Limiting Behaviour of Barabási-Albert Random Graphs

Candidate Number: 321325

March 15, 2012

Abstract

Many popular existing random graph algorithms produce graphs for which the vertices have approximately equal degrees. This differs from many real-world examples modelled by graphs where as the number of vertices grows, infrequent, largely connected vertices, called hubs, become more common. We consider a number of different algorithms used to generate random graphs. Of particular interest is the Barabási-Albert random graphs which uses the idea of preferential attachment to produce hubs, mimicking the real world data. We construct a sequence of (dependent) stochastic processes in which we are able to embed the degree sequences of Barabási-Albert random graphs. Using this result, the limiting behaviour of the embedded sequences are investigated and extended to the random graph structures. We run large-scale simulations of the generalised Barabási-Albert algorithm and examine whether the generated results match both the theoretical results we develop, and the empirical data we analyse.

Contents

Chapter 1

Motivation

1.1 Uses of Graphical Modelling

Deterministic graphs are very useful as a means to model various real-world networks that focus on connections between interacting objects [6]. Research has been undertaken in using these models for a wide range of situations. In 1959, Erdős and Rényi first considered the idea of using random graphs to model the properties of such real-world networks [10], and since then a number of different models have been suggested.

Consider the degrees of each vertex of a graph, that is, the number of edges attached to it. We save a formal definition until Section 2.1. One of the properties shared by many of the real-world models was that the sequence formed by the degrees of the vertex follow, what Barabási and Albert called in their original paper, a *scale-free* distribution [5].The theoretical model behind this suggests that as the number of vertices tends to infinity, the proportion of vertices with degree j is of the order $\mathrm{O}(j^{-r})$ for some r, which empirically appears to be in the interval $(2, 4)$. Since our models deal with real world behaviour where the notion of a graph with an infinite number of vertices is impossible, we instead consider this behaviour for large graphs. We consider one such example where we can observe such a property in section 1.2

Graphs which experts suspect have this property underlying their behaviour include:

- Citation networks [13], where vertices represent academic papers in a field and the edges between them represent the later paper referencing the earlier one;
- Biochemical reactions [15], where vertices represent chemicals found in human bodies and an edge exists if a reaction would commonly occur between them in the human body;
- Actor filmographies [5], where vertices represent actors and edges represent movies in which two actors collaborate;
- Internet topologies [1], where vertices represent websites and edges hyperlinks between them.

One of the features of these models is that as a system grows, instead of increasingly high-degree vertices becoming rare as earlier models predicted, it is more likely to see clusters of vertices gathered around a small number of these highly connected vertices (or ”hubs”). In the examples referenced above, these make intuitive sense representing well-regarded or influential articles, reactive chemicals, established actors and popular websites respectively.

Earlier algorithms for generating random graphs fail to give rise to such a property. This motivated Barabási and Albert to propose a preferential attachment that correctly models such scale-free distribution [5].

1.2 Internet Data

Due to the interest in the applications of scale-free networks, there are a large collection of data sets available online for study. One of these, data collected by Eötvös University [14] is collated by data robots on a weekly basis, showing the source and destination of links between websites.

The data comes in the form of 16,704,672 directed arcs, showing the source of each of the links lying within the top level domains ".eu" and ".hu," that is 1,845,262 distinct Europe-wide and Hungarian websites.

The in-degree and out-degree of a website are the number of links going to and from it respectively, which we define formally in section 2.1. From our data, both the in-degree and the out-degree of each website can be calculated and then tested to see if, as Barabási first suggested, they adopt the structure of a scale-free network.

The following analysis, including the plotting of any graph, was undertaken using R, and the functionality contained in the package igraph [9]. The code is available in Appendix A.1.

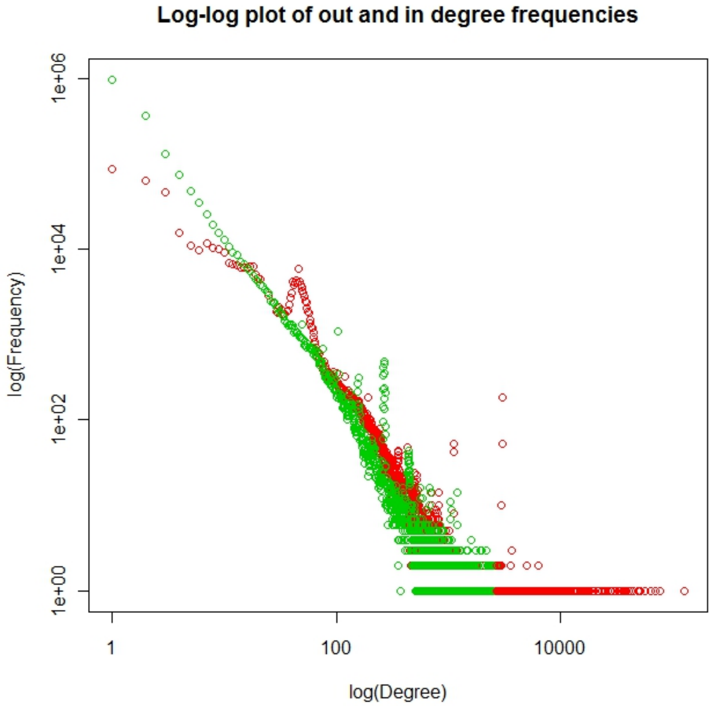


Figure 1.1: Log-log plot of out and in degree frequencies, where red points indicate out-degrees and green points indicate in-degrees

Let $R^+(j)$ represent the number of vertices which have out-degree j, and similarly $R^-(j)$ the number of vertices with in-degree j. We are trying to fit a model of the form

$$R^+(j) \sim j^{-r_1} \qquad \text{and} \qquad R^-(j) \sim j^{-r_2}.$$

Note that this is equivalent to finding a linear relationship on a log-log plot, which we present in Figure 1.1.

The red points, indicating out-degrees show a weaker linear relationship, if at all, however, the in-degrees show significant promise for further study. When considering the meaning of the data, this seems reasonable. Our scale-free models appear when there are large hubs which gather a lot of attention, and in the internet, these hubs will not be websites which are the source of many hyperlinks, but are the destination of hyperlinks. The prominent, but anomalous, kink in the out-degrees distribution is worth further study but not part of this project. Instead, we choose to analyse the in-degree for evidence of a power law distribution.

The first thing to notice is the noise for large degrees. There do seem to be a large number of websites that fall outside the power-law for large degrees. The website for which we have the highest in-degree (http://bannercsere.topextra.hu) is not a proper website, but instead a hosting service for banner advertisements that appear on other websites and link back to itself. The fact that large websites of this sort consist of many different kinds gives a possible cause of a high variance.

We will attempt to fit a linear model of the form

$$\log(1 + R^{-}(j)) = c - r\log(1 + j)$$

to see if the assumption of the scale-free properties of in-degrees is suitable. We give a more sophisticated means of fitting a power-law in Section 4.2. The addition of 1 to the variable is to prohibit the model from breaking down when either of the variables equalling 0. However, this method is only suitable for models where we expect the residuals to have a constant variance. This is clearly not the case in this model, so we restrict any analysis to the domain in which the assumption of homoscedasticity is more reasonable ($j < \mathrm{e}^6 \approx 400$).

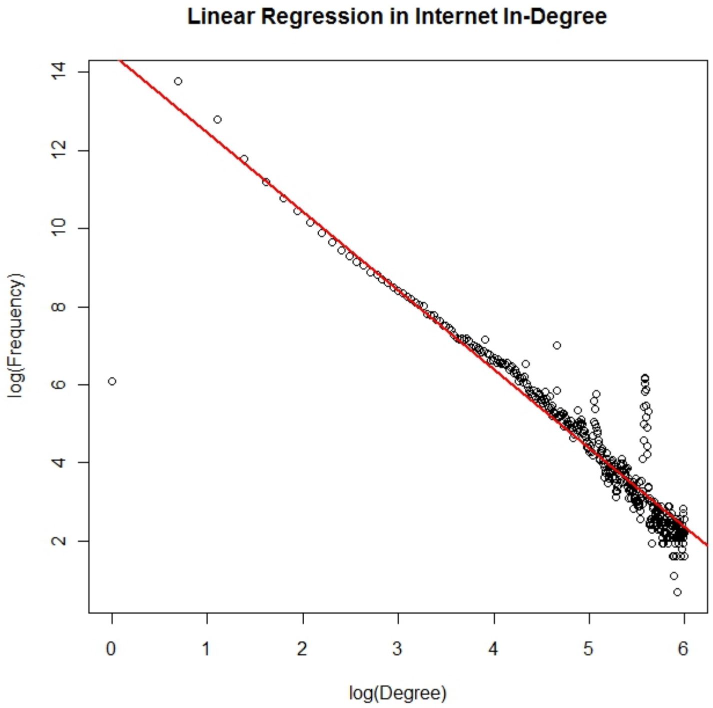


Figure 1.2: Linear regression line of log(Degree) against the corresponding log(Frequency)

The fitted linear model can be seen in Figure 1.2, and we show the corresponding diagnostic plots in Figure 1.3.

The diagnostic plots show the model is mostly acceptable, apart from one data point. This data point represents the number of websites with no hyperlinks linking into them, and is such a

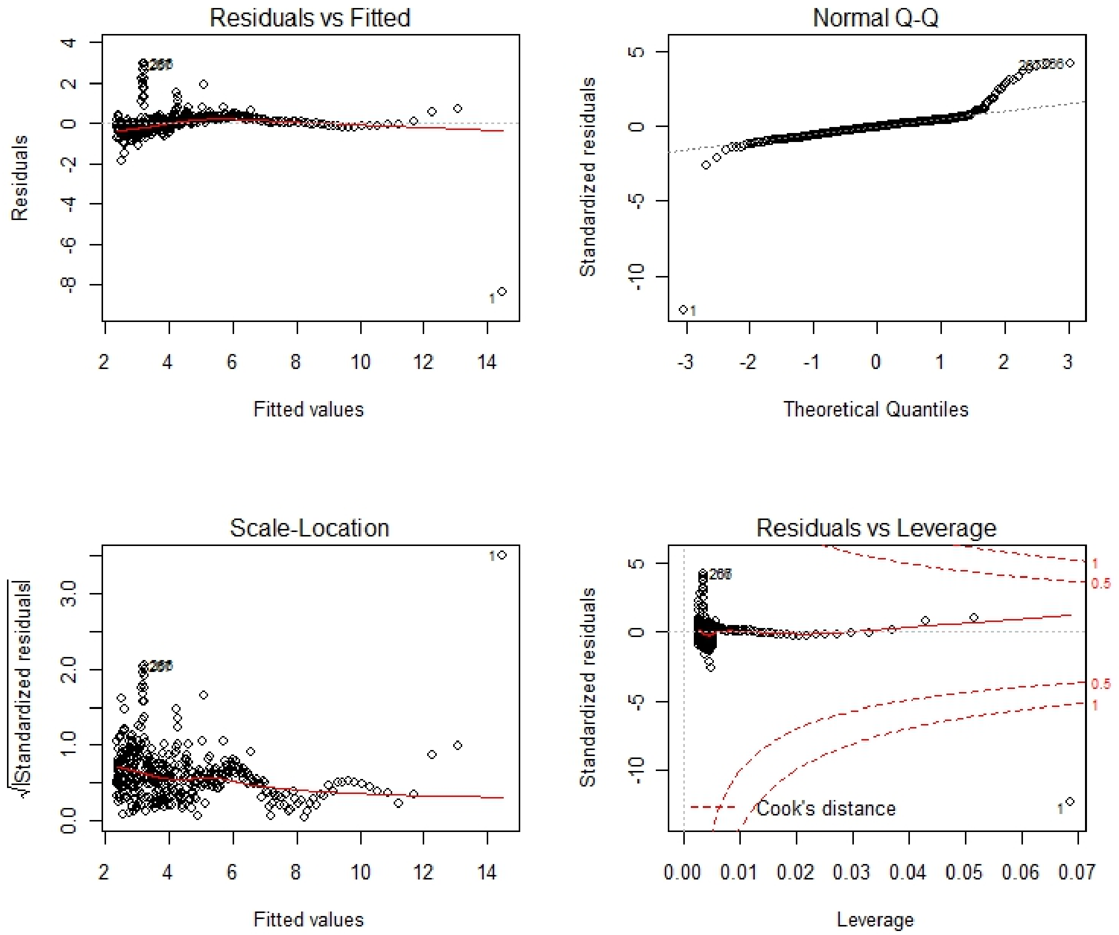


Figure 1.3: Diagnostic plots for the linear model fitted

heavy outlier because of the relative sparsity of such websites, compared to the large amount with one, two or otherwise few links.

The other concern may be the Q-Q plots showing a lack of normality for higher data values. This is unfortunate, but the Cook's distance and leverage of those data points who exhibit non-normality points mostly justify keeping them in the analysis, especially because of the large amounts of data otherwise discarded. It is worth noting, however, that although the non-normality of the standardised residuals is a concern for the legitimacy of parameter estimation, from observing the graph, any weakness of our model is not hiding a different, non-linear underlying structure. This data is worthy of further, more rigorous, study, but that is beyond the scope of this analysis, where we seek only to demonstrate the linearity of this relationship.

The coefficients of this linear model are given by

$$\begin{aligned} &\log(R^{-}(j)+1) \approx C - 2.018\log(j+1) \\ \implies &R^{-}(j) \approx C' j^{-2.018}, \end{aligned}$$

where C, C' are simply constants. This represents the manner of scale-free behaviour that we are after. A larger statistical analysis performed on a data set of nearly 800 million websites, representing more global sites, shows that this same behaviour perserves (they found the power coefficient was 2.0 with a standard error of around 0.1 [5]). The fact that this power property is preserved at different level of depths in the same subject is motiviation for the nomenclature of a *scale-free* models.

1.3 Overview

In the rest of this dissertation, we consider the scale-free nature of degree sequences for graphs. This is a property that Barabási and Albert [5] noticed in a large amount of models, for example in the destinations of hyperlinks. We have replicated their finding above using more modern data. We seek to analyse graphs of this structure. In particular, we aim to explore an algorithm which creates graphs with a scale-free limiting degree distribution.

In Chapter 2 we consider a number of different algorithms used to generate random graphs. Of particular interest is the Barabási-Albert random graphs (Algorithm 2.3) which uses the idea of preferential attachment. Using these algorithms, we provide plots of various realisations of such random graphs, and suggest why other popular random graph models without preferential attachment are not suitable for modelling a scale-free network.

In Chapter 3, we mimc the derivation of some key properties of the generalised Barabási-Albert random graphs that Athreya et al. [3] used. We construct a sequence of (dependent) stochastic processes that we are able to embed the degree sequences of Barabási-Albert random graphs into. We use results from applied probability and martingale theory [2, 17] to investigate the limiting behaviour of the embedded sequences and extend thsee results to the random graphs.

In Chapter 4, we investigate some of the results we had earlier proved. We run large-scale simulations of the generalised Barabási-Albert algorithm and examine whether the generated results match both the theoretical results we develop, and the empirical scale-free internet data we have analysed.

There is also an appendix, which lists the original code written for this project, including the commands used to generate simulations, analyse data and plot graphs.

The study of random graphs, and scale-free networks, is a very wide field which we are just touching on here. There are many more data sets which exhibit such a structure, which could be studied using more sophisticated techniques. There are a number of variations to the preferential attachment random graphs presented here which maintain the scale-free distributions. We introduce one such model, but concentrate on deriving results for the original algorithm. These other algorithms could be theoretically analysed using their underlying stochastic embeddings. Also, the simulations undertaken are restricted to one set of parameters. Further simulations could be undertaken to give an understanding of the output of the algorithm more clearly. Some convergence results developed in this paper show convergence to a limit, but do not give an easy way to calculate that limit. Further simulation could also provide a means of estimating these results.

Chapter 2

Random Graph Models

2.1 Preliminary Definitions

We will employ the notation that for $n \in \mathbb{N}$,

$$[n] := \{i \leq n : i \in \mathbb{N}\}.$$

For a set S, we denote the distinct, unordered pairs by

$$S^{(2)} := \{\{u, v\} : u \neq v; u, v \in S\}.$$

By using graphs, we wish to model relationships between any numbers of pairs of points, which we will represent pictorially as dots, with lines between them representing a potential interaction. Although other definitions can be used, we will formally define a graph G as an ordered pair $(V(G), E(G))$, where $V(G)$ is a (normally) finite set called the *vertices* and $E(G)$ is a collection of elements of $V(G)^{(2)}$, not necessarily distinct, called the *edges*. (Other texts call this a multigraph, and restrict graphs to be those multigraphs with distinct edges [6].) The *order* of a graph, denoted $|G|$, is the number of vertices $|V(G)|$.

For a given vertex $v \in V(G)$ we define the *degree* of the vertex by the multiplicity of edges with either of their endpoints at v

$$\deg_G(v) := |\{\{u, v\} \in E(G) : u \in V(G)\}|.$$

We say that two graphs G_1, G_2 are *isomorphic*, written $G_1 \cong G_2$, if they are the same up to a relabelling of their vertices. More formally, this occurs if there exists an isomorphism $\varphi : V(G_1) \to V(G_2)$ such that

$$\forall \{v_1, v_2\} \in V(G_1)^{(2)}, |\, \{\{v_1, v_2\} \in E(G_1)\} \,| = |\, \{\{\varphi(v_1), \varphi(v_2)\} \in E(G_2)\} \,|$$

Given a graph G, we define the *adjacency matrix* of G, $A(G)$, as

$$A := (a_{ij})_{i,j \in V(G)},$$

where the entries are the multiplicites of the edges between any pair of vertices, that is

$$a_{uv} := |\{\{u, v\} \in E(G)\}|.$$

Two vertices $u, v \in V(G)$ are *adjacent* if $a_{uv} \neq 0$, that is, at least one edge connects them.

The *complete graph* on n vertices, written K_n, is defined as any graph isomorphic to the graph on n vertices with the adjacency matrix that has non-diagonal entries equal to one, that is, the graph where each vertex is attached to each other with exactly one edge.

A *regular* graph is a graph whose vertices all have equal degree. The *regular ring lattice*, as seen in Figure 2.1, on n vertices with each vertex having degree $2k \leq n$ is the graph with vertex set $[n]$ and adjacency matrix

$$A = (a_{ij})_{i,j \leq n},$$

where

$$a_{ij} = \begin{cases} 0 & k < |i-j| < n-k \qquad \text{or} \qquad i = j \\ 1 & \text{otherwise.} \end{cases}$$

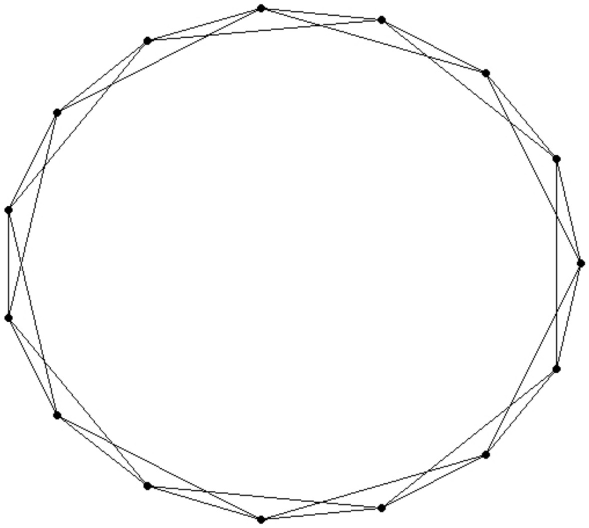


Figure 2.1: The regular ring lattice on 15 vertices, with each vertex having degree 4

Related to the concept of a graph, a *directed graph* D is intuitively a graph whose edges are ordered pairs, so that each edge has a source and a target. Formally, this means that D is an ordered pair $(V(D), E(D))$ where $V(D)$ is the (normally) fiinite vertex set. In a directed graph, $E(D)$ is a collection of (not necessarily distinct) ordered pairs of distinct elements of $V(D)$. Elements of $E(D)$ are called the *arcs* or *edges*.

We define the *out-degree* of a vertex $v \in V(D)$ as the multiplicity of edges with v as a source, that is

$$\deg^+(v) := |\{(v, u) \in E(D)\}|,$$

and the *in-degree* of a vertex $v \in V(D)$ as the multiplicty odf edges directed towards v,

$$\deg^-(v) := |\{(u, v) \in E(D)\}|.$$

When we consider directed graphs, we generalise the idea of an *adjacency matrix* to represent the edge structure. The adjacency matrix of a directed graph D is the matrix

$$A(D) := (a_{ij})_{i,j \in [|D]},$$

where

$$a_{ij} := |(v_i, v_j) \in E(D)|.$$

2.2 Erdős-Rényi Random Graphs

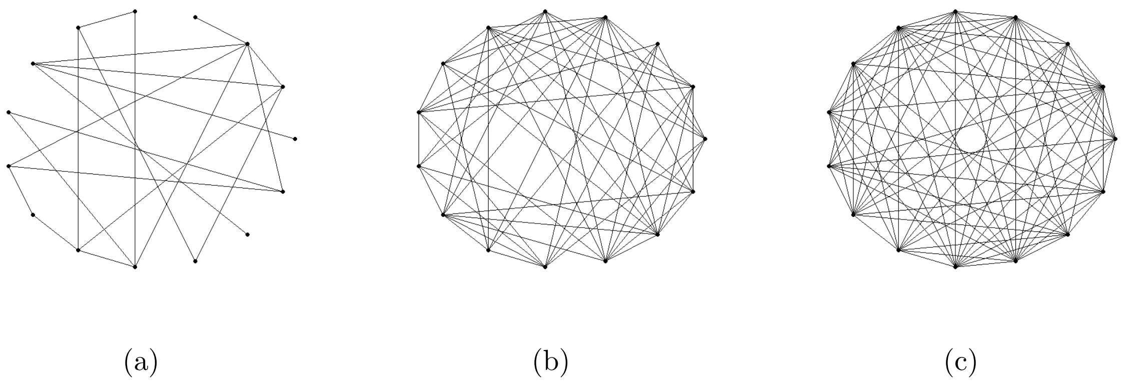
The simplest model first studied in the field of random graphs was introduced by Erdős and Rényi [10] and simply sampled uniformly between all graphs with n vertices and e edges. Gilbert [11] made a similar, if slightly more sophisticated model also with two parameters – the number of vertices n and a probability parameter $p \in [0, 1]$. Since the results of graphs generated this way are so similar, the algorithm that Gilbert proposed kept the name of Erdős and Rényi.

An Erdős-Rényi graph is then produced using the algorithm with parameters n and p:

Algorithm 2.1 (Erdős-Rényi Algorithm for Random Graphs)**.**

- *Create the empty graph on n labelled vertices, that is $V(G) = \{v_i : 1 \leq i \leq n\}$ and $E(G) = \emptyset$*
- *For each pair of vertices $v_i, v_j \in V(G)^{(2)}$, add the edge $\{v_i, v_j\}$ to the edge set with probability p indepedently of each other pair.*

We show simulations from this algorithm with various parameters in figure 2.2.



(a) (b) (c)

Figure 2.2: Graphs generated using the Erdős-Rényi algorithm with $n = 15$ and $p =$ (a) 0.2, (b) 0.5, (c) 0.8

There are a number of reasons why this model is unsuitable for some of the real world models that we may wish to analyse. Firstly, it does not allow for the addition of new vertices, instead fixing this at the start. We also have that each pair is equally likely to be connected, independent of other vertices. Although in fields such as combinatorics this is an advantage, for modelling certain situations this is undesirable. For example, in the model of collaborations in successful movies, lesser known actors are more likely to be paired with more established performers in the starring role. The Erdős-Rényi model fails to take this into account since each pairing is equally likely.

2.3 Strogatz-Watts Random Graphs

Watts and Strogatz [16] noticed that certain models which lent themselves to graphical modelling had properties which lay between fully deterministic graphs and the fully random models which Erdős and Rényi produced. Their model solved a problem known as the *small-world phenomenon*, that stated that even in large networks, there exist short paths between almost all pairs of vertices. Their algorithm produces random graphs by taking a regular graph with the property that no two vertices are too far apart, and modifying each edge with a given probability. Formally, the algorithm on n vertices is given by the cluster parameter $2k$, an even integer, and disorder parameter $p \in [0, 1]$:

Algorithm 2.2 (Strogatz-Watts Algorithm for Random Graphs)**.**

- *Create the regular ring lattice on n vertices with each vertex having degree $2k$.*
- *For each of the nk edges of the graph generated this way, with probability p randomly rewire it, that is replace its endpoints with two non-adjacent vertices sampled uniformly from the set of all non-adjacent pairs.*

We show simulations from this algorithm with various parameters in figures 2.3 and 2.4.

At $p = 0$, the algorithm produces the non-random regular ring lattice with each vertex having degree $2k$. At $p = 1$, the graph produced is sampled uniformly from among the graphs on n vertices and nk edges. In the intermediate values of p, we have a graph which maintains some of the properties of the regular graph, but simulates real-world noise more accurately.

2.4 Barabási-Albert Random Graphs

Athreya et al [3] suggested a generalised version of a model first proposed by Barabási and Albert [5] that work on the basis of preferential attachment. That is, unlike previous models, when considering a given vertex, the likelihood of it being adjacent any other given vertex is neither independent nor uniformly distributed but instead depends on the degree of existing vertices.

Our algorithm has two parameters, $\beta \in \mathbb{R}^+$ and a probability mass function (pmf) $\{p_i\}_{i \in \mathbb{N}}$ on the natural numbers (which we defined as *not* including zero). Given these parameters, we define an algorithm:

Algorithm 2.3 (Generalised Barabási-Albert Algorithm for Random Graphs)**.**

- *Define $G_0 \cong K_2$ with vertices v_1 and v_2.*
- *On the n^{th} iteration of this algorithm, we transform G_{n-1} to G_n, by the following:*
  - *Add the vertex v_{n+2}*
  - *Define the probability mass function $f : \{v_1 \dots v_{n+1}\} \to [0, 1]$ by*

$$f(v_i) := \frac{\deg_{G_{n-1}}(v_i) + \beta}{\displaystyle\sum_{j=1}^{n+1} \Big(\deg_{G_{n-1}}(v_j) + \beta\Big)}$$

  - *Generate a realisation v_{n*} from the pmf f and d_n from the pmf p.*

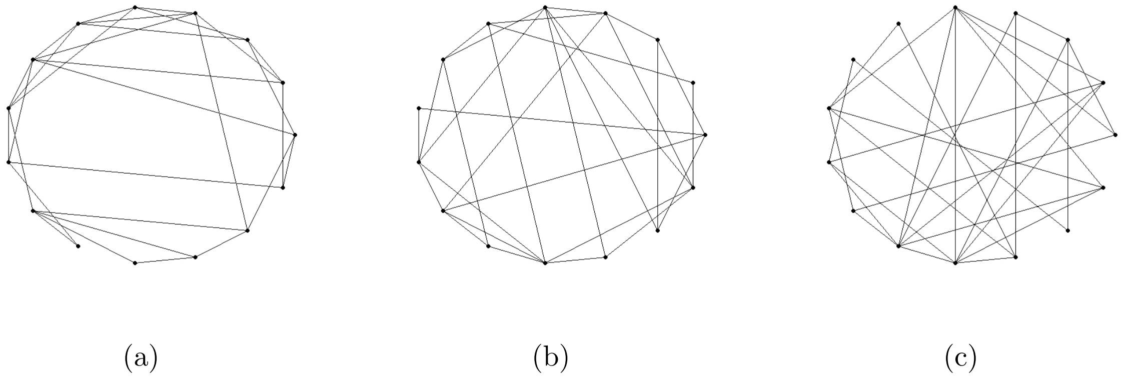


Figure 2.3: Graphs generated using the Strogatz-Watts algorithm with $n = 15$, $k = 4$ and intermediate $p =$ (a) 0.2, (b) 0.5, (c) 0.8

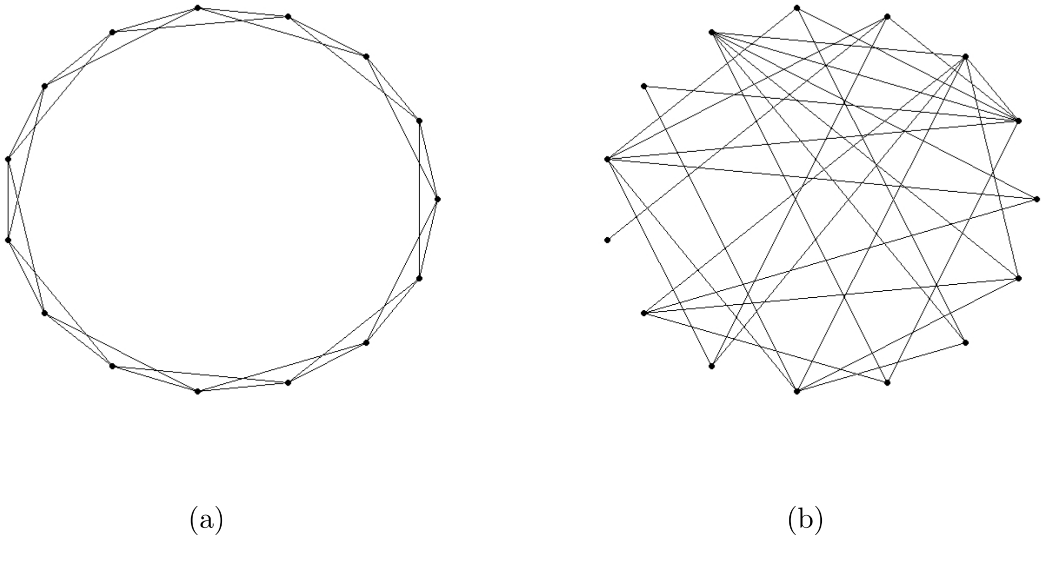
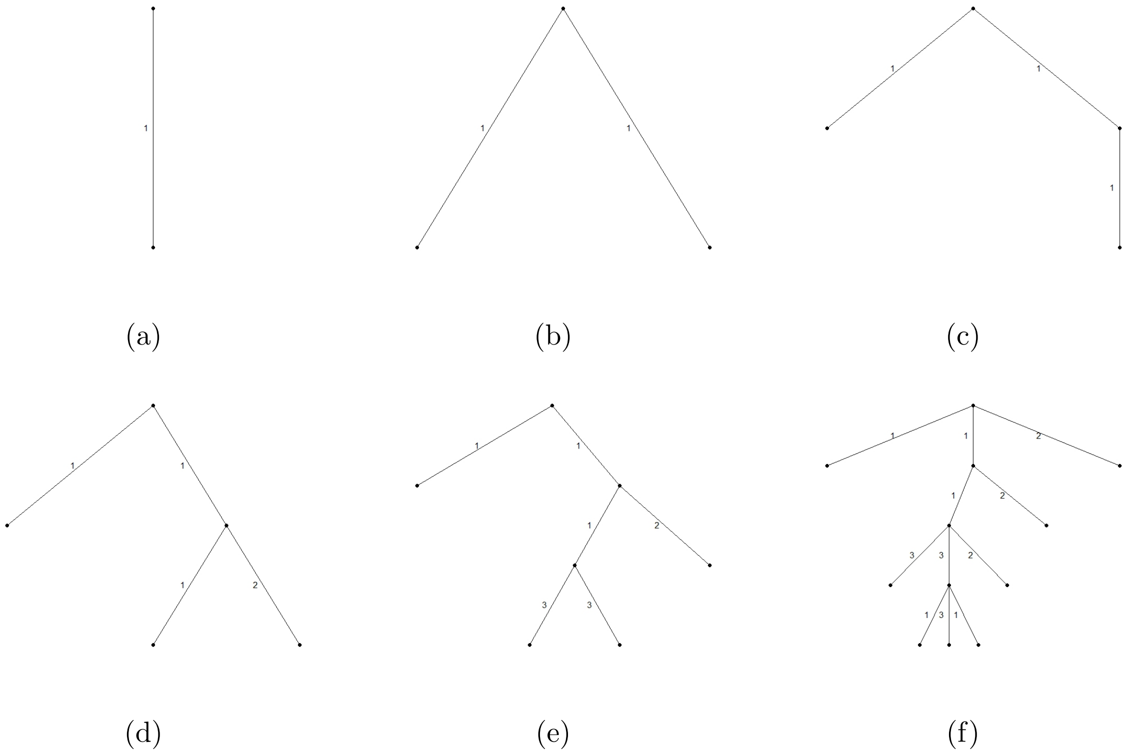


Figure 2.4: Graphs generated using the Strogatz-Watts algorithm with $n = 15$, $k = 4$ with (a) no randomness, i.e. $p = 0$ and (b) entirely random, i.e. $p = 1$

– *Attach vertex v_{n+2} to v_{n*} with d_n edges.*

This generates a sequence of graphs $(G_i)_{i\in\mathbb{N}\cup\{0\}}$, each of which is formed by attaching a new vertex to exactly one other, favouring attachments with those who are already highly connected.

We show iterations within the same simulation in figure 2.5, and three differerent simulations with the same parameter in 2.6.



(a) (b) (c)

(d) (e) (f)

Figure 2.5: Successive iterations of an output from a Barabási-Albert algorithm with edge distribution Geometric($\frac{1}{2}$) and n= (a) 0, (b) 1, (c) 2, (d) 3, (e) 5, (f) 10
, with the numbers on the edges showing the multiplicity of that edge

Note that algorithm 2.3 will only be able to generate a tree, as each vertex can only be attached to exactly one vertex which was produced earlier in the process. This is a clear weakness when we're modelling real-world models, which often have cycles. Athreya et al suggested a means to convert a tree to a general graph [3], which they claim preserve a number of the qualities of the Generalised Barabási-Albert algorithm. We include here the algorithm, and a number of simulations of this model in Figure 2.7, but do not analyse any of the theoretical results.

On top of the parameters required to generate a random Barabási-Albert tree, we require a probability distribution, called the grouping distribution, on the positive integer, $(q_i)_i \in \mathbb{N}$.

Algorithm 2.4.

- *Use algorithm 2.3 to generate a random tree T on n vertices from the generalised Barabási-Albert algorithm*

- *Generate a series of realisations* $(Q_j)_{j\leq N}$ *from the probability distribution* (q_i)*, where* N *will be a random variable, generated such that*

$$\sum_{j\leq(N-1)} Q_j < n \leq \sum_{j\leq N} Q_j.$$

- *Let us define the cumulative sums of* $(Q_j)_{j\leq N}$ *as*

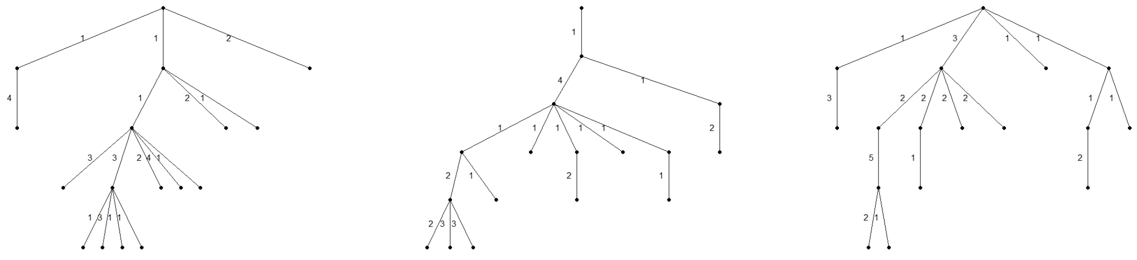
$$Q_i^+ = \sum_{j\leq i} Q_j,$$

and then we also define the partition of vertices

$$V_i = \{v_a : Q_i^+ < a \leq Q_{i+1}^+\}$$

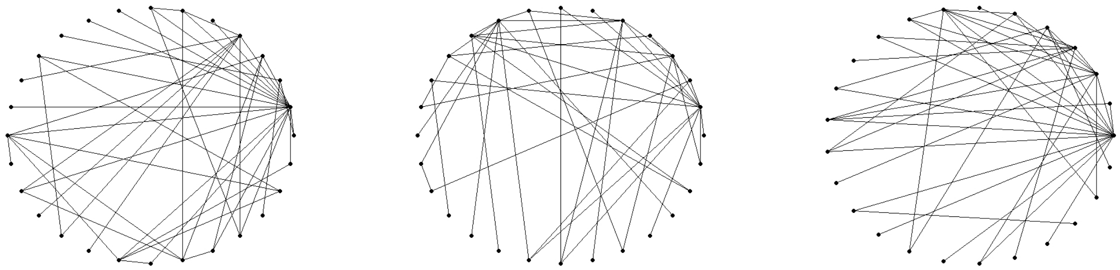
- *For the vertex set* $\{v_1', v_2' \ldots v_N'\}$*, define the adjacency matrix of our graph* G *having entries with the number of edges connecting the two vertex sets which share an index, that is*

$$a_{st} = |\{\{u,v\} \in E(G) : u \in V_s, v \in V_t\}|.$$



(a) (b) (c)

Figure 2.6: Three examples of graphs generated using the Barabási-Albert algorithm with edge distrubition Geometric($\frac{1}{2}$) and n = 15



(a) (b) (c)

Figure 2.7: Three examples of graphs generated using the Barabási-Albert algorithm, converted to a non-tree graph using algorithm 2.4 with grouping distrubition 1+Poisson(1) and $n = 50$

Chapter 3

Branching Processes and Embeddings

3.1 Markov Branching Processes

We define a *Markov branching process* Z_t with the following parameters [2]:

- initial distribution z;
- lifetime parameter $\lambda > 0$;
- offspring distribution $p' = (p'_i)_{i \in \mathbb{N} \cup 0}$;
- immigration parameter $\beta \geq 0$;
- (if $\beta > 0$) immigration distribution $p = (p_k)_{k \in \mathbb{N} \cup 0}$.

If $\beta = 0$, then we define our Markov branching process as a continuous time Markov chain with state space the non-negative integers, initial state with distribution z and Q-matrix $Q = (q_{ij})_{i,j \geq 0}$ where

$$q_{ij} = \begin{cases} -\lambda i(1 - p'_i) & i = j \\ \lambda i p'_{j-i+1} & i > 0, j \geq i - 1 \\ 0 & \text{otherwise.} \end{cases}$$

If $\beta > 0$ we say that our Markov process is *with immigration* and define it as follows. We construct a Poisson process $(N_t)_{t \geq 0}$, called the *immigration process*, with rate β. Its jump times

$$T_i = \sup\{t \in \mathbb{R}^+ : N_t < i\}$$

are called the *immigration times*. We construct a sequence of Markov branching processes without immigration $(Z_t^{(i)})_{i=0}^{\infty}$ where the initial distribution of $Z_t^{(i)}$ is z for $i = 0$ and the immigration distribution p for $i > 0$. Then, the Markov branching process with immigration is defined as

$$Z_t := \sum_{i=0}^{\infty} Z_{t-t_i}^{(i)} \mathbf{1}_{\{t > t_i\}}. \tag{3.1}$$

An intuitive system modelled by this process is a collection of particles which behave identically. The initial number of particles is distributed according to z and more arrive according to our immigration parameters. Each particle has lifetime with distribution $\text{Exp}(\lambda)$ and upon their death leave in their place zero, one or more identical copies according to the offspring distribution, p'.

We restrict our attention to a specific class of the Markov branching processes. We call such a process, Z_t *preferential* if its parameters satisfy

$$p'_j = p_{j-1} \quad \forall j \geq 1 \tag{3.2}$$

$$p_0 = p'_0 = p'_1 = 0. \tag{3.3}$$

At any time, the jump in our branching process is defined as

$$\Delta Z_t := Z_t - Z_{t^-}$$

where $Z_{t^-} = \lim_{t' \uparrow t} Z_{t'}$. Note that due to restriction 3.3 we have that $\Delta Z_t \geq 0$. Define an *event time* τ in a preferential branching process as any time such that $\Delta_\tau > 0$. The event times τ correspond exactly to the times in which either an immigration or an offspring occurs in Z_t, so we define a binary random variable for each event time τ_i, E_{τ_i} which takes value 1 at times which correspond to offspring, and 0 at times which correspond to immigration.

Then, for each $i \geq 1$,

$$\begin{aligned}
\mathbb{P}(\Delta Z_{\tau_i} = d) &= \mathbb{P}(\Delta Z_{\tau_i} = d | E_{\tau_i} = 0)\mathbb{P}(E_{\tau_i} = 0) + \mathbb{P}(\Delta Z_{\tau_i} = d | E_{\tau_i} = 1)\mathbb{P}(E_{\tau_i} = 1) \\
&= p_d \mathbb{P}(E_{\tau_i} = 0) + p'_{d+1}\mathbb{P}(E_{\tau_i} = 1) \\
&= p_d(\mathbb{P}(E_{\tau_i} = 0) + \mathbb{P}(E_{\tau_i} = 1)) \\
&= p_d.
\end{aligned}$$

That is, for any preferential Markov branching process, the size of the jump is independent of the manner of event it corresponds to.

3.2 Embedding Theorem

Theorem 3.1 (Embedding Theorem)**.** *Consider a sequence of preferential Markov branching processes with immigration parameter β and lifetime parameter $\lambda = 1$. Let its immigration distribution be $(p_j)_{j \geq 1}$ and since our process is preferential, we have $p'_{j+1} = p_j$. Define the sequence $(D_t^{(i)})_{t \geq 0}^{i \in \mathbb{N}}$ by*

- *Start $(D_t^{(1)})_{t \geq 0}$, $(D_t^{(2)})_{t \geq 0}$ and let $\tau_{-1} = \tau_0 = 0$;*
- *Inductively define following terms by*
  - *After n processes have started, let τ_{n-1} be the next event time in any of the processes $(D_{t-\tau_{i-2}}^{(i)})_{t \geq \tau_{i-2}}$ for $i \leq n$. Suppose it occurs in the process $i = I_n$;*
  - *Define $\Delta_{n+1} = D_{\tau_{n-1} - \tau_{I_n 2}}^{(I_n)} - D_{[\tau_{n-1} - \tau_{I_n 2}]^-}^{(I_n)}$;*
  - *Start process $(D_t^{(n+1)})_{t > 0}$ at d_{n+1} and evolve it independently of the existing processes.*

Let G_n be a random graph on $n+2$ vertices generated according to algorithm 2.3, the generalised Barabási-Albert algorithm. Then

$$Z_n = \left\{ D_{(\tau_n - \tau_{i-2})}^{(i)} : 1 \leq i \leq n+2 \right\}$$

$$\tilde{Z}_n = \left\{ \deg_{G_n}(v_i) : 1 \leq i \leq n+2 \right\}$$

have the same distribution.

This property allows us to embed the degree sequence of our graph into a sequence of dependent Markov branching processes, which are often more workable. Because of this property, we will refer the sequence $(D_t^{(i)})_{t\geq 0}^{i\in\mathbb{N}}$ as defined above as a *preferential embedding.*

Proof. We prove this result by induction on n. The base case, $n = 0$, holds since

$$Z_0 = \tilde{Z}_0 = (1, 1)$$

Suppose our claim holds true for all $n' < n$.

Let $z \in \mathbb{N}^{n+1}$ and $y \in \mathbb{N}^{n+2}$. Note that if the first $(n + 1)$ elements of y differ from z at more than one index, or by an amount not equal to y_{n+2} then, considering the conditional probability mass functions,

$$P_{Z_n|Z_{n-1}=z}(y) = P_{\tilde{Z}_n|\tilde{Z}_{n-1}=z}(y) = 0.$$

Then we can restrict ourselves to the case which we only have one difference between the two variables consistent with our construction algorithms. We define the random variables I_n to be the index of the value in y that differs from z, and Δ_n to be the amount by which this differs. The observed values of these will be denoted i_n, d_n respectively.

Consider $P_{Z_n|Z_{n-1}=z}(y)$. Since each process in a preferential embedding has independent and identically distributed offspring and immigration parameters, we must have that I_n, Δ_n conditionally independent given $Z_{n-1} = z$. Thus,

$$P_{Z_n|Z_{n-1}=z}(y) = P_{\Delta_n|Z_{n-1}=z}(d_n)P_{I_n|Z_{n-1}=z}(i_n),$$

where

$$P_{\Delta_n|Z_{n-1}=z}(d_n) = p_{d_n}.$$

Furthermore, since we are considering preferential Markov branching processes,

$$\begin{aligned} P_{I_n|Z_{n-1}=z}(i_n) &= \frac{d^{(i_n)}_{\tau_n - \tau i_n - 2} + \beta}{\sum_{j=1}^{n+1}\left(d_{\tau_n - \tau_{j-2}} + \beta\right)} \\ &= \frac{z_{i_n} + \beta}{\sum_{j=1}^{n+1}(z_j + \beta)} \end{aligned}$$

by the competing clocks model of exponential lifetimes.

Consider now $P_{\tilde{Z}_n|\tilde{Z}_{n-1}=z}(y)$. By construction of Algorithm 2.3, we have that I_n, Δ_n are conditionally independent given $\tilde{Z}_{n-1} = z$. Thus

$$P_{\tilde{Z}_n|\tilde{Z}_{n-1}=z}(y) = P_{\Delta_n|\tilde{Z}_{n-1}=z}(d_n)P_{I_n|\tilde{Z}_{n-1}=z}(i_n)$$

where

$$P_{\Delta_n|\tilde{Z}_{n-1}=z}(d_n) = p_{d_n}$$

and

$$\begin{aligned} P_{I_n|Z_{n-1}=z}(i_n) &= \frac{\deg_{G_{n-1}}(v_{i_n}) + \beta}{\sum_{j=1}^{n+1}\left(\deg_{G_{n-1}}(v_j) + \beta\right)} \\ &= \frac{z_{i_n} + \beta}{\sum_{j=1}^{n+1}(z_j + \beta)} \end{aligned}$$

This gives us

$$P_{Z_n|Z_{n-1}=z}(y) = P_{\tilde{Z}_n|\tilde{Z}_{n-1}=z}(y).$$

Since, by the inductive hypothesis we have that Z_{n-1} and $\tilde{Z}_{n-1}$ are identically distributed, we conclude

$$\forall y \in \mathbb{N}^{(n+2)} P_{Z_n}(y) = P_{\tilde{Z}_n}(y)$$

as required.

□

3.3 Limiting Behaviour of Individual Vertex Degrees

Let the event times of a preferential embedding $(D_t^{(i)})_{t\geq 0}^{i\in\mathbb{N}}$ be denoted by $(\tau_j)_{j\in\mathbb{N}}$ with the size of the jump at successive jump times is given by $(\Delta_{\tau_j})_{j\in\mathbb{N}}$. Define the sequence of random variables $(\Lambda_n)_{n\in\mathbb{N}\cup\{0\}}$ by

$$\begin{aligned}\Lambda_0 &= 2 + 2\beta,\\ \Lambda_n &= \Lambda_{n-1} + (2\Delta_{\tau_n} + \beta).\end{aligned}$$

Then,

$$\begin{aligned}\Lambda_n &= 2 + 2\beta + \sum_{i=1}^{n}(\beta + 2\Delta_{\tau_i})\\ &= (n+2)\beta + \sum_{i=1}^{n+2} D^{(i)}_{\tau_n - \tau_{i-2}}\end{aligned}$$

Let $(\mathcal{F}_n)_{n\in\mathbb{N}}$ be the filtration defined by the σ-algebras generated by

$$\{D^{(i)}_{t-\tau_{j-2}} : \tau_{i-2} \leq t \leq \tau_n\}_{1\leq i\leq n} \cup \{\Delta_{\tau_i}\}_{1\leq i\leq n}$$

When we condition of $\mathcal{F}_n$, each term of Λ_n represents a parameter for an exponentially distributed random variable measuring time until the next event time in $D^{(i)}$. That is, the immigration parameter β for each of the $(n+2)$ vertices and the lifetime parameter for each running process equal to its current value.

By considering the completing clocks model of exponential lifetimes, we note that the random variables $\tau_{n+1} - \tau_n$ are distributed according to a $\text{Exp}(\Lambda_n)$ distribution when we condition on $\mathcal{F}_n$.

Lemma 3.2. *Let*

$$M_n := \tau_n - \sum_{j=1}^{n} \frac{1}{\Lambda_{j-1}}$$

Consider $\{M_n\}_{n\geq 1}$ with respect to the filtration $\{\mathcal{F}_n\}_{n\geq 1}$. Then this is a L^2-bounded martingale.

Proof. By construction of the filtration, each term is measurable in the required σ-algebra. We also have that $(M_n)_{n\in\mathbb{N}}$ is a martingale since

$$\begin{aligned}\mathbb{E}[M_n|\mathcal{F}_n] &= \mathbb{E}\left[\tau_{n+1} - \sum_{j=1}^{n+1}\frac{1}{\Lambda_{j-1}}\middle|\mathcal{F}_n\right]\\ &= \tau_n - \sum_{j=1}^{n+1}\frac{1}{\Lambda_{j-1}} + \mathbb{E}[\tau_{n+1}-\tau_n|\mathcal{F}_n]\\ &= \tau_n - \sum_{j=1}^{n}\frac{1}{\Lambda_{j-1}}\\ &= M_{n-1}.\end{aligned}$$

In particular, $\mathbb{E}[M_n] = M_0 = 0$.

We show that this martingale is bounded in L^2 by showing that each term has finite variance. First, we derive a useful result about covariances that for $i < j$

$$\begin{aligned}\mathrm{Cov}(M_i - M_{i-1}, M_j - M_{j-1}) &= \mathbb{E}[(M_i - M_{i-1})(M_i - M_{i-1})] - \mathbb{E}[M_i - M_{i-1}]\mathbb{E}[M_j - M_{j-1}]\\ &= \mathbb{E}\Big[(M_i - M_{i-1})\mathbb{E}[M_j - M_{j-1}|\mathcal{F}_{j-1}]\Big]\\ &= 0.\end{aligned}$$

Next, we consider the variance itself:

$$\begin{aligned}\mathrm{Var}(M_n) &= \sum_{j=1}^{n}\mathrm{Var}(M_j - M_{j-1}) + \sum_{1\le i<j\le n}\mathrm{Cov}(M_i - M_{i-1}, M_j - M_{j-1})\\ &= \sum_{j=1}^{n}\bigg(\mathbb{E}\Big[(M_j - M_{j-1})^2\Big] - \Big(\mathbb{E}[M_j - M_{j-1}]\Big)^2\bigg)\\ &= \sum_{j=1}^{n}\mathbb{E}\Big[(\tau_j - \tau_{j-1} - \frac{1}{\Lambda_{j-1}})^2\Big].\end{aligned}$$

But, by the tower property:

$$\begin{aligned}\mathrm{Var}(M_n) &= \sum_{j=1}^{n}\bigg(\mathbb{E}\Big[\mathbb{E}[(\tau_j - \tau_{j-1})^2|\mathcal{F}_{j-1}]\Big] - mathbbE\Big[\frac{2}{\Lambda_{j-1}^2}\Big] + \mathbb{E}\Big[\frac{1}{\Lambda_{j-1}^2}\Big]\bigg)\\ &= \sum_{j=1}^{n}\mathbb{E}\Big[\frac{1}{\Lambda_{j-1}^2}\Big]\\ &= \sum_{j=1}^{n}\mathbb{E}\Big[\frac{1}{\Lambda_{j-1}}\mathbb{E}[\tau_j - \tau_{j-1}|\mathcal{F}_{j-1}]\Big].\end{aligned}$$

Since we know that the term in the conditional expectation is given by a known exponential distribution, we can rewrite it using the integral definion of an expectation:

$$\begin{aligned}
\mathrm{Var}(M_n) &= \sum_{j=1}^{n} \mathbb{E}\Big[\int_0^\infty x\mathrm{e}^{-x\Lambda_{j-1}}\mathrm{d}x\Big] \\
&= \sum_{j=1}^{n} \mathbb{E}\Big[\int_0^\infty x\mathrm{e}^{-x(2+2\beta)}\left(\mathrm{e}^{-2x\Delta_{\tau_1}}\mathrm{e}^{-x\beta}\right)^{j-1}\mathrm{d}x\Big] \\
&\leq \int_0^\infty x\mathrm{e}^{-x(2+2\beta)}\sum_{j=1}^{n}\left(\mathbb{E}\Big[\mathrm{e}^{-2x\Delta_{\tau_1}}\Big]\mathrm{e}^{-x\beta}\right)^{j-1}\mathrm{d}x \\
&= \int_0^\infty \frac{x\mathrm{e}^{-x(2+2\beta)}}{1-\mathbb{E}\Big[\mathrm{e}^{-2x\Delta_{\tau_1}}\Big]\mathrm{e}^{-x\beta}}\mathrm{d}x
\end{aligned}$$

Since at both limits, the integrand *is* integrable, it must be finite. Since the variance of each term is bounded by a constant independent of n, we conclude that the martingale is bounded in L^2 as required. □

Lemma 3.3. *Suppose $\mu := \sum jp_j < \infty$. Then $\tau_n \uparrow \infty$ almost surely as $n \to \infty$*

Proof. Define the parameter $\theta := \frac{1}{2\mu=\beta}$

Since we have that $\{M_n\}$ is a bounded martingale with respect to the filtration $\mathcal{F}_n$, and hence by Doob's Forward Convergence Theorem that it converges almost surely [17].

Note that

$$\Lambda_j = (2+2\beta) + \sum_{i=1}^{j}(2\Delta_{\tau_i}+\beta) \leq \sum_{i=1}^{j}(2\Delta_{\tau_i}+\beta)$$

So by Kolmogorov's Strong Law of Large Numbers, we have that

$$\frac{1}{j}\sum_{i=1}^{j}(2\Delta_{\tau_i}+\beta) \to 2\mu+\beta \quad \text{a.s.}$$

as $j \to \infty$. Then

$$\Lambda_j \leq \frac{j}{\theta}$$

for large j, so we have that for there exists $N \in \mathbb{N}$ such that for $n \geq N$ we get

$$\sum_{j=1}^{n}\frac{1}{\Lambda_j} \geq \sum_{j=1}^{N}\frac{1}{\Lambda_j} + \theta\left(\sum_{j=N+1}^{n}\frac{1}{j}\right) \to \infty \quad \text{as} \quad n \to \infty$$

However, since

$$M_n = \tau_n - \sum_{j=1}^{n}\frac{1}{\Lambda_{j-1}}$$

is bounded, this means that

$$\tau_n \uparrow \infty \quad \text{as} \quad n \to \infty$$

as required.

□

Lemma 3.4. *Suppose* $\sum_{j=1}^{\infty}(j \log j) < \infty$*. Then there exists a real random variable* Y *such that*

$$\lim_{n\to\infty}\left(\tau_n - \sum_{j=1}^{n}\left(\frac{\theta}{j}\right)\right)$$

exists almost and equals Y*.*

Proof. By the Lemma 3.3, the random variables

$$M_n = \tau_n - \sum_{j=1}^{n}\frac{1}{\Lambda_{j-1}}$$

form a martingale with respect to the filtration $\mathcal{F}_n$ which is bounded in L^2. Therefore, by Doob's Forward Convergence Theorem we have that it converges almost surely to a random variable, say M_∞.

Karlin and Athreya [4] proved that for

$$S_j = c + \sum_{i=1}^{j} U_i,$$

where U_i are independent, identically distributed random variables with non-zero mean, and c is a positive constant, we have that

$$\lim_{n\to\infty}\left(\sum_{j=1}^{n}\left(\frac{1}{S_j} - \frac{1}{j\mathbb{E}[U_1]}\right)\right) \quad \text{exists} \iff \mathbb{E}[U_1|\log(U_1)|] < \infty \tag{3.4}$$

Recall

$$\Lambda_j = 2 + 2\beta + \sum_{i=1}^{j}(\beta + 2\Delta_{\tau_i}),$$

where Δ_{τ_i} are independently, indentically distributed variables, and that

$$\mathbb{E}[\Delta_{\tau_i}\log(\Delta_{\tau_i})] = \sum_{j=1}^{\infty} j\log(j)p_j < \infty$$

by our assumptions.

We can apply equation (3.4) to our random variable to get that for some random variable Y', almost surely

$$\sum_{j=1}^{n}\left(\frac{1}{\Lambda_j} - \frac{\theta}{j}\right) \to Y' \quad \text{as} \quad n \to \infty.$$

Setting

$$\begin{aligned}
Y :&= M_\infty + Y' + \frac{1}{2+2\beta} \\
&= \lim_{n\to\infty}\left(\tau_n - \sum_{j=1}^{n}\frac{1}{\Lambda_{j-1}}\right) + \lim_{n\to\infty}\left(\sum_{j=1}^{n}\left(\frac{1}{\Lambda_j} - \frac{\theta}{j}\right)\right) \\
&= \lim_{n\to\infty}\left(\tau_n - \sum_{j=1}^{n}\left(\frac{\theta}{j}\right)\right)
\end{aligned}$$

which exists almost surely by the algebra of limits, as required.

□

Lemma 3.5. *Suppose Z_t is a preferential Markov branching process with $Z_0 \geq 1, \mathbb{E}[Z_0] < \infty$. Then*

$$\lim_{t\to\infty} Z_t \mathrm{e}^{-\mu t}$$

converges almost surely and is supported on $(0, \infty)$

Proof. First consider $\beta = 0$ and $Z_0 = 1$. Let $\phi(t) := \mathbb{E}[Z_t]$. Conditioning on the first event gives us

$$\begin{aligned}
\phi(t) &= \mathrm{e}^{-t} + \int_0^t \mathrm{e}^{-s}\mathbb{E}[Z_t|\tau_1 = s]\mathrm{d}s \\
&= \mathrm{e}^{-t} + \int_0^t \mathrm{e}^{-s}\sum_{j=1}^{\infty} p_j \mathbb{E}[Z_t|\tau_1 = s, \Delta_{\tau_1} = j]\mathrm{d}s \\
&= \mathrm{e}^{-t} + \int_0^t \mathrm{e}^{-s}\sum_{j=1}^{\infty} p_j (j+1)\phi(t-s)\mathrm{d}s \\
&= \mathrm{e}^{-t} + \int_0^t \mathrm{e}^{r-t}(\mu+1)\phi(r)\mathrm{d}r.
\end{aligned}$$

Hence

$$\begin{aligned}
&\mathrm{e}^t\phi(t) = 1 + (\mu+1)\int_0^t \mathrm{e}^r\phi(r)\mathrm{d}r \\
\implies &\left\{\begin{array}{l} \phi(0) = \mathbb{E}[Z_0] = 1 \\ \phi'(t) = \mu\phi(t) \end{array}\right. \\
\implies &\phi(t) = \mathrm{e}^{\mu t}. \qquad (3.5)
\end{aligned}$$

Let $M_t := Z_t\mathrm{e}^-\mu t$. We aim to show that this is a martingale with respect to the natural filtration $\mathcal{F}_t := \sigma(Z_s, s \leq t$ and sufficient information about the geneology of our branching process). Let $s \leq t$, then

$$\begin{aligned}
\mathbb{E}[M_t|\mathcal{F}_s] &= \mathrm{e}^{-\mu t}\mathbb{E}[\sum_{j=1}^{Z_s}\text{number of descendents at time t of individual j at time}|\mathcal{F}_\int] \\
&= \mathrm{e}^{-\mu t}\sum_{j=1}^{Z_s}\mathbb{E}[\text{number of descendents at time t of individual j at time}] \\
&= \mathrm{e}^{-\mu t}\sum_{j=1}^{Z)s}\mathrm{e}^{\mu(t-s)}
\end{aligned}$$

by Equation 3.5. So,

$$\mathbb{E}[M_t|\mathcal{F}_s] = \mathrm{e}^{-\mu t} Z_s \mathrm{e}^{\mu(t-s)} = M_s.$$

That is we have the martingale property as required. We can therefore apply Doob's Forward Convergence Theorem [17] to show that it the limit exists in this case.

Let the *discretisation* of a branching process $(Z_t)_{t\geq 0}$, denoted by $(\widetilde{Z_n})_{n\in\mathbb{N}}$ be the discrete time stochastic process defined by taking the value of the continuous time branching process at unit intervals, that is $\widetilde{Z_n} = Z_n$ for $n \in \mathbb{N}$. Note that this is a Galton-Watson branching process with the offspring distribution equal to the distribution of Z_1. Specifically, this means that the mean of the offspring distribution of the discretised process is given by

$$\begin{aligned}\mathbb{E}[Z_1] &= \phi(1) \\ &= \mathrm{e}^{\mu}. \end{aligned} \tag{3.6}$$

If $(Z_t)_{t\geq 0}$ is preferential, then it is non-decreasing and we have that $\widetilde{Z_n}$ is supercritical, that is that the mean of its offspring distribution, m, is at least one. Lyons et al [12] documented the Kesten-Stigum theorem that for a supercritical Galton-Watson branching process with mean offspring m and

$$\lim_{n\to\infty} \frac{\widetilde{Z_n}}{m^n} = W,$$

then

$$\mathbb{E}[W] = 1 \iff \sum_{j\geq 1} p_j j \log j < \infty. \tag{3.7}$$

By our assumption, we know that the sum is bounded. Furthermore, in this case, we know that our mean from Equation 3.6, so from Equation 3.7 we conclude

$$\mathbb{E}[\lim_{n\to\infty} \widetilde{Z_n} \mathrm{e}^{-\mu n}] = 1$$

Since our branching process is monotonic, and so well-behaved between the integer steps in our discretised process, we can extend this result to the continuous-time Markov branching process, which shows that the limit cannot be trivial, that is

$$\lim_{t\to\infty} Z_t \mathrm{e}^{-\mu t} > 0,$$

as required for the case when $\beta > 0$ and $Z_0 = 1$.

We can extend this to any value of Z_0, by considering the branching process as a sum of Z_0 independent, identically distributed Markov branching processes starting at 1. Let $(Z_t^{(i)})_{t\geq 0}$ be such a collection of processes. Then

$$\begin{aligned}\lim_{t\to\infty} Z_t e^{-\mu t} &= \lim_{t\to\infty} \sum_{i=0}^{Z_0} Z_t^{(i)} e^{-\mu t} \\ &= \sum_{i=0}^{Z_0} \lim_{t\to\infty} Z_t^{(i)} e^{-\mu t} \\ &\in (0, \infty),\end{aligned}$$

by our earlier, simpler case.

Consider now the case in which we allow immigration. Then we have from the definition of Markov branching processes, Equation 3.1, that

$$Z_t := \sum_{i=0}^{\infty} Z_{t-T_i}^{(i)} \mathbf{1}_{\{t>t_i\}},$$

where $Z_{t-T_i}^{(i)}$ are independent, identically distributed random variables and T_i are the immigration times modelled by the arrival times of a Poisson process with parameter β. Let

$$W_s^{(i)} := \sup_{t\geq 0} Z_t^{(i)}(e)^{-\mu t}.$$

Since the $(Z_t^{(i)})_{t\geq 0}$ have the same distribution, we use W_s as a random variable with this distribution when we are not concerned about the branching process it comes from.

Since T_i are the arrival times of a Poisson process, we can write them as the sum of the interarrival times

$$T_i = \sum_{j=1}^{i} (T_i - T_{i-1}),$$

where these are distributed according to $\text{Exp}(\beta)$. Consider the moment generating function of these random variable,

$$\begin{aligned} M_{T_i}(s) &:= \mathbb{E}[\mathrm{e}^{sT_i}] \\ &= \mathbb{E}[\mathrm{e}^{s\sum_{j=1}^{i}(T_i - T_{i-1})}] \\ &= \prod_{j=1}^{i} \mathbb{E}[\mathrm{e}^{s(T_i - T_{i-1})}] \\ &= \prod_{j=1}^{i} M_{(T_i - T_{i-1})}(s), \end{aligned}$$

where these are each independent exponential distributions, so

$$M_{T_i}(s) = \left(\frac{\beta}{\beta - s}\right)^i \tag{3.8}$$

Consider

$$\begin{aligned} \mathbb{E}\left[\sum_{i=0}^{\infty} W_s^{(i)}\right] &= \mathbb{E}[W_s]\mathbb{E}\left[\sum_{i=0}^{\infty} \mathrm{e}^{-\mu T_i}\right] \\ &= \mathbb{E}[W_s]\left(\sum_{i=0}^{\infty}\left(\frac{\beta}{\beta+\mu}\right)^i\right) \quad \text{by Equation 3.8} \\ &= \mathbb{E}[W_s]\left(1 + \frac{\beta}{\mu}\right) \\ &< \infty, \end{aligned}$$

where the finiteness of $\mathbb{E}[W_s]$ comes from Proposition 2.6 in Athreya et al [3]. From this, we derive that

$$\sum_{i=0}^{\infty} W_s^{(i)} \mathrm{e}^{-\mu t} < \infty \quad \text{almost surely} \tag{3.9}$$

We consider

$$\lim_{t\to\infty} Z_t \mathrm{e}^{-\mu t} = \lim_{t\to\infty} \left(\sum_{i=0}^{\infty} Z^{(i)}_{t-T_i} \mathbf{1}_{\{T_i \leq t\}} \mathrm{e}^{-\mu(t-T_i)} \right) \mathrm{e}^{-\mu T_i}.$$

We can apply the dominated convergence theorem, with our choice of dominant

$$W_s \mathrm{e}^{-\mu T_i}$$

justified by the boundedness we derived in Equation 3.9, so we have

$$\begin{aligned} \lim_{t\to\infty} Z_t \mathrm{e}^{-\mu t} &= \sum_{i=0}^{\infty} \lim_{t\to\infty} \left(Z^{(i)}_{t-T_i} \mathbf{1}_{\{T_i \leq t\}} \mathrm{e}^{-\mu(t-T_i)} \right) \mathrm{e}^{-\mu T_i} \\ &\leq \sum_{i=0}^{\infty} W^{(i)}_s \mathrm{e}^{-\mu T_i}, \end{aligned}$$

so our sequence of scaled branching processes converges to a finite limit.

If we consider $(Z_t)_{t\geq 0}$ and form a new preferential branching process by considering the case in which we follow only the original Z_0 particles and their offspring, and ignore the effects of immigration, we have a branching process Y_t. We have already shown that such a process converges to a limit on $(0, \infty)$. Furthermore, Y_t which is less than or equal to Z_t for all values of t, so we conclude

$$\lim_{t\to\infty} Z_t \mathrm{e}^{-\mu t} \leq \lim_{t\to\infty} Y_t \mathrm{e}^{-\mu t} < 0,$$

so the limit is on $(0, \infty)$ almost surely as required.

□

Theorem 3.6 (Limiting Behaviour for Preferential Branching Processes)**.** *Suppose* $\sum_{j=1}^{j=\infty} (j \log j) p_j < \infty$ *and let*

$$\alpha := \frac{\mu}{2\mu + \beta}$$

. Let $(D^{(i)}_t)^{i\in\mathbb{N}}_{t\geq 0}$ *be a preferential embedding. Then*

$$\lim_{n\to\infty} \left(\frac{D^{(i)}_{\tau_n}}{n^{\alpha}} \right)$$

converges almost surely to a random variable γ_i *on* $(0, \infty)$

Proof. Consider

$$D^{(i)}_{\tau_n - \tau_{i-2}} n^{-\alpha} = D^{(i)}_{\tau_n - \tau_{i-2}} \mathrm{e}^{-\mu(\tau_n - \tau_{i-2})} \cdot \mathrm{e}^{\mu \tau_n} n^{-\alpha} \cdot \mathrm{e}^{-\mu \tau_{i-2}}$$

We aim to show that this is the product of variables that converge almost surely to a random variable in $(0, \infty)$, so that by the algebra of limits the limit of the product must also converge to a random variable on $(0, \infty)$.

The term $\mathrm{e}^{-\mu \tau_{i-2}}$ does not depend on n and so we need only consider the rest of the product Note that by Lemma 3.5, we have that

$$\lim_{t\to\infty} Z_t \mathrm{e}^{-\mu t}$$

exists almost surely under some conditions, which we have here. We can apply this to $Z = D^{(i)}$. Also, by Lemma 3.3 we have that $\tau_n \uparrow \infty$ as $n \to \infty$

$$\lim_{n\to\infty} D_{\tau_n-\tau_{i-2}} \mathrm{e}^{-m(\tau_n-\tau_{i-2})}$$

exists and is in $(0,\infty)$ as required.

Recall from Lemma 3.4 that

$$\lim_{n\to\infty}\left(\sum_{j=1}^{n}\left(\frac{1}{j(2\mu+\beta)}\right)\right)\to Y$$

for $Y\in(0,\infty)$.

$$\begin{aligned}\mathrm{e}^{\mu\tau_n}n^{-\alpha} &= \mathrm{e}^{\mu\tau_n-\alpha\log(n)}\\ &\to \mathrm{e}^{\mu Y+\alpha\gamma},\end{aligned}$$

where γ is Euler's constant, defined as

$$\lim_{n\to\infty}\left(\sum_{j=1}^{n}\frac{1}{j}-\log(n)\right)$$

This also exists in $(0,\infty)$, and so we conclude that the entire product converges almost surely to a limit in the same interval.

□

Corollary 3.7 (Limiting Vertex Distribution of Graphs)**.** *Let $(G_n)_{n\in\mathbb{N}\cup\{0\}}$ be a sequence of graphs generated using the Algorithm 2.3. Let $\mu=\sum jp_j$ be the mean number of edges added on each iteration and let $\alpha=\frac{\mu}{2\mu+\beta}$. Suppose that $\sum(j\log j)p_j<\infty$, then for each $i\geq 1$ we have that*

$$\lim_{n\to\infty}\frac{1}{n^{\alpha}}\deg_{G_n}(v_i)$$

converges almost surely to a random limit $\gamma_i\in(0,\infty)$.

Proof. Follows directly from Theorem 3.1 and Corollary 3.6 □

3.4 Limiting Behaviour of Empirical Degree Distribution

The previous result informs us about the probability that an individual vertex has a particular degree as we increase the number of vertices in a random graph. However, the properties that we were particularly interested in was that the number of vertices with a particular degree follows a power law. To do so, we prove two key results, firstly we prove that the empirical degree distribution converges in probability to a particular distribution; then after we prove that such a distribution follows the power law we are after. Again, we will do this in the context of preferential branching processes, and then use the Embedding theorem to carry this behaviour over to our Barabási-Albert random graphs.

Consider a preferential embedding $D:=(D_t^{(i)})_{t\geq 0}^{i\in\mathbb{N}}$. Let

$$R_j(n):=|\{i:D^{(i)}_{\tau_n-\tau_{i-2}}=j\}|,$$

that is, the number of branching processes within our embedding that are equal to j. Our first theorem gives us the limiting behaviour of the $\frac{1}{n}R_j(n)$, and is proved by writing it as a series of differences, most of which converge to zero. We prove the limiting behaviour of these terms as lemmas, then proceed to the main theorem.

Define the probability mass function

$$f_j(y) := \mathbb{P}(D_y^{(1)} = j),$$

which since each of the degrees has the same distribution, is also the probability mass function of each of the other Markov branching processes in ur embedding.

Lemma 3.8. *Let $D := (D_t^{(i)})_{t\geq 0}^{i\in\mathbb{N}}$ be a preferential embedding with offspring distribution $(p_j)_{j\in\mathbb{N}}$ such that $\mu := \sum jp_j < \infty$ and $\sum j\log(j)p_j < \infty$. Recall $\theta := (2\mu+\beta)^{-1}$. Then*

$$\begin{aligned}J_1(n) &:= \frac{1}{n}\sum_{i=1}^{n+2}\left(\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta\log\frac{n}{(i-2)\vee 1}} = j\}\right)\\ &\to 0\end{aligned}$$

in probability as $n\to\infty$.

Proof. Let $\epsilon > 0$. Consider the set

$$S_\epsilon(n) := \left\{|\tau_n - \tau_{i-2} - \theta\log\frac{n}{(i-2)\vee 1}| : n\epsilon + 2 \leq i \leq n+2\right\}.$$

For $\delta > 0$, denote by $A_\epsilon(n,\delta)$ the event that

$$\sup_n S_\epsilon(n) > \delta.$$

From lemma 3.4, we know that

$$\tau_n - \sum_{j=1}^{n}\frac{\theta}{j}$$

converges almost surely to a limit, say, Y. Then

$$\begin{aligned}\tau_n - \theta\log n &= \left(\tau_n - \sum_{j=1}^{n}\frac{\theta}{j}\right) + \theta\left(\sum_{j=1}^{n}\frac{1}{j} - \log n\right)\\ &\to Y + \theta\gamma\end{aligned}$$

almost surely, where γ is Euler's constant. Since this is a convergent series, it must also be Cauchy, so we have that

$$\lim_{\delta\downarrow 0}\lim_{n\to\infty}\mathbb{P}[A_\epsilon(n,\delta)] = 0. \tag{3.10}$$

Note that

$$\begin{aligned}n\epsilon + 2 \leq i \leq n+2 &\implies 0 \leq \log\frac{n}{i-2} \leq -\log\epsilon\\ &\implies 0 \leq \theta\log\frac{n}{i-2} \leq -\theta\log\epsilon\end{aligned}$$

Consider a typical element of the sum $J_1(n)$, say

$$\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}} = j\}.$$

We aim to show that the absolute value of this has sufficiently small expectation, and so by the method of first moments, it must converge to zero in probability.

$$\begin{aligned}\mathbb{E}\left[\left|\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}} = j\}\right|\right] &= \mathbb{P}\left(\left|\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}} = j\}\right| = 1\right) \\ &\leq \mathbb{P}\left(\left|D^{(i)}_{\tau_n-\tau_{i-2}} - D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}}\right| \geq 1\right)\end{aligned}$$

Note that

$$\mathbb{P}(A_1) = \mathbb{P}(A_1 \cap A_2) + \mathbb{P}(A_1 \cap A_2^c) \leq \mathbb{P}(A_1 \cap A_2) + \mathbb{P}(A_2^c).$$

Applying this principle to the previous probability:

$$\mathbb{P}\left(\left|D^{(i)}_{\tau_n-\tau_{i-2}} - D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}}\right| \geq 1\right) \leq \mathbb{P}\left(\left|D^{(i)}_{\tau_n-\tau_{i-2}} - D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}}\right| \geq 1 \cap A^c_\epsilon(n,\delta)\right) + \mathbb{P}(A_\epsilon(n,\delta))$$

Recall, by the definition of the event $A_\epsilon(n,\delta)$, if it does *not* hold, we have that

$$S_\epsilon(n) \leq \delta \implies \left(\theta \log \frac{n}{(i-2)\vee 1} - \delta\right) \vee 0 \leq \tau_n - \tau_{i-2} \leq \theta \log \frac{n}{(i-2)\vee 1} + \delta$$

Note that we also have that

$$0 \leq \theta \log \frac{n}{(i-2)\vee 1} \leq -\theta \log \epsilon,$$

so

$$\begin{aligned}\mathbb{P}\left(\left|D^{(i)}_{\tau_n-\tau_{i-2}} - D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}}\right| \geq 1\right) &\leq \mathbb{P}\left(\left|D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}+\delta} - D^{(i)}_{\left(\theta \log \frac{n}{(i-2)\vee 1}-\delta\right)\vee 0}\right| \geq 1 \cap A^c_\epsilon(n,\delta)\right) + \\ &\qquad \mathbb{P}(A_\epsilon(n,\delta) \\ &\leq \sup_{x\in[0,-\theta\log\epsilon]} \mathbb{P}(\{D^{(i)}_{x+\delta} - D^{(i)}_{(x-\delta)\vee 0} \geq 1\} \cap A^c_\epsilon(n,\delta)) + \mathbb{P}(A_\epsilon(n,\delta)) \\ &:= d^{(i)}_\epsilon(n,\delta)\end{aligned}$$

Note that since D is non-explosive, we have that

$$\lim_{\delta\downarrow 0} \mathbb{P}\left(\{D^{(i)}_{x+\delta} - D^{(i)}_{(x-\delta)\vee 0} \geq 1\}\right) = 0$$

for all x. Because of this, we can see that

$$\lim_{\delta\downarrow 0}\lim_{n\to\infty} d^{(i)}_\epsilon(n,\delta)$$

exists, and equals zero.

Now,

$$\begin{aligned}
\mathbb{E}[|J_1(n)] &= \mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^{n+2}\left(\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta\log\frac{n}{(i-2)\vee 1}} = j\}\right)\right|\right] \\
&\le \frac{1}{n}\sum_{i=1}^{n+2}\mathbb{E}\left[\left|\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta\log\frac{n}{(i-2)\vee 1}} = j\}\right|\right] \\
&\le \frac{n\epsilon+2}{n} + \frac{1}{n}\sum_{i=n\epsilon+2}^{n+2}\mathbb{E}\left[\left|\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} - \mathbf{1}\{D^{(i)}_{\theta\log\frac{n}{(i-2)\vee 1}} = j\}\right|\right] \\
&= \epsilon + \frac{2}{n} + (1-\epsilon)d^{(i)}_{\epsilon}(n,\delta)
\end{aligned}$$

Then

$$\begin{aligned}
\lim_{\delta\downarrow 0}\lim_{n\to\infty}\mathbb{E}[|J_1(n)] &\le \lim_{\delta\downarrow 0}\lim_{n\to\infty}\left(\epsilon + \frac{2}{n} + (1-\epsilon)d^{(i)}_{\epsilon}(n,\delta)\right) \\
&= \epsilon + (1-\epsilon)\lim_{\delta\downarrow 0}\lim_{n\to\infty} d^{(i)}_{\epsilon}(n,\delta) \\
&= \epsilon
\end{aligned}$$

Since $|J_1(n)| \ge 0$, and we fixed ϵ arbitrarily, we conclude that

$$\lim_{n\to\infty}|J_1(n)| = 0 \quad \text{in probability,}$$

as required.

□

Lemma 3.9. *Let $D := (D^{(i)}_t)^{i\in\mathbb{N}}_{t\ge 0}$ be a preferential embedding with offspring distribution $(p_j)_{j\in\mathbb{N}}$ such that $\mu := \sum jp_j < \infty$ and $\sum j\log(j)p_j < \infty$. Recall $\theta := (2\mu+\beta)^{-1}$. Then*

$$\begin{aligned}
J_2(n) &:= \frac{1}{n}\sum_{i=1}^{n+2}\left(\mathbf{1}\{D^{(i)}_{\theta\log\frac{n}{(i-2)\vee 1}} = j\} - f_j\left(\theta\log\frac{n}{(i-2)\vee 1}\right)\right) \\
&\to 0
\end{aligned}$$

in probability as $n\to\infty$.

Proof. Let $\epsilon > 0$ and let

$$\phi_i(n) := \mathbf{1}\{D^{(i)}_{\theta\log\frac{n}{(i-2)\vee 1}} = j\} - f_j(\theta\log\frac{n}{(i-2)\vee 1}),$$

so

$$J_2(n) = \frac{1}{n}\sum_{i=1}^{n+2}\phi_i(n).$$

Furthermore,

$$\begin{aligned}\mathbb{E}[\mathbf{1}\{D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}} = j\}] &= \mathbb{P}(D^{(i)}_{\theta \log \frac{n}{(i-2)\vee 1}} = j) \\ &= f_j(\theta \log \frac{n}{(i-2)\vee 1}),\end{aligned}$$

which gives us

$$\mathbb{E}[\phi_i(n)] = 0. \tag{3.11}$$

By Equation 3.11, $\phi_i(n)$ is centred on zero, so applying Chebyshev's inequality gives

$$\begin{aligned}\mathbb{P}(|J_2(n)| > \epsilon) &\le \frac{\mathrm{Var}(J_2(n))}{\epsilon^2} \\ &= \frac{\mathbb{E}[J_2(n)^2]}{\epsilon^2}.\end{aligned}$$

We now work out the second moment.

$$\begin{aligned}\mathbb{E}[J_2(n)^2] &= \frac{1}{n^2}\mathbb{E}\left[\left(\sum_{a=1}^{n+2}\phi_a(n)\right)^2\right] \\ &= \frac{1}{n^2}\mathbb{E}\left[\sum_{a=1}^{n+2}\phi_a(n)^2 + \sum_{a=1}^{n+2}\sum_{b\neq a}\phi_a(n)\phi_b(n)\right] \\ &= \frac{1}{n^2}\mathbb{E}\left[\sum_{a=1}^{n+2}\phi_a(n)^2\right] + \frac{1}{n^2}\sum_{a=1}^{n+2}\sum_{b\neq a}\mathbb{E}\left[\phi_a(n)\phi_b(n)\right].\end{aligned} \tag{3.12}$$

Consider now for $a \neq b$,

$$\mathbb{E}\left[\phi_a(n)\phi_b(n)\right].$$

Without loss of generality, consider the case where $a > b$.

$$\mathbb{E}\left[\phi_a(n)\phi_b(n)\right] = \mathbb{E}\left[\mathbb{E}[\phi_a(n)|\mathcal{F}_b]\phi_b(n)\right]$$

by the Tower property. However, the multiplicity of the jump times of a process is independent of an earlier filtration, so

$$\begin{aligned}\mathbb{E}\left[\phi_a(n)\phi_b(n)\right] &= \mathbb{E}\left[\mathbb{E}[\phi_a(n)]\phi_b(n)\right] \\ &= 0.\end{aligned}$$

Returning to Equation 3.12,

$$\begin{aligned}\mathbb{E}[J_2(n)^2] &= \frac{1}{n^2}\sum_{a=1}^{n+2}\phi_a(n)^2 \\ &\le \frac{n+2}{n^2} \le \frac{1}{n}\end{aligned}$$

since $|\phi_a(n)| \le 1$. So, we conclude

$$\begin{aligned}\mathbb{P}(|J_2(n)| > \epsilon) &\le \frac{\mathbb{E}[J_2(n)^2]}{\epsilon^2} \\ &\le \frac{1}{n\epsilon^2} \\ &\to 0\end{aligned}$$

as $n \to \infty$ as required.

□

Lemma 3.10. *Let $D := (D_t^{(i)})_{t \geq 0}^{i \in \mathbb{N}}$ be a preferential embedding with offspring distribution $(p_j)_{j \in \mathbb{N}}$ such that $\mu := \sum j p_j < \infty$ and $\sum j \log(j) p_j < \infty$. Recall $\theta := (2\mu + \beta)^{-1}$. Then*

$$J_3(n) := \frac{1}{n} \sum_{i=1}^{n+2} \left(f_j \left(\theta \log \frac{n}{(i-2) \vee 1} \right) - \int_0^1 f_j(-\theta \log x) \mathrm{d}x \right)$$
$$\to 0$$

as $n \to \infty$.

Proof. Consider

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^{n+2} f_j(\theta) \log \frac{b}{(i-2) \vee 1} &= \frac{2 f_j(\theta \log n)}{n} + \frac{1}{n} \sum_{i=1}^{n} f_j(\theta \log \frac{n}{i}) \\ &= \frac{2 f_j(\theta \log n)}{n} + \frac{1}{n} \sum_{i=1}^{n} f_j(-\theta \log \frac{i}{n}) \end{aligned}$$

However,

$$\frac{1}{n} \sum_{i=1}^{n} f_j(-\theta \log \frac{i}{n})$$

is an approximation to the Riemann integral, and so as $n \to \infty$,

$$\frac{1}{n} \sum_{i=1}^{n} f_j(-\theta \log \frac{i}{n}) \to \int_0^1 f_j(-\theta \log x) \mathrm{d}x.$$

So,

$$\begin{aligned} J_3(n) &= \frac{1}{n} \sum_{i=1}^{n+2} \left(f_j \left(\theta \log \frac{n}{(i-2) \vee 1} \right) - \int_0^1 f_j(-\theta \log x) \mathrm{d}x \right) \\ &= \frac{2}{n} \left[f_j(\theta \log n) - \int_0^1 f_j(-\theta \log x) \right] + \frac{1}{n} \sum_{i=1}^{n} f_j \left(-\theta \log \frac{i}{n} \right) - \int_0^1 f_j(-\theta \log x) \mathrm{d}x \\ &\to 0 \end{aligned}$$

as $n \to \infty$ as required.

□

Theorem 3.11. *Let $D := (D_t^{(i)})_{t \geq 0}^{i \in \mathbb{N}}$ be a preferential embedding with offspring distribution $(p_j)_{j \in \mathbb{N}}$ such that $\mu := \sum j p_j < \infty$ and $\sum j \log(j) p_j < \infty$. Recall $\theta := (2\mu + \beta)^{-1}$. Define*

$$\pi_j := \frac{1}{\theta} \int_0^\infty f_j(y) \mathrm{e}^{-y/\theta} \mathrm{d}y$$

Then, for $j \geq 1$, we have that

$$\frac{R_j(n)}{n} \to \pi_j$$

in probability as $n \to \infty$

Proof. Note that by construction, we have

$$\begin{aligned}\frac{R_j(n)}{n} &= \frac{1}{n}\sum_{i=1}^{n+2}\mathbf{1}\{D^{(i)}_{\tau_n-\tau_{i-2}} = j\} \\ &= J_1(n) + J_2(n) + J_3(n) + \frac{n+1}{n}\int_0^1 f_j(-\theta\log x)\mathrm{d}x\end{aligned}$$

By substituting

$$y := -\theta\log x$$

into the integral, we see that

$$\frac{R_j(n)}{n} = J_1(n) + J_2(n) + J_3(n) + \frac{n+1}{n\theta}\int_0^1 f_j(y)\mathrm{e}^{-y/\theta}\mathrm{d}y \tag{3.13}$$

As we take $n \to \infty$, the first three terms converge to zero in probability (or stronger), as we showed in Lemmas 3.8-3.10. We can derive the algebra of limits for convergence in probability by combining the algebra of limits for almost sure convergence and A13.2(e) in Williams [17], which states that if $X_n \to X$ in probability, then every subsequence of of (X_n) contains a further subsequence along which it converges almost surely to X. Applying this algebra of limits to equation 3.13, we conclude that as $n \to \infty$,

$$\begin{aligned}\frac{R_j(n)}{n} &\to \frac{1}{\theta}\int_0^1 f_j(y)\mathrm{e}^{-y/\theta}\mathrm{d}y \\ &= \pi_j\end{aligned}$$

in probability, as required.

□

3.5 Scale-Free Property

We know now what the degree sequence converges to. We aim to show that the π_j follow the power-law that we seek in our model.

Theorem 3.12. *Recall*

$$\pi_j := \frac{1}{\theta}\int_0^\infty p_j(y)\mathrm{e}^{-\frac{y}{\theta}}\mathrm{d}y.$$

Suppose

$$\sum_{j\geq 1} j^{2+\frac{\beta}{m}}p_j \leq \infty.$$

Then, for $s \geq 0$, we have that

$$\sum_{j\geq 1} j^s\pi_j < \infty \iff s < 2 + \frac{\beta}{m}$$

Proof. Let $0 \leq s \leq 2 + \frac{\beta}{\mu}$. We have by assumption that

$$\sum_{j\geq 1} j^{2+\beta/\mu} p_j < \infty.$$

Since $r := 2 + \frac{\beta}{\mu} > 1$, by Proposition 2.6 in Athreya et al [3] that for some limit C_r that

$$\begin{aligned} C_r &:= \mathbb{E}[\sup_{t\geq 0} Z_t^r \mathrm{e}^{-\mu t r}] \\ &= \mathbb{E}[(\sup_{t\geq 0} Z_t \mathrm{e}^{-\mu t})^r] \\ &< \infty. \end{aligned}$$

In paticular, for each t,

$$\mathbb{E}[Z_t^r] \leq C_r \mathrm{e}^{\mu t r}. \tag{3.14}$$

Recall from Lemma 3.5 that there exists a limit W such that

$$Z_t \mathrm{e}^{-\mu t} \to W$$

almost surely as $t \to \infty$ and so by the dominated convergence theorem [17], with the dominant

$$\sup_{t\geq 0} Z_t^r \mathrm{e}^{-\mu t r},$$

we conclude

$$\begin{aligned} \frac{\mathbb{E}[Z_t^r]}{\mathrm{e}^{\mu t r}} &= \mathbb{E}[Z_t^r \mathrm{e}^{-\mu t r}] \\ &= \mathbb{E}[(Z_t \mathrm{e}^{-\mu t})^r] \\ &\to \mathbb{E}[W^r] \end{aligned}$$

as $t \to \infty$, so that for sufficiently large $t \geq t_0(r)$ we also have

$$\mathbb{E}[Z_t^r] \geq c_r \mathrm{e}^{\mu t r} \tag{3.15}$$

for some c_r.

Consider

$$\begin{aligned} \sum_{j\geq 1} j^s \pi_j &= \sum_{j\geq 1} \left[(2\mu+\beta) j^s \int_0^\infty \mathrm{e}^{-(2\mu+\beta)y} f_j(y) \mathrm{d}y \right] \\ &= (2\mu+\beta) \int_0^\infty \mathrm{e}^{-(2\mu+\beta)y} \sum_{j\geq 1} j^s f_j(y) \mathrm{d}y \end{aligned}$$

using Tonelli's theorem to justify the interchange of sum and integral. Note that

$$\begin{aligned} \sum_{j\geq 1} j^s f_j(y) &= \mathbb{E}[Z_y^s] \\ &\in [c_s \mathrm{e}^{\mu y s}, C_s \mathrm{e}^{\mu y s}], \end{aligned}$$

where the boundedness comes from Equations 3.14 and 3.15.

So,

$$\sum_{j\geq 1} j^s \pi_j \leq (2\mu+\beta)C_s \int_0^\infty \mathrm{e}^{-(s\mu+\beta-\mu s)y}\mathrm{d}y \tag{3.16}$$

$$\text{and} \quad \sum_{j\geq 1} j^s \pi_j \geq (2\mu+\beta)c_s \int_{t_0(s)}^\infty \mathrm{e}^{-(2\mu+\beta-\mu s)y}\mathrm{d}y \tag{3.17}$$

The expressions given in equations 3.16 and 3.17 are finite if and only if

$$2\mu+\beta-\mu s > 0$$
$$\iff s < 2 + \frac{\beta}{\mu},$$

as required.

□

Corollary 3.13 (Power Law for the Limiting Degree Distribution)**.** *Let $(G_n)_{n\in\mathbb{N}\cup\{0\}}$ be a sequence of graphs generated using Algorithm 2.3. Let $\mu = \sum jp_j$ be the mean number of edges added on each iteration. Suppose that $\sum(j\log j)p_j < \infty$, and let*

$$d_n(j) := \frac{|\{v \in G_n : deg_{G_n}(v) = j\}}{n}.$$

Then

$$d_n(j) \to d_\infty(j)$$

as $n \to \infty$, where

$$d_\infty(j) = \mathrm{O}(j^{-(3+\frac{\beta}{\mu})}).$$

Proof. The version of this theorem for the preferential embeddings come from Theorems 3.11 and 3.12. Applying the embedding theorem, Theorem 3.1, allows us to transfer this result to the underlying graphs.

□

Chapter 4

Simulation

We have considered the limiting behaviour of the graph properties that occur during the generation of the Barabási-Albert random graphs, that is the individual vertex degrees and the distribution of the vertex degrees. We now turn to iterated simulation methods to gain an intuitive grasp of these results.

There are two parameters we must define when running a simulation, that is the value of β and the probability distribution that determines the number of edges we add at each step. $(p_i)_{i\in\mathbb{N}}$. For simplicity, we will fix these values at β equal to 0, and p_i generated according to a Geometric distribution with parameter $\frac{1}{2}$, that is for $n \geq 1$, that

$$\mathbb{P}(\Delta = n) = \frac{1}{2^n}.$$

4.1 Convergence of Individual Degree Distribution

The result that we developed in section 3.3, stated that

$$\lim_{n\to\infty} \frac{1}{n^\alpha} \deg_{G_n}(v_i)$$

exists almost surely, and converges to a random limit $\gamma_i \in (0, \infty)$, where

$$\alpha = \frac{\mathbb{E}[\Delta]}{2\mathbb{E}[\Delta] + \beta}.$$

In any case in which β equals 0, this simply reduces to $\frac{1}{2}$. Note first an important corollary to this theorem.

Corollary 4.1. *Let $(G_n)_{n\in\mathbb{N}\cup\{0\}}$ be a sequence of graphs generated using the preferential attachment method. Let $\mu = \sum jp_j < \infty$ be the mean number of edges added on each iteration and let $\alpha = \frac{\mu}{2\mu+\beta}$. Suppose that $\sum(j \log j)p_j < \infty$, then for each $i \geq 1$ we have that*

1.

$$\frac{1}{n^z} \deg_{G_n}(v_i) \to 0$$

almost surely, for $z > \alpha$

2.

$$\frac{1}{n^z}\deg_{G_n}(v_i) \to \infty$$

almost surely, for $z < \alpha$

Proof. Suppose $z > \alpha$, then we can write $z = \alpha + z^*$ for $z^* > 0$

$$\begin{aligned}\frac{1}{n^z}\deg_{G_n}(v_i) &= \frac{1}{n^{\alpha+z^*}}\deg_{G_n}(v_i)\\ &= \frac{1}{n^{z^*}}\left(\frac{1}{n^\alpha}\deg_{G_n}(v_i)\right)\\ &\to 0\cdot\gamma_i\\ &= 0\end{aligned}$$

by the algebra of limits.

Now suppose $z < \alpha$, then we can write $z = \alpha - z^*$ for $z* > 0$

$$\begin{aligned}\frac{1}{n^z}\deg_{G_n}(v_i) &= \frac{1}{n^{\alpha-z^*}}\deg_{G_n}(v_i)\\ &= n^{z^*}\left(\frac{1}{n^\alpha}\deg_{G_n}(v_i)\right)\\ &\to \infty\end{aligned}$$

since n^{z^*} is unbounded. □

When referring to one of the initial two vertices, as the following simulations will do, we will call the degrees, scaled by the appropriate power of the number of iterations, the *scaled initial vertex degree*, and refer to the limit defined above as the *scaled initial vertex degree limit*, which is a well-defined random variable and exists on $(0, \infty)$ almost surely.

We can use these results to see empirically that our behaviour is what the theory would predict. Given a simulation of the Barabási-Albert model, the degree of vertex v_1 is plotted in a number of realisations from the simulation against n, that is, the number of iterations of the algorithm that we use. In each case, it is scaled by $\frac{1}{n^z}$ for different values of z

The unboundedness and convergence to zero in Figure 4.1 as well as the apparent convergence in Figure 4.2 backs up the theory for these parameters.

4.2 Limiting Degree Distribution

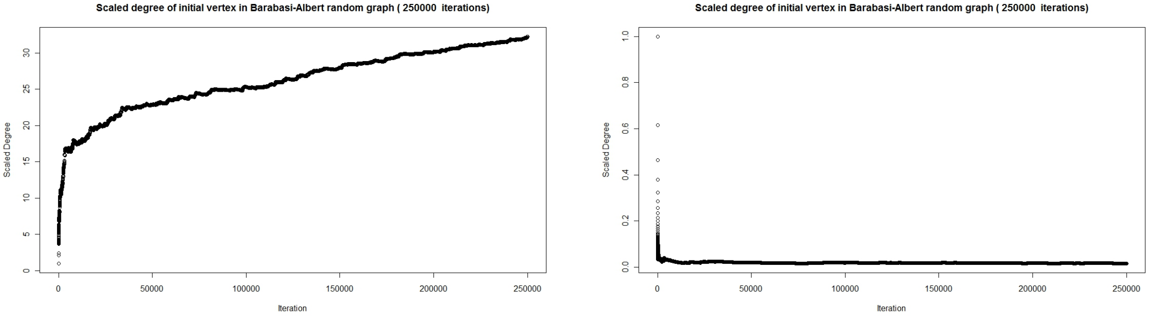
The other random variable whose limiting distribution we considered was the number of vertices which have a particular degree, that is

$$R_n(j) = |\{v \in V(G) : \deg_{G_n}(v) = j)\}|$$

as we increase n by running more iterations of Algorithm 2.3.

The code in Appendix A.2, namely the DegSeqSim() functon, was used to generate a large random graph which we would expect to follow the power law. The parameters are set as before, with

$$\beta = 0,$$



(a) (b)

Figure 4.1: Scaled initial vertex degrees for the first 250,000 iterations with $z =$ (a) 0.3, showing unboundedness and (b) 0.7, showing convergence to zero

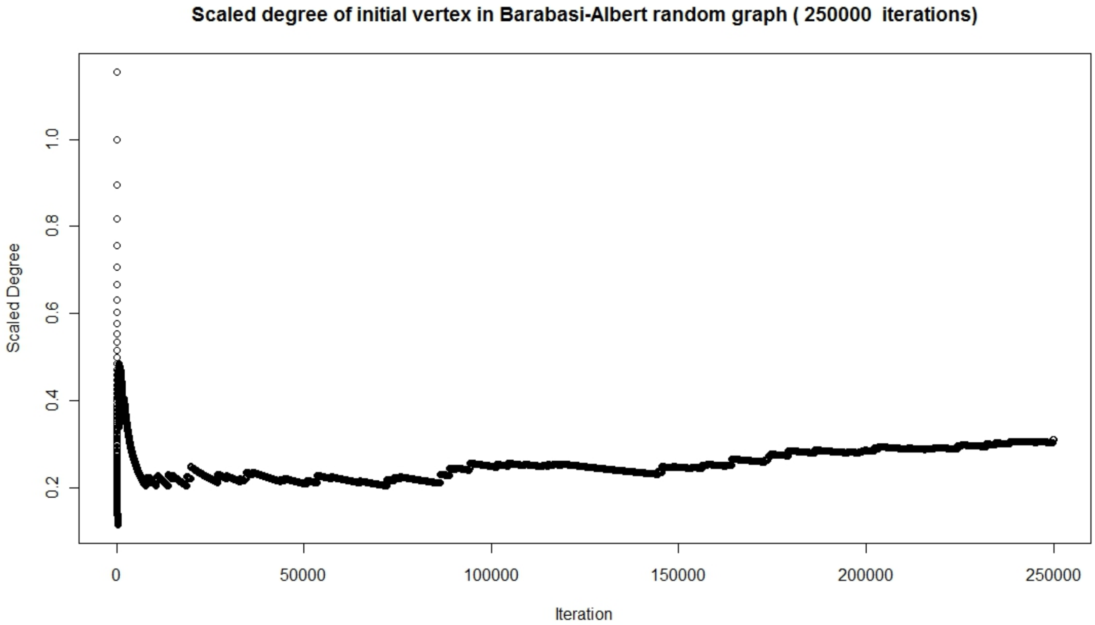


Figure 4.2: Scaled initial vertex degrees for the first 250,000 iterations with z $= \alpha = 0.5$, showing convergence to a random, bounded, non-zero limit

and

$$p_j = \frac{1}{2^j}$$
$$\implies \sum_{j\geq 1} j p_j := \mu = 2.$$

The algorithm was run with num equal to 250,000 so we end up with a graph with 250,002 vertices.

Theorem 3.13 suggests that this should be distributed according to a power law, so we plot it on log-log axes. If our hypothesis of a scale-free network is true, we should expect to see linearity. We produce this graph in Figure 4.3. We repeat the graph of the data we analysed in Section 1.2 in Figure 4.4. There is a strong similarity in shape despite the fact that the number of data points are an order of magnutde different. This provides strong motivation for the use of the Barabási-Albert random graph model.

The initial data points are slightly non-linear and give cause for concern. However, it is reasonable for there to be slight fluctuations from the power-law in extreme values of the distribution, since it is the tail behaviour that we are interested in. Clauset et al. [8] suggested a power law model that includes an extra parameter in its estimation, $x_{\min}$, and then uses maximum likelihood estimation to fit the existing model to the tail values above our value of $x_{\min}$. R code for this was produced by Clauset and Dubroca [7] and will be used to analyse the data. We first give a short summary of the method studied.

The first aim is to estimate $x_{\min}$ and discarding all points beneath this threshold value. We can then consider our points as coming from the discrete case of the exponential distribution with probability mass fuction

$$f_X(i) := \mathbb{P}(X = i) = \begin{cases} 0 & i < x_{\min} \\ \frac{i^{-r}}{\zeta(r, x_{\min})} & \text{otherwise} \end{cases}$$

where r is a parameter to be estimated, and $\zeta(r, x_{\min})$ is a normalising constant. For our work, where we are happy to work up to a constant, its exact value is not necessary, but it takes the form of the generalised Zeta function, namely

$$\zeta(r, x_{\min}) = \sum_{i\geq 0} (i + x_{\min})^{-r}.$$

In order to obtain a maximum likelihood estimator, we estimate the probability mass function (pmf) for values beneath $x_{\min}$ with their own probability parameter p_k, that is the proportion of data points which have that value, and scale our estimated power law accordingly, that is

$$\phi_X(i; x_{\min}) = \begin{cases} p_i & i < x_{\min} \\ & \\ Ci^{-r} & i \geq x_{\min} \end{cases} \tag{4.1}$$

By construction of this model, the true likelihood will increase as we allow $x_{\min}$ as it introduces more parameters to estimate, and this will hide the fact that we can nonetheless explain much of the variance with fewer parameters because of the power law. To balance between these, we fit a model for each value of $x_{\min}$ in our data set, and choose as the estimator $\widehat{x}_{\min}$ that minimises the Kolmogorov-Smirnov statistic, namely

$$\sup_{x\geq x_{\min}} |\Phi_X(x) - O_X(x)|, \tag{4.2}$$

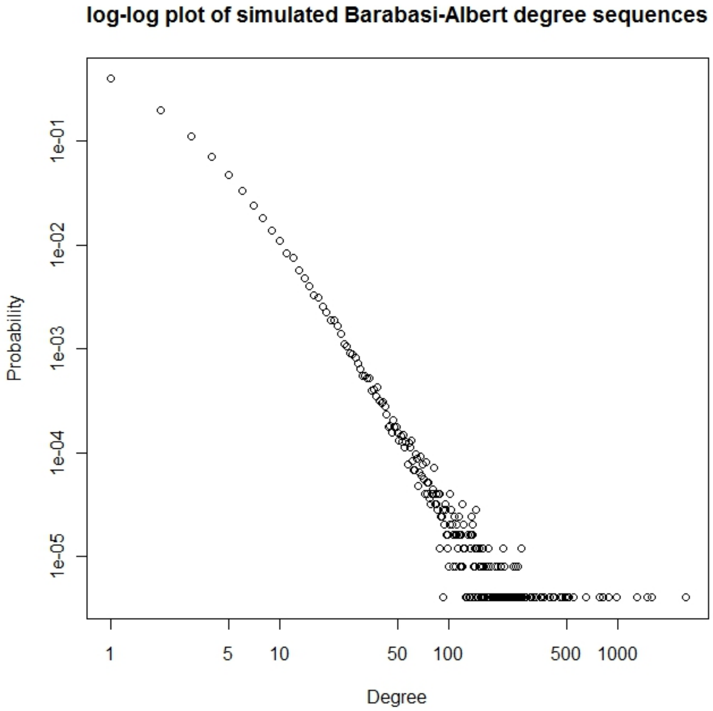


Figure 4.3: A log-log graph showing the frequency of vertices with specific degrees in a Barabási-Albert random graph

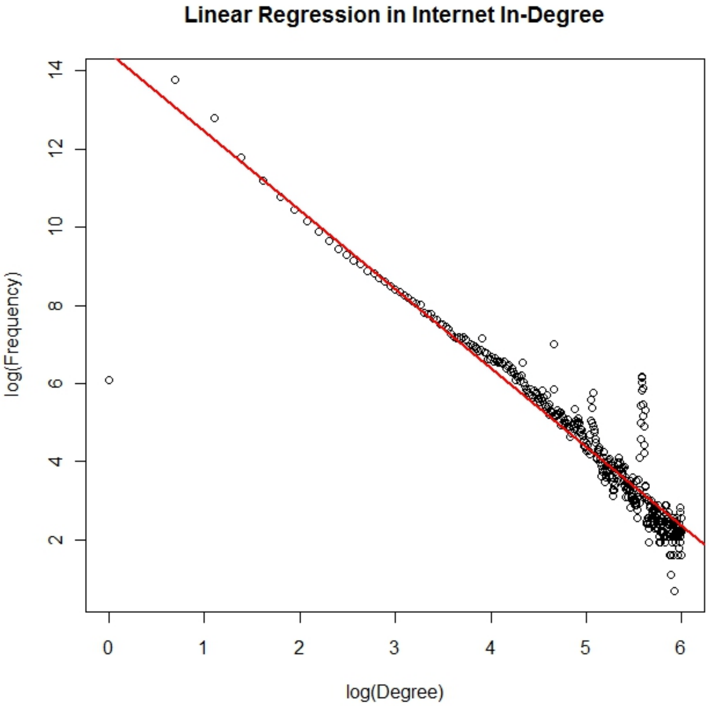


Figure 4.4: Fitted line for the power-law of the internet data set analysed in section 1.2

where Φ_X is the cumulative density function (cdf) of the fitted model and O_X is the emprirical cdf of the observed values. If we estimate the value of $x_{\min}$ to be too small, the model will be a poor fit and the statistic will be large. If we estimate it to be too large, then the number of data points over which the model is fit will be small and will give us less accurate results. This method of minimising the Kolmogorov-Smirnov test statistic aims to find the best compromise between these effects.

For a given value of $x_{\min}$, the estimation of the scaling parameter r is simply a case of maximum likelihood estimation. In cases, such as the degree sequence, when we are attempting to fit a model to a discrete case, it cannot be solved analytically, and must either be solved numerically or by approximating the data to fitting the underlying continuous model. We do the latter to derive a general estimator for the scale parameter, which we will use for this model, and later for our Kolmogorov-Smirnov goodness of fit test.

Theorem 4.2. *Let $x = (x_i)_{i=1}^n$ be a data set which we assume comes from an underlying power-law structure. By making the approximation that they are rounded to the nearest integer and are instead from a continuous power-law distribution we can get an estimate for the maximum likelihood of the scale factor given by*

$$\widehat{r} = 1 + n \left(\sum_{x_i \geq x_{\min}} \log \frac{x_i}{x_{\min} - \frac{1}{2}} \right)^{-1}$$

Proof. We mimic the proof given by Clauset et al [8].

Recall the definition of the distribution of the number of vertices with degree i, $\phi_X(i)$, we are trying to fit from Equation 4.1.

The log-likelihood is given by

$$\begin{aligned} \mathcal{L}(r; x) &= \log \prod_{x_i \geq x_{\min}} \phi_X(x_i) \\ &= \sum_{x_i \geq x_{\min}} \log \frac{x_i^{-r}}{\zeta(r, x_{\min})} \\ &= -r \sum_{x_i \geq x_{\min}} \log x_i - n \log \zeta(r, x_{\min}). \end{aligned}$$

We wish to find the value of r for which this is maximal.

$$\frac{\partial \mathcal{L}}{\partial r} = - \sum_{x_i \geq x_{\min}} \log x_i - n \frac{\zeta_r(r, x_{\min})}{\zeta(r, x_{\min})}, \tag{4.3}$$

where $\zeta_r(r, x_{\min})$ denotes the partial derivative of $\zeta(r, x_{\min})$ with respect to r.

Setting Equation 4.3 equal to zero, we find that our maximum likelihood estimator $\widehat{r}$ occurs at the solution of

$$\frac{\zeta_r(\widehat{r}, x_{\min})}{\zeta(\widehat{r}, x_{\min})} = -\frac{1}{n} \sum_{x_i \geq x_{\min}} \log x_i. \tag{4.4}$$

Consider a function of the form $f : t \to t^{-r}$. Call its antiderivative F and note that its differentiable. Then

$$\int_{x_{\min} - \frac{1}{2}}^{\infty} f(t) \mathrm{d}t = \frac{(x_{\min} - \frac{1}{2})^{-r+1}}{r - 1}, \tag{4.5}$$

but, equating the same integral in a different way,

$$\begin{aligned}\int_{x_{\min}-\frac{1}{2}}^{\infty} f(t)\mathrm{d}t &= \sum_{x=x_{\min}}^{\infty} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} f(x)\mathrm{d}x \\ &= \sum_{x=x_{\min}}^{\infty} F(x+\frac{1}{2}) - F(x-\frac{1}{2}).\end{aligned}$$

Consider the Taylor expansion of these two terms. Since they aboth have the same terms, up to sign differences, and we have that one is alternating, we can simplify this to

$$F(x+\frac{1}{2}) - F(x-\frac{1}{2}) = f(x) + \frac{1}{24}f''(x) + \delta^{(4)}(x),$$

where $\delta^{(4)}(x)$ is a term consisting only terms which are fourth or higher derivatives of f. Substituting the original function $f(x) = x^{-2}$ in, we get

$$\begin{aligned}F(x+\frac{1}{2}) - F(x-\frac{1}{2}) &= x^{-r} + \frac{x^{-r-2}}{r(r+1)} + \mathrm{O}(x^{-4}) \\ &= x^{-r}(1+\mathrm{O}(x^{-2})).\end{aligned}$$

So,

$$\int_{x_{\min}}^{\infty} f(t)\mathrm{d}t = \sum_{x\geq x_{\min}} x^{-r}(1+\mathrm{O}(x^{-2})) \tag{4.6}$$

$$= \zeta(r, x_{\min})(1+\mathrm{O}(x_{\min}^{-2})). \tag{4.7}$$

Combining Equations 4.5 and 4.7 gives us

$$\zeta(r, x_{\min})(1+\mathrm{O}(x_{\min}^{-2})) = \frac{(x_{\min}-\frac{1}{2})^{1-r}}{r-1}.$$

Assuming that $x_{\min}$ is large enough that we can neglect $x_{\min}^{-2}$ compared to 1, we can write this as

$$\begin{aligned}\zeta(r, x_{\min}) &= \frac{(x_{\min}-\frac{1}{2})^{1-r}}{r-1} \\ \implies \zeta_r(r, x_{\min}) &= -\frac{(x_{\min}-\frac{1}{2})^{1-r}}{r-1}\left(\log{(x_{\min}+\frac{1}{2})} - \frac{1}{r-1}\right).\end{aligned}$$

Using these results to solve equation 4.4, we find that it simplifies to

$$\log{(x_{\min}-\frac{1}{2})} + \frac{1}{r-1} = \frac{1}{n}\sum_{x_i\geq x_{\min}} log x_i,$$

which simplifies to gives us the maximum likelihood estimator we were after.

□

Returning to our original data set of the degree sequence for vertices of a Barabasi-Albert random graph, we estimate both of our parameters, using the code provided by Clauset and Dubroca [7] and also in appendix A.3. The estimated power law is then given by

$$R_n(j) \sim Cj^{-2.82} for j \geq 20.$$

This is plotted alongside the data in Figure 4.5.

To ensure we have fit a correct model, we run a Kolmogorov-Smirnov test. The test statistic, worked out using Equation 4.2 where Φ_X is the cdf of the fitted power law and O_X the observed value of the data points above our value of $x_{\min}$ is given by 0.0071, recalling that we only included the 5,895 data points which are above 20. Figure 4.6 shows the Kolmogorov-Smirnov test statistic of a number of simulated distributions, so we achieve an estimate of the p-value given by 0.416. This number is typical for a correctly fitted model so we do not reject the power law distribution, making the simulated data set faithful to the proven theorems.

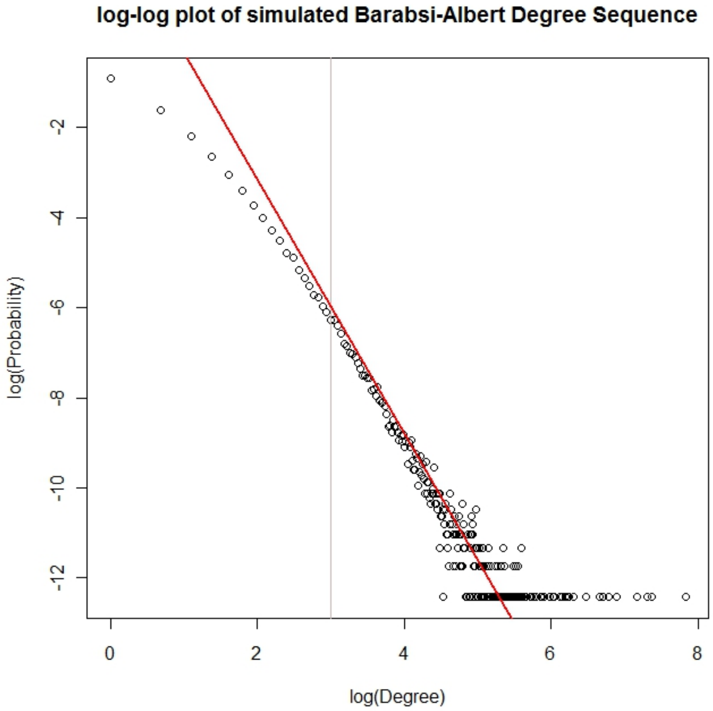


Figure 4.5: A log-log graph showing the frequency of vertices with specific degrees in a Barabasi-Albert random graph, with the power-law model fitted in red above the minimum value in grey.

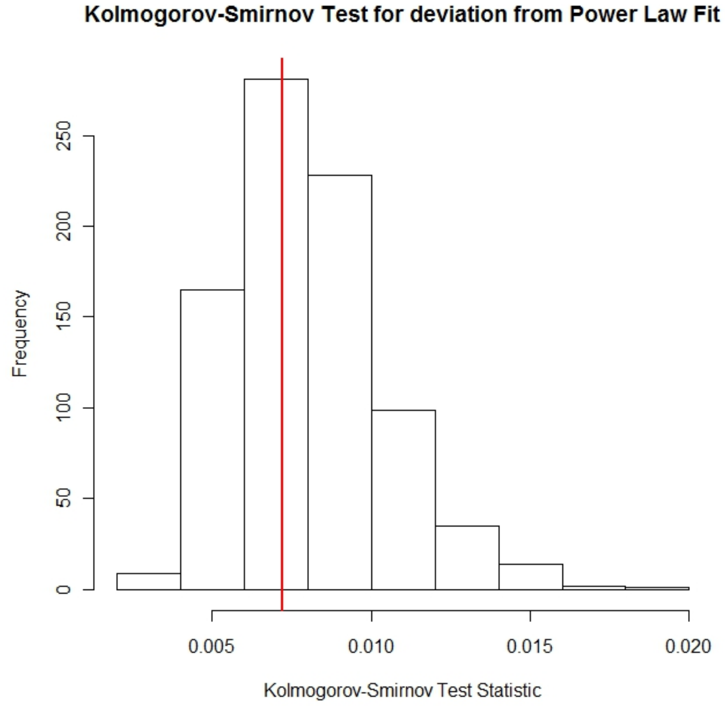


Figure 4.6: Histogram of simulated Kolmogorov-Smirnov test statistics for data sets drawn from a power-law distribution. The red line (x=0.00719) shows the test statistic of the simulated data set giving a p-value of 0.416.

Appendix A

Code

During the process of completing this project, R was used to produce both the data analysis results, the plots for the simulated graphs and also undertake the large scale simulations. Below is the code created for this purpose.

A.1 Data Analysis

To undertake the data analysis, the package igraph [9] was used, to facilitate the storage and manipulation of the large graphs. Listed below is the supplementary R code needed.

```
#Load package to allow use of read.graph and degree commands
library(igraph)

#Read graph
G<-read.graph("graph.txt",format="ncol",directed=TRUE)

#Outputs a vector of the in and out degrees of each vertex
oin<-degree(G,mode="in")
oout<-degree(G,mode="out")

#Produces a frequency table of the degrees
tin<-table(oin)
tout<-table(oout)

#Read in the degrees and frequencies, which we will be plotting
din<-as.integer(row.names(tin))
dout<-as.integer(row.names(tout))
vin<-as.integer(tin)
vout<-as.integer(tout)

#Combine it to one vector to facilitate plotting
dtot<-c(dout,din)
vtot<-c(vout,vin)
type<-c(rep(1,length(dout)),rep(2,length(din)))

plot(dtot,vtot,log="xy",col=type+1)

#Deciding which model to plot
deg<-log(din+1)
freq<-log(vin+1)
range<-which(deg<6)
```

```
#Fitting model and plotting it with data
power.lm<-lm(freq[range]~deg[range])
power.lm
plot(deg[range],freq[range])
abline(power.lm,col=2,lwd=2)

#Plot diagnostic plots for our linear model
par(mfrow=c(2,2))
plot(power.lm)
```

data.R

A.2 Random Graphs

This section details the package of algorithms and constants that I assembled for use in this project. These include the algorithms used for generating graphs according to a number of different algorithms, plotting them and analysing specific various random variables that arise when constructing the graphs.

```
#################################################################
#
# Contents
#
#################################################################

#Parameters
#1.....  Init()
#2.....  EdgeDist()
#3.....  GroupSim()

#General Purpose
#4.....  DiscreteSim()
#5.....  PlotGraph()
#6.....  PrepTree()
#7.....  PlotTree()

#Random Graph Models

#8.....  RESim()
#9.....  SWSim()
#10....  UpdateMatrix()
#11....  MultUpdate()
#12....  BASim()
#13....  TreeConv()

#Limiting Behaviour Simulations
#14....  InitDegSim()
#15....  MultInitDegSim()
#16....  DegSeqSim()

#################################################################
#
# Provides a matrix of the adjacency matrix of K2
#
#################################################################
Init<-function() {
 matrix(c(0,1,1,0), nrow=2, ncol=2)
```

```
}

#################################################################################
#
# Distribution of number of edges added in Barabasi-Albert
#
#################################################################################
EdgesDist <- function() {
 rgeom(1,0.5)+1
}

######################################################################################
#
# Distribution of number group length
#
######################################################################################
GroupSim <- function() {
 1+rpois(1,1)
}

####################################################################################
#
# Uses the method of inversion to simulate discrete variables from
# a given probability distribufunction
#
####################################################################################
DiscreteSim<-function(vecWeights) {
 #Normalise Weights Vector
 vecProbDist<-vecWeights/sum(vecWeights)

 #Generate random variable
 numUnif<-runif(1)

 #Run through probabilities until it totals more than numUnif
 intMaxResult<-length(vecProbDist)
 numRunTotal<-0
 intCount<-1
 while(numRunTotal < numUnif && !(intCount > intMaxResult)) {
  numRunTotal<-numRunTotal+vecProbDist[intCount]
  intCount<-intCount+1
 }
 #Return correct result
 intRes<-intCount-1
 intRes
}

######################################################################################
#
# Plot a graph with vertices at equal intervals around a circle
#
######################################################################################

PlotGraph<-function(adjMat) {
 noVert<-dim(adjMat)[1]
 #Plotting points equally around the unit circle
 angle<-numeric(0)
 for(i in 1:noVert) {
  angle<-c(angle,((i-1)*(2*pi))/(noVert))
 }
 x<-cos(angle)
 y<-sin(angle)
 plot(x,y,pch=19,fg="white",col.axis="white", col.lab="white")
 #Adding necessary edges
 for(i in 1:noVert) {
  for(j in 1:noVert) {
   if(adjMat[i,j]) {
    lines(c(x[i],x[j]),c(y[i],y[j]))
   }
  }
 }
}

######################################################################
#
# Rewrite the Adjacency Matrix of a tree in a more manageable form
# for plotting (Not intended to be called without PlotTree)
#
######################################################################
PrepTree<-function(adjMat,start=1,x=0,y=1,width=1) {

 #Find edges adjacent to vertex not already plotted
 dep<-which(adjMat[,start]>0 & c(1:(dim(adjMat)[1])) > start)
 noDep<-length(dep)

 #Add coordinates of plot to list
 plotx<-x+width/2
 ploty<-y
 origInd<-start

 #Using recursion, adding the coordinates of each of the adjacent
 #vertices to list
 if (noDep > 0) {

  nextLevelDep<-numeric(0)

  for(i in dep) {
   s<-sum(adjMat[,i]>0 & c(1:dim(adjMat)[1])>i)
   nextLevelDep<-c(nextLevelDep,max(1,s))
  }

  widthvec<-nextLevelDep/sum(nextLevelDep)*width

  for(i in dep) {
   n<-sum(dep<=i)
   v<-PrepTree(adjMat, i, x+sum(widthvec[0:(n-1)]), y-1, widthvec[n])
   plotx<-c(plotx,v[1:(length(v)/3)])
   ploty<-c(ploty,v[((length(v)/3)+1):(length(v)*(2/3))])
   origInd<-c(origInd,v[(1+(length(v)*(2/3))):length(v)])
  }
 }

 #Output concatenated list of (x,y) coordinates to plot
 c(plotx,ploty,origInd)
}

######################################################################
#
# Plot a graphical model of the tree given its adjacency matrix
```

```
#
############################################################################

PlotTree<-function(adjMat,text=1) {
 #Run preparatory script on adjacency matrix
 v<-PrepTree(adjMat)

 #Separate the x and y coordinate strings
 plotx<-v[1:(length(v)/3)]
 ploty<-v[((length(v)/3)+1):(length(v)*(2/3))]
 origInd<-v[(1+(length(v)*(2/3))):length(v)]

 #Plot each of them, hiding the coordinates axes
 plot(plotx,ploty,pch=19,fg="white",col.axis="white", col.lab="white")

 #For each vertex, find the vertices it is adjacent to and construct an
 #edge between them
 for(i in 1:length(plotx)) {
  x<-plotx[i]
  y<-ploty[i]
  j<-(i+1)
   while(j<=length(ploty) & ploty[j]+1 <= y) {
    if(ploty[j]+1 == y) {
    lines(c(x,plotx[j]),c(y,ploty[j]))
    if(text) {
     text.x<-mean(c(x,plotx[j]))
     text.y<-mean(c(y,ploty[j]))
     text(text.x,text.y,adjMat[origInd[i],origInd[j]],pos=2)
    }
    }
    j<-j+1
  }
 }
}



############################################################################
#
# Algorithm 2.1 - Erdos-Renyi Random Simulation
#
############################################################################
RESim <- function(noVert=100, prob=0.5,plot=1) {
 #Begin with the empty graph on noVert vertices
 adjMat<-matrix(rep(0,noVert^2),nrow=noVert,ncol=noVert)

 #For each pair, add that edge with probability prob
 for(i in 1:(noVert-1)) {
  for(j in (i+1):noVert) {
    if(runif(1)<prob) {
     adjMat[i,j]<-1
     adjMat[j,i]<-1
    }
  }
 }

 if(plot) {
 PlotGraph(adjMat)
 }
 adjMat

}
########################################################################
#
# Algorithm 2.2 - Storgatz and Watts model for Random Graph
#
########################################################################
SWSim<-function(noVert=100,cluster=1,prob=0.5,plot=1) {
#Initialise matrix as k-local ring
 matSim<-matrix(rep(0,noVert^2),ncol=noVert,nrow=noVert)
 for(i in 1:noVert) {
  for(j in 1:cluster) {
   vertAtt<-(i+j)
   if(vertAtt > noVert) {
    vertAtt <- vertAtt%%noVert
   }
   matSim[i,vertAtt]<-1
   matSim[vertAtt,i]<-1
  }
 }

#Randomly rewire edges with probability p
 for(i in 1:noVert) {
  for(j in 1:cluster) {
   if(runif(1)<prob) {

    #Sample from zero non-diagonal entries of adjacency matrix
    W<-which(!matSim & (c(1:noVert^2)-1)%%(noVert+1)!=0)
    v<-sample(W,1)

    #Unattach old edges
    vertAtt<-(i+j)
    if(vertAtt > noVert) {
     vertAtt <- vertAtt%%noVert
    }
    matSim[i,vertAtt]<-0
    matSim[vertAtt,i]<-0

    #Attach new edges
    y<-v%%noVert
    if(y==0) {
     y<-noVert
    }
    x<-1+((v-y)/noVert)
    matSim[x,y]<-1
    matSim[y,x]<-1

   }
  }
 }

 if(plot) {
  PlotGraph(matSim)
 }

 matSim
}
```

```
######################################################################
#
# Performs an iteration of the preferential attachment model
#
######################################################################
UpdateMatrix <-function(matOrig,parab=0) {

 intDim<-dim(matOrig)[1]

 #Determine the vertex to add new edges to
 vecWeight<-colSums(matOrig)+parab
 intSelVec<-DiscreteSim(vecWeight)

 #Determine the number of edges to add
 intEdges<-EdgesDist()

 #Create new rows and columns for matrix
 vecNewCol<-numeric(intDim)
 vecNewCol[intSelVec]<-intEdges
 vecNewRow<-numeric(intDim)
 vecNewRow[intSelVec]<-intEdges
 #Update Matrix
 vecMatUpd<-numeric(0)
 for(i in 1:intDim) {
  vecMatUpd<-c(vecMatUpd,matOrig[,i],vecNewRow[i])
 }
 vecMatUpd<-c(vecMatUpd,vecNewCol,0)
 matUpd<-matrix(vecMatUpd,nrow=intDim+1,ncol=intDim+1)
 matUpd
}

#########################################################################
#
# Algorithm 2.3 - Generalised Barabasi-Albert Algorithm
# Produces a graph output of a simulation after n iterations
#
#########################################################################

BASim<-function(n=10,parab=0,graph=1) {

 matOrig<-Init()
 for(count in 1:n) {
  matOrig<-UpdateMatrix(matOrig,parab)
 }

 #Either output the plotted graph, or the simulated adjacency matrix
 if(graph) {
  PlotTree(matOrig)
 }

 matOrig

}

#########################################################################
#
# (Algorithm 2.4)
# Convert tree-based simulation to general model
#
```

```
########################################################################
TreeConv<-function(matTree) {

 #Initialise Variables
 intVertices<-dim(matTree)[1]

 #Generate groups of vertices
 vecVertGroups<-c(1,1)
 while(sum(vecVertGroups) < intVertices) {
  vecVertGroups<-c(vecVertGroups, min(GroupSim(),intVertices-sum(vecVertGroups)))
 }
 intUpdVertices<-length(vecVertGroups)

 #Combine grouped columns
 vecCombEntries<-numeric(0)
 count<-1
 for(group in vecVertGroups) {
  for(row in 1:intVertices) {
   vecRelGroup<-count:(count+group-1)
   vecCombEntries<-c(vecCombEntries,sum(matTree[row,vecRelGroup]))
  }
  count<-count+group
 }
 matDemiUpd<-matrix(vecCombEntries,nrow=intVertices,ncol=intUpdVertices)

 #Combine grouped rows
 vecCombEntries<-numeric(0)
 count<-1
 for(group in vecVertGroups) {
  for(col in 1:intUpdVertices) {
   vecRelGroup<-count:(count+group-1)
   vecCombEntries<-c(vecCombEntries,sum(matDemiUpd[vecRelGroup,col]))
  }
  count<-count+group
 }

 matUpdated<-matrix(vecCombEntries,nrow=intUpdVertices,ncol=intUpdVertices)
 matUpdated
}

#######################################################################
#
# Simulates the degree of vertex v_1 scaled by an appropriate power
#
#######################################################################
InitDegSim <- function(len=5,parab=0,scale=0.5,plot=1) {
 #Initialising Variables
 v<-numeric(len)
 deg<-c(1,1)
 no<-2

 for(j in 1:len) {
  #Updating Vector
  v[j]<-deg[1]/(j^scale)

  #Determining which vertex to add new edges to
  thresh<-(deg[1]+parab)/(sum(deg)+(no*parab))
  u<-runif(1)
  if(u<thresh) {
   i<-1
  } else {
   i<-2
  }

  #Adding edges and vertices
  deg[i]<-deg[i]+EdgesDist()
  deg[2]<-deg[2]+EdgesDist()
  no<-no+1

 }
 v
 if(plot) {
  title<-paste("Scaled degree of initial vertex in Barabasi-Albert random graph (",len," iterations)")
  plot(v,main=title,xlab="Iteration",ylab="Scaled Degree")
 }

 v

}
####################################################################################
#
# Simulates the limit of the scaled v_1 distribution multiple times
#
####################################################################################
MultInitDegSim <- function(num=100,len=3000,parab=0,scale=0.5,plot=1) {

 #Run simulation keeping track of vertex 1 and "others"

 vecDeg<-numeric(0)
 for(i in 1:num) {
  vecDeg<-c(vecDeg,InitDegSim(len,parab,scale,plot=0))
 }


 if(plot) {
  title<-paste("Empirical Distribution of scaled Initial Vertex Degree (",length(vecDeg)," simulations)")
  hist(vecDeg,main=title,xlab="Scaled Degree Limit",freq=0)
 }

 vecDeg
}
####################################################################################
#
# Degree Sequence Simulation
#
####################################################################################
DegSeqSim <- function(num=100,parab=0,plot=1) {

 #Initialise Variablse
 DegSeq<-numeric(num+2)
 DegSeq[1]<-1
 DegSeq[2]<-1

 for(j in 1:num) {
  #Add the number of edges to the two relevant vertices
  edges<-EdgesDist()
```

```
    dist<-v[1:(j+1)]+parab
    i<-DiscreteSim(dist)
    DegSeq[i]<-DegSeq[i]+edges
    DegSeq[j+2]<-edges
  }

  if(plot) {
    table<-table(v)
    x<-as.integer(row.names(table))
    y<-as.integer(table)
    plot(x,y,log="xy")
  }

v

}
```

BASum.R

A.3 Simulation Analysis

We also used the data generated by simulations provided in Appendix A.2 to generate a large data set generated by Algorithm 2.3, the Generalised Barabási-Albert algorithm. As well as code provided by Clauset et al [7,8] which gives us the plfit function to estimate power-law parameters, we used the following code to analyse the output of such a simulation.

```
#Get data, and save it to avoid having to repeat simulation
DegSeqSim(250000,plot=FALSE)
dput(.last.Value,"output.txt")

#Read data in, and plot a log-log graphy
vecDegree<-dget("output.txt")
tabDeg<-table(vecDegree)

y<-as.vector(tabDeg)/250000
x<-as.vector(as.integer(row.names(tabDeg)))
plot(x,y,log="xy", xlab="Degree",ylab="Probability",main="log-log plot of simulated Barabasi-Albert degree sequences")

#Use plfit function from Clauset 2009 paper to estimate parameters
param<-plfit(vecDegree)
xmin<-as.numeric(param[1])
power<-as.numeric(param[2])

#Plot the simulated data with regression
plot(log(x),log(y),xlab="log(Degree)",ylab="log(Probability)",main="log-log plot of simulated Barabsi-Albert Degree Sequence")
abline(v=log(20),col=colors()[482])
abline(a=2.5,b=-2.82,col=2,lwd=2)

#Initialise parameters for Kolmogorov-Smirnov Goodness of Fit Test
teststat<-param[3]
teststat
noSampled<-sum(y[which(x >= 20)])*250000
noSampled

#Normalising Constant given by HurwitzZeta[2.82,20] worked out from Wolfram Alpha
```

```r
c<-405.675

#Simulate models from the fitted distribution to work out p-value

output<-numeric(0)

#Simulate Model
for(i in 1:1000) {
u<-runif(noSampled)
top<-max(u)
v<-c(0)
while(sum(v)<top) {
 A<-c*((length(v)+xmin-1)^(-power))
 v<-c(v,A)
}

w<-cumsum(v)

#Work out empirical cdf
sample<-numeric(noSampled)
for(i in 1:noSampled) {
 sample[i]<-xmin+max(which(w<u[i]))-1
}
vecVal<-unique(sort(sample))
noUniq<-length(vecVal)
min(sample)
g<-numeric(noUniq)
for(i in c(1:noUniq)) {
 g[i]<-length(which(sample<=vecVal[i]))/noSampled
}


#Refit parameter if need be (keeping threshold constant)
fitPower<-1+noSampled*(sum(log(sample/19.5)))^(-1)

#Work out cdf of model
h<-numeric(max(vecVal))
for(i in c(xmin:max(vecVal))) {
 h[i]<-(i^-fitPower)
}
h<-h/sum(h)
h<-cumsum(h)

j<-numeric(noUniq)
n<-c(1:noUniq)

output<-c(output,max(abs(g[n]-h[vecVal])))
}

#Output plot and test statistic
hist(output,xlab="Kolmogorov-Smirnov Test Statistic",main="Kolmogorov-Smirnov Test for deviation from Power Law Fit")
abline(v=teststat,col=2,lwd=2)
p<-length(which(output<teststat))/length(output)
p
```

simanal.R

Bibliography

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Modern Phys.*, 74(1):47–97, 2002.

[2] K. B. Athreya and P. E. Ney. *Branching Processes.* Springer-Verlag, 1972.

[3] Krishna B. Athreya, Arka P. Ghosh, and Sunder Sethuraman. Growth of preferential attachment random graphs via continuous-time branching processes. *Proc. Indian Acad. Sci. Math. Sci.*, 118(3):473–494, 2008.

[4] Krishna B. Athreya and Samuel Karlin. Limit theorems for the split times of branching processes. *J. Math. Mech.*, 17:257–277, 1967.

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

[6] Béla Bollobás. *Modern Graph Theory.* Number 184 in Graduate Texts in Mathematics. Graduate Texts in Mathematics, 1998.

[7] Aaron Clauset and Laurent Dubroca. Power-law distributions in empirical data (r script) - http://tuvalu.santafe.edu/ aaronc/powerlaws/.

[8] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51:661–703, 2009.

[9] Gábor Csárdi and Tamás Nepusz. igraph library for r - http://igraph.sourceforge.net/, August 2010.

[10] P. Erdős. On the evolution of random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[11] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):pp. 1141–1144, 1959.

[12] Russell Lyons, Robin Pemantle, and Yuval Peres. Conceptual proofs of l log l criteria for mean behavior of branching processes. *The Annals of Probability*, 23(3):pp. 1125–1138, 1995.

[13] Alexander M. Petersen, Fengzhong Wang, and H. Eugene Stanley. Methods for measuring the citations and productivity of scientists across time and discipline. *Phys. Rev. E*, 81:036114, Mar 2010.

[14] Erdős WebGraph Project. http://web-graph.org, February 2012.

[15] Andrew K. P. Taggart, Shu-Chun Teng, and Virginia A. Zakian. As a cell cycle-regulated activator of telomere-bound telomerase. *Science*, 297(5583):1023–1026, August 2002.

[16] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of /'small-world/' networks. *Nature*, 393(6684):440–442, June 1998.

[17] D. Williams. *Probability with Martingales*. Cambridge University Press, 9 edition, 2005.