon the results in [KM15] who show an upper bound of $O((\log n)^{\frac{32}{(3-\beta)(5-\beta)}+1})$ on the diameter and $O((\log n)^{\frac{64}{(3-\beta)(5-\beta)}+1})$ on the second largest component. As we will see in Theorem 5.3, however, the lower bound on the diameter is only $\Omega(\log n)$. It is an open problem to show a tight result for $\frac{1}{2} < \alpha < 1$. For the case $\alpha > 1$, we bridge this gap in the next section.

5.5 Logarithmic Upper Bound

In this section, we show that the diameter of the hyperbolic random graph is $O(\log n)$, if $\beta > 3$, or, equivalently, $\alpha > 1$. The intuition behind the analysis in this section is opposite to the approach in the section before. Instead of showing that there exist short paths to the center of the graph from all nodes, we show that all shortest paths terminate after $O(\log n)$ steps since they reach the boundary of D_R. This holds because for every node v, their largest degree neighbor is (in expectation) of smaller degree than v itself. Thus, a shortest path visits successively nodes of smaller and smaller degree, until it cannot continue.

In this section, we prove all intermediate results using the Poisson point process (PPP), see Section 5.3. We begin by showing that each node's largest degree neighbor is of small degree, or, equivalently, is in a small layer. Here, we have to deal with an additional technicality: When sampling a shortest path, we already have uncovered a neighbor of the current node. We therefore strengthen our result by conditioning on this information. Given a node v and a *forbidden* neighbor f, we denote by the random variable $Y(v, f)$ the largest layer in which v has a neighbor that is not f. If v has no other neighbors than f, we set $Y(v, f) = 0$. We show the following.

Lemma 5.5. *There exist constants $\varepsilon, \delta > 0$ such that for all $i \geqslant \varepsilon$*

$$\mathbb{E}\left[Y(v, f) \mid v \in L_i, f \in L_j\right] \leqslant (1-\delta)i.$$

Proof. We first compute the probability that v has no neighbors in layer x, without conditioning on f. This happens when no nodes are sampled in the area $L_x \cap B_v(R)$. Recall that in the PPP, it holds $\Pr[\exists u \in S] = 1 - \exp(-n\mu(S))$, see Section 5.3. Thus, by Lemma 5.2,

$$\begin{aligned}\Pr[|\Gamma(v) \cap L_x| = 0] &= \exp(-n\mu(L_x \cap B_v(R))) \\ &= \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})x}). \qquad (5.3)\end{aligned}$$

We now compute the probability that all neighbors $\Gamma(v) \setminus f$ are below layer m. For this, we apply the definition of conditional probability to obtain

$$\Pr[Y(v, f) < m \mid v \in L_i, f \in L_j] = \frac{\Pr[Y(v, f) < m \wedge f \in L_j \mid v \in L_i]}{\Pr[f \in L_j \mid v \in L_i]}.$$

Recall that $f \in L_j$ is shorthand for whether v has a neighbor in L_j, and therefore $\Pr[f \in L_j \mid v \in L_i] = \Pr[|\Gamma(v) \cap L_j| > 0]$. We now consider the two cases (i) $j < m$ and (ii) $j \geqslant m$. Since each layer is independent in the PPP, we obtain for case (i)

$$\begin{aligned}\Pr[Y(v, f) < m \mid v \in L_i, f \in L_j] &= \Pr[\forall x \geqslant m\colon |\Gamma(v) \cap L_x| = 0] \cdot \frac{\Pr[|\Gamma(v) \cap L_j| > 0]}{\Pr[|\Gamma(v) \cap L_j| > 0]} \\ &= \prod_{x \geqslant m} \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})x}) \qquad (5.4) \\ &= \exp(-\Theta(1) \cdot \sum_{x \geqslant m} e^{\frac{i}{2}-(\alpha-\frac{1}{2})x}) \\ &= \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})m}),\end{aligned}$$

since the sum is geometric. For case (ii), we condition on the fact that $f \in L_j$ for $j \geqslant m$. This means that v has *exactly* one neighbor in L_j, and all other neighbors are below layer m. Thus, we obtain

$$\Pr[Y(v,f) < m \mid v \in L_i, f \in L_j] = \frac{\Pr[|\Gamma(v) \cap L_j| = 1]}{\Pr[|\Gamma(v) \cap L_j| > 0]} \cdot \prod_{\substack{x \geqslant m \\ x \neq j}} \Pr[|\Gamma(v) \cap L_x| = 0].$$

Recall that the number of nodes in an area S is distributed as $\mathrm{Po}(n\mu(S))$ in the PPP. Thus,

$$\begin{aligned}\Pr[|\Gamma(v) \cap L_j| = 1] &= \Pr[\mathrm{Po}(n\mu(L_j \cap B_v(R))) = 1] \\ &= n\mu(L_j \cap B_v(R)) \cdot \exp(-n\mu(L_j \cap B_v(R))) \\ &= n\mu(L_j \cap B_v(R)) \cdot \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})j}).\end{aligned}$$

On the other hand, for $\Pr[|\Gamma(v) \cap L_j| > 0]$ we apply the inequality $1 - e^{-x} \leqslant x$ (c. f. Lemma 2.1) to obtain

$$\Pr[|\Gamma(v) \cap L_j| > 0] = 1 - \exp(-n\mu(L_j \cap B_v(R))) \leqslant n\mu(L_j \cap B_v(R))$$

Plugging this together, we have in case (ii)

$$\begin{aligned}\Pr[Y(v,f) < m \mid v \in L_i, f \in L_j] &\geqslant \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})j}) \cdot \prod_{\substack{x \geqslant m \\ x \neq j}} \Pr[|\Gamma(v) \cap L_x| = 0] \\ &= \prod_{x \geqslant m} \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})x}).\end{aligned}$$

Thus, we obtain the same result as in Equation (5.4).

Finally, to compute the expectation of $Y(v,f)$, we sum over the CCDF as in Equation (2.2). This yields

$$\begin{aligned}\mathbb{E}\left[Y(v,f) \mid v \in L_i, f \in L_j\right] &= \sum_{m=1}^{\infty} \Pr[Y(v,f) \geqslant m \mid v \in L_i, f \in L_j] \\ &= \sum_{m=1}^{\infty} (1 - \Pr[Y(v,f) < m \mid v \in L_i, f \in L_j]) \\ &\leqslant \sum_{m=1}^{\infty} (1 - \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})m})).\end{aligned}$$

Since the first $\frac{1}{2\alpha-1}i$ terms of the sum are close to 1, we simply overestimate them with 1. For the remaining part of the sum, we again apply the inequality $1 - e^{-x} \leqslant x$ and obtain

$$\begin{aligned}\mathbb{E}\left[Y(v,f) \mid v \in L_i, f \in L_j\right] &\leqslant \frac{i}{2\alpha-1} + \sum_{m=\frac{1}{2\alpha-1}i}^{\infty} \left(1 - \exp(-\Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})m})\right)\\ &\leqslant \frac{i}{2\alpha-1} + \sum_{m=\frac{1}{2\alpha-1}i}^{\infty} \Theta(1) \cdot e^{\frac{i}{2}-(\alpha-\frac{1}{2})m}\\ &\leqslant \frac{i}{2\alpha-1} + \Theta(1).\end{aligned}$$

To prove the claim, we choose $\varepsilon > 0$ as a large enough constant and $\delta > 0$ as a small enough constant. Then, since $i \geqslant \varepsilon$ by assumption and $\alpha > 1$, it holds $\frac{1}{2\alpha-1} + \Theta(\frac{1}{i}) \leqslant 1 - \delta$. □

We note that the same result can be achieved when there is no forbidden node f by an analogous computation. In this case, we simply write $Y(v)$ instead of $Y(v,f)$.

Assume we now fix some vertex u and sample an arbitrary shortest path $\pi = [u = V_0, V_1, V_2, \ldots]$. We want to obtain a bound on the length $|\pi|$ that holds with high probability. Unfortunately, this process is hard to analyze exactly since it is governed by many dependencies. For example, V_2 may not be connected to V_0, as otherwise, π is not a shortest path. We may, however, analyze an alternative process that is closely related. To this end, consider the following sequence of random variables, also called a *random walk*.

$$(Y_i)_{i\geqslant 1}, \quad Y_1 := Y(V_0), \quad Y_i := Y(V_{i-1}, V_{i-2}) \text{ if } i \geqslant 2$$

Recall that $Y(V_{i-1}, V_{i-2})$ denotes the largest layer in which V_{i-1} has a neighbor that is not V_{i-2}. It is therefore immediate that V_i is always in a smaller or equal layer than Y_i, since Y_i denotes the highest layer in which the shortest path can continue. Further, if $Y_i = 0$, then $|\pi| \leqslant i$ as V_{i-1} has no further neighbors apart from V_{i-2}. Recall now that a node's neighborhood is monotone in its radial coordinate, i. e., the smaller r_v, the more neighbors v has, see Lemma 3.4. Thus, we may overestimate the radial coordinates of the nodes $V_0, V_1, \ldots$ and obtain an upper bound on $|\pi|$, since each node in π may only gain additional neighbors.

A natural candidate for this overestimation is to use the upper bounds given by $Y(\cdot)$. For example, we know that V_0 has no neighbors in layers above $Y(V_0)$, thus we may overestimate that V_1 has radial coordinate $R - Y(V_0)$. The next lemma formalizes this intuition by giving a random walk $(X_i)_{i\geqslant 1}$ that *dominates* $(Y_i)_{i\geqslant 1}$, i. e., it holds $\Pr[X_i \geqslant x] \geqslant \Pr[Y_i \geqslant x]$ for all i and x.

Lemma 5.6. *Consider the random walk $(X_i)_{i \geqslant 1}$ with $X_i \in \mathbb{N}$, $X_1 := Y_1$ and distribution*

$$\begin{aligned} \Pr[X_{i+1} \geqslant j \mid X_i = \ell] &= 1 - \exp(-ce^{\frac{\ell}{2}-(\alpha-\frac{1}{2})j}) && \textit{if } \ell > 0, \\ \Pr[X_{i+1} = 0 \mid X_i = 0] &= 1 && \textit{otherwise.} \end{aligned} \tag{5.5}$$

If c is a large enough constant, this random walk dominates $(Y_i)_{i \geqslant 1}$.

Proof. We prove this statement by giving a coupling of the two random walks such that X_i will always be greater or equal than Y_i. By definition, this holds for X_1 and Y_1. We now assume inductively that such a coupling exists for $1, \ldots, i$ and show how to produce it for $i + 1$.

We first reveal $\Gamma(V_i)$, i. e., all neighbors of V_i. Obviously, the shortest path formed by the vertices $V_1, \ldots, V_i$ will continue on a node from $\Gamma(V_i) \setminus V_{i-1}$ or end at V_i. Consider now a fresh instance of a hyperbolic random graph in which no nodes have been sampled yet. Assume that in this new instance, we place a node V_i' at position $(R - X_i, \varphi_{V_i})$. Observe that this node has a smaller or equal radius than V_i, since by induction we know that $X_i \geqslant Y_i$. Further, we have that by Lemma 3.4, $B_{V_i'}(R) \cap B_0(R)$ is a superset of $B_{V_i}(R) \cap B_0(R)$. Since we are in the PPP, vertices in the additional area $B_{V_i'}(R) \cap B_0(R) \setminus B_{V_i}(R)$ may be sampled independently from $B_{V_i}(R) \cap B_0(R)$ since these regions are disjoint.

Thus, we may couple the neighborhood of V_i' to contain copies of all nodes $\Gamma(V_i)$ and possibly more. Therefore, the largest layer containing a neighbor of V_i' that is not the copy of V_{i-1} is $Y(V_i', V_{i-1}) \geqslant Y(V_i, V_{i-1})$. And as derived in Lemma 5.5, there is a constant c such that $Y(V_i', V_{i-1})$ is distributed as

$$\Pr[Y(V_i', V_{i-1}) \geqslant j \mid X_i = \ell] \leqslant 1 - \exp(-ce^{\frac{\ell}{2}-(\alpha-\frac{1}{2})j}).$$

This agrees with Equation (5.5), and since $Y(V_i', V_{i-1}) \geqslant Y(V_i, V_{i-1})$, so is $X_{i+1} \geqslant Y_{i+1}$. □

Observe that Lemma 5.5 shows that by definition of $(X_i)_{i \geqslant 1}$, it also holds

$$\mathbb{E}\left[X_{i+1} \mid X_i\right] \leqslant (1 - \delta)X_i,$$

if X_i is at least a large enough constant ε. In other words, $(X_i)_{i \geqslant 1}$ has a so-called *multiplicative drift* towards 0 while it is above some constant layer ε.

We now finally turn to analyzing the length of the random walk $(X_i)_{i \geqslant 1}$ until it reaches 0, and thus, by our explanations above, the length $|\pi|$ of a shortest path. Let $T := \min\{i \mid X_i = 0\}$ be the random variable describing the number of iterations until X_i hits 0. We bound T by a multiplicative drift theorem as presented by Lehre and Witt [LW13, Theorem 7] and originally developed by Doerr and Goldberg [DG10,

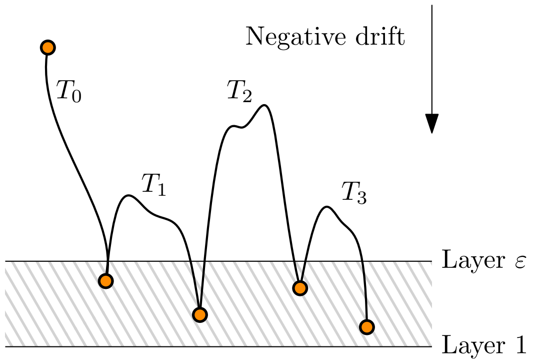


Figure 5.3: A sketch of the diameter proof. Until layer ε, there is a negative drift on the next node of the random walk. Between layer ε and layer 1, there is a constant probability for the walk to end. Thus, the overall walk visits the marked area $O(\log n)$ times. Naively, the random walk is thus of length $O(\log^2 n)$, however, Lemma 5.7 shows that it is with high probability of length $O(\log n)$.

Theorem 1] for the analysis of evolutionary algorithms. For the sake of completeness, we restate their result.

Theorem 5.4 (from [LW13; DG10]). *Let $(X_i)_{i\geqslant 1}$ be a stochastic process over some state space $\{0\} \cup [x_{\min}, x_{\max}]$, where $x_{\min} > 0$. Suppose that there exists some $0 < \delta < 1$ such that $\mathbb{E}[X_{i+1} \mid X_0, \dots, X_i] \leqslant (1-\delta)X_i$. Then, for the hitting time $T := \min\{i \mid X_i = 0\}$ it holds*

$$\Pr[T \geqslant \tfrac{1}{\delta}(\ln(X_0/x_{\min}) + r) \mid X_0] \leqslant e^{-r} \text{ for all } r > 0.$$

Unfortunately, in our case, the multiplicative drift vanishes once $X_i < \varepsilon$. We are therefore only interested when the random walk falls below ε. Thus, we map all points $x < \varepsilon$ to 0 and set $X_0 \leqslant R$ and $x_{\min} = \varepsilon$. Using Lemma 5.5 this shows that

$$\Pr[T \geqslant \tfrac{1}{\delta} \cdot (\log\log n - \log\varepsilon + r)] \leqslant e^{-r}. \tag{5.6}$$

Hence, by setting $r = 4\log n$ we obtain that with probability $1 - O(n^{-4})$, the random walk $(X_i)_{i\geqslant 1}$ ends after $O(\log n)$ steps below ε.

Once X_i crosses ε, we consider two possibilities: Either, the random walk ends; or it continues at ε. For the latter case, recall that by Lemma 3.4, a node's neighborhood is monotone in the radial coordinate. Increasing X_i to ε thus results in a dominating random walk. Again, by Theorem 5.4, its stopping time T is distributed as

$$\Pr[T \geqslant \tfrac{1}{\delta} \cdot r] \leqslant e^{-r}. \tag{5.7}$$

As we prove later, the probability for X_i to end is constant if it is below ε. By the Chernoff bound, the random walk therefore visits at most $O(\log n)$ nodes below layer ε

before stopping w. h. p.. A naive application of Equations (5.6) and (5.7) thus yields that $(X_i)_{i\geqslant 1}$ is w. h. p. of length $O(\log^2 n)$, see Figure 5.3. It is, however, possible to improve this result. The reason is that when adding together $O(\log n)$ random variables that are exponentially distributed, most of them will be of constant size. Thus, intuitively, the main contribution to the sum comes from just one variable achieving a value of $\Omega(\log n)$; whereas all others are small. In the following, we prove this intuition and thereby show that if $(X_i)_{i\geqslant 1}$ drops below layer ε not more than $O(\log n)$ times, then the total length of the random walk is still w. h. p. $O(\log n)$.

Lemma 5.7. *Let $(T_i)_{i=1\dots x}$ be $x = c\log n$ independent random variables, each with distribution as in Equation* (5.7)*. Then, with probability* $1 - O(n^{-5})$, $\sum_{i=1}^{x} T_i \leqslant O(\log n)$.

Proof. As we only know an exponential tail bound but not the exact distribution of T_i, we instead investigate the random variables T_i' whose distribution is given by

$$\Pr[T_i' \geqslant r] = \exp(-\delta r).$$

Note that T_i' dominates T_i, therefore it suffices to find a tail bound on $T^* := \sum_i T_i'$. Since T^* is a sum of X exponentially distributed variables with equal mean, the distribution of T^* is an Erlang distribution [EHP00] and we have

$$\Pr[T^* \geqslant t] = \sum_{i=0}^{x-1} \frac{1}{i!} e^{-\delta t} (\delta t)^i. \tag{5.8}$$

To estimate this term, we observe that a random variable P that is Poisson distributed with mean δt has probability mass function $\Pr[P = i] = (\delta t)^i e^{-\delta t} \frac{1}{i!}$. This term equals the summands in Equation (5.8), and we can therefore write

$$\Pr[T^* \geqslant t] = \sum_{i=0}^{x-1} \Pr[P = i] = \Pr[P < x].$$

By a Chernoff bound for Poisson variables [MU05], we can estimate this with

$$\Pr[P < x] \leqslant \frac{e^{-\delta t}(e\delta t)^x}{x^x},$$

as long as $x = c\log n \leqslant \delta t$. Choosing $t = c'\log n$ large enough, we obtain

$$\begin{aligned}\Pr[T^* \geqslant t] &\leqslant \frac{e^{-\delta c'\log n}(e\delta c'\log n)^{c\log n}}{(c\log n)^{c\log n}} \\ &= n^{-\delta c'} \left(\frac{e\delta c'}{c}\right)^{c\log n} \\ &= n^{-\delta c' + c\log\left(\frac{e\delta c'}{c}\right)} \leqslant n^{-5}.\end{aligned}$$

□

Using our auxiliary lemmas, we can prove that the diameter of the hyperbolic random graph is $O(\log n)$ if $\beta > 3$, or, equivalently, $\alpha > 1$.

Theorem 5.2. *Let $\beta > 3$. Then, the diameter of the giant component in the hyperbolic random graph is $O(\log n)$ with probability $1 - O(n^{-2})$.*

Proof. Consider any node v and a shortest path beginning in v. The length of the shortest path is dominated by the length of the random walk $(X_i)_{i \geqslant 1}$ as defined in Equation (5.5). Let $s = c \log n$ for some large enough constant c, and let ε be a large enough constant. By Lemma 5.5 and Equation (5.6), $X_s < \varepsilon$ with probability $1 - O(n^{-4})$. The probability that $X_{s+1} = 0$ is then by Equation (5.5)

$$\begin{aligned}\Pr[X_{s+1} = 0 \mid X_s < \varepsilon] &= \Pr[X_{s+1} < 1 \mid X_s < \varepsilon] \\ &\geqslant \exp(-ce^{\frac{\varepsilon}{2} - (\alpha - \frac{1}{2})}) \\ &= \Theta(1).\end{aligned}$$

Thus, for a large enough constant c', the probability that the random walk $(X_i)_{i \geqslant 1}$ returns more than $c' \log n$ times to a value $< \varepsilon$ is at most n^{-5}. Consequently, we may apply Lemma 5.7 and obtain that with probability $1 - O(n^{-5})$, the length of the walk $(X_i)_{i \geqslant 1}$ is $O(\log n)$. Thereby, the length of a shortest path from any node v in the Poisson point model is at most $O(\log n)$ with a probability of at least $1 - O(n^{-5})$. By the union bound, it thus holds that all $O(n^2)$ shortest paths in the graph have length at most $O(\log n)$ with probability $1 - O(n^{-3})$. Finally, since we have done the analysis in the Poisson point model, we have that the probability that the shortest path in the hyperbolic random graph is $O(\log n)$ with probability at least $1 - O(n^{-2})$. □

Let us conclude this section by mentioning that the case $2 < \beta \leqslant 3$ remains an open problem: Even though we provided a better polylogarithmic bound on the diameter, the lower bound in the next section only produces a path of length $\Omega(\log n)$. These bounds leave an open gap, and as of today it remains unclear whether the diameter is truly logarithmic or in fact polylogarithmic.

The main problem in proving a better bound in that case lies within the outer band B_O. Consider a similar approach as in Theorem 5.2: If $\beta < 3$, then the random walk Y_i has a multiplicative drift towards the center of D_R. While this sounds promising, there is a significant problem. In our case, we may simply discard the dependencies of previously visited nodes. After all, they only may exclude potential areas of D_R to contain neighbors and discarding them increases the (expected) layer of the next vertex.

This estimation does not hold the other way around: If the random walk is to reach the center of the graph, we may not discard these dependencies as doing so decreases the length of the random walk. Thus, one has to consider the dependencies of previously visited nodes. This is difficult, since the influence of an earlier node depends on its (angular) distance from the current node, and its radial coordinate. Imagine, for instance, that $X_1 = R/2$, and $X_2 = 1$. We would now like to sample X_3 only depending on X_2. We know, however, that X_3 cannot be in a large layer like $R/2$. Otherwise, the node represented by X_3 would likely have been connected to the node represented by X_1, and thus X_2 cannot be the layer containing the largest neighbor of the node in X_1.

What makes matters even worse is that one needs to consider these conditions of *all* preceding nodes, not only the last. This combination of factors makes an analysis technically challenging. Comparing to other random graph models (see Table 5.1), however, it would seem highly surprising if the diameter is indeed polylogarithmic. Furthermore, the upper bound on the diameter $O((\log n)^{\frac{2}{3-\beta}})$ in Theorem 5.1 increases as $\beta \rightarrow 3$; however, for $\beta > 3$, the diameter is $O(\log n)$ by Theorem 5.2. While such an abrupt phase transition seems unnatural, however, it is not completely unreasonable: The largest component in hyperbolic random graphs for $2 < \beta < 3$ is of linear size, whereas for $\beta > 3$ it is only of polynomial order. The lower connectivity might disconnect long paths and therefore decrease the diameter overall. Nevertheless, we believe this unlikely and conclude this section with the following conjecture.

Conjecture 1. *The diameter of the hyperbolic random graph with power-law exponent $2 < \beta \leqslant 3$ is $O(\log n)$ with high probability.*

5.6 Logarithmic Lower Bound

Kiwi and Mitsche [KM15] provide a proof for the existence of a path component of length $\Theta(\log n)$ with high probability. In this section, we present a slightly simpler proof that there exists a component with diameter of $\Omega(\log n)$. We achieve this by considering $\Theta(\log n)$ subsequent sectors of angle $\Theta(\frac{1}{n})$, such that each sector contains exactly one node in layer L_1, and no further nodes. We can show that such a sequence of sectors occurs at least once in the graph w. h. p., and that it forms a path of length $\Omega(\log n)$ without shortcuts. In the case where $2 < \beta < 3$, we can further show that this path component is connected to B_I. This proves the intuition that the giant component has a diameter of at least $\Omega(\log n)$, which is not obvious a priori.

Theorem 5.3. *Let $\beta > 2$. Then, there exists a component in the hyperbolic random graph with diameter $\Omega(\log n)$ with probability $1 - O(n^{1-\frac{\beta}{2}})$. If $\beta < 3$, this is the giant component.*

Proof. Let $\varepsilon := (\frac{1}{2} - \frac{1}{4\alpha})$. Observe that for $\alpha > \frac{1}{2}$, we have $\varepsilon > 0$. Consider the hyperbolic random graph model, i.e. *not* the Poisson point process. We first show that with high probability, there are no nodes in $B_0(\varepsilon R)$. For this, we observe that $\mu(B_0(\varepsilon R)) = \Theta(1) \cdot \exp(-(\frac{\alpha}{2} + \frac{1}{4})R) = o(1)$ by Equation (3.6). Thus, we may apply

Lemma 2.2 to obtain

$$\begin{aligned}\Pr[\nexists v \in B_0(\varepsilon R)] &= (1-\mu(B_0(\varepsilon R)))^n \\ &\geqslant \exp(-\Theta(1)\cdot e^{R/2}\cdot e^{-(\frac{\alpha}{2}+\frac{1}{4})R}) \\ &\geqslant 1-\Theta(1)\cdot e^{(\frac{1}{4}-\frac{\alpha}{2})R} \\ &= 1-\Theta(n^{-(\alpha-\frac{1}{2})}).\end{aligned}$$

It is important to perform this computation in the hyperbolic random graph model, as the probability that there are no nodes in $B_0(\varepsilon R) \neq \emptyset$ is smaller than $1-n^{-\frac{1}{2}}$ for some values of α. Thus, a direct application of the Poisson point process will result in a useless tail bound. Instead, we condition in the Poisson point process on the fact that there are no nodes in $B_0(\varepsilon R)$. Then, if the same holds in the hyperbolic random graph, we again recover the same distribution of nodes by simply applying the Poisson point process to the area $D_R \setminus B_0(\varepsilon R)$ instead of D_R. The expected number of nodes in the PPP is then

$$\begin{aligned}\mathbb{E}\left[|\mathcal{P}_n \setminus B_0(\varepsilon R)|\right] &= n\cdot\mu(B_0(R)\setminus B_0(\varepsilon R)) \\ &= n\cdot(1-\Theta(e^{\alpha(\varepsilon R-R)})) \\ &= n\cdot(1-\Theta(e^{-\alpha R(\frac{1}{2}+\frac{1}{4\alpha})})) \\ &= n\cdot(1-\Theta(e^{-\alpha\frac{R}{2}-\frac{R}{4})})) \\ &= n\cdot(1-\Theta(n^{-(\alpha+\frac{1}{2})})) \\ &= n-o(1).\end{aligned}$$

Thus, the penalty term is still equal to $\Theta(n^{\frac{1}{2}})$:

$$\begin{aligned}\Pr[|\mathcal{P}_n \setminus B_0(\varepsilon R)| = n] &= (n-o(1))^n\exp(-n+o(1))\frac{1}{n!} \\ &\geqslant \Theta(1)\cdot(n-o(1))^n\exp(-n+o(1))n^{-n-\frac{1}{2}}e^n \\ &= \Theta(n^{-\frac{1}{2}})\cdot(1-o(\tfrac{1}{n}))^n \\ &\geqslant \Theta(n^{-\frac{1}{2}}).\end{aligned}$$

In the following, we therefore may condition on the fact that there are no nodes in $B_0(\varepsilon R)$; and switch to the Poisson point process. Next, we compute the probability that a shortest path of length $\Omega(\log n)$ appears in a certain area. At the end, we amplify this probability by repeating the experiment independently multiple times to arrive at our desired result.

To this end, similarly to Theorem 5.1, we now partition the disk D_R into $\Theta(n)$ sectors

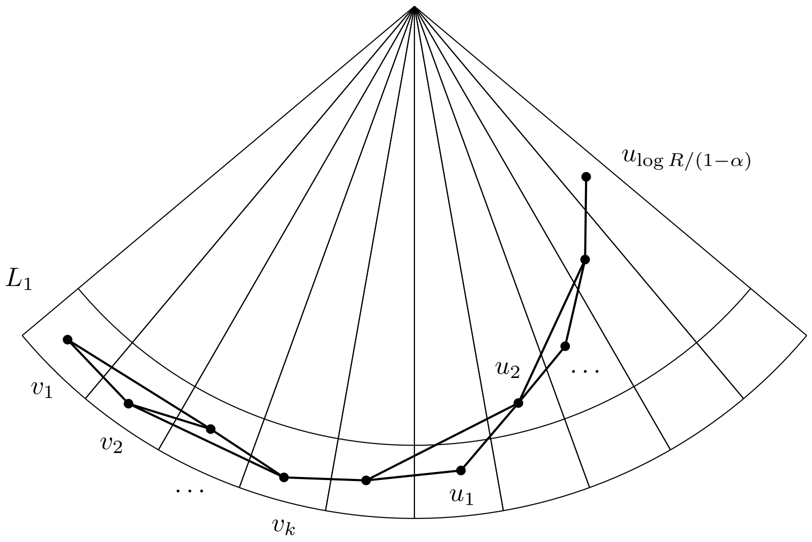


Figure 5.4: Proof illustration for Theorem 5.3. The disk D_R is partitioned into sectors of angle $\frac{1}{n}$. Nodes $v_1, \dots, v_k$ in neighboring sectors form a path component of length $\Theta(k)$. If $\alpha < 1$, the path is also connected via the nodes $u_1, \dots, u_{\frac{\log R}{1-\alpha}+c}$ to B_I.

of equal angle $\varphi := e^{-R/2} = \Theta(\frac{1}{n})$. Then, two nodes $u, v \in L_1$ in neighboring sectors have angular distance at most $2e^{-R/2}$ and are therefore by Lemma 3.2 connected. On the flip side, if two nodes are at least 6 sectors apart, they are not connected, since their angle is $6e^{-R/2} > 2e^{-R/2+1}(1 + O(e^{-R}))$.

Consider now k consecutive sectors, where k is to be fixed later. Let p_1 be the probability that such a sector contains exactly one node in L_1. Then, $p_1 \geqslant (e^{-R/2} \cdot ne^{-\alpha}) = e^{-\Theta(1)}$, i.e. a constant bounded away from 0. Let p_2 be the probability that all neighbors of this node lie in L_1 as well. This is again $e^{-\Theta(1)}$ by Lemma 3.3. We name these nodes $v_1, \dots, v_k$. As argued above, k such nodes form a shortest path of length $\Omega(k)$. We now argue that when $\beta < 3$, this path is also connected to the core of the hyperbolic random graph by exposing a path $u_1, \dots, u_h$ to the inner band B_I, where $h = O(\log \log n)$. Figure 5.4 contains an illustration of the proof.

The probability that sectors $k + 1$ to $k + c + 1$ also each contain exactly one node in L_1 is again $e^{-\Theta(1)}$, if c is constant. From here, we expose a path to the inner band B_I as follows. Assume we have a node $u_i \in L_i$ in sector $k + c + i$. Assume further u_i is to the right of all previous sectors. Then, we consider the probability that u_i has a neighbor to the right in layer L_{i+1}, while we still condition on the fact that all nodes $v_1, \dots, v_k$ have no neighbors above L_1 as stated before. By Lemma 5.1, a node $u_{i+1} \in L_{i+1}$ is not connected to any of the nodes $v_1, \dots, v_k$ if $\Delta\varphi_{u_{i+1}, v_k} > 6e^{\frac{i-R}{2}}$. Similarly, it is connected to u_i if $\Delta\varphi_{u_i, u_{i+1}} \leqslant \frac{2}{e}e^{\frac{2i-R}{2}}$. Since all nodes $v_1, \dots, v_k$ are to the left of u_i and there are

c sectors between, u_{i+1} may fall into an angular range of

$$\tfrac{2}{e}e^{\frac{2i-R}{2}} - 6e^{\frac{i-R}{2}} + ce^{-R/2} \geqslant \Theta(1)\cdot e^{i-\frac{R}{2}},$$

if c is large enough. Therefore, the probability that node u_i has a neighbor in layer L_{i+1} that is not connected to $v_1,\ldots,v_k$, is at least

$$1-\exp(-\Theta(1)\cdot n\cdot e^{-\alpha i}e^{i-\frac{R}{2}}) = 1-\exp(-\Theta(1)\cdot e^{(1-\alpha)i}) = \Theta(1).$$

Such a path to B_I is of length at most $\frac{\log R}{1-\alpha}+c = O(\log\log n)$. In total, the probability that $v_1,\ldots,v_k$ exist as described above; and that they are connected to B_I is thereby $e^{-\Theta(k+\log\log n)}$; or just $e^{-\Theta(k)}$ in the case where $\beta \geqslant 3$.

It remains to compute how often we can repeat this experiment independently. Consider a node in the outermost layer $v \in L_1$. Since we assumed that $v_1,\ldots,v_k$ have no neighbors in layers above 1, we have uncovered the whole area $B_0(R)\cap B_{v_i}(R)$ for all i. The largest angular distance such a node v can have to one of its neighbors is by Lemma 3.2

$$\Delta\varphi \leqslant 2e^{-\frac{\varepsilon R}{2}}(1\pm O(e^{-\varepsilon R})) \leqslant O(n^{-\varepsilon}), \tag{5.9}$$

where $\varepsilon = (\frac{1}{2}-\frac{1}{4\alpha})$ as chosen in the beginning. This holds since we condition on the fact that there are no nodes in $B_0(\varepsilon R)$.

We thus expose at most an angle of $O(\frac{k}{n}+n^{-\varepsilon}+\log\log n\cdot\frac{1}{n}(\log n)^{1/(1-\alpha)})$ of the graph. Therefore, if $\frac{k}{n} < n^{-\varepsilon}$, we can repeat this experiment independently $\Omega(n^{\varepsilon})$ times. The probability that all of them fail is at most

$$(1-e^{-\Theta(k+\log\log n)})^{\Omega(n^{\varepsilon})} \leqslant \exp(-e^{-\Theta(k)}\Omega(n^{\varepsilon})) = \exp(-n^{\Omega(1)}),$$

if $k=\Theta(\log n)$ is chosen small enough. This proves the claim. □

5.7 Conclusion

We derive a new polylogarithmic upper bound on the diameter of hyperbolic random graphs for the case $2<\beta<3$; and show that it is $O(\log n)$ if $\beta>3$. We further prove a logarithmic lower bound. This immediately yields lower bounds for any broadcasting protocol that has to reach all nodes. Processes such as bootstrap percolation or rumor spreading therefore must run at least $\Omega(\log n)$ steps until they inform all nodes in the giant component. In particular, this result stands in contrast to the average distance of two nodes in the hyperbolic random graph, which is of order $\Theta(\log\log n)$ [BKL15; ABF15]. This implies the existence of a path that is exponentially longer than the average path.

It remains an open problem to find a matching upper bound on the diameter in the case $2<\beta\leqslant 3$, but we conjecture that it is of order $O(\log n)$ as well. A natural direction to expand this research is to investigate rumor spreading on hyperbolic random graphs. Even though there exists a significant body of research on rumor spreading in other social network models, hyperbolic random graphs are largely unexplored in this context. The only work known to us in this direction is by Candellero and Fountoulakis [CF16], who study bootstrap percolation in this model. An interesting question in this context is whether the constant clustering of this model affects rumor spreading protocols in a positive or negative way. Previously inspected scale-free graph models have subconstant clustering.

6 Efficient Hyperbolic Embeddings

This chapter is based on joint work with Thomas Bläsius, Tobias Friedrich and Sören Laue [Blä+16]. It contains a more detailed discussion of possible improvements of the algorithm over the conference version.

6.1 Introduction

In this chapter, we construct and implement a new maximum likelihood estimation algorithm that embeds scale-free graphs in the hyperbolic space. That is, given a list of edges, we seek hyperbolic coordinates of nodes such that close pairs in the embedding are likely to have an edge. All previous approaches of similar embedding algorithms require a runtime of $\Omega(n^2)$. Our algorithm achieves quasilinear runtime, which makes it the first algorithm that can embed networks with hundreds of thousands of nodes in less than one hour on commodity hardware. We demonstrate the performance of our algorithm on artificial and real networks. In all typical metrics like Log-likelihood and greedy routing our algorithm discovers embeddings that are very close to the ground truth.

It is well known in the visualization community that hierarchical or tree-like structures can be well represented in a hyperbolic space [SKP16]. There are three general approaches to embed a network in the hyperbolic space:

- A popular way to obtain hyperbolic coordinates for the nodes of a network is embedding a spanning tree of the network in hyperbolic space [WR02; Wal04; Mun98]. As trees can be embedded perfectly, this is a very efficient way to map a network and has been used for interactive network browsers, which allow assigning more display space to the interesting portions of a network [LRP95; LR96]. The result might reduce visual clutter and help focus, but it ignores most structural details of the network. Nodes which are close in graph distance are not necessarily close in hyperbolic space. In fact, clusters and most local structures are not preserved.

- Another approach is determining shortest path distances and finding an embedding where metric distances match the graph distances. Computing the all-pair-shortest-path matrix can be done with the well established Euclidean data analysis method Multidimensional Scaling (MDS) [CC00], which has been translated to hyperbolic geometry [CE16]. Due to the quadratic size of the distance matrix, this approach only works in practice for graphs with a few hundred nodes [AS15b]. To reduce the runtime, it is possible to (randomly) select a small subset of the pairwise distances [ST08; VS14; Zha+11].

- Our objective is slightly different. Instead of preserving distances between nodes, we aim at inferring the *popularity* (reflected by radial coordinates) and *similarity* (reflected by angular coordinates) of all nodes [Pap+12]. The reason why connections between vertices exist can be twofold: Either, the two vertices are similar, which holds e. g. for close friends in social networks; or for geographically close autonomous systems (AS) in the Internet graph. On the other hand, a connection may be present due to the popularity of one end vertex: For instance, many people follow Lady Gaga on Twitter; but most are arguably not very similar to her. Embedded shortest path distances lose this information. Our goal is to recover these details using the most likely embedding assuming a hyperbolic nature of the graph in the first place. For this, we use the random network model of Krioukov et al. [Kri+10].

Maximum Likelihood Estimation Embedding of Graphs in Hyperbolic Space. We focus on the last-mentioned approach of maximum likelihood estimation (MLE) algorithms, i.e., we want to find the node coordinates in the network by maximizing the probability that the network is produced by some underlying hyperbolic model. Boguñá et al. [BPK10] were the first to find such an embedding for the Internet graph ($m = 58\,416$ connections between $n = 23\,752$ autonomous systems) in the hyperbolic space. It is impressive that greedy navigation along these hyperbolic coordinates is almost maximally efficient. On average, such greedy paths are just 10% longer than the shortest path found in the network. However, the described method to discover the hyperbolic coordinates "require[s] substantial manual intervention and do[es] not scale to large networks" [Kri+10]. A general algorithm for embedding a network in a hyperbolic space was later presented by Papadopoulos et al. [PPK15]. Their HyperMap algorithm is an approximate maximum likelihood estimation (MLE) algorithm. They demonstrate their algorithm on synthetic networks with $n = 5\,000$ nodes and $m = 20\,000$ edges and a subset of the aforementioned Internet graph with $n = 8\,220$ nodes. The asymptotic runtime was improved in a subsequent paper from $O(n^3)$ to $O(n^2)$ [PAK15]. The authors present no runtime measurements [PPK15; PAK15], but their HyperMap code on our machine requires more than 1.5 hours for a graph of size 2 000 (cf. Section 6.6.3).

Improvements to HyperMap have been suggested. For instance, Wang, Li, Jin, Xiong, and Wu [Wan+16a] use a community detection algorithm for the coarse layout of the nodes; and an MLE to find precise positions. Alanis-Lobato, Mier, and Andrade-Navarro [AMA16] take a different approach by embedding the graph using their Laplacian. Both these methods, however, still require a running time of $\Omega(n^2)$.

Our New Hyperbolic Embedder. We design and implement a new algorithm for computing hyperbolic MLE embeddings of massive networks (Section 6.5).[1] Compared to previous approaches that need $\Omega(n^2)$ runtime, our algorithm runs in quasilinear runtime. To this end, we developed several new techniques. First, we use an analytical approach to compute the expected angles between pairs of high-degree nodes based on their number of common neighbors. In contrast to [PAK15], this approach does not rely on expensive numerical computations, making it fast in practice. The resulting angle distance matrix is then fed to a spring embedder that finds good positions for high-degree nodes in linear time. For small degree nodes, we substantially improve runtime by using the geometric data structure of Bringmann et al. [BKL15] that allows traversing nodes of close proximity in expected amortized constant time.

This enables us to embed significantly larger graphs than before. For instance, we computed on commodity hardware in under one hour a hyperbolic embedding of the Amazon product recommendation network which has over 300 000 nodes. To evaluate the quality of our embedding, we conduct large-scale experiments on 6 250 generated graphs and compare our embedding with the ground truth data (Section 6.6). We observe that in typical metrics like Log-likelihood and greedy routing, our algorithm achieves embeddings that are competitive with the original.

Furthermore, we investigate the performance of two classical methods of embedding graphs in the Euclidean space, namely spring embedders and maximum variance unfolding, when applied to the hyperbolic space (Sections 6.3 and 6.4). We find that both of them can work under some strong assumptions, but generally fail to translate to large real-world graphs.

6.2 Preliminaries

We rely on the notion and results introduced in Chapters 2 and 3. Recall that the distance between two nodes x, y in the hyperbolic plane is

$$\operatorname{dist}(x,y) := \cosh^{-1}(\cosh(r_x)\cosh(r_y) - \sinh(r_x)\sinh(r_y)\cos(\varphi_x - \varphi_y)).$$

In this chapter, we steer away from the step model and consider the *binomial* model of the hyperbolic random graph. There, every two vertices u, v are connected with a probability p depending on their distance:

$$p_{uv} := p(\operatorname{dist}(u,v)) = \Big(1 + \exp(\tfrac{1}{2T} \cdot (\operatorname{dist}(u,v) - R))\Big)^{-1}. \tag{6.1}$$

Here, T is a parameter regulating the importance of the underlying geometry: When $T \to 0$, we again recover the step model, where an edge $\{u, v\}$ is present if and only if $\operatorname{dist}(u,v) \leqslant R$. For $T > 0$, long-range edges are possible (but unlikely). Typically, one

[1] Our code is available at `https://hpi.de/friedrich/research/hyperbolic`.

assumes $0 \leqslant T < 1$. This yields a random graph depending on 4 parameters: n, R (or C), α, and T. Recall further that $\Gamma(v)$ is the set of neighbors of v, and δ refers to the average degree of G.

Further, given a graph $G = (V, E)$ and any mapping from nodes to hyperbolic coordinates $\{r_i, \varphi_i\}_{i=1}^n$, we judge the quality of this embedding using the *Log-likelihood*

$$\mathcal{L}(\{r_i, \varphi_i\}_{i=1}^n \mid G) := \sum_{\{u,v\} \in E} \log(p(\text{dist}(u, v))) + \sum_{\{u,v\} \notin E} \log(1 - p(\text{dist}(u, v))),$$

where the hyperbolic distances dist are taken with respect to the coordinates $\{r_i, \varphi_i\}_{i=1}^n$. To determine the quality of a specific node v, we write

$$\mathcal{L}(v) := \sum_{u \in \Gamma(v)} \log(p(\text{dist}(u, v))) + \sum_{u \notin \Gamma(v)} \log(1 - p(\text{dist}(u, v))), \tag{6.2}$$

so that we have $\mathcal{L}(\{r_i, \varphi_i\}_{i=1}^n \mid G) = \frac{1}{2} \sum_{v \in V} \mathcal{L}(v)$.

Our goal is to devise an algorithm which, given only the network structure (i.e. a list of edges) of a generated hyperbolic random graph, can output hyperbolic coordinates close to the original embedding. As an additional requirement, we would like that the algorithm is robust to noise (i.e. works reasonably well even if the supplied graph was not hyperbolic).

Before presenting our algorithm, we revisit two popular embedding techniques in the Euclidean plane and investigate their performance when applied to the hyperbolic setting.

6.3 Spring Embedder

A heavily used technique to embed graphs in the Euclidean plane is the force-directed method (also called spring embedder) [Kob13], which works roughly as follows. For every edge one assumes an attractive force pulling its end vertices toward each other, and for every pair of vertices one assumes a repulsive force pushing them away. The algorithm starts with some initial drawing (e.g., by choosing random positions) and computes for each vertex the total force acting on it. Then, all vertices are moved by a small step according to these forces. This is iterated until a stable configuration is reached.

In a drawing generated by a spring embedder, edges are usually short and non-adjacent vertices are usually far away from each other. Moreover, the repulsive forces lead to a somewhat uniform distribution of the vertices in the available space. Note that these are exactly the properties we wish to obtain for our embeddings in the hyperbolic plane. It thus seems natural to adapt spring embedders to the hyperbolic geometry, which actually has been done before by Kobourov and Wampler [KW05].

In the following we discuss why the straight-forward way of implementing a spring embedder in the hyperbolic plane does not work in our setting. In Section 6.3.2, we present several adaptations that lead to good results at least for smaller graphs.

6.3.1 Difficulties in the Hyperbolic Plane

To understand the difficulties in the hyperbolic plane, first consider the following artificial situation in the Euclidean plane. Assume v is a vertex only connected to u; and assume the current drawing is stable except that v is far away from u. Now when v moves towards u, it also approaches other vertices it is not connected to, which then push v back towards the direction it came from. This is not a problem, however, as there are usually only few vertices close enough to v for their force to be noticeable. Moreover, vertices on the opposite side of v support the movement towards u.

In the hyperbolic plane, an analogous situation works out differently. The geodesic line between v and u contains points with smaller radius, such that v first moves almost directly towards the origin. In turn, the distance to *all* other nodes decreases, which immediately pushes v back to a position with larger radius. Thus, even bad embeddings are stable.

Judging from the pictures presented by Kobourov and Wampler [KW05], it seems that they did not encounter these issues in their spring embedder. This can be explained by the fact that the radii they use are all rather small, which can be deduced from the presented drawings by observing that the vertices are very well separated from the boundary of the Poincaré disk (which is only true for very small radii). However, for such small radii the hyperbolic plane behaves very similar to the Euclidean plane. We note that using small radii is reasonable for visualizing small graphs using a fish-eye view. However, as the radii in a hyperbolic random graph grow logarithmically with an increasing number of vertices, this is not suitable for our purpose.

6.3.2 Fixing the Spring Embedder

We circumvent the above described problems by treating the two components of a coordinate, i.e., the radius and the angle, more or less independently. More precisely, let u and v be two vertices. Assume without loss of generality that $0 \leqslant \varphi_u < \varphi_v \leqslant \pi$, i.e., increasing φ_u moves u towards v. We define the forces $F_\varphi^u(v)$ and $F_r^u(v)$ acting on the angle and on the radius, respectively, as

$$F_\varphi^u(v) = \begin{cases} 1 - p(\operatorname{dist}(u, v)) & \text{if } \{u, v\} \in E, \\ -p(\operatorname{dist}(u, v)) & \text{otherwise, and} \end{cases}$$

$$F_r^u(v) = \begin{cases} -(1 - p(\operatorname{dist}(u, v))) & \text{if } \{u, v\} \in E, \\ p(\operatorname{dist}(u, v)) & \text{otherwise.} \end{cases}$$

Recall that $p(\mathrm{dist}(u,v))$ denotes the probability that u and v with hyperbolic distance $\mathrm{dist}(u,v)$ are adjacent. The total forces F_φ^u and F_r^u for the vertex u are defined as

$$F_\varphi^u = \sum_{v \in V \setminus \{u\}} F_\varphi^u(v), \quad \text{and} \quad F_r^u = \sum_{v \in V \setminus \{u\}} F_r^u(v).$$

After computing these forces for each vertex $u \in V$, it is moved from (r_u, φ_u) to $(r_u + c_r F_r^u, \varphi_u + c_\varphi F_\varphi^u)$. The values for c_φ and c_r are chosen such that $\max_{u \in V}\{c_\varphi F_\varphi^u\} = \varphi_{\max}$ and $\max_{u \in V}\{c_r F_r^u\} = r_{\max}$ holds for the parameters $\varphi_{\max}$ and $r_{\max}$, which basically ensures that no angle and no radius is changed by more than $\varphi_{\max}$ and $r_{\max}$, respectively.

Note that $F_\varphi^u(v)$ is positive if u and v are adjacent and thus $F_\varphi^u(v)$ contributes to decreasing the angle between u and v (as we assumed $0 \leqslant \varphi_u < \varphi_v \leqslant \pi$), which coincides with the desired behaviour. On the other hand $F_r^u(v)$ is always negative if u and v are connected and positive otherwise. This can have the counter-intuitive effect that v contributes to moving u towards the origin although u and v are connected and v is farther away from the origin than u, which increases the difference between their radii. However, unless u and v have almost the same angle, this actually moves u closer to v (with respect to hyperbolic distance) and thus has the desired effect.

Before we discuss the choices for the parameters $\varphi_{\max}$ and $r_{\max}$, we want to point out some potential issues (and how to fix them). First note that edge probability $p(d(u,v))$ depends on the radius R and on the parameter T, which we both estimate as described in Section 6.5.1. Note that for $T \to 0$ (or for constant T with increasing R), the edge probability converges to the step function, i.e., $p(d(u,v)) \to 1$ if $d(u,v) \leqslant R$ and $p(d(u,v)) \to 0$ otherwise. This has two undesirable effects. First, if u and v are only just close enough (in case they are adjacent) or only just sufficiently far apart (in case they are not connected), then there are no forces that work towards keeping it like that. Second, vertices that are way too close (but not adjacent) or way too far apart (but adjacent) have roughly the same influence as vertices that are only slightly too close or slightly too far apart. Both effects are in particular bad in the early stages of the algorithm. To resolve this issue, we start with the rather large value $T = 0.3R$ in the first iteration and decrease it linearly. More precisely, in the ith iteration out of I iterations in total, we set $T = 0.3R \times \max\{0.05, (1 - i/I)\}$. We note that using a linear dependency on R is reasonable as this leads to roughly the same shape of the function $p(xR)$ for $x \in [0,2]$ independent of R.

As a second potential issue, note that handling the angle independently from the radius leads to huge jumps in terms of hyperbolic distance for vertices with large radius (unless $\varphi_{\max}$ is unreasonably small). Such large jumps are usually undesirable in the Euclidean plane (and do not occur without a large change to at least one coordinate). In the hyperbolic plane, we however allow these large jumps, as forbidding them leads to exactly the problems described in Section 6.3.1.

For the final issue, that also leads us to the parameters $\varphi_{\max}$ and $r_{\max}$, first consider a vertex u moving through the Euclidean plane towards its desired location. If there is another non-adjacent vertex v on its way, then getting close to v leads to potentially large repulsive forces. However, u usually does not get stuck because of these forces as u and v can get around each other by a slight movement in opposite directions orthogonally to the actual movement of u. In the hyperbolic plane, two vertices with the same angle are close to each other no matter what their radius is. Thus, while changing the angle of u to get it to its desired value, u necessarily comes close to every other vertex whose angle is between φ_u and the position u aims for. Thus, the algorithm is much more likely to get stuck in a local minimum than a spring embedder in the Euclidean plane.

We use two strategies to circumvent this issue. The first is to simply allow rather large changes to the coordinates (i.e., use large values for $\varphi_{\max}$ and $r_{\max}$), which makes it possible to jump out of local minima. To make sure that the algorithm still converges to a stable position, we decrease $\varphi_{\max}$ and $r_{\max}$ for later iterations. More precisely, we use $\varphi_{\max} = \pi$ and $r_{\max} = R$ in the first iteration and decrease both values linearly down to 0.

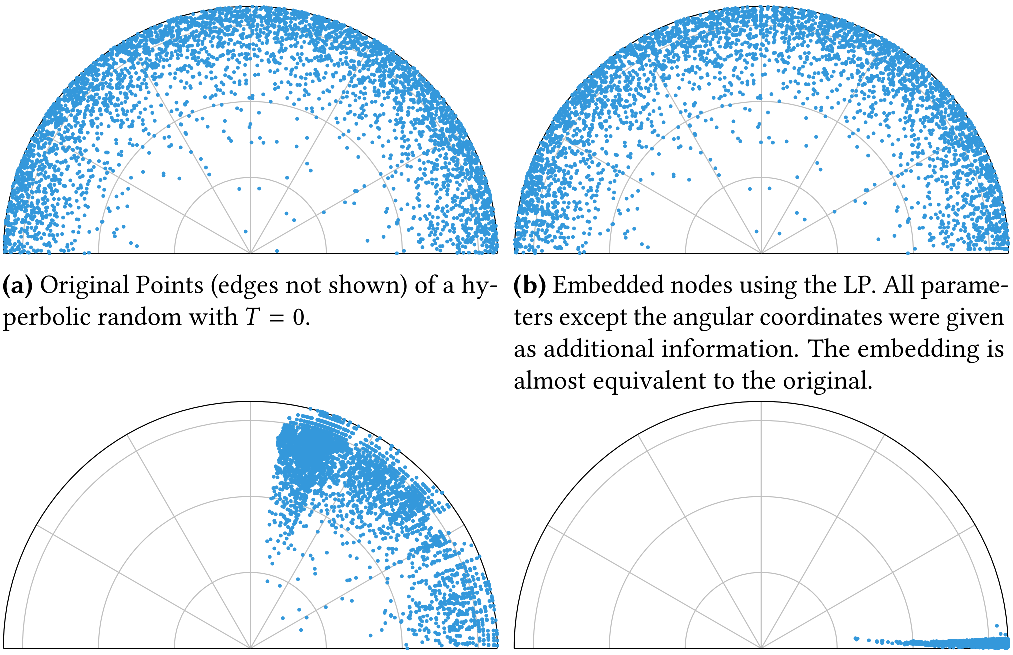
The second strategy is to simulate some kind of velocity. In the above example, this can help u to get past v as the repulsive force of v may slow u down instead of actually pushing it back to where it came from. A simple way to achieve such a notion of velocity is as follows. Assume F_φ^u is the force acting on u in iteration i. Then in iteration $i + 1$, we compute the new force as before and add cF_φ^u to it, where c is 1 in the first iteration and decreases linearly down to 0.2 in the last iteration.

To conclude this section, we have seen that there are several reasons why spring embedders work less well in the hyperbolic plane than in the Euclidean plane. We suggested potential solutions for these problems and we see in Section 6.6 that our spring embedder actually performs reasonably well at least on small to medium sized instances. Moreover, we see in Section 6.5.2 how techniques described above can be reused to embed the core of a larger graph.

6.4 Maximum Variance Unfolding

Another popular method for embedding graphs into the Euclidean plane is maximum variance unfolding (MVU) [WS06]. This is essentially a semidefinite program whose objective function spreads out nodes while using constraints to keep neighbors close together. In the one-dimensional case it is equivalent to an LP.

The use-case in the hyperbolic geometry is similar: Nodes shall have distance $< R$ if they have an edge, and distance $\geqslant R$ otherwise. It is possible to encode this into the



(a) Original Points (edges not shown) of a hyperbolic random with $T = 0$.

(b) Embedded nodes using the LP. All parameters except the angular coordinates were given as additional information. The embedding is almost equivalent to the original.

(c) Embedded nodes using the LP with estimated radial coordinates (See Section 6.5.1). The quality of the LP solution quickly degrades.

(d) Embedded nodes using the LP with all other parameters given. The graph was generated using $T = 0.5$. The embedding is essentially unusable.

Figure 6.1: First phase of the LP. Since nodes are placed in $[0, \pi]$, half of D_n is hidden.

following LP:

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{j=1}^{n} \varphi_j & \\ \text{subject to} & \varphi_i - \varphi_j \leqslant \theta(r_i, r_j), & i, j = 1, \ldots, n, \text{if } \{i, j\} \in E \\ & \varphi_j - \varphi_i \leqslant \theta(r_i, r_j), & i, j = 1, \ldots, n, \text{if } \{i, j\} \in E \\ & 0 \leqslant \varphi_i \leqslant \pi & i = 1, \ldots, n \\ & \varphi_v = 0, & \text{for some starting node } v \end{array}$$

where $\theta(r_i, r_j)$ is the maximal angular distance such that nodes $\operatorname{dist}(i, j) \leqslant R$, i. e.

$$\theta(r_i, r_j) = \arccos\left(\frac{\cosh(r_i)\cosh(r_j) - \cosh(R)}{\sinh(r_i)\sinh(r_j)}\right). \tag{6.3}$$

The LP has a caveat: It is only able to spread nodes on the half circle $[0, \pi]$; since for larger angular coordinates the hyperbolic distances start decreasing again, which is

Algorithm 1 Fast Embedding Algorithm

```
Input: Undirected connected Graph $G = (V, E)$
Output: Hyperbolic coordinates $(r_i, \varphi_i)_{i=1}^n$
Estimate global parameters $n, R, \alpha, T$; and radial coordinates $r_i$ ▷ See Section 6.5.1
Partition nodes into layers such that $v \in L_i \Leftrightarrow \deg(v) \in [2^i, 2^{i+1} - 1]$
Embed Core (all nodes in layers $\geqslant \frac{\log n}{2}$) ▷ See Section 6.5.2
for $i = \frac{\log n}{2} - 1 \ldots 0$ do
    for $r = 1 \ldots \log n$ do
        for all $v \in \bigcup_{j \geqslant i} L_j$ do
            Embed $v$ by optimizing its Log-likelihood ▷ See Sections 6.5.3 and 6.5.4
```

not encodable in the LP. This can be fixed, however, using a small trick: First, embed all nodes on a half-circle with an arbitrary starting node v. Then, pick the node u in the embedding with angular coordinate closest to $\frac{\pi}{2}$; and embed the graph again using u as the starting node.

This yields all nodes that belong in the lower half of D_n: If w has angular distance at least $\frac{\pi}{2}$ from u in the second embedding, we set $\varphi_w = \varphi_w + \pi$ in the first embedding.

This simple method works surprisingly well on generated hyperbolic random graphs that are drawn from the step model, when given all global parameters and radial coordinates, see Figures 6.1a and 6.1b. It is, however, extremely volatile to the quality of the estimated parameters; and it fails completely when used on a real graph or even a graph generated by the binomial model, see Figures 6.1c and 6.1d. The reason is that the LP has a constraint for each edge in the graph: If there is just one long-range edge, the MVU can no longer unfold the graph and all nodes are mapped to an extremely small range of angular coordinates. This behavior persists even after adding different error terms for edges; and we were not able to make this approach work on noisy data.

6.5 The Embedder

Our embedding algorithm is inspired by the Metropolis-Hastings Algorithm from [BPK10]. Algorithm 1 contains a bird's eye view over the whole algorithm. Detailed description of individual steps follow in the next sections.

The algorithm proceeds in three phases: First, it estimates all parameters that are computationally easy to guess. This includes the radial coordinates of all nodes, see Section 6.5.1.

In the second phase, high-degree nodes are embedded by considering their common neighbors. Producing a good initial ordering of nodes in inner layers is crucial for the success of the algorithm since nodes in all subsequent layers are typically placed close to their neighbors in higher layers. This step is described in Section 6.5.2.

In the third phase, the algorithm embeds the rest of the graph layer-wise. To embed a layer L_i, we iterate over all nodes $v \in L_i$. In each iteration, $O(\log n)$ angular coordinates for v are sampled; and v is moved to the position with the best Log-likelihood, see Sections 6.5.3 and 6.5.4. This is repeated $\log n$ times per layer. While this step is similar to HyperMap [BPK10; PAK15; PPK15], we improve upon their algorithm by achieving an amortized polylogarithmic runtime per node as compared to their linear runtime. Our overall algorithm thus runs in $O(n \cdot \operatorname{polylog}(n))$.

6.5.1 Parameter Estimation

To bootstrap the embedding algorithm, the global graph parameters have to be known: The original number of nodes n, the radius R of the disk D_R, the parameter α adjusting the power-law exponent; and the parameter T adjusting the clustering. These values are required for instance for evaluating the probability that two nodes are connected (c. f. Equation (6.1)) which in turn is needed to produce the Log-likelihood. In the following, we give some brief explanations on how each parameter is guessed.

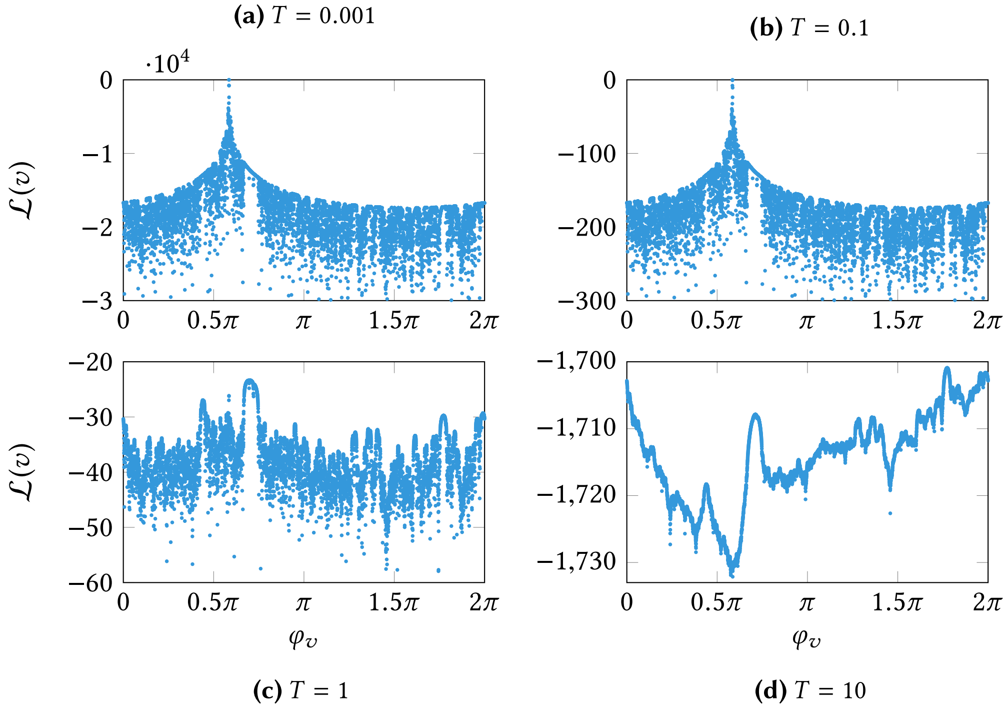
Estimating *n*. Algorithm 1 expects a connected graph as input, since disconnected components can be placed anywhere in the graph as there is no adjacency information.

The hyperbolic random graph, however, does typically not produce a connected graph. For power-law exponents $2 < \beta < 3$, its giant component is of size $\Theta(n)$ [BFM13; BFM14]; and for $\beta \geqslant 3$ the graph breaks up into components of order $o(n)$. Unfortunately, the leading constant of the size of the giant component is yet unknown; and a numerical estimation is hard since it is governed by a non-linear system of equations together with other parameters [BPK10].

We have found experimentally that the majority of nodes missing from the giant component are of degree 0. Surprisingly, the most effective and robust method for estimating the number of these nodes was by simply extrapolating from the number of 1- and 2-degree nodes. Let $\hat{n} \cdot f(k)$ be the number of nodes of degree k, where $\hat{n}$ is the total number of nodes in the input graph. Then, we estimate n simply by setting $n := \hat{n}(1 + \max\{0, 2f(1) - f(2)\})$.

Estimating *α*. The parameter α adjusts the power-law exponent β of the hyperbolic random graph via the functional behavior $\beta = 2\alpha + 1$ [GPP12]. We estimate β from the cumulative degree distribution using the classical algorithm by Clauset et al. [CSN09].

Estimating *T*. Recall that this parameter adjusts the importance of the underlying geometric structure. It has recently been observed, however, that T does not have a big influence on the quality of the embedding [PAK15]. For small T, the fitness landscapes look virtually the same (up to rescaling), see Figures 6.2a and 6.2b. In these cases, the attractive forces of neighbors dominate and the fitness is high whenever a



(c) $T = 1$

(d) $T = 10$

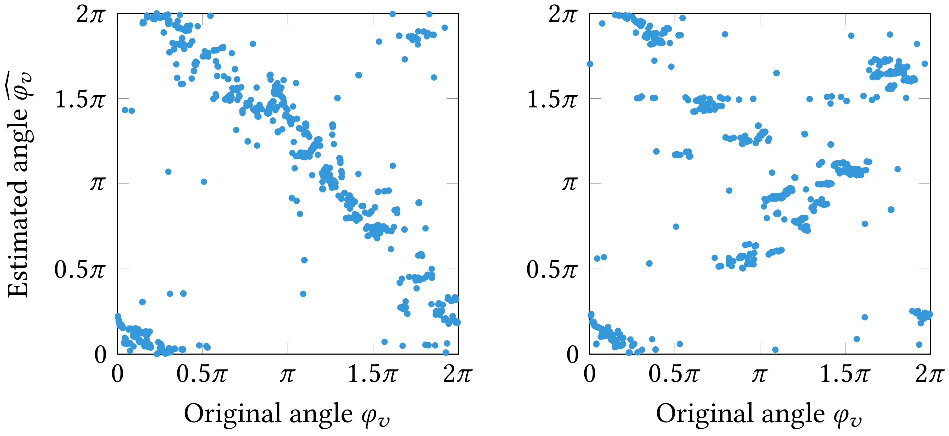
Figure 6.2: Fitness landscape of a node v for different values of T.

node is close to their neighbors. We found that setting T to a small fixed value like 0.1 produces good results.

Increasing T emphasizes non-neighbors. The algorithm then places nodes in an area where there are few non-neighbors, while essentially disregarding the information from neighbors, see Figure 6.2d. Even though there is a short intermediate transition of the fitness landscape as can be seen in Figure 6.2c, our experiments suggested that setting T to a small value—even if the graph was generated using a large T—produced cleaner embeddings. For instance, Figure 6.3 contains the original vs. embedded angle of two embeddings where one has been computed using the original value of $T = 0.7$ and the other with $T = 0.1$. The algorithm performs better when using $T = 0.1$, even though the original T that has been used to generate the graph was large.

Estimating R and r_i. We estimate these values using the above determined parameters. Good analytical estimates have been derived in previous work [BPK10]:

$$R = 2\log\left(\frac{4n^2\alpha^2 T}{|E|\cdot\sin(\pi T)(2\alpha-1)^2}\right), \quad r_i = \min\left\{R, 2\log\left(\frac{2n\alpha T}{\deg(i)\cdot\sin(\pi T)(\alpha-\frac{1}{2})}\right)\right\}.$$



(a) Our algorithm sets $T = 0.1$ for any input and is able to reconstruct the original ordering of nodes fairly well.

(b) Choosing $T = 0.7$ and scanning the whole range $[0, 2\pi)$ for the best Log-likelihood of a node results in a much worse embedding.

Figure 6.3: Original angular coordinates vs. embedded angular coordinates for a generated hyperbolic random graph with $T = 0.7$.

6.5.2 Embedding the Core

Laying out large-degree nodes (also called the *core* of the graph) is critical for the overall performance of the embedding. We consider all nodes v with radial coordinates $r_v < R/2$ to be in the core, of which there are $\Theta(n^{1-\alpha})$ w. h. p., see Section 4.4. If the node ordering of the core is roughly correct, the algorithm yields excellent embeddings. One the other hand, if the core was embedded poorly, the remaining steps can not salvage this. We therefore put considerable care into embedding the core correctly.

HyperMap [PPK15] uses the number of common neighbors of large degree nodes to infer their relative angles: For two nodes u, v they determine $c_{uv} = |\Gamma(u) \cap \Gamma(v)|$ and numerically compute the angle $\varphi(c_{uv}, r_u, r_v)$ that maximizes the likelihood that the nodes u, v have c_{uv} common neighbors. This approach is robust since the number of common neighbors of large degree nodes is tightly concentrated around its expected value. Determining the likelihood numerically, however, is a computationally expensive operation.

To overcome this, we analytically derive an approximate expression for the relative angle of two nodes up to constant factors. Using this, we present a spring embedder that embeds the core based on the estimated pair-wise angle differences.

Estimating the Angle-Differences. To estimate the relative angle between two nodes, we use their inferred radial coordinates and the number of their common neighbors. We perform this computation in the step model; however, we have experimentally found that our results hold up well in the binomial model.

Let u, v be the two nodes whose (expected number of) common neighbors we wish to compute. They have radii r_u and r_v, respectively, and a relative angle of $\Delta\varphi_{u,v}$. W. l. o. g., assume that $r_u \leqslant r_v$. Consider now a third node w. We compute the probability that w is connected to both u and v. Under the assumption that $r_u + r_w \geqslant R$ and $r_v + r_w \geqslant R$, we know from Lemma 3.2 that this only holds if

$$\begin{aligned}\Delta\varphi_{u,w} &\leqslant 2e^{\frac{1}{2}(R-r_u-r_w)}(1+\Theta(e^{R-r_u-r_w})), \qquad \text{and}\\ \Delta\varphi_{v,w} &\leqslant 2e^{\frac{1}{2}(R-r_v-r_w)}(1+\Theta(e^{R-r_v-r_w})). \end{aligned} \tag{6.4}$$

Assume $r_v + r_w \geqslant R$ does not hold. In this case, the distance between v and w is obviously at most R and thus they are connected. Moreover, note that in this case the right hand side of the above formula increases with increasing R and thus the inequality is satisfied for any angle $\Delta\varphi_{v,w}$ if R is sufficiently large. Thus, under the assumption that R is sufficiently large, we may use Equation (6.4).

Observe now that for large enough radii r_w, the node w is not connected to either u or v (unless $\Delta\varphi_{u,v} \leqslant O(\frac{1}{n})$). On the other hand, when $R - r_v - r_w = \Omega(1)$, w is connected with constant probability to both u and v. Thus, depending on the radius r_w, there is a "good" fraction of the angular coordinates $[0, 2\pi)$ where w will be connected to both nodes, and a "bad" fraction where it will be connected to only one or neither of u, v. We call the probability to be connected to both nodes $p_g(r_w)$.

As discussed, $p_g(r_w) = 1 \Leftrightarrow r_w = R - r_v \pm \Theta(1)$. We label this critical value of r_w with r_1. On the other hand, $p_g(r_w) = 0$ holds when $\theta(r_u, r_w) + \theta(r_v, r_w) \leqslant \Delta\varphi_{u,v}$, since then there is no possible angle for φ_w where it is connected to both nodes u, v, see Equation (6.3). The critical value r_0 for which this number becomes positive is when $\theta(r_u, r_w) + \theta(r_v, r_w) = \Delta\varphi_{u,v}$ and thereby

$$\begin{aligned}\Delta\varphi_{u,v} &= 2e^{\frac{1}{2}(R-r_u-r_0)}(1\pm\Theta(e^{R-r_u-r_0})) + 2e^{\frac{1}{2}(R-r_v-r_0)}(1\pm\Theta(e^{R-r_v-r_0}))\\ &= \Theta(1)\cdot e^{\frac{1}{2}(R-r_u-r_0)}.\end{aligned}$$

Solving for r_0, this holds whenever $r_0 = \min\{R, R - r_u - 2\log(\Delta\varphi_{u,v}) \pm \Theta(1)\}$.

For values $r_1 \leqslant r_w \leqslant r_0$, the regions in which w connects to u, v both increase as in Equation (6.4). Thus, the intersection of these regions increases as $p_g(r_w) \sim e^{-r_w/2}$. To determine the function up to constants, we set

$$1 = p_g(r_1) = A \cdot e^{-r_1/2} + B, \qquad \text{and} \qquad 0 = p_g(r_0) = A\cdot e^{-r_0/2} + B.$$

Solving this system of equations, we obtain that $p_g(r_w) = \Theta(1)\cdot(e^{\frac{1}{2}(r_1-r_w)} - e^{\frac{1}{2}(r_1-r_0)})$. Thus, we may compute the probability that an arbitrary node is connected to both u

and v using the cumulative distribution function and p_g. We thereby have

$$\begin{aligned}
\Pr[w \sim u, v] &= \int_0^R \rho(r) \cdot p_g(r)\,\mathrm{d}r \\
&= \Pr[r_w \leqslant r_1] + \Theta(1) \cdot \int_{r_1}^{r_0} e^{\alpha r - \alpha R} \cdot (e^{\frac{1}{2}(r_1 - r)} - e^{\frac{1}{2}(r_1 - r_0)})\,\mathrm{d}r \\
&= e^{\alpha r_1 - \alpha R} + \Theta(1) \cdot \left[e^{\alpha r - \alpha R} \cdot (\tfrac{1}{\alpha - \frac{1}{2}} e^{\frac{1}{2}(r_1 - r)} - \tfrac{1}{\alpha} e^{\frac{1}{2}(r_1 - r_0)}) \right]_{r_1}^{r_0} \\
&= \Theta(1) \cdot e^{\alpha r_0 - \alpha R + \frac{1}{2}(r_1 - r_0)}.
\end{aligned}$$

Hence, the expected number of common neighbors of u and v is

$$c_{uv} = \Theta(1) \cdot \exp(\tfrac{R}{2} + (\tfrac{1}{2} - \alpha) r_u - \tfrac{1}{2} r_v) \cdot \Delta\varphi_{u,v}^{1-2\alpha}.$$

To find the angle $\varphi(c_{uv}, r_u, r_v)$ maximizing the Log-likelihood in the step model, we observe that the number of common neighbors of u, v is a binomial random variable: There exists a set $S \subseteq D_n$ in which each node is connected to both u, v and each node in $D_n \setminus S$ connected to at most one of u, v. Since the maximum likelihood estimator for binomial random variables is the number of successes divided by the number of trials, we obtain the maximum likelihood for $\Delta\varphi_{u,v}$ by rearranging above equation.

$$\varphi(c_{uv}, r_u, r_v) = \Theta(1) \cdot c_{uv}^{\frac{1}{1-2\alpha}} \cdot \exp(-\tfrac{1}{2} r_u + (\tfrac{1}{2-4\alpha})(r_v - R)).$$

To obtain actual values for $\Delta\varphi_{u,v}$ we first simply omit the constant factor hidden by $\Theta(1)$ in the above expression. To obtain reasonable angles, observe that the largest angle should likely be π. To obtain this, one can simply rescale all values of $\varphi(c_{uv}, r_u, r_v)$ with the same constant factor such that the maximum is π. As this is prone to errors if outliers exist, we instead scale all angles by the same constant such that their median is $\pi/2$. Angles that are larger than π after this scaling are then set to π. Preliminary experiments showed that using the logarithm of the above expression for initially computing $\Delta\theta(u, v)$ (before the scaling) improved the robustness of our algorithm.

Embedding According to the Estimated Angles. In this section, we assume that we know the desired angle $\Delta\varphi_{u,v}$ between any pair of vertices u and v in the core. Our goal is to assign an angle to each vertex that realizes these differences as good as possible. To this end, we use a 1-dimensional spring embedder (see Section 6.3 for a short introduction to spring embedders) that basically works as follows. We start with random initial angles. Then in each iteration, we consider every pair u, v of vertices. If the the current angle between u and v is larger than $\Delta\theta(u, v)$ we get an attractive force, otherwise we get a repulsive force. W. l. o. g., we assume $0 \leqslant \varphi_u < \varphi_v \leqslant \pi$ (the other cases work symmetrically). Moreover, let $\mathrm{err}(u, v) = \varphi_v - \varphi_u - \varphi(c_{uv}, r_u, r_v)$.

The force $F_u(v)$ acting on u due to v is then given by

$$F_u(v) = \begin{cases} -\mathrm{err}(u,v)^2 & \text{if } \mathrm{err}(u,v) \leqslant 0, \\ \mathrm{err}(u,v)^2 & \text{if } 0 < \mathrm{err}(u,v) \leqslant \frac{\pi}{2}, \textit{and} \\ (\pi - \mathrm{err}(u,v))^2 & \text{if } \frac{\pi}{2} < \mathrm{err}(u,v) \leqslant \pi. \end{cases}$$

To interpret this formula, first note that $\mathrm{err}(u,v) < 0$ holds if the current angle is too small. Thus, $F_u(v)$ is negative (pushing u away from v) and the strength of the force increases quadratically in the distance to the desired angle. Conversely, if the current angle is too large, we get a repulsive force increasing quadratically in the distance to the desired angle as long as this distance is at most $\pi/2$. For larger distances, the strength of the force actually decreases again. This has the following reason. Imagine the extreme case that u and v have angle π between them but actually want to have a very small angle. Then it does not really matter whether the angle of u increases or decreases as it comes closer to v not matter what. Thus, we do not really want a very strong force in one of the two directions, which is the reason why we decrease the strength of attractive forces when $\mathrm{err}(u,v)$ becomes very large.

Similar to Section 6.3, the total force acting on u is defined as

$$F_u = \sum_{v \in V \setminus u} F_u(v)$$

and the new angle of u is obtained by setting $\varphi_u = \varphi_u + cF_u$. The value for c is again chosen such that the maximum step size does not exceed a parameter $\theta_{\max} := \max_{u \in V}\{cF_u\}$.

Due to the 1-dimensionality of this spring embedder, we encounter a similar problem as for the hyperbolic spring embedder in Section 6.3: to move a vertex u to a specific position, it necessarily has to pass through all vertices in between and there is no second dimension that could be used to get around them. This leads to strong repulsive forces hindering u in getting to the desired position and we observed in our experiments that the algorithm often gets stuck in a local minimum. As before, we use velocity and a rather large step size $\theta_{\max}$ to circumvent this issue. Preliminary experiments showed that we obtain good results using the following parameters. We set $\theta_{\max} = 0.55\pi$ in the first iteration, decreasing it linearly down to 0 in the final iteration. For the velocity assume F_u is the force from iteration i. Then we add cF_u to the force in iteration $i + 1$ where c is 1 in the first iteration and linearly decreases down to 0.5 in the last iteration. Since there are $\Theta(n^{1-\alpha})$ nodes in the core [FK15a], the total runtime of the spring embedder is $O(k \cdot n^{2-2\alpha})$, where k is the number of iterations. Choosing $k = O(n^{2\alpha-1})$, we achieve a runtime of $O(n)$.

The performance of this algorithm depends on the randomly chosen initial angles.

To be able to compare core embeddings, we define a score S as

$$S = \sum_{u \in V} \sum_{v \in V \setminus u} |F_u(v)|$$

A smaller score then indicates a better embedding. We define s_{opt} as the score that is obtained when the spring embedder is initialized with the original coordinates. We then say that a core embedding is *good*, if it has a score $s \leqslant 1.2 \cdot s_{\text{opt}}$. Each graph thus has a certain probability that the core embedding is good, depending on the randomly chosen initial positions. To further increase the probability of getting a good embedding for the core, we run the spring embedder 5 times with different initial angles and use the best result, which boosts the probability of getting a good embedding to 95% for the *worst* of over 6 000 randomly generated hyperbolic random graphs (see Section 6.6 for the experimental setup). This suggests that the spring embedder is rather robust, i.e. we rarely encounter initial drawings that lead to bad results.

6.5.3 Computing the Log-likelihood efficiently

A further key ingredient to achieve a quasilinear runtime is to improve the runtime of the Log-likelihood computation $\mathcal{L}(v)$. Recall that $\mathcal{L}(v)$ was defined as

$$\mathcal{L}(v) := \sum_{u \in \Gamma(v)} \log(p_{uv}) + \sum_{u \notin \Gamma(v)} \log(1 - p_{uv}),$$

see Equation (6.2). By a naive implementation, one needs $\Omega(n)$ time to compute the Log-likelihood of a single node and thus at least $\Omega(n^2)$ for the whole graph. A more careful inspection, however, allows for a significant speedup.

First, observe that the total number of edges in a hyperbolic random graph is of order $O(n)$ in expectation; so the term $\sum_{u \in \Gamma(v)} \log(p_{uv})$ can be computed in amortized constant time. To speed up the computation of the second summand, we observe that the term $\log(1 - p_{uv})$ is very close to 0 whenever $\text{dist}(u, v) \gg R$, since

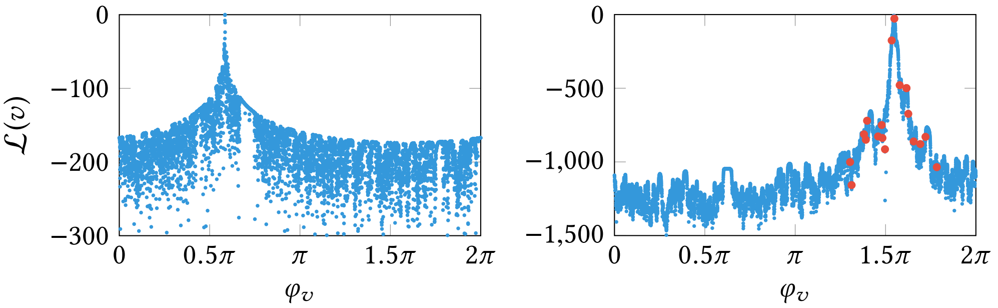
$$p_{uv} := (1 + \exp(\tfrac{1}{2T}(\text{dist}(u, v) - R)))^{-1} \approx \exp(-\tfrac{1}{2T}(\text{dist}(u, v) - R)),$$

and by a Taylor series for $p_{uv} \to 0$ we get

$$\log(1 - p_{uv}) = -p_{uv} - O(p_{uv}^2) \approx -\exp(-\tfrac{1}{2T}(\text{dist}(u, v) - R)).$$

This implies that non-neighbors that are far away from v barely contribute to its Log-likelihood. If, on the other hand, $\text{dist}(u, v) \ll R$, we have by Lemma 2.3

$$p_{uv} \approx 1 - \exp(\tfrac{1}{2T}(\text{dist}(u, v) - R)) \to 1, \tag{6.5}$$



(a) Exemplary fitness landscape for a node v with 3 neighbors. Both methods for computing the fitness landscape exhibit no visible difference in the plot.

(b) Fitness landscape of a node v and the coordinates at which the efficient algorithm samples the fitness. Red points indicate the sampled angles.

Figure 6.4: Fitness landscape of a node v computed with the efficient algorithm.

and thus

$$\log(1 - p_{uv}) \approx \log(1 - (1 - \exp(\tfrac{1}{2T}(\mathrm{dist}(u, v) - R)))) = \tfrac{1}{2T}(\mathrm{dist}(u, v) - R).$$

Thus, it suffices to take into account non-neighbors with low distance from u while either ignoring or coarsely approximating the influence of far away non-neighbors on the Log-likelihood. To this end, we implemented the geometric data structures introduced by Bringmann et al. [BKL15]. These were originally used to generate hyperbolic random graphs in linear time by partitioning the disk D_n into suitably sized cells. To compute the Log-likelihood of a node, one can then compare it directly with nodes in neighboring cells (that have a big influence on the Log-likelihood); while averaging over all nodes in far away cells. As shown in [BKL15], this runs in amortized time $O(1)$. We need an extra $O(\log n)$ factor to update the cells whenever a node is moved during the embedding algorithm.

Figure 6.4a shows the fitness landscapes of a node v; computed once via the classical exact $\Omega(n)$ method, and once using our amortized $O(\log n)$ method. Both methods exhibit no visible differences in the plot; and we found that the relative error made by the fast Log-likelihood computation is $\leqslant 1.0025$ at all coordinates except one, where it was $\leqslant 1.02$.

6.5.4 Finding the Optimal Angle

To find a good angular coordinate for a node v, previous algorithms typically scan the whole range $[0, 2\pi)$ at resolution $\frac{2\pi}{n}$; and evaluate at each angle the Log-likelihood $\mathcal{L}(v)$. This incurs another factor $\Omega(n)$ on the overall runtime.

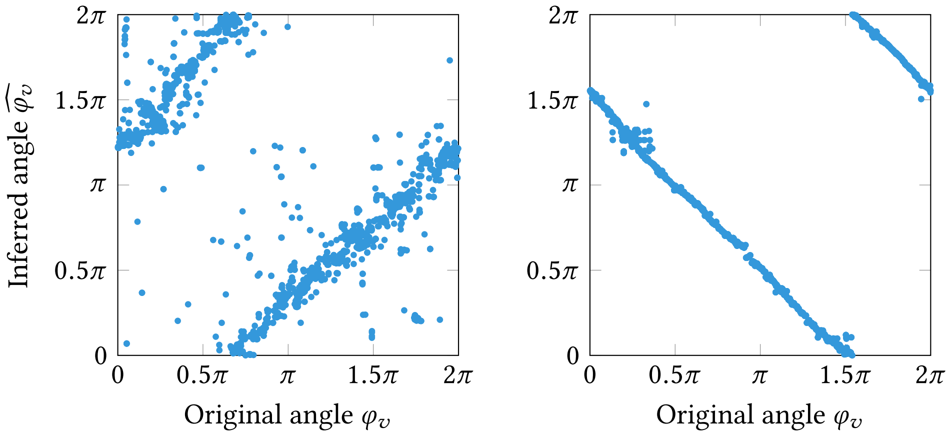


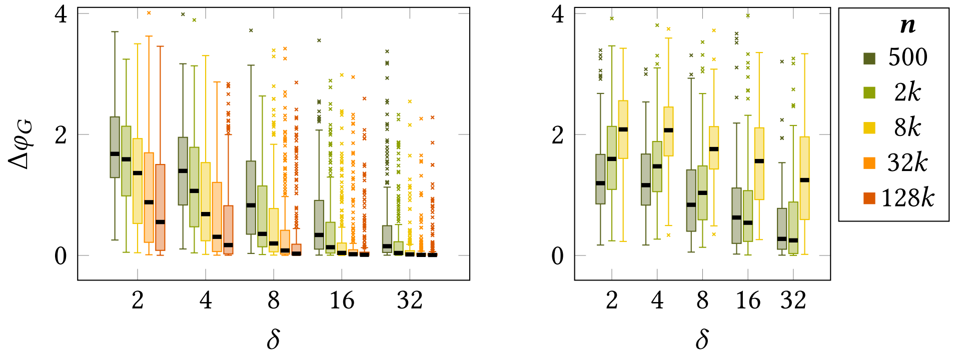
Figure 6.5: The plots correspond to embeddings with average squared deviation $\Delta\varphi_G = 0.44$ (left) and $\Delta\varphi_G = 0.01$ (right). For each vertex v the plot contains one point with x-coordinate φ_v (angle of v in the original embedding) and y-coordinate $\widehat{\varphi_v}$ (angle in the computed embedding). The embedding is considered good if the plot resembles the identity function $f(x) = x$ up to cyclic shift and rotation.

To save on this, we sample only few points around a region where a node has their maximum likelihood. To determine this region, we observe that the coarse likelihood landscape for a node v (for small T) is governed by the position of v's neighbors. Furthermore, neighbors with large radii have a larger influence on the fitness landscape, as the hyperbolic distance to these nodes increases more quickly than to neighbors with small radial coordinates. Hence, v needs to be placed close to its embedded low-degree neighbors.

Ignoring non-neighbors for now, we achieve this by computing a weighted average over the angles of all neighbors of v. Let $u_1, \ldots, u_k$ be the embedded neighbors of v. Then, v's angle is computed as follows.

$$\varphi_v = \arctan\left(\frac{\sum_{i=1}^{k} \exp(r_{u_i}) \cdot \sin(\varphi_{u_i})}{\sum_{i=1}^{k} \exp(r_{u_i}) \cdot \cos(\varphi_{u_i})}\right)$$

To take non-neighbors into consideration, we then randomly sample $O(\log(n))$ points around this angle and use the one with the smallest Log-likelihood. Figure 6.4b shows the fitness landscape of an exemplary node u, as well as the randomly sampled angles. As can be seen, the heuristic typically finds good candidates whose angles are close to the optimal angle.



(a) Our main algorithm.

(b) Our hyperbolic spring embedder.

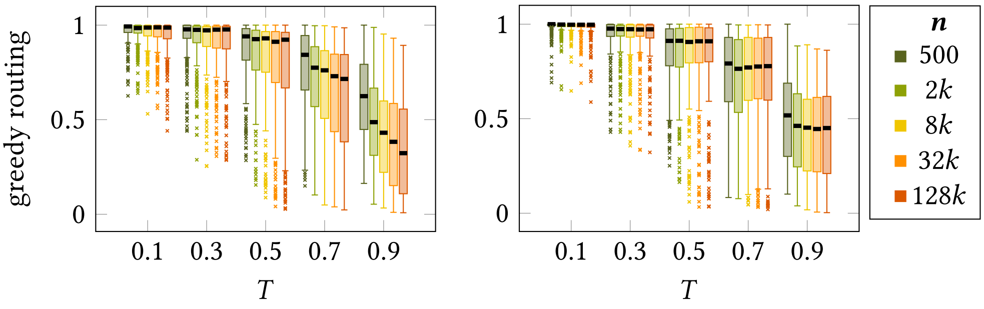
Figure 6.6: Each data point in the box plot represents the value of $\Delta\varphi_G$ for a single graph G (y-axis) depending on the average degree a (x-axis). Smaller values are better. The graphs are grouped by their sizes.

6.6 Experiments

To evaluate the quality of our algorithm, we sampled 10 different hyperbolic random graphs for every combination of the following parameters: $\alpha \in \{0.55, 0.65, 0.75, 0.85, 0.95\}$, $T \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\delta \in \{2, 4, 8, 16, 32\}$, $n \in \{500, 2\,000, 8\,000, 32\,000, 128\,000\}$. This results in a total of 6 250 graphs. For each of these graphs, we computed the following statistics: Log-likelihood, success ratio of greedy routing and the average squared deviation in the original angle vs. estimated angle plot. We present the most insightful statistics in standard box plot form: A box contains 50% of all data points closest to the median; which is marked black. The size of the box is called interquartile range (IQR). Data points are considered outliers if they have distance more than 1.5× IQR to the box. The whiskers depict the closest data point to the box that is not an outlier.

6.6.1 Quality

A popular way to judge whether an embedding makes sense is to plot the embedded angular coordinates against the original generated coordinates. If the result resembles a straight line (that might have a cyclic shift), then the relative ordering of nodes has been reconstructed well in the embedding. Two examples for such plots are shown in Figure 6.5. To allow for comparisons that scale to a large amount of graphs, we derive the following quality measure. For a vertex v let $\Delta\varphi_v$ be the quadratic difference between φ_v in the original embedding and φ_v in the computed embedding. For a graph $G = (V, E)$, the value $\Delta\varphi_G = \sum_{v \in V} \Delta\varphi_v / n$ then describes the average squared deviation in G.



(a) Our main algorithm.

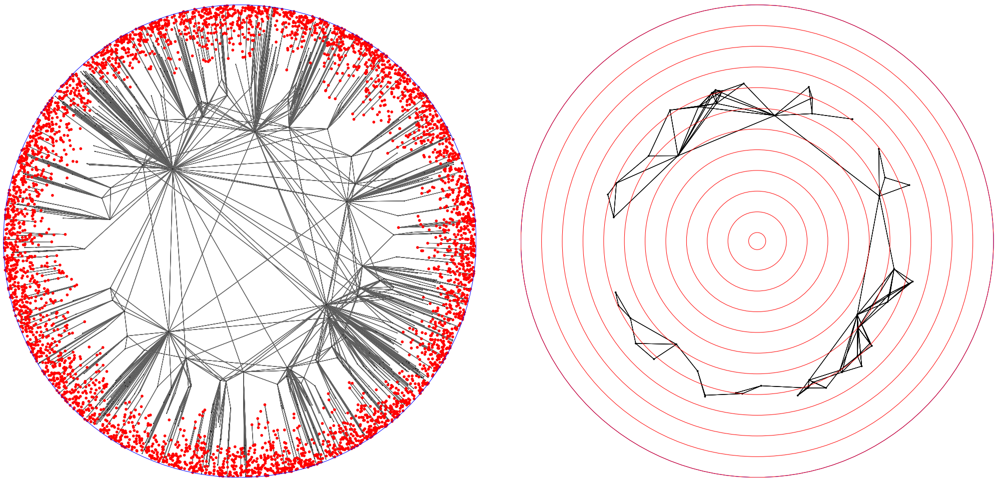
(b) Originally generated embeddings.

Figure 6.7: The success ratio of greedy routing (x-axis) depending on the value of T (y-axis) grouped with respect to the number of vertices (colors).

The box plot in Figure 6.6a plots $\Delta\varphi_G$ against the average degree δ; grouped by the size of the graph. In this and all other plots, we average over all parameters that are not explicitly grouped by. Observe that $\Delta\varphi_G$ is high if the average degree is small, as the few existing edges are not sufficient to uniquely determine the single best embedding. Thus, several embeddings may be equally good. In fact, for small δ, our algorithm finds an embedding with a Log-likelihood very close to the Log-likelihood of the original embedding (the mean values for large graphs with $\delta = 2$ are $-2.39 \cdot 10^5$ for the embedding and $-2.19 \cdot 10^5$ for the original, respectively, while the corresponding values for $\delta = 16$ are $-1.78 \cdot 10^6$ and $-1.16 \cdot 10^6$). For an average degree of 8, the mean value for $\Delta\varphi_G$ of all medium sized ($n = 8\,000$) and large ($n = 128\,000$) graphs is 0.2 and 0.03, respectively. For comparison, note that the plots in Figure 6.5 correspond to graphs with values 0.01 and 0.44. Also note that our algorithm performs particularly well on large graphs, which was the goal we aimed for.

For comparison with the spring embedder, see Figure 6.6b. As the spring embedder is too slow on larger graphs, we only ran the experiments on graphs up to size $n = 8\,000$. Note that the quality of the spring embedder decreases for increasing graph size. In contrast, it performs comparatively well on small graphs (and in some cases actually better than our main algorithm) while it is heavily outperformed on the medium sized graphs. Hence, the spring embedder is a reasonable option for graphs with up to 1 000 vertices, while our main algorithm is the better option for larger graphs.

A quality measure previously used for hyperbolic embeddings is the success ratio of greedy routing. Figure 6.7a shows this ratio for the embeddings generated by our algorithm depending on the parameter T, grouped by the size of the graph. Observe that the ratio is close to 100% for small values of T but drops significantly for larger values. This is unfortunate as real world graphs are considered to have fairly large values of T, e.g., $T = 0.7$ was used for the embedding of the Internet graph [BPK10]. Though this particular embedding allows greedy routing with success ratio 97%, the



(a) When the average degree δ is small for the graph generation, most nodes (shown red) are not part of the giant. It becomes hard to infer the original number of nodes based on the few that remain in the giant component.

(b) When $\beta \to 3$, all nodes are pushed away from the center of D_R. The core thus attains a sparse, circular structure, for which the algorithm is not tailored.

Figure 6.8: Hard corner cases for the embedding algorithm.

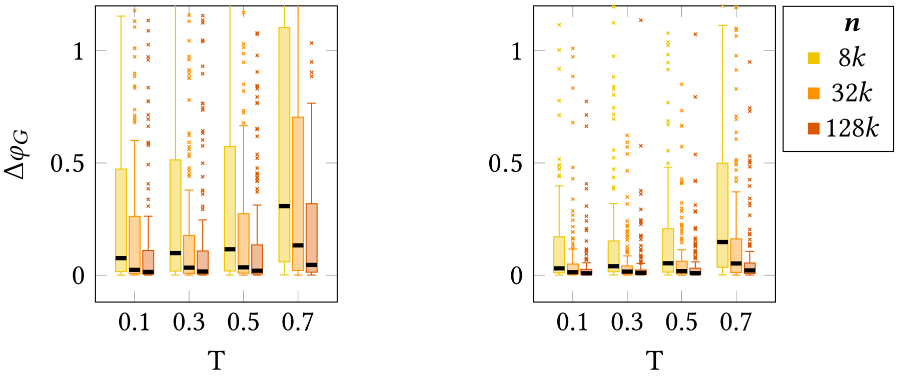
ratios of around 80% we obtain for $T = 0.7$ seem to reflect the typical behavior of random hyperbolic graphs much better; see Figure 6.7b.

Note that these observations imply that maximizing the Log-likelihood will not necessarily lead to the desired result in terms of greedy routing. Conversely, optimizing the embedding for greedy routing will probably not lead to an embedding that is close to the original embedding of a hyperbolic random graph. Hence, we do not see the low success ratios our embeddings achieve for large T as a weakness but rather as a strength as it matches the behavior of the original embedding.

6.6.2 Further Work

Even though the algorithm produces meaningful embeddings overall, we observed that certain parameter combinations lead to bad embeddings more often than others. In particular, this happens when (i) T is close to 1, (ii) δ is small or (iii) β is close to 3. Case (i) poses an inherent problem: If T was chosen large during the graph generation, random edges become more prevalent while the geometry plays a background role. Thus, it is natural that it is hard to embed these graphs meaningfully.

The other two cases are less intuitive. In case (ii), the average degree is small. This leads to the generated graph having a small giant component. For instance, when $\delta = 2$, $\beta = 2.1$ and $T = 0.1$, a generated graph with 5 000 nodes only has 800 in its giant



(a) Evaluation using all data.

(b) Evaluation excluding degenerate cases.

Figure 6.9: Reevaluation of the experiments for embeddings, where hard cases $\delta = 2$ and $\beta = 2.9$ were discarded, see Section 6.6.2. This reveals that the embeddings of non-degenerate instances are of high quality, and most bad embeddings stem from hard corner cases.

component, see Figure 6.8a. Since only the giant is fed to the embedding algorithm, this results in a severe reduction of information. Consequently, the algorithm infers wrong parameters R, r_i which leads to a significantly different embedding than the ground truth. Note that when the algorithm is supplied the correct values of R, r_i, it produces again embeddings of high quality. As of now, however, we are not aware of a robust method that can infer these parameters in this degenerate case.

In the case (iii) when $\beta \to 3$, a different problem arises: Increasing β corresponds to shifting all nodes away from the center. Consequently, the core has a ring-like shape: Most high-degree nodes are only connected to few other nodes in the core; which results in few common neighbors, see Figure 6.8b. As the core embedder in Section 6.5.2 relies on a dense matrix of common neighbor information, it fails to produce a good initialization which leads to a bad embedding in the end. While rings can in principle be embedded well with classical spring embedders; those fail for dense cores. Thus, a refined core embedding algorithm that switches between these methods could improve upon the quality in this case. We argue, however, that this case is degenerate since such a ring-like structure most likely does not appear in the core of real-world graphs.

Figure 6.9 shows the performance of our algorithm on non-degenerate cases. It is apparent that on large graphs, our algorithm performs extremely well if the generation parameters are non-degenerate. On the other hand, this shows a clear road map on how the algorithm can be improved to achieve even better results overall.

6.6.3 Runtime

A key contribution of our algorithm is its significant improvement on the runtimes compared to previous approaches. We performed runtime experiments on commodity hardware, i.e. a 2.7 GHz Core i7 with 8 GB of RAM. Figure 6.10 shows the runtimes depending on n. Note that compared to available algorithms these are fairly quick: Graphs of size 20 000 can be embedded in under two minutes. We even embedded graphs of size 330 000 in under one hour, see Section 6.6.4. For comparison, the reference algorithm HyperMap [PAK15; PPK15] needs over 1.5 hours for a graph of size 2 000.

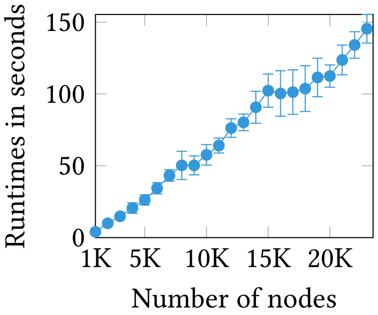


Figure 6.10: Runtimes for the embedding algorithm. Error bars show the standard deviation.

6.6.4 Embedding a Real-World Graph

As a proof of concept, we embed the Amazon product recommendation network [YL15]. It has $n = 334\,863$ nodes with an average degree of 5.53, the degree distribution follows a power-law with exponent $\beta = 3.6$ and the average clustering coefficient is 0.4. The nodes represent products available on Amazon, and an edge $\{u, v\}$ is present if product u is recommended together with product v. Product categories define ground truth communities in this graph.

The embedding took 50 minutes on a single 2.7 GHz Core i7. While the number of nodes is too large to visually inspect the whole graph, we have plotted the nine largest communities in Figure 6.11. Most nodes belonging to a single community are mapped close together; which suggests that the hyperbolic embedding might be a useful tool in discovering hidden communities in a large network.

6.7 Conclusion

We designed and implemented a new algorithm for embedding complex networks into the hyperbolic plane. Connected nodes are typically placed close-by, whereas disconnected nodes have a large hyperbolic distance. Compared to previous algorithms, our algorithm is the first to achieve a quasilinear runtime. This enables us to embed significantly larger graphs than before. Further, as we experimentally validated, our algorithm produces embeddings close to the ground truth; especially when either the number of nodes n or the average degree δ is large. This is evidenced by the fact that the average angular error for embedded nodes becomes as small as 0.03 for $n = 128\,000$ and $\delta = 8$.

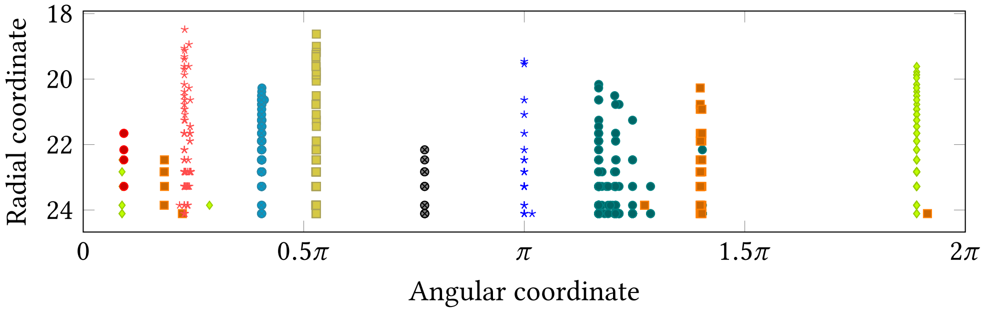


Figure 6.11: The nine largest communities in the amazon product recommendation network. For clarity, only the 893 nodes that belong to a single community are shown. Nodes belonging to the same community are typically placed nearby, even though the embedding algorithm had no knowledge of the ground truth communities.

Our work was focused on presenting a proof of concept. The next logical step is to use this embedding algorithm to learn new information about the behavior of real-world graphs. Such embeddings were used before to produce efficient greedy routing [BPK10], but other applications come to mind. For instance, a geographical representation of nodes opens new possibilities for finding clusters [Wan+16b]. In fact, a different embedding algorithm reverses this idea by first computing clusters in the graph and then inferring node positions based on the found clusters [Wan+16a].

A different direction is to use the embedding for visualization of massive networks. In fact, the hyperbolic plane was often used for visualization purposes in the past [LRP95; LR96; Mun98; WR02; VS14]. Due to their size, classical methods typically struggle with finding a visual representation of the network that still conveys meaningful information. While the currently produced plots still only work for medium-size graphs before they become too cluttered, this may be easily improved by e. g. (i) hiding "unimportant" edges as in [BPK10], or (ii) providing a Focus+Context-like graph browser that allows for changing the coordinate origin as in [LRP95; LR96]. Such tools magnify different regions of the graph while still placing the inspected nodes into the general graph context.

Finally, graph algorithms on hyperbolic random graphs that require knowledge of the geometrical representation can be invoked once we obtain the graph embedding. For instance, it has been shown that on hyperbolic random graphs, structures such as matchings and independent sets may be found more efficiently than on general graphs [BFK16].

7 Conclusion & Outlook

In this thesis, we contributed to the understanding of the hyperbolic random graph model by analyzing various structural properties. We showed that cliques of polynomial size appear when the power law exponent is $2 < \beta < 3$, and of logarithmic size when $\beta \geqslant 3$. Moreover, we analyzed the diameter of hyperbolic random graphs and found that it is at least logarithmic, and at most polylogarithmic when $2 < \beta < 3$. For the case $\beta > 3$, we even computed a tight result on the diameter of $\Theta(\log n)$.

From an algorithmic perspective, we presented efficient algorithms for finding cliques in this model. Further, we designed and implemented an embedding algorithm that finds hyperbolic coordinates for a given graph using only the graph structure.

With most fundamental structural properties of hyperbolic random graphs settled, there are multiple possible directions for future work. In order to advance the understanding of complex networks, one has to relate the structural properties of hyperbolic random graphs to their real-world counterparts. As of right now, it seems that many structural properties like power law degree distribution and clustering are similar. It is crucial, however, to falsify the model by finding properties in which hyperbolic random graphs differ from real-world graphs. Only then is it possible to refine this model (or find new models) that reflect the natural behavior in a more precise manner.

From a more theoretical perspective, setting aside the obvious goal of improving known bounds, there are many other possible research directions. The notion of hyperbolicity of networks, for instance, is not new. Gromov hyperbolicity [Gro87; Bow91] is a measure defined on a metric space and used in particular for graphs. Consider three points (vertices) u, v, w in the metric space (graph). The space (graph) then has hyperbolicity h, if for every point on a geodesic (shortest path) between two of the points u, v, w at least one other geodesic (shortest path) can be reached within distance h. It is immediate that every graph with diameter D has a Gromov hyperbolicity of at most $D/2$. The hyperbolic plane, on the other hand, has constant hyperbolicity due to its constant negative curvature. It is tempting to conjecture that the same must thus hold for hyperbolic random graphs due to the underlying geometric space. As has been shown, however, there exist path components of length $\Omega(\log n)$ in the hyperbolic random graph. We conjecture that the proof may be extended to yield cycles of length $\Omega(\log n)$ that contain no shortcuts. Such a cycle immediately bounds the Gromov hyperbolicity from below: Consider three equidistant points on the cycle as vertices u, v, w, and let z be the node that lies between u and v. To reach the path $u \to w$ or $v \to w$ from z, one needs at least $\Omega(\log n)$ hops.

Conjecture 2. *Hyperbolic random graphs have Gromov hyperbolicity* $\Omega(\log n)$ *with high probability.*

Since this is close to the diameter of the graph, such a result would essentially state that hyperbolic random graphs are not (Gromov) hyperbolic.

A different direction is to investigate so-called *unit disk graphs* (UDGs) [CCJ91]. Their definition is essentially a deterministic analog of hyperbolic random graphs in the step model (c. f. Definition 3) in the following sense. A graph G is a hyperbolic UDG if there exist an $R \in \mathbb{R}$ and an assignment $f\colon V \to \mathbb{H}^2$ such that $\operatorname{dist}(f(u), f(v)) \leqslant R \Leftrightarrow \{u, v\} \in E$. Notice that every graph generated by the step model is a hyperbolic UDG with probability 1. While the behavior of such graphs is well understood in the euclidean case, they are essentially unstudied in the hyperbolic case. We took the first step with Theorem 4.6 by proving that cliques in hyperbolic UDGs can be found in polynomial time if the underlying geometry, i. e., the function f is known. Since this algorithm was inspired by the euclidean analog, it is a natural question to ask which other results translate to this case.

One of the most fundamental problems is to decide whether a given graph is a (hyperbolic) UDG. Unfortunately, in the euclidean analog, this is NP-hard to determine. The NP-hardness proof extensively uses the following fact. In euclidean UDGs, constant-sized induced cycles can be embedded in the euclidean plane such that they may geometrically enclose a node not connected to the cycle. Since the hyperbolic plane behaves locally like the euclidean plane, this result should translate immediately if R can be chosen small enough. Even when R is chosen as a constant in n, the above property still holds and we believe that it is possible to adapt the same proof from the euclidean case. In that case, however, one has to put considerable care into the proof details.

For the case where $R = \Omega(\log n)$—which is the case in the step model of hyperbolic random graphs—it is not possible to enclose a node in a cycle of length $\leqslant n$ without having the node connect to the cycle. Therefore, it is not possible to directly translate the euclidean NP-hardness proof. This leaves the possibility open that an efficient algorithm for detecting hyperbolic UDGs might exist in this case. We believe, however, that due to the complexity of this problem this is most likely not the case, and that there likely exists another approach showing NP-hardness of the problem.

Conjecture 3. *Given a graph G, it is* NP*-hard to decide whether G is a hyperbolic UDG.*

Finally, from a practical perspective it would be interesting to apply the embedding algorithm from Chapter 6 to various real-world networks. Such an embedding might reveal new information about the data. For instance, close disconnected nodes suggest missing links in a social network, or hidden dependencies in a biological network. To this end, it might be helpful to construct a more accessible tool to reach different research areas working with complex networks.

Bibliography

[ABF15] M. A. Abdullah, M. Bode, and N. Fountoulakis. **Typical distances in a geometric model for complex networks**. 2015. arXiv: `1506.07811 [math.CO]`.

[ACL00] W. Aiello, F. R. K. Chung, and L. Lu. **A random graph model for massive graphs**. In: *32nd Symp. Theory of Computing (STOC)*. 2000, pp. 171–180.

[ACL01] W. Aiello, F. R. K. Chung, and L. Lu. **A random graph model for power law graphs**. *Experimental Mathematics* 10:1 (2001), pp. 53–66.

[AMA16] G. Alanis-Lobato, P. Mier, and M. A. Andrade-Navarro. **Efficient embedding of complex networks to hyperbolic space via their laplacian**. *Scientific Reports* 6 (2016).

[AOK15] R. Aldecoa, C. Orsini, and D. Krioukov. **Hyperbolic graph generator**. *Computer Physics Communications* 196 (2015), pp. 492–496.

[Ald91a] D. Aldous. **The continuum random tree I**. *The Annals of Probability* (1991), pp. 1–28.

[Ald91b] D. Aldous. **The continuum random tree II: an overview**. *Stochastic analysis* 167 (1991), pp. 23–70.

[Ald93] D. Aldous. **The continuum random tree III**. *The Annals of Probability* (1993), pp. 248–289.

[AS15a] N. Alon and J. H. Spencer. **The probabilistic method**. John Wiley & Sons, 2015.

[AS15b] D. M. Asta and C. R. Shalizi. **Geometric network comparisons**. In: *31st Conf. Uncertainty in Artificial Intelligence (UAI)*. 2015, pp. 102–110.

[Bar16] A.-L. Barabási. **Network Science**. 2016. URL: `http://barabasi.com/book/network-science`.

[BA99] A.-L. Barabási and R. Albert. **Emergence of scaling in random networks**. *Science* 286:5439 (1999), pp. 509–512.

[BM06] G. Bianconi and M. Marsili. **Number of cliques in random scale-free network ensembles**. *Physica D: Nonlinear Phenomena* 224:1 (2006), pp. 1–6.

[BFK16] T. Bläsius, T. Friedrich, and A. Krohmer. **Hyperbolic random graphs: separators and treewidth**. In: *24th European Symp. Algorithms (ESA)*. 2016, 15:1–15:16.

[Blä+16] T. Bläsius, T. Friedrich, A. Krohmer, and S. Laue. **Efficient embedding of scale-free graphs in the hyperbolic plane**. In: *24th European Symp. Algorithms (ESA)*. 2016, 16:1–16:18.

[BFM13] M. Bode, N. Fountoulakis, and T. Müller. **On the giant component of random hyperbolic graphs**. In: *7th European Conf. Combinatorics, Graph Theory and Applications.* 2013, pp. 425–429.

[BFM14] M. Bode, N. Fountoulakis, and T. Müller. **The probability that the hyperbolic random graph is connected**. Available online at `www.math.uu.nl/~Muell001/Papers/BFM.pdf`. 2014.

[BKC09] M. Boguñá, D. Krioukov, and K. C. Claffy. **Navigability of complex networks**. *Nature Physics* 5:1 (2009), pp. 74–80.

[BPK10] M. Boguñá, F. Papadopoulos, and D. Krioukov. **Sustaining the internet with hyperbolic mapping**. *Nature Communications* 1 (2010), p. 62.

[Bol98] B. Bollobás. **Random graphs**. Springer, 1998.

[BC88] B. Bollobás and F. R. K. Chung. **The diameter of a cycle plus a random matching**. *SIAM J. Discrete Mathematics* 1:3 (1988), pp. 328–333.

[BJR07] B. Bollobás, S. Janson, and O. Riordan. **The phase transition in inhomogeneous random graphs**. *Random Structures & Algorithms* 31:1 (2007), pp. 3–122.

[Bol+01] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. **The degree sequence of a scale-free random graph process**. *Random Structures & Algorithms* 18:3 (2001), pp. 279–290.

[Bow91] B. H. Bowditch. **Notes on gromov's hyperbolicity criterion for path-metric spaces**. In: *Group theory from a geometrical viewpoint.* 1991.

[BKL15] K. Bringmann, R. Keusch, and J. Lengler. **Geometric inhomogeneous random graphs**. 2015. arXiv: `1511.00576 [cs.SI]`.

[Bul15] P. Bullen. **Dictionary of Inequalities**. CRC Press, 2015.

[CF16] E. Candellero and N. Fountoulakis. **Bootstrap percolation and the geometry of complex networks**. *Stochastic Processes and their Applications* 126:1 (2016), pp. 234–264.

[Che+11] J. Cheng, Y. Ke, S. Chu, and M. T. Özsu. **Efficient core decomposition in massive networks**. In: *27th Intl. Conf. Data Engineering (ICDE).* 2011, pp. 51–62.

[CL02a] F. R. K. Chung and L. Lu. **Connected components in random graphs with given expected degree sequences**. *Annals of Combinatorics* 6:2 (2002), pp. 125–145.

[CL02b] F. R. K. Chung and L. Lu. **The average distances in random graphs with given expected degrees**. *Proceedings of the National Academy of Sciences* 99:25 (2002), pp. 15879–15882.

[CCJ91] B. N. Clark, C. J. Colbourn, and D. S. Johnson. **Unit disk graphs**. *Annals of Discrete Mathematics* 48 (1991), pp. 165–177.

[CSN09] A. Clauset, C. R. Shalizi, and M. E. J. Newman. **Power-law distributions in empirical data**. *SIAM Review* 51:4 (2009), pp. 661–703.

[CE16] J. R. Clough and T. S. Evans. **Embedding graphs in Lorentzian spacetime**. 2016. arXiv: `1602.03103 [physics.soc-ph]`.

[CC00] T. F. Cox and M. A. Cox. **Multidimensional Scaling**. CRC press, 2000.

[DS86] W. H. E. Day and D. Sankoff. **Computational complexity of inferring phylogenies by compatibility**. *Systematic Biology* 35:2 (1986), pp. 224–229.

[Die12] R. Diestel. **Graph Theory**. 4 ed. Springer, 2012.

[DG10] B. Doerr and L. A. Goldberg. **Drift analysis with tail bounds**. In: *11th Intl. Conf. Parallel Problem Solving From Nature (PPSN)*. 2010, pp. 174–183.

[DHH10] S. Dommers, R. van der Hofstad, and G. Hooghiemstra. **Diameters in preferential attachment models**. *J. Statistical Physics* 139:1 (2010), pp. 72–107.

[DA09] D. Dubhashi and A. Panconesi. **Concentration of Measure for the Analysis of Randomized Algorithms**. Cambridge University Press, 2009.

[EHP00] M. Evans, N. Hastings, and B. Peacock. **Statistical Distributions**. 4th ed. Wiley-Interscience, 2000.

[Fou15] N. Fountoulakis. **On a geometrization of the chung–lu model for complex networks**. *Journal of Complex Networks* (2015), cnu049.

[FPS12] N. Fountoulakis, K. Panagiotou, and T. Sauerwald. **Ultra-fast rumor spreading in social networks**. In: *23rd Symp. Discrete Algorithms (SODA)*. 2012, pp. 1642–1660.

[FK12] T. Friedrich and A. Krohmer. **Parameterized clique on scale-free networks**. In: *23rd Intl. Symp. Algorithms and Computation (ISAAC)*. 2012, pp. 659–668.

[FK15a] T. Friedrich and A. Krohmer. **Cliques in hyperbolic random graphs**. In: *34th IEEE Conf. Computer Communications (INFOCOM)*. 2015, pp. 1544–1552.

[FK15b] T. Friedrich and A. Krohmer. **On the diameter of hyperbolic random graphs**. In: *42nd Intl. Coll. Automata, Languages and Programming (ICALP)*. 2015, pp. 614–625.

[FK15c] T. Friedrich and A. Krohmer. **Parameterized clique on inhomogeneous random graphs**. *Discrete Applied Mathematics* 184 (2015), pp. 130–138.

[FSS13] T. Friedrich, T. Sauerwald, and A. Stauffer. **Diameter and broadcast time of random geometric graphs in arbitrary dimensions**. *Algorithmica* 67:1 (2013), pp. 65–88.

[GMB14] P. Gao, H. Miao, and J. S. Baras. **Social network ad allocation via hyperbolic embedding**. In: *53rd IEEE Conf. Decision and Control (CDC)*. 2014, pp. 4875–4880.

[Gar+16] G. García-Pérez, M. Boguñá, A. Allard, and M. Á. Serrano. **The hidden hyperbolic geometry of international trade: world trade atlas 1870–2013**. *Scientific Reports* 6 (2016), p. 33441.

[GKT05] D. Gibson, R. Kumar, and A. Tomkins. **Discovering large dense subgraphs in massive graphs**. In: *31st Intl. Conf. Very Large Databases (VLDB)*. 2005, pp. 721–732.

[Gro87] M. Gromov. **Hyperbolic groups**. In: *Essays in group theory*. Springer, 1987, pp. 75–263.

[GPP12] L. Gugelmann, K. Panagiotou, and U. Peter. **Random hyperbolic graphs: degree sequence and clustering**. In: *39th Intl. Coll. Automata, Languages and Programming (ICALP)*. 2012, pp. 573–585.

[Hof16] R. van der Hofstad. **Random graphs and complex networks**. Lecture notes available from `http://www.win.tue.nl/~rhofstad/NotesRGCN.pdf`. 2016.

[HK73] J. E. Hopcroft and R. M. Karp. **An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs**. *SIAM Journal on Computing* 2:4 (1973), pp. 225–231.

[JŁN10] S. Janson, T. Łuczak, and I. Norros. **Large cliques in a power-law random graph**. *J. Applied Probability* 47:4 (2010), pp. 1124–1135.

[JŁR11] S. Janson, T. Łuczak, and A. Rucinski. **Random graphs**. Vol. 45. John Wiley & Sons, 2011.

[Jen06] J. Jensen. **Sur les fonctions convexes et les inégalités entre les valeurs moyennes**. *Acta Mathematica* 30:1 (1906), pp. 175–193.

[KM15] M. Kiwi and D. Mitsche. **A bound for the diameter of random hyperbolic graphs**. In: *12th Workshop Analytic Algorithmics and Combinatorics (ANALCO)*. 2015, pp. 26–39.

[KM16] M. Kiwi and D. Mitsche. **Spectral Gap of Random Hyperbolic Graphs and Related Parameters**. 2016. arXiv: `1606.02240 [math.PR]`.

[KY15] P. Klein and N. E. Young. **On the number of iterations for dantzig-wolfe optimization and packing-covering approximation algorithms**. *SIAM Journal on Computing* 44:4 (2015), pp. 1154–1172.

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

[Kli98] M. Kline. **Calculus: an intuitive and physical approach**. Courier Corporation, 1998.

[Kob13] S. G. Kobourov. **Force-directed drawing algorithms**. In: *Handbook of Graph Drawing and Visualization*. CRC press, 2013, pp. 383–408.

[KW05] S. G. Kobourov and K. Wampler. **Non-euclidean spring embedders**. *IEEE Transactions on Visualization and Computer Graphics* 11:6 (2005), pp. 757–767.

[Kri+10] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá. **Hyperbolic geometry of complex networks**. *Physical Review E* 82:3 (2010), p. 036106.

[LRP95] J. Lamping, R. Rao, and P. Pirolli. **A focus+context technique based on hyperbolic geometry for visualizing large hierarchies**. In: *13th ACM Conf. Human Factors in Comp. Systems (CHI)*. 1995, pp. 401–408.

[LR96] J. Lamping and R. Rao. **The hyperbolic browser: a focus+context technique for visualizing large hierarchies**. *J. Visual Languages & Computing* 7:1 (1996), pp. 33–55.

[LW13] P. K. Lehre and C. Witt. **General drift analysis with tail bounds**. 2013. arXiv: `1307.2559 [cs.NE]`.

[LK14] J. Leskovec and A. Krevl. **SNAP Datasets: Stanford Large Network Dataset Collection**. `http://snap.stanford.edu/data`. 2014.

[LPW08] D. A. Levin, Y. Peres, and E. L. Wilmer. **Markov Chains and Mixing Times**. AMS, 2008.

[Li+05] L. Li, D. Alderson, J. C. Doyle, and W. Willinger. **Towards a theory of scale-free graphs: definition, properties, and implications**. *Internet Mathematics* 2:4 (2005), pp. 431–523.

[LK07] D. Liben-Nowell and J. M. Kleinberg. **The link-prediction problem for social networks**. *J. Association for Information Science and Technology* 58:7 (2007), pp. 1019–1031.

[Lin76] T. Lindvall. **On the maximum of a branching process**. *Scandinavian Journal of Statistics* (1976), pp. 209–214.

[LMP15] M. von Looz, H. Meyerhenke, and R. Prutkin. **Generating random hyperbolic graphs in subquadratic time**. In: *26th Intl. Symp. Algorithms and Computation (ISAAC)*. Springer, 2015, pp. 467–478.

[LP49] R. D. Luce and A. D. Perry. **A method of matrix analysis of group structure**. *Psychometrika* 14:2 (1949), pp. 95–116.

[MN04] C. U. Martel and V. Nguyen. **Analyzing Kleinberg's (and other) small-world models**. In: *23rd Symp. Principles of Distributed Computing (PODC)*. 2004, pp. 179–188.

[McD98] C. McDiarmid. **Concentration**. In: *Probabilistic methods for algorithmic discrete mathematics*. Springer, 1998, pp. 195–248.

[MU05] M. Mitzenmacher and E. Upfal. **Probability and computing: Randomized algorithms and probabilistic analysis**. Cambridge University Press, 2005.

[Mun98] T. Munzner. **Exploring large graphs in 3D hyperbolic space**. *IEEE Computer Graphics and Applications* 18:4 (1998), pp. 18–23.

[New01] M. E. J. Newman. **Clustering and preferential attachment in growing networks**. *Physical Review E* 64 (2001), p. 025102.

[New03] M. E. J. Newman. **The structure and function of complex networks**. *SIAM Review* 45:2 (2003), pp. 167–256.

[NR06] I. Norros and H. Reittu. **On a conditionally Poissonian graph process**. *Advances in Applied Probability* 38:1 (2006), pp. 59–75.

[Olv10] F. W. J. Olver. **NIST handbook of mathematical functions**. Cambridge University Press, 2010.

[PAK15] F. Papadopoulos, R. Aldecoa, and D. Krioukov. **Network geometry inference using common neighbors**. *Physical Review E* 92:2 (2015), p. 022807.

[Pap+12] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguñá, and D. Krioukov. **Popularity versus similarity in growing networks**. *Nature* 489:7417 (2012), pp. 537–540.

[Pap+10] F. Papadopoulos, D. Krioukov, M. Boguñá, and A. Vahdat. **Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces**. In: *29th IEEE Conf. Computer Communications (INFOCOM)*. 2010, pp. 2973–2981.

[PPK15] F. Papadopoulos, C. Psomas, and D. Krioukov. **Network mapping by replaying hyperbolic growth**. *IEEE/ACM Transactions on Networking* 23:1 (2015), pp. 198–211.

[Pen03] M. D. Penrose. **Random Geometric Graphs**. Oxford University Press, 2003.

[Pet14] U. Peter. **Random graph models for complex systems**. PhD thesis. ETH Zürich, 2014.

[RS98] M. Raab and A. Steger. **"Balls into bins" – a simple and tight analysis**. In: *2nd Intl. Workshop on Randomization and Computation (RANDOM)*. 1998, pp. 159–170.

[RW10] O. Riordan and N. Wormald. **The diameter of sparse random graphs**. *Combinatorics, Probability & Computing* 19:5-6 (2010), pp. 835–926.

[Ros08] B. Rossman. **On the constant-depth complexity of k-clique**. In: *40th Symp. Theory of Computing (STOC)*. 2008, pp. 721–730.

[Ros10a] B. Rossman. **Average-case complexity of detecting cliques**. PhD thesis. Massachusetts Institute of Technology, 2010.

[Ros10b] B. Rossman. **The monotone complexity of k-clique on random graphs**. In: *51st Symp. Foundations of Computer Science (FOCS)*. 2010, pp. 193–201.

[SM98] R. Samudrala and J. Moult. **A graph-theoretic algorithm for comparative modeling of protein structure**. *Journal of Molecular Biology* 279:1 (1998), pp. 287–302.

[SBS12] M. Á. Serrano, M. Boguñá, and F. Sagués. **Uncovering the hidden geometry behind metabolic networks**. *Molecular BioSystems* 8:3 (2012), pp. 843–850.

[SKB08] M. Á. Serrano, D. Krioukov, and M. Boguñá. **Self-similarity of complex networks and hidden metric spaces**. *Physical Review Letters* 100:7 (2008), p. 078701.

[ST08] Y. Shavitt and T. Tankel. **Hyperbolic embedding of internet graph for distance estimation and overlay construction**. *IEEE/ACM Transactions on Networking* 16:1 (2008), pp. 25–36.

[Söd02] B. Söderberg. **General formalism for inhomogeneous random graphs**. *Physical review E* 66:6 (2002), p. 066121.

[SKP16] E. Stai, V. Karyotis, and S. Papavassiliou. **A hyperbolic space analytics framework for big network data and their applications**. *IEEE Network* 30:1 (2016), pp. 11–17.

[Tso+13] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli. **Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees**. In: *19th Intl. Conf. Knowledge Discovery and Data Mining (KDD)*. 2013, pp. 104–112.

[Váz03] A. Vázquez. **Growing network with local rules: preferential attachment, clustering hierarchy, and degree correlations**. *Physical Review E* 67:5 (2003), p. 056104.

[VS14] K. Verbeek and S. Suri. **Metric embedding, hyperbolic space, and social networks**. In: *30th Symp. Computational Geometry (SoCG)*. 2014, p. 501.

[Wal04] J. A. Walter. **H-MDS: a new approach for interactive visualization with multidimensional scaling in the hyperbolic space**. *Information Systems* 29:4 (2004), pp. 273–292.

[WR02] J. A. Walter and H. J. Ritter. **On interactive visualization of high-dimensional data using the hyperbolic plane**. In: *8th ACM Intl. Conf. Knowledge Discovery and Data Mining (SIGKDD)*. 2002, pp. 123–132.

[WY05] Y. Wang and Y.-N. Yeh. **Polynomials with real zeros and Pólya frequency sequences**. *J. Combinatorial Theory, Series A* 109:1 (2005), pp. 63–74.

[Wan+16a] Z. Wang, Q. Li, F. Jin, W. Xiong, and Y. Wu. **Hyperbolic mapping of complex networks based on community information**. *Physica A: Statistical Mechanics and its Applications* 455 (2016), pp. 104–119.

[Wan+16b] Z. Wang, Q. Li, W. Xiong, F. Jin, and Y. Wu. **Fast community detection based on sector edge aggregation metric model in hyperbolic space**. *Physica A: Statistical Mechanics and its Applications* 452 (2016), pp. 178–191.

[WS98] D. J. Watts and S. H. Strogatz. **Collective dynamics of 'small-world' networks**. *Nature* 393:6684 (1998), pp. 440–442.

[WS06] K. Q. Weinberger and L. K. Saul. **Unsupervised learning of image manifolds by semidefinite programming**. *Intl. J. Computer Vision* 70:1 (2006), pp. 77–90.

[YL15] J. Yang and J. Leskovec. **Defining and evaluating network communities based on ground-truth**. *Knowledge and Information Systems* 42:1 (2015), pp. 181–213.

[Zha+14] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al. **Named data networking**. *ACM SIGCOMM Computer Communication Review* 44:3 (2014), pp. 66–73.

[Zha+11] X. Zhao, A. Sala, H. Zheng, and B. Y. Zhao. **Efficient shortest paths on massive social graphs**. In: *7th Intl. Conf. Collaborative Computing (CollaborateCom)*. 2011, pp. 77–86.

List of Publications

Journal Articles

[1] **Efficient key pathway mining: combining networks and omics data**. *Integrative Biology* 4 (2012), pp. 756–764. Joint work with N. Alcaraz, T. Friedrich, T. Kötzing, J. Müller, J. Pauling, and J. Baumbach.

[2] **Parameterized clique on inhomogeneous random graphs**. *Discrete Applied Mathematics* 184 (2015), pp. 130–138. Joint work with T. Friedrich.

Articles in Refereed Conference Proceedings

[3] **Ratfish: A file sharing protocol provably secure against rational users**. In: *15th European Symp. Research in Computer Security (ESORICS)*. 2010, pp. 607–625. Joint work with M. Backes and O. Ciobotaru.

[4] **Efficient algorithms for extracting biological key pathways with global constraints**. In: *Genetic and Evolutionary Computation Conf. (GECCO)*. 2012, pp. 169–176. Joint work with J. Baumbach, T. Friedrich, T. Kötzing, J. Müller, and J. Pauling.

[5] **Parameterized clique on scale-free networks**. In: *23rd Intl. Symp. Algorithms and Computation (ISAAC)*. 2012, pp. 659–668. Joint work with T. Friedrich.

[6] **De-anonymization of heterogeneous random graphs in quasilinear time**. In: *22nd European Symp. Algorithms (ESA)*. 2014, pp. 197–208. Joint work with K. Bringmann and T. Friedrich.

[7] **Cliques in hyperbolic random graphs**. In: *34th IEEE Conf. Computer Communications (INFOCOM)*. 2015, pp. 1544–1552. Joint work with T. Friedrich.

[8] **On the diameter of hyperbolic random graphs**. In: *42nd Intl. Coll. Automata, Languages and Programming (ICALP)*. 2015, pp. 614–625. Joint work with T. Friedrich.

[9] **Unbounded discrepancy of deterministic random walks on grids**. In: *26th Intl. Symp. Algorithms and Computation (ISAAC)*. 2015, pp. 212–222. Joint work with T. Friedrich and M. Katzmann.

[10] **Hyperbolic random graphs: separators and treewidth**. In: *24th European Symp. Algorithms (ESA)*. 2016, 15:1–15:16. Joint work with T. Bläsius and T. Friedrich.

[11] **Efficient embedding of scale-free graphs in the hyperbolic plane**. In: *24th European Symp. Algorithms (ESA)*. 2016, 16:1–16:18. Joint work with T. Bläsius, T. Friedrich, and S. Laue.