Analysis of a Bipartite Network of Movie Ratings and Catalogue Network Growth Models

Mariano Beguerisse Díaz

St Cross College

University of Oxford

A thesis submitted for the degree of

M.Sc. in Mathematical Modelling and Scientific Computing

September 2008

I Mariano Beguerisse Díaz hereby declare that the content of this dissertation is entirely my own work (except where otherwise indicated), that it has not been submitted for a degree of any other university, and that all the assistance I have received has been fully acknowledged.

Mariano Beguerisse Díaz
St Cross College
September 5, 2008.

"Nothing in life is to be feared, it is only to be understood.
Now is the time to understand more, so that we may fear less."
Maria Sklodowska-Curie

"All beings going and remaining not at all."
Heraclitus

Acknowledgements

I would like to thank the following people for the valuable help and support in the realisation of this project:

My supervisors Mason Porter and Jukka-Pekka Onnela for introducing me to networks science, being patient while I learned, providing data, comments, criticism and encouragement. I am really happy I had the opportunity to work with them and hope to have more in the future. Kathryn Gillow who took special care to make sure that we had everything that we needed for the course and my classmates from the MSc for making it even better. Peter J. Mucha, Brian Harrington, David Smith and Doug Plato for their very useful comments and suggestions. The British Foreign and Commonwealth Office and the British Council for funding my studies through the Chevening Scholarship programme. Eficiencia Informativa for their constant support and help. The friends I have been fortunate enough to collect during the course of my life.

Above all, to my family who have never stopped showing me their love and mean everything to me.

Abstract

Network science is a rapidly growing field that draws important results from mathematics, physics, computer science, sociology, and many other disciplines. There are many problems in nature and man made systems that involve interactions between large number of agents which take place over a non-trivial topology. These problems lend themselves naturally and successfully to a network representation. Of particular interest are the models that deal with growth and evolution of networks because the vast majority of the systems represented by them are not static. This work is concerned about systems with two different types of interacting constituents known as bipartite networks.

This thesis is structured as follows: In Chapter 1 a network is defined as a graph and a brief introduction to the concepts used throughout this work is given. We describe the well-known network growth model of Preferential Attachment [2] and a model of the evolution of a bipartite network whose agent quantities are fixed [9]. In Chapter 2 we study data from Netflix, an online movie rental service whereby users can give ratings to movies they rent. We show how this system can be represented as a network and analyse some of its properties. The probability distribution of the number of ratings of users and movies follows a power-law distribution with an exponential cutoff, which indicates saturation in the number of ratings that a movie can receive or a user give. We also found that movies and users in the system form densely connected neighbourhoods. Chapter 3 is concerned with the development of network growth and evolution models which attempt to explain the growth and evolution of networks with saturation and a limited number of agents. We develop a network growth model in which the agents are drawn from fixed catalogues. An exact analytical solution to the model can sometimes be found, an approximate one using asymptotics in other cases and numerically in general. The results given by this model describe what is observed in simulated networks and show some of the characteristics observed in the Netflix network.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to networks

Networks and their representation

A *network* or *graph* is a collection of items, called *nodes* or *vertices* and their relations, called *links* or *edges.* We represent a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$ is the set of M vertices and $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$ is the set of E edges. We say that two nodes are connected or are *neighbours* if there is an edge between them [22]. A directed edge is one in which there is a precise direction of the relation, the edge goes from node v_i to node v_j. A network that has type of edges is called a directed network, if it does not then it is called undirected. When the edges between two nodes has a specific values assigned to them, we say it is a weighted network. An example of this type of networks might be cities and their distances between them. When the network is undirected, then the values of the edges are 1 or 0. All of the previous examples are also known as unipartite networks. A path is an ordered sequence of nodes in which there is one edge between consecutive nodes and its length is the number of edges contained in the sequence. A geodesic path from node v_j to v_i is the shortest possible of all paths that go from v_j to v_i. On Figure 1.1 we can see two graphical representations of networks. Nodes are represented by the circles and the lines that connect them represent the edges.

The usual way to work with a network is through its adjacency matrix $\mathbf{G} \in \mathbb{R}^{M \times M}$ [13]. An entry in this matrix $\mathbf{G}(i, j)$ represents the value of the edge between nodes v_i and v_j. If the network is undirected then $\mathbf{G}$ is symmetric. The entries of G^2 are the number of paths of length two between nodes, in general G^n gives the paths of length n between the nodes [13].

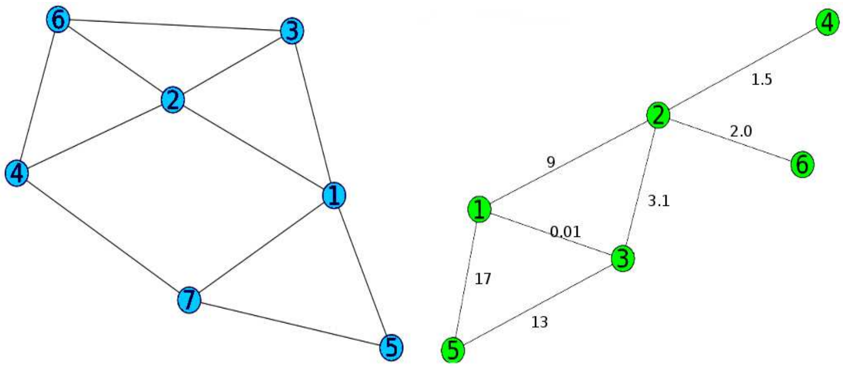

Figure 1.1: Graphical representation of a network (left), circles represent nodes and lines connecting them represent edges. (right) A weighted network.

Node Degrees and degree distribution

The *degree* of a node v_j, denoted as k_j is the number of edges that have one end attached to the node [22]. If N_k denotes the number of nodes in a network with degree k, then $p_k = N_k/M$ is the fraction of the nodes of the network with degree k, or the probability of choosing a node at random with degree k. A histogram of the values of p_k is called the *probability distribution function* (PDF) or *degree distribution* of the network. The form of this function can tell us many things about the network we are studying. It can tell us if there is a non-trivial structure in the relations of the nodes, or if there are nodes that are of particular importance, how many. When one is working with networks, it is usual to use the *cumulative distribution function* (CDF):

$$P_k = \sum_{j=1}^{k} p_j,$$

because it eliminates some of the fluctuations in the tails of the PDF caused by finite size of the network [22]. Some distributions often found in networks include the Poisson distribution found in Erdös-Réyni random graphs, power-law distributions, $p_k \sim k^{-\alpha}$, found on networks like the World Wide Web and some social networks and the exponential distribution $p_k \sim e^{-\lambda k}$ which arises in some growing networks [7]. One can usually detect power-law and exponential distributions by plotting the CDF in double or single logarithmic scales because the appear as straight lines [22].

Bipartite networks

A *bipartite network* $\mathcal{G} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$ has two different types of nodes: $\mathcal{V}_1 = \{v_{1_1}, v_{1_2}, \dots\ v_{1_U}\}$ and $\mathcal{V}_2 = \{v_{2_1}, v_{2_2}, \dots v_{2_M}\}$ of sizes U and M respectively called *partite sets* and the edges only lie between nodes of different type [13]. Figure 1.2 shows the graphical representation of a bipartite network in which $\mathcal{V}_1 = \{1, 2, 3, 4, 5\}$ and $\mathcal{V}_2 = \{A, B, C\}$, and edges only connect letters to numbers. This type of network is used to study networks of movies and actors, members of boards of directors and companies, or club membership. A typical example in the social sciences are the so-called affiliation networks where nodes of individuals are related to nodes of artifacts, which can be clans, tribes or associations [2, 22]. Bipartite networks have two node degree counts N_{1k} and N_{2j}, and two degree distributions $p_{1_k} = N_{1k}/U$ and $p_{2_j} = N_{2j}/M$. As in the unipartite case we can visualise the degree distribution through the histograms of N_{1k} and N_{2j} or the cumulative distribution function of each type. A bipartite network can always be mapped into two unipartite networks, one of each type of node. In these mapped or projected networks two nodes are connected if they share at least one neighbour.

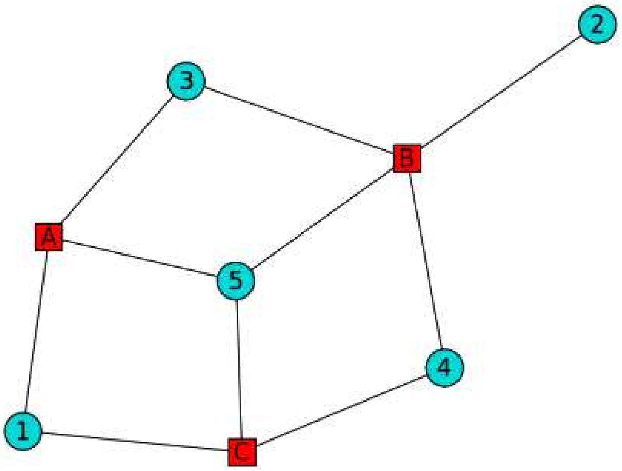

Figure 1.2: A bipartite network where nodes of letters can only be connected to nodes of numbers and vice versa.

We can still represent $\mathcal{G}$ through an adjacency matrix $\mathbf{G} \in \mathbb{R}^{U \times M}$. Though we will generally not be able to square $\mathbf{G}$, if we multiply it by its transpose $\mathbf{G}\mathbf{G}^T$ or $\mathbf{G}^T\mathbf{G}$, we to get two matrices of size $U \times U$ and $M \times M$. These matrices are the adjacency matrices of the projected weighted networks and their entries give the number of paths of length 2 between nodes of the same type in $\mathcal{G}$. If we continue to multiply by $\mathbf{G}$ and $\mathbf{G}^T$ many times over we will get the number of paths of length $2n$ between nodes of the same type and the paths of length $2n + 1$ between nodes of different types. An important interpretation of this is that the projection $\mathbf{G}\mathbf{G}^T$ yields the adjacency matrix of a unipartite network in which nodes connected to each other are the ones that are attached to common nodes in $\mathcal{G}$, for example in an affiliation network, two persons would be connected in the projected network of individuals if they have membership to at least one society in common.

Clustering

Earlier we stated that two nodes are neighbours if an edge lies between them. If a node in a network has two neighbours and those nodes are also connected to each other, together they form a triangle. Triangles are very important to find out if a network contains clusters of nodes that are more connected among each other than with the rest [22]. A measure of this "cliquishness" of the neighbourhood of a node used by Watts and Strogatz [30] is the *clustering coefficient* of a node defined as:

$$C_3(i) = \frac{2t_i}{k_i(k_i-1)}, \tag{1.1}$$

where t_i is the number of triangles that contain node v_i, and it is divided by the total number of possible triangles that include v_i which is $k_i(k_i-1)/2$. The clustering coefficient of an entire network is defined as the mean of all the individual coefficients:

$$C_3 = \frac{1}{M}\sum_{i=1}^{M} C_3(i). \tag{1.2}$$

The clustering coefficient can be very useful to detect networks with meaningful social structures.

There is also a definition of the clustering coefficient which does not use triangles but squares. A square in a network appears when two neighbours of a node have a common neighbour different from the node in question. Lind *et al* [17] define a clustering coefficient for a node v_i using the observed number of squares $C_4(i)$ as:

$$C_4(i) = \frac{\sum_{h,m} q_{i_{mh}}}{\sum_{m,h}\left[(k_m-\eta_{i_{mh}})(k_h-\eta_{i_{mh}})+q_{i_{mh}}\right]}. \tag{1.3}$$

The numerator is the sum of $q_{i_{mh}}$, the number of squares that include v_m, v_h and v_i The denominator is the sum of the total possible number of squares that can contain v_m, v_h and v_i. The degrees of n_m, n_h are k_m, k_n respectively, and $\eta_{i_{mh}} = (q_{i_{mh}}+\theta_{mn}+1)$ where $\theta_{mh} = 1$ if v_m and v_h are connected and zero otherwise [17]. As with C_3, the coefficient for the entire network is

$$C_4 = \frac{1}{M}\sum_{i=1}^{M} C_4(i). \tag{1.4}$$

Random graphs

A random graph is a network in which the edges are placed randomly between the nodes. One of the earliest models of random graphs is the one proposed by Erdös and Réyni. In their model they have a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{v_1, \dots, v_M\}$ and every node n_i is connected to any other node with probability p, and not connected with $(1-p)$. They defined $G_{M,p}$ as the set of all possible graphs of size M and probability p [2]. Figure 1.3 shows a realisation of a network from $G_{M,p}$ where $M = 20$ and $p = 0.2$. The mean degree $z = (M-1)p$ of a graph like this is just the probability of the existence of a node times the number of possible edges that a node can have. The degree distribution $\{p_k\}$ of a the graph is given by:

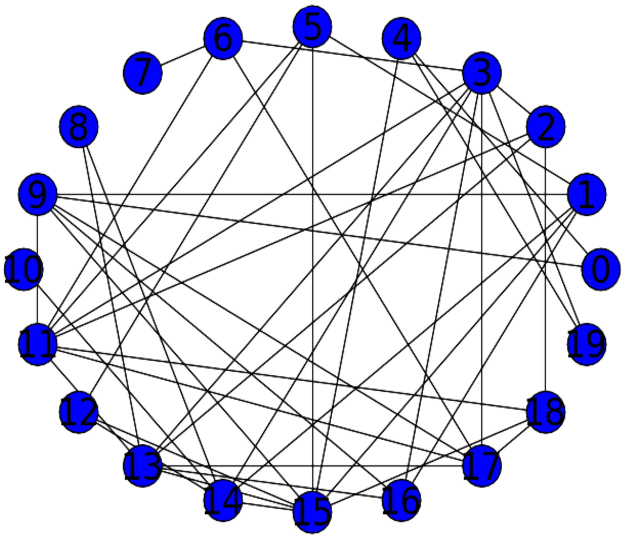

Figure 1.3: Random graph generated using the model defined by Erdös and Réyni with $M = 20$ and $p = 0.2$.

$$p_k = \binom{M}{k} p^k (1-p)^{M-k},$$

the binomial distribution. As $M \to \infty$:

$$p_k = \frac{z^k e^{-z}}{k!}, \tag{1.5}$$

which is the Poisson distribution [22]. The random graph model is important for this project because it served as a motivation for the development of some of the ideas that were used in this work, such as random growing networks, preferential attachment, and the catalogue growth model to name a few [2, 30].

1.2 Network growth

In many real-life problems, networks do are static: the number of nodes and edges may be in constant change. The way in which this happens determines the properties of a network [22]. In the following sections we give a brief introduction to some important models of network growth and evolution.

1.2.1 Preferential attachment

One of the early models of network growth is the one observed by Derek Price who in 1965 studied citation networks of scientific papers. These are directed networks in which nodes are papers and edges are citations among them [26]. Price found that most papers that get cited are by papers published within a decade of their publication. There are very few papers that got citations years after being published, most of them very influential papers or reviews. Those papers cite many of the papers that will be "dead" after a while, and summarise the research front of the time. Price hypothesised if the more a paper receives citations, the higher the probability that it will be cited in the future [26]. He called this conjecture *cumulative advantage*. He also wondered if classic papers could be detected in an automatic way just by the number of citations they receive [26]. One of the key findings in his analysis of citation networks was that the in-degree (how many times a paper is cited) and the out-degree (how many papers it cites) of a paper follow power-law distributions [22] (For a brief introduction to power laws, see Appendix B). The model that Price found produced these distributions, it started with a directed citation network $\mathcal{G}$ with n_0 vertices. New vertices were added to the network with an average out-degree of m. The probability of a node v_i of receiving an edge of the new nodes is proportional to its in-degree k_i plus one :

$$P(v_i) = \frac{k_i + 1}{\sum_i [k_i + 1]}.$$

This is so nodes with zero in-degree are able to receive edges. Price justified it saying that the publication of the paper is equivalent to a citation (by itself) [22].

The idea of cumulative advantage was retaken by Barabási and Albert some years later in a paper where they analysed the World Wide Web and some other networks, where they also found power-law degree distributions [3, 22]. In their paper they proposed a model of network growth in which the probability with which node receives new edges is proportional to its degree. They renamed the model *preferential attachment*, which is the name that is most widely used now for these types of models [3, 4]. There are some key differences between the Barabási-Albert (BA) model and Price's. The BA model assumes an undirected network, eliminating the need to distinguish between in and out-degree [22]. In such a network it is unnecessary to add a constant to the degrees of the nodes like in Price's model, because the network is undirected and all the nodes in the seed network and thereafter have degrees greater than zero. In the BA model a node is added in every time-step that must have exactly m edges, not on average like in Price's model, and as before, the initial network $\mathcal{G}$ must have at least $m + 1$ nodes. Figure 1.4 shows one realisation of a BA network with 250 nodes where each new node has one edge.

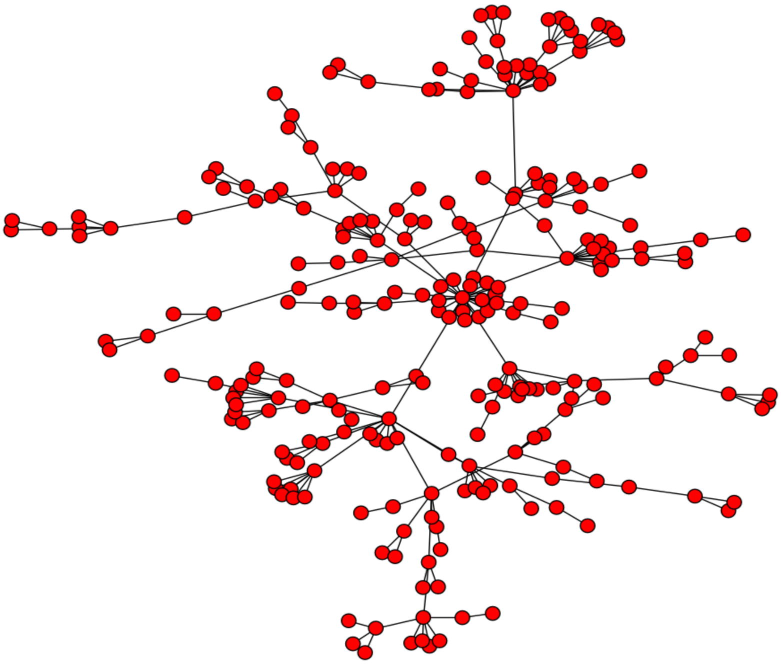

Figure 1.4: Graphical representation of a Barabási-Albert network with 250 nodes and $m = 1$.

If $N_k(t)$ is the number of nodes with degree k at a time t (the network grows by one node at every time-step). The probability that a new edge is attached to a node with degree k is

$$P_k = \frac{A_k N_k(t)}{\sum_j A_k N_j(t)}. \tag{1.6}$$

The "kernel" of the system is A_k is called linear if $A_k = k$. When $A_k = k^\gamma$, then we say that the kernel is sublinear when $\gamma < 1$ and superlinear when $\gamma > 1$. A *master equation* is a set of ODEs that describe the evolution of the probability that $N_k(t)$ will have a certain value. If we denote by $A = \sum_j jN_j(t)$ then the master equation for the growth of the network is [15]:

$$\frac{\mathrm{d}N_k}{\mathrm{d}t} = \frac{A_{k-1}}{A}N_{k-1} - \frac{A_k}{A}N_k + \delta_{k1}, \qquad k = 1, 2, \ldots \tag{1.7}$$

The first term in the right hand side is the number of nodes that go from having degree $k-1$ to degree k, the second term is for the nodes that go from k to $k+1$ and the third term is the Kronecker delta to take into account the entry of new nodes to the network. When we use the linear kernel, the denominator of equation (1.6) is just the total number of endpoints of edges in the network and $\sum_j jN_j(t) = 2t$ [15]. We can solve equation (1.7) using an arbitrary initial condition, this is justified because the long term behaviour of the network is being modelled so the initial condition is irrelevant [16]. The solutions are linear in t so we can express them as $N_k(t) = n_k t$ where

$$n_1 = \frac{2}{3}, \qquad n_k = n_{k-1}\frac{k-1}{k+2}, \qquad k \geq 2. \tag{1.8}$$

This recursive relation can be solved for all k:

$$n_k = \frac{4}{k(k+1)(k+2)}, \tag{1.9}$$

which can be re-written in terms of the Gamma function (see equation (A.7) on Appendix A):

$$n_k = \frac{4\Gamma(k)}{\Gamma(k+3)}. \tag{1.10}$$

This function is the discrete analog of the power-law $f(x) \sim x^{-3}$ [16]. Figure 1.5 shows in log-log coordinates the complement of the CDF $P(X \geq x) = 1 - F(x)$, of a network that was allowed to grow following the BA model until it reached 10,000 nodes with one edge per added node. In red we can see the fit of the observed degree distribution to a power-law. The fit shows shows that the exponent in the power-law is $\gamma \approx 3.0082$, as the calculations had predicted. The fit was made using the goodness-of-fit method outlined by Clauset *et al.* [7] In this case, the exponent in the power-law is three. However, Krapivsky *et al.* have shown that it can be tuned to any value larger than two by means of small adjustments in the attachment probabilities [15]. For example if we let $A_1 = \alpha$ and $A_k = k$ when $k > 1$ then $n_k \sim k^{-\nu}$ where

$$\nu = \frac{3 + \sqrt{1 + 8\alpha}}{2}.$$

More general forms of the BA model include the use of sublinear and superlinear kernels. In the sublinear case we assume that $A(t) = \sum_k k^\gamma N_k(t)$ is linear in t, e.i.

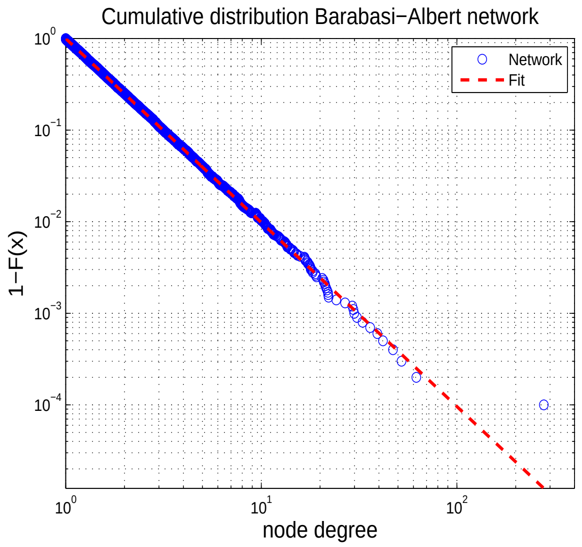

Figure 1.5: Degree distribution of a network created according to the BA model.

$A(t) = \mu t$, $\mu \in \mathbb{R}$. Equation (1.7) has the solution $N_k(t) = n_k t$ where

$$n_k = \frac{\mu}{k^\gamma} \prod_{j=1}^{k} \left(1 + \frac{\mu}{j^\gamma}\right)^{-1}. \tag{1.11}$$

The solution to this equation is a stretched exponential [15]

$$n_k \sim k^\gamma e^\beta,$$

where $\beta \in \mathbb{R}$ [2]. When the superlinear kernel $A_k = k^\gamma \quad \gamma > 1$ is used, then a discretised version of equation (1.7) can be solved. If $1 < \gamma < 2$ there will be a dominant node with a nonzero probability of having a finite fraction of all links in the network. When $\gamma \geq 2$ then the dominant has a nonzero probability of being connected to every other node in the network [2, 22].

1.3 Network rewiring

There are some problems in which the network does not change its size in time, i.e. the numbers of nodes and edges remain constant but its structure does not. The process known as wiring is when a new edge that connects two nodes is added to the network. Node rewiring means that an existing edge changes one of its end points from one node to another but the other one remains. Networks in which rewiring often occurs are bipartite affiliation networks. Rewiring represents, for example, when people change their affiliation from one society or club to a different one.

1.3.1 A network rewiring model

Network rewiring models have been studied thoroughly by Evans and Plato [9, 10]. Let $\mathcal{G} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$ be a bipartite network with two different types of nodes: individuals $\mathcal{V}_1 = \{v_{1_1}, v_{1_2}, \ldots\ v_{1_E}\}$ and artifacts $\mathcal{V}_2 = \{v_{2_1}, v_{2_2}, \ldots v_{2_N}\}$ of sizes E and N respectively. Each individual holds exactly one edge that is attached to one of the artifact nodes, so there are always E edges in the network and the mean degree of the artifacts is E/N at all times. Every time step, an artifact node is chosen at random with probability Π_R and re-attached to an artifact node with independently chosen with probability Π_A. The chosen artifact can be the same which was originally detached. Figure 1.6 shows graphically how this process works. The number of artifact nodes with degree k at any time in the network $N_k(t)$, and the degree probability distribution $p_k(t) = N_k(t)/N$. The master equation that describes the evolution of the artifact degree distribution is:

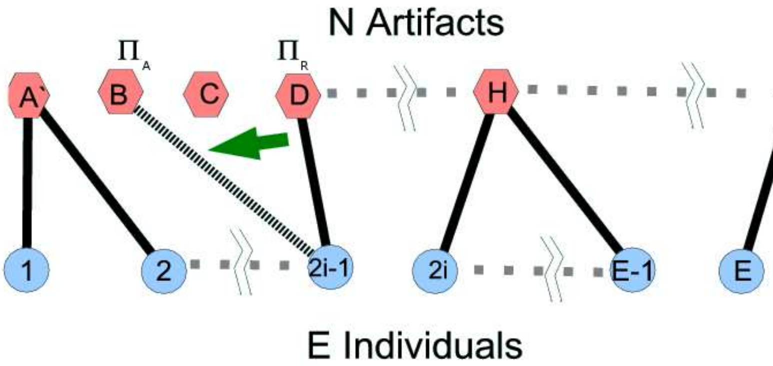

Figure 1.6: Description of the rewiring process of the edges from one artifact to another. One edge is detached from artifact node D and rewired to B. (Image from [10], used with permission)

$$\begin{aligned} N_k(t+1) - N_k(t) = {} & N_{k+1}(t)\Pi_R(k+1,t)[1-\Pi_A(k+1,t)] \\ & - N_k(t)\Pi_R(k,t)[1-\Pi_A(k,t)] \\ & - N_k(t)\Pi_A(k,t)[1-\Pi_R(k,t)] \\ & + N_{k-1}(t)\Pi_A(k-1,t)[1-\Pi_R(k-1,t)]. \end{aligned} \tag{1.12}$$

The positive terms in the equations account for the arrival of edges from the artifacts with degree $k+1$ that lose an edge and the ones with degree $k-1$ that gain an edge. The negative terms account for the nodes with degree k that have either gained or lost an edge. The terms $(1 - \Pi)$ represent the edges that are removed from one artifact and re-attached back to it. The probability with which edges are chosen for removal and attachment are:

$$\Pi_R(k,t) = \frac{k}{E}, \qquad \Pi_A(k,t) = \frac{p}{N} + (1-p)\frac{k}{N}. \tag{1.13}$$

Nodes are chosen to have an edge removed with probability Π_R proportional to their degree. The arrival probabilities Π_A involve a parameter $p \in (0,1)$ which means that an edge is re-wired using uniform attachment with probability p and preferential attachment with probability $(1-p)$.

Evans and Plato show that equation (1.12) has an exact solution for the probabilities (1.13) [10]. This equation represents a Markov process for $N_k(t)$ and its solutions are given in terms of the eigenfunctions of the system $\omega^{(m)}(k)$ and generating functions $G^{(m)}(x)$. The solution is:

$$N_k(t) = \sum_{k=0}^{E} c_m(\lambda_m)^t \omega^{(m)}(k). \tag{1.14}$$

The generating function $G(x,t)$ is:

$$\begin{aligned} G(x,t) &= \sum_{k=0}^{E} x^k N_k(t) \\ &= \sum_{m=0}^{E} c_m(\lambda_m)^t G^{(m)}(x), \end{aligned} \tag{1.15}$$

and the generating functions $G^{(m)}(x)$ are

$$G^{(m)} = \sum_{k=0}^{E} x^k \omega^{(m)}(k),$$

which can be written in terms of the ${}_2F_1$ hypergeometric function (a brief introduction to the hypergeometric functions and some of its properties are given in Appendix A)

$$G^{(m)} = (1-x)^m {}_2F_1(a+m, b+m; c; x), \tag{1.16}$$

where

$$a = \frac{pE}{[(1-p)N]}, \qquad b = -E, \qquad c = 1 + a + \frac{b}{1-p}.$$

In Appendix A the needed properties of the hypergeometric function are reviewed. The eigenvalues of the process are:

$$\lambda_m = 1 - m\frac{p}{E} - m(m-1)\frac{(1-p)}{E^2}. \tag{1.17}$$

One important result is that when $p \leq E^{-1}$ we can see from the eigenvalues that there will be an artifact that will receive all the edges in the network eventually. When $p \gg E^{-1}$ then we get a power-law with cutoff $p_k \sim k^{-1}e^{-\xi k}$ [10], which behave initially like a power-law distribution but show exponential decay in the tails [7].

Chapter 2

A bipartite network of movie ratings

Netflix[1] is DVD rental company in the United States that allows its costumers to give a rating to the movies they rent through their Internet page. Netflix has a system that uses these ratings to "predict" the ratings their costumers would give to unseen films and based on that, it makes recommendations [6]. In 2006 Netflix issued a contest called the Netflix Prize[2], in which it challenged the public to develop a system that can make better predictions than their own. To help the contestants the company released a dataset consisting of more than 100 million ratings of nearly 18,000 movies by about one million costumers. In this chapter we describe and analyse this dataset.

2.1 Dataset description

The Netflix dataset consists of 100,480,507 ratings in the integers from 1 to 5 of 17,770 movies made by 480,189 system users from late 1999 to the end of 2005. Each entry in the dataset consists of a movie ID, a user ID, the value of the rating and the date it was entered into the system.

The users of the system form a set that grows as more people become costumers. This is important to keep in mind to normalise data whenever necessary to avoid biases in the results. On the left image of Figure 2.1 we can see how the number of users active in the system grows. On the right image we see the average number of ratings entered to the system.

The dataset displays recurring behaviour. The number of ratings introduced into the system shows strong dependence on the day of the week. This is probably a

[1] *http://www.netflix.com*

[2] *http://www.netflixprize.com*

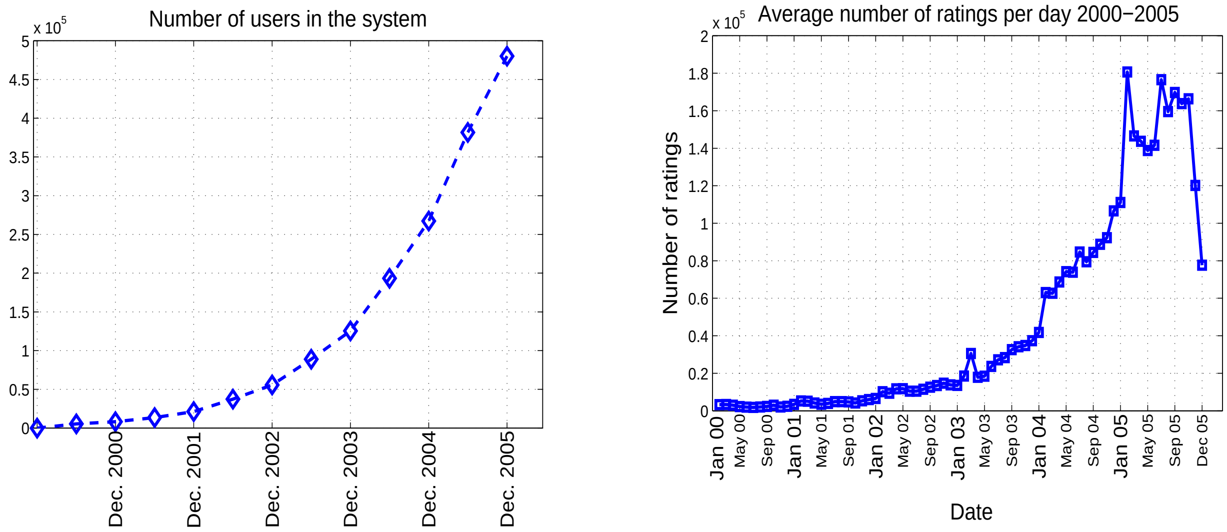

Figure 2.1: Monthly average number daily ratings and total number of users in the Netflix dataset from January 2000 to December 2005.

consequence of the work and leisure habits of people that normally follow weekly patterns. Tuesdays and Wednesdays were the days in which the system received the most ratings. In Figure 2.2 we can see how the number of ratings received a day for an

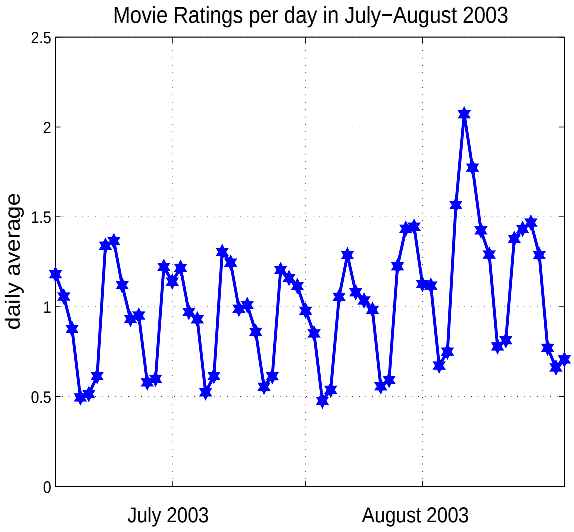

Figure 2.2: Ratings per day divided by the daily average for July and August 2003.

interval of two months in 2003 follows a . Similar patterns are observed throughout the whole dataset.

2.2 The Netflix dataset as a network

The Netflix dataset can be described as a bipartite network with users and movies as different node types, and weighted edges to represent the ratings. If the set of U users is $\mathcal{U} = \{u_1, u_2, \ldots, u_U\}$, the set of M movies is $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$ and the set of ratings $\mathcal{E} = \{e_{(1,1)}, e_{(1,2)}, \ldots, e_{(U,M)}\}$, then we can represent the dataset as the network $\mathcal{N} = (\mathcal{U}, \mathcal{M}, \mathcal{E})$. The adjacency matrix of the network is $\mathbf{G} \in \mathbb{R}^{U \times M}$, where $\mathbf{G}(i,j)$ is the rating of user i to movie j and zero if it does not exist. As a weighted bipartite network, $\mathcal{N}$ has two degree distributions, one for the users and one for the movies. Most of the analysis done in this chapter will be done on subsets of the original data. This is done for two reasons: To see how the structure of the network is and changes at different times and because it is a very large dataset it can take a computer a very long time to compute results.

With the adjacency matrix $\mathbf{G}$ of the network or subsets of it, we can also construct the projected networks of movies and users. These unipartite networks will have, for example in the projected network of movies, an edge between any two movies that have been rated by the same user. In the projected network of users, two of them will be connected if they have rated at least one movie in common. We can construct these networks simply by multiplying the adjacency matrix by its transpose $\mathbf{G}\mathbf{G}^T$ to get the network of users and $\mathbf{G}^T\mathbf{G}$ for the network of movies. The value of an edge in any of the projected networks is the sum of the products of the ratings.

2.2.1 Degree distributions

In this section we will look the degree distributions of the different node types found in the network. Figure 2.3 shows the degree distributions observed in a subset of the data corresponding to one day. In this subset there were 33,528 ratings of 4,819 movies by 6,671 users and the average user rated 5 movies. In the plots, we that the degree distribution looks like a power-law that gets disrupted in the lower end by alterations due to of the finite size of the network [29]. Accordingly, in this project we will use CDFs instead, in which the effects of finite size networks are not as prominent as in the PDFs. A legitimate claim of a power-law cannot be based on graphical evidence alone because it can lead to serious errors in the interpretations [5]. The method we will use to confirm the shape of the distributions is the one outlined by Clauset *et al* [7], which relies maximum likelihood estimators. In Figure 2.4 we see the CDF of the degree distributions from the same date as in Figure 2.3 and their power-law fits. The fit seems reasonably agree with the data for about a decade (a power of 10) and a half

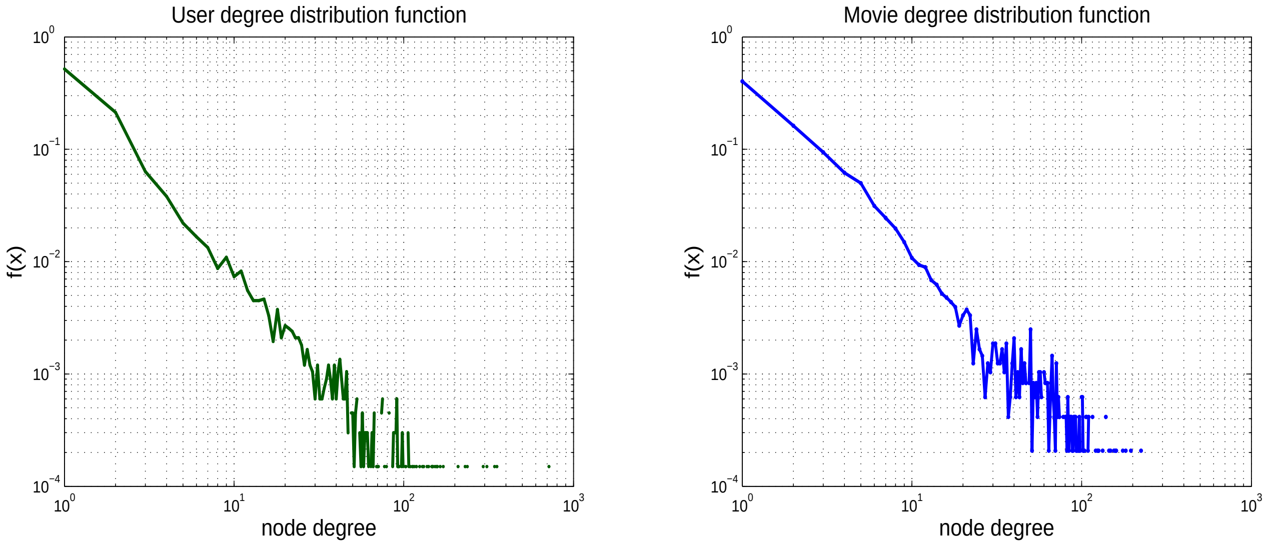

Figure 2.3: Probability distribution functions of user (left) and movie (right) degrees from the subset of data from August 15, 2003.

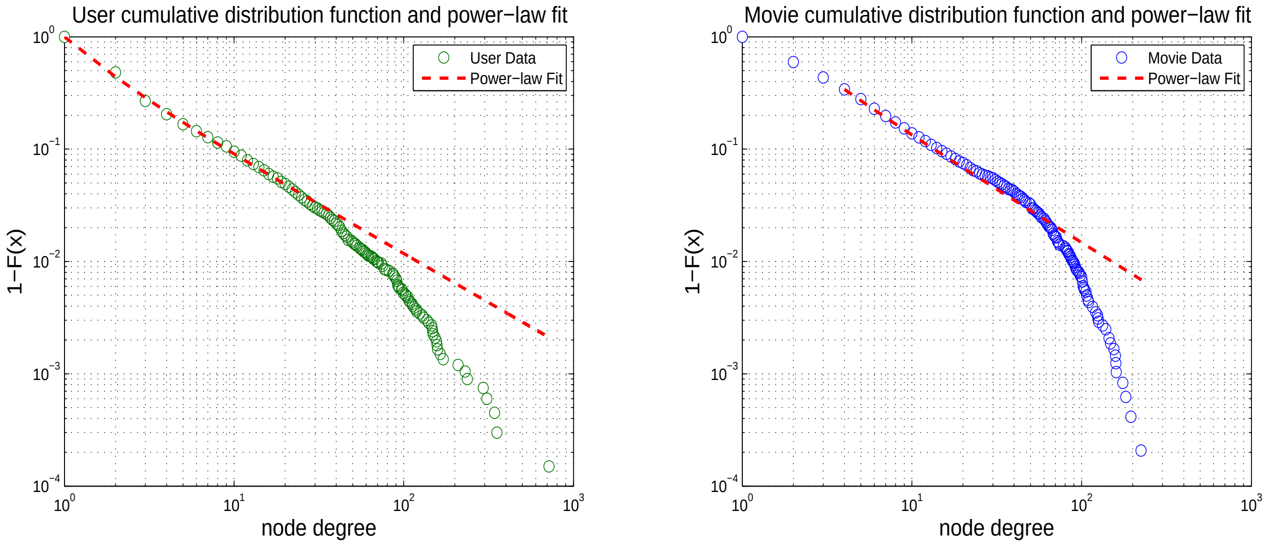

Figure 2.4: Cumulative distribution functions of the user (left) and movie (right) degrees from August 15, 2002 and their power-law fits.

of data, but the tails decay much faster as they would in a power-law distribution. There appears to be some a cut-off which bounds the degree of the nodes. There are a number of CDFs that display that behaviour, such as like the power-law with a cut-off,

$$F(x) \sim cx^{-a}e^{-bx}, \tag{2.1}$$

the stretched exponential

$$F(x) \sim x^{b-1}e^{-\lambda x^b},$$

and the general power-law with two crossover points:

$$F(x) \sim \begin{cases} x_1^{-a} & 1 \leq x < x_1, \\ x^{-a} & x_1 \leq x < x_2, \\ x_2^{-a}e^{-b(x-x_2)} & x_2 \leq x. \end{cases}$$

The crossover points x_1 and x_2 can be determined using maximum likelihood estimators [5] or the Kolmogorov-Smirnov statistic which measures maximum discrepancies between CDFs [7]. The distribution that we found was a better fit of the Netflix data snapshot from August 15, 2002 was the power-law with cut-off defined in equation (2.1). Figure 2.5 shows the fits to the user and movie CDFs. The fit for the user nodes looks closer to a normal power-law rather than one with cut-off. This is confirmed by the fitting parameters that have values of a and b from equation (2.1) are: $a = 1.063$ and $b = 5.055 \times 10^{-7}$. The movie node degree distribution in the plot showed a good agreement to the fit with coefficient values $a = 0.7968$ and $b = 0.01089$. However, tests done on several days of the data do not rule out the power-law with exponential

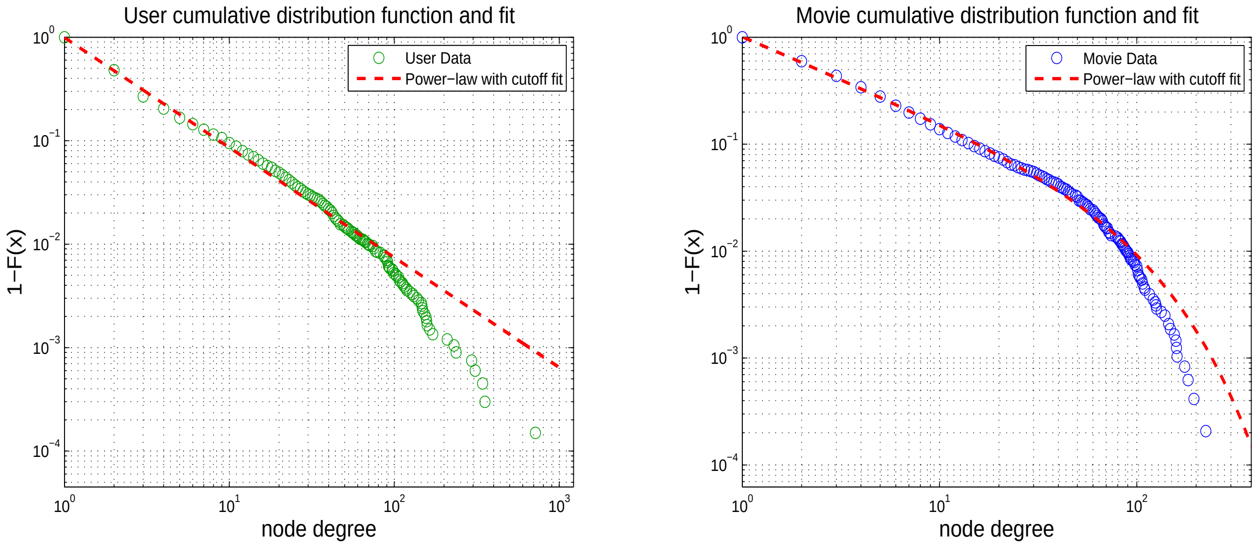

Figure 2.5: Power-law with cut-off fit to user and movie degree distributions.

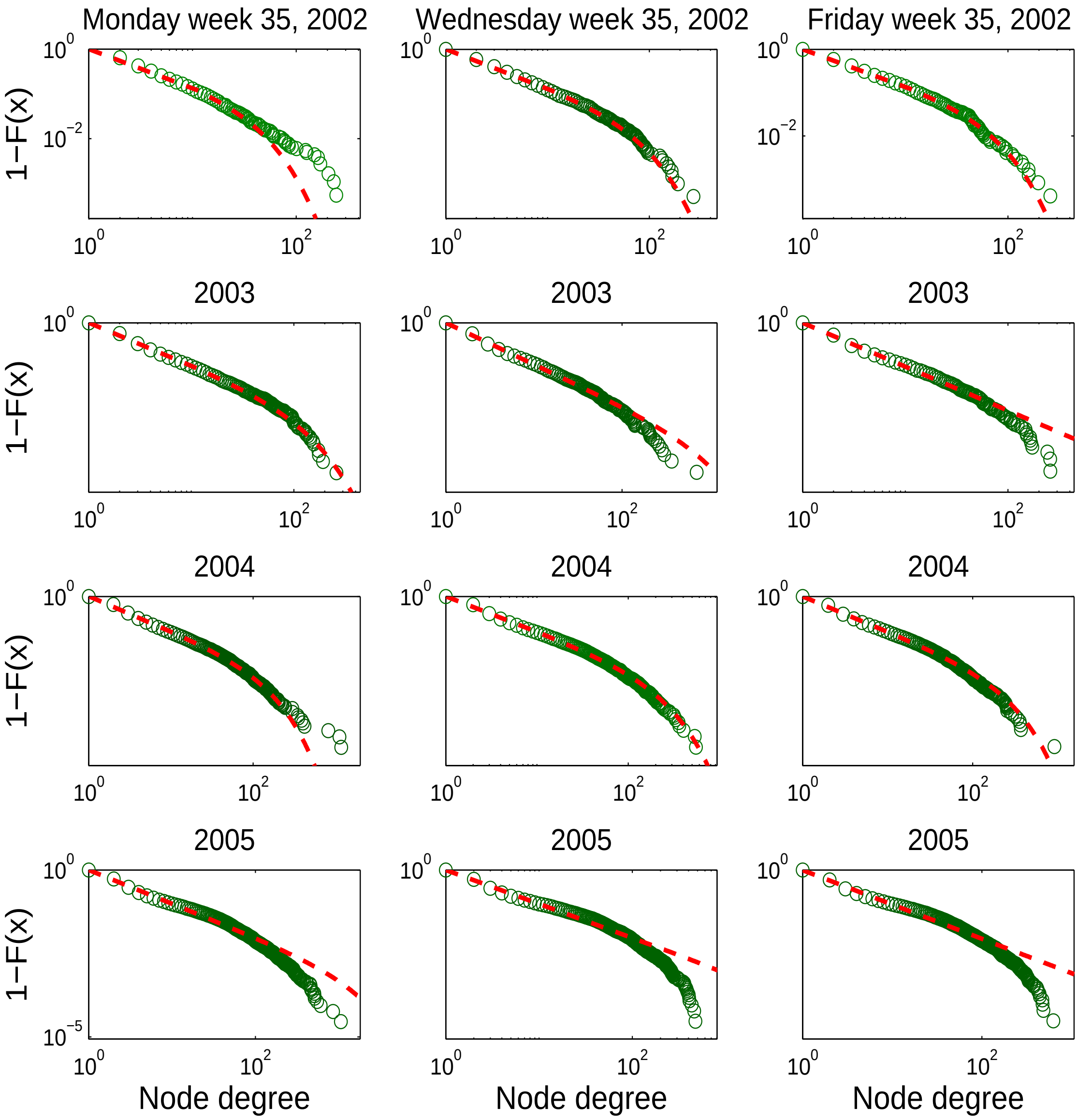

Figure 2.6: Fits of user degree distributions for three different days during week 35 of years 2002-2005.

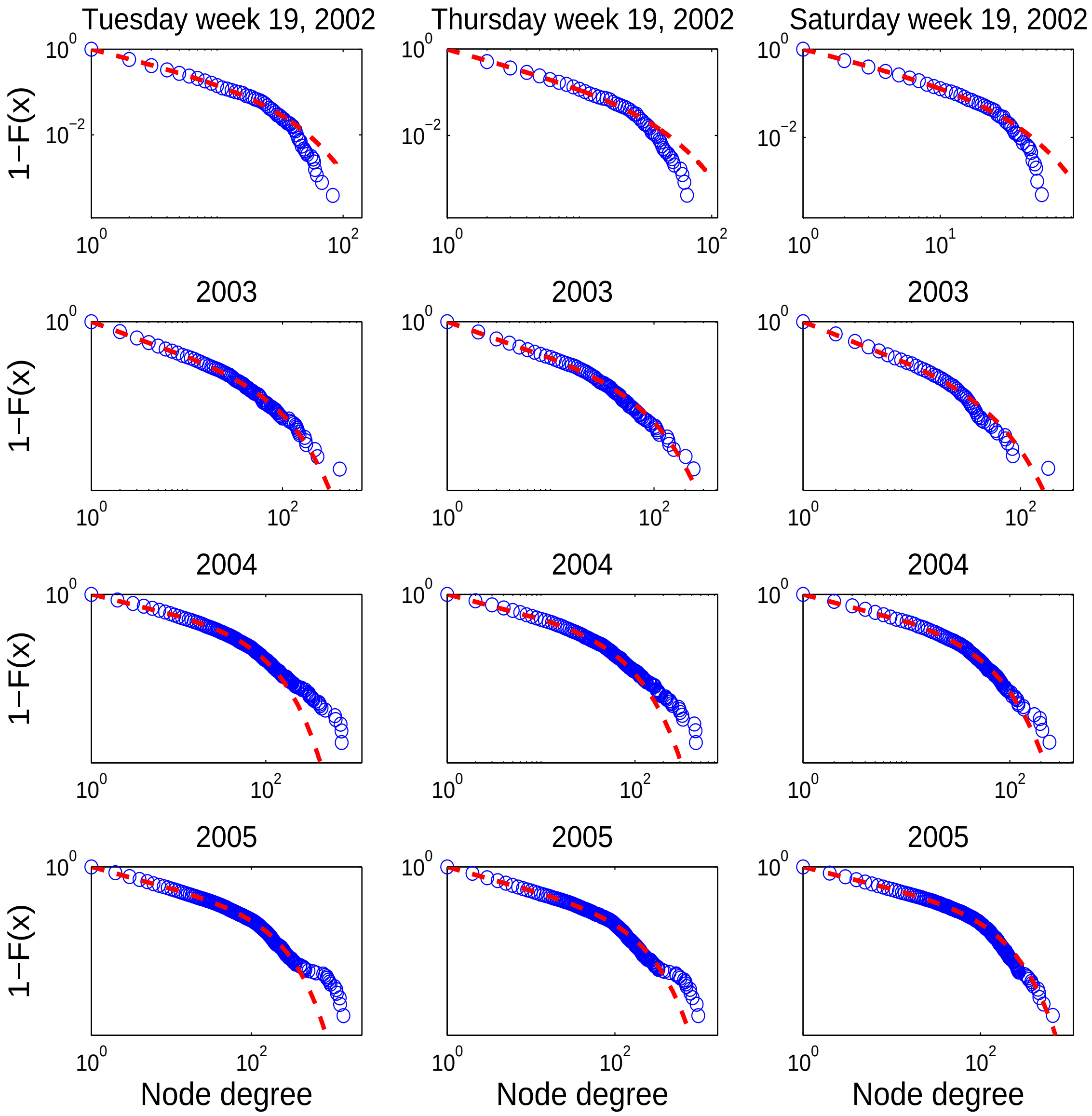

Figure 2.7: Fits of movie degree distributions for three different days during week 19 of years 2002-2005

hypothesis. In Figures 2.6 and 2.7 we show a few examples of the results obtained from fitting the CDFs of the movies and the users respectively to the power-law with cut-off distribution. These images show a few of the results obtained from the subsets of the data from different dates. Each column belongs to one day of the week and each column to a year. Something curious that we noted in the movie degree distribution from the data, is that around weeks 18-20 (about the end of April and beginning of May), the degree distribution has a deformation, a "kick" in the nodes higher degrees. This can be clearly seen in the bottom left image of Figure 2.7 that corresponds to the Tuesday of week 19 of 2005. This was also observed in the data at other dates, but around these weeks it was particularly prominent.

As mentioned in the previous section and is expressed in Figure 2.2, the dataset displays recurrent behaviour in the number of ratings in the database depending on the day of the week. We will look for differences in the degree distributions of movies and users on all weekdays that will allow us to distinguish one day from the other just by looking at the distributions. In Tables 2.2 and 2.1 we show the mean and variance fitting parameters of movie and user CDFs to the form of equation (2.1) by day of the week for the all the ratings between January 2000 and December 2005, Although there is not too much variation in the values for the different days of the week, we do note what we had already seen in the description of the data (Figure 2.2). Tuesdays and Wednesdays, the days that have more ratings, have their movie degree distributions closer to each other than to the rest of the weekdays. Mondays, Thursdays and Fridays, that have similar number of ratings per day, also have degree distributions closer to each other. While in the days with less activity, Saturday and Sunday the degree distributions are also close. See Figure 2.8 for a plot of these distributions using the mean values of the fits. In the users' data from Table 2.2 we can also see some differences in the coefficients. As with the movie degree distribution, Tuesday and Wednesday are closer together than the to rest of the days. Saturday and Sunday also appear together. To see the degree distributions of the users and the movies follow weekly patterns is something that was to be expected given the previous data descriptions.

2.2.2 Clustering coefficients

To understand better the structure of the network, we now turn to clustering coefficients to find out how well connected are nodes in their neighbourhoods and what is the impact of highly connected nodes. As explained in Section 1.1, bipartite networks cannot have triangles because two nodes of the same type cannot be neighbours. For

	Movie degree distribution fit parameters					
	a		b		c	
	mean	var	mean	var	mean	var
Monday	0.6616	0.0285	0.0707	0.0093	1.0843	0.012
Tuesday	0.6481	0.0276	0.0553	0.006	1.0651	0.0073
Wednesday	0.6458	0.0205	0.0596	0.0082	1.0698	0.0112
Thursday	0.6436	0.0235	0.0688	0.0107	1.0789	0.0155
Friday	0.6544	0.0239	0.0695	0.0101	1.0789	0.0146
Saturday	0.6775	0.0262	0.074	0.0123	1.0822	0.0226
Sunday	0.6748	0.0335	0.0822	0.0137	1.0937	0.0211

Table 2.1: Fitting parameters of movie degree distribution per weekday from 2000 to 2005 to a power-law with cut-off model.

	User degree distribution fit parameters					
	a		b		c	
	mean	var	mean	var	mean	var
Monday	0.8426	0.0672	0.0123	0.00019842	1.0159	0.00064487
Tuesday	0.8789	0.0600	0.0118	0.00038765	1.0139	0.00090404
Wednesday	0.8769	0.0694	0.0119	0.00035937	1.0124	0.00086279
Thursday	0.8681	0.0664	0.0102	0.00015413	1.0071	0.00064731
Friday	0.8661	0.0626	0.0097	0.00012747	1.0046	0.00063994
Saturday	0.7838	0.0387	0.0122	0.0002455	1.0126	0.00082575
Sunday	0.7497	0.0368	0.0132	0.00010061	1.0194	0.00076648

Table 2.2: Fitting parameters of user degree distribution per weekday from 2000 to 2005 to a power-law with cut-off model.

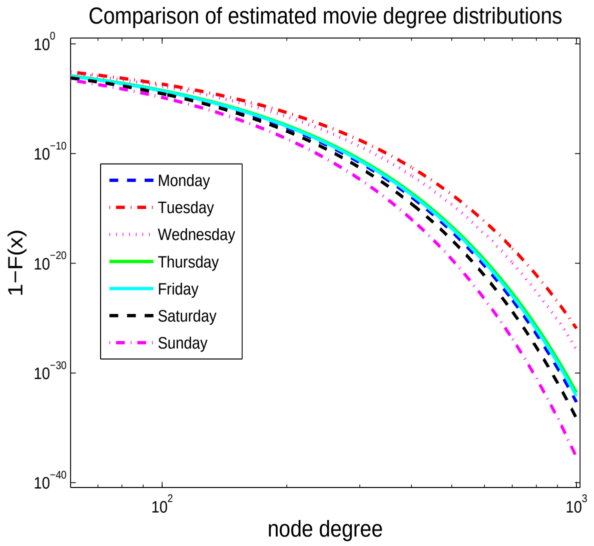

Figure 2.8: Plot of the distributions using fitting parameters of different weekdays.

that reason we cannot calculate the clustering coefficients $C_3(i)$ or C_3 on $\mathbf{G}$. Squares do exist in a bipartite network, so we can calculate $C_4(i)$ and C_4. We remember the definition of $C_4(i)$ for a node v_i:

$$C_4(i) = \frac{\sum_{h,m} q_{i_{mh}}}{\sum_{m,h} \left[(k_m - \eta_{i_{mh}})(k_h - \eta_{i_{mh}}) + q_{i_{mh}}\right]},$$

where $q_{i_{mh}}$ is the observed number of squares for two neighbours v_n, v_m of v_i, and the denominator is the possible number of squares. In this case we will never have an edge between two neighbours of v_i so $\theta = 0$ and $\eta_{i_{mh}} = q_{i_{mh}} + 1$. In Figure 2.9 we can see the values of $C_4(i)$ attained by the nodes of movies and users in the bipartite network. Although the nodes are in the same network, they are shown separately so we can compare users to users and movies to movies. The values of $C_4(i)$ we observe are quite small. This is because there may be, for example, a movie watched by many people who have not other movies in common but that one. This can happen when a movie is so popular that people with normally incompatible taste in films (or no common neighbours) rate it. Such films are not part of many squares but the number of possible squares that could include them is quite high because of its degree, so the value of their $C_4(i)$ and its neighbours' will be lower. This makes the value of the clustering coefficient of the whole network on that day be $C_4 = 0.0014$.

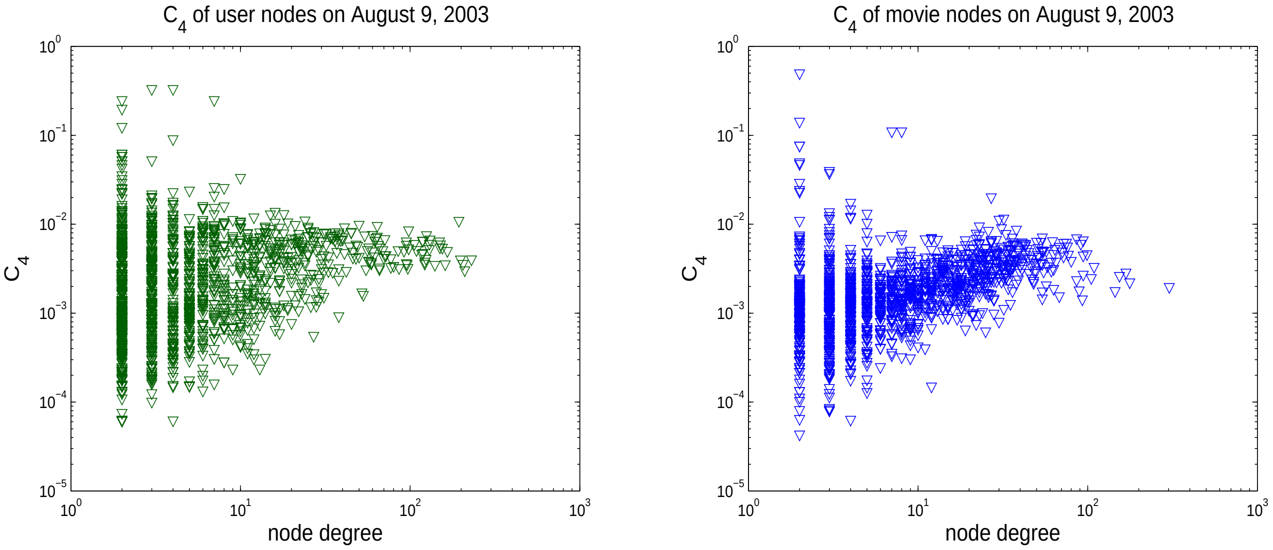

Figure 2.9: Clustering coefficients C_4 for the users (left) and movies (right) in the bipartite network of ratings on August 9, 2003.

We also look at the clustering coefficient of the projected networks of users $\mathbf{GG}^T$ and movies $\mathbf{G}^T\mathbf{G}$. Here triangles do form and we can calculate $C_3(i)$ for all nodes and C_3 for the projected networks. In Chapter 1 we gave the definition of $C_3(i)$:

$$C_3(i) = \frac{2t_i}{k_i(k_i - 1)},$$

where k_i is the degree of v_i and t_i is the maximum possible number of triangles that include v_i. The coefficient of the entire network is

$$C_3 = \frac{1}{M}\sum_{i=1}^{M} C_3(i).$$

Users in their projected network $\mathbf{G}\mathbf{G}^T$ go from having as degree the number of movies seen by them, to have a degree that is equal to the number of all users who have seen at least one of their movies . A zero degree, which was nonexistent in $\mathbf{G}$, is possible when a node has degree 1 and its neighbour has also degree 1 in the bipartite network. Although the degrees of the nodes change and normally the number of connections increases greatly in the projected networks, the graph remains sparse. For example, in the daily snapshots of year 2003 we found that the fraction of entries used in the matrix goes from 0.001015 in $\mathbf{G}$ to 0.01681 in $\mathbf{G}\mathbf{G}^T$ for the users, and to 0.05413 in $\mathbf{G}^T\mathbf{G}$ for the movies. In Figure 2.10 we show an example of the clustering coefficient of all the users and movies in the subset of Netflix ratings corresponding to August 9, 2003. The value of the coefficient for the entire network of users is $C_3 = 0.7904$

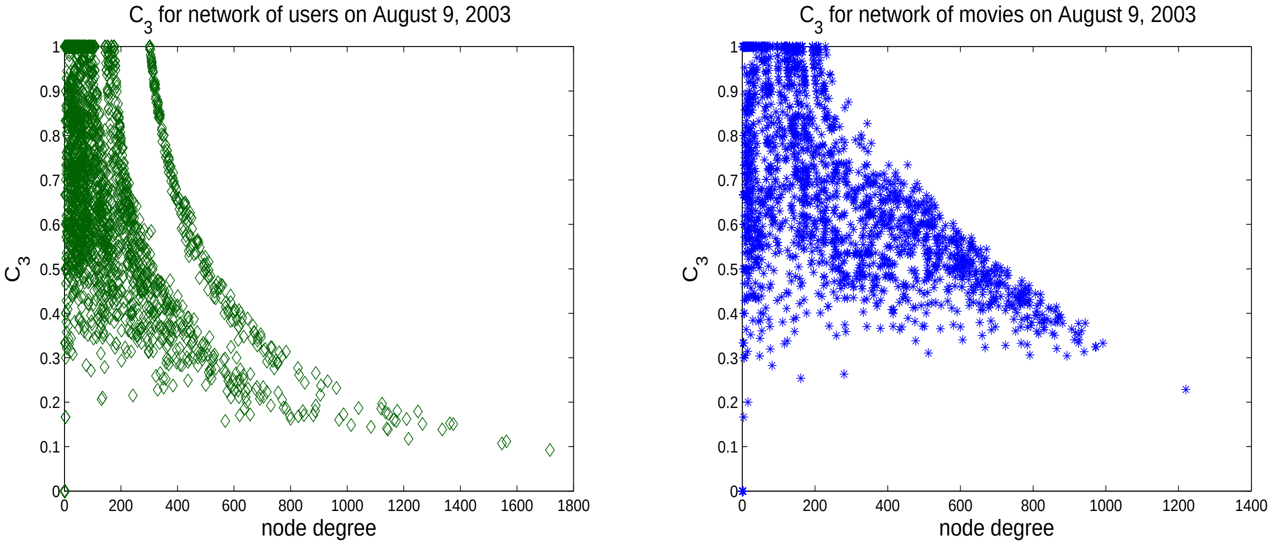

Figure 2.10: Clustering coefficient C_3 for projected networks of users (left) and movies (right) on August 9, 2003

and the value of the coefficient of a random graph of similar size (5,073 nodes and 507,368 edges) is $C_{3Rand} = 0.01943$. This is sign that there is an underlying structure in the network that is very different from random. There are many nodes up to degree 200 that are completely connected, and many more that are in very highly connected neighbourhoods. The value of the coefficient for the network of movies on the same day is $C_3 = 0.7195$, and for a random graph of the same size (3,694 nodes and 587,560 edges) is $C_{3Rand} = 0.0430$. In here we can also appreciate many movies that are very connected, up to degree 300 with $C_3(i) = 1.0$.

In the plot of the users' C_3 in Figure 2.10 we can see a separate set of nodes that stand apart from their peers and have visible higher values of $C_3(i)$ than all the other nodes with the same degree, this may be the effect of a dominant movie in the network. On the day of the example shown, the movie with the highest degree in the bipartite network was *The Bourne Identity* with 303 ratings. Removing that movie's entries from $\mathbf{G}$, changes the clustering coefficient of the users network $\mathbf{GG^t}$ to $C_3 = 0.7737$ (0.0167 lower than the original value observed in the original network). If we plot the values of C_3 for the projected network without this film, the group of nodes that had a higher value of $C_3(i)$ has been integrated to the rest of the nodes (see Figure 2.11), which means that *The Bourne Identity* is responsible for a great deal of triangles in the projected network. This behaviour was not observed in the network of movies, because it is very unlikely to find a user who has rated a significant proportion of the films enough to alter the structure of the projected network of movies.

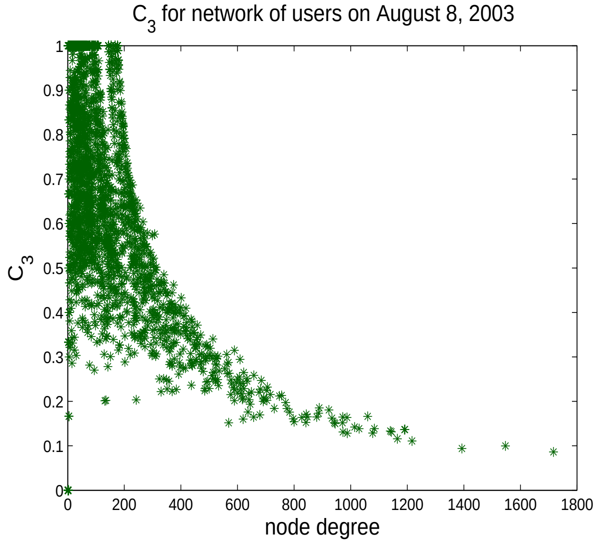

Figure 2.11: Clustering coefficients C_3 of the projected network of users without the top film on August 9, 2003.

	Degree	**Title**	**Year**
1	232,944	*Miss Congeniality.*	2000
2	216,596	*Independence Day.*	1996
3	200,832	*The Patriot.*	2000
4	196,397	*The Day After Tomorrow.*	2004
5	193,941	*Pirates of the Caribbean: The Curse of the Black Pearl.*	2003
6	193,295	*Pretty Woman.*	1990
7	181,508	*Forrest Gump.*	1994
8	181,426	*The Green Mile.*	1999
9	178,068	*Con Air.*	1997
10	177,556	*Twister.*	1996

Table 2.3: The 10 most rated movies in the Netflix dataset.

2.2.3 Ratings of movies

In networks grown using preferential attachment, nodes that have many more edges than the rest have a higher probability of receiving more edges from new incoming nodes. These networks have been subject of much study in recent years [3, 4, 22, 26]. Given that the Netflix data can be viewed as a social network (or more properly, an affiliation network) and that we have observed power-law distributions with cut-off in the movie degrees, we want to see how does the degree of a movie grow in time. In the information of the ratings we also have the date in which the user rated the movie, this is very valuable information because we can use it to tell the age of a node and we are able describe the evolution of every node as the network develops.

We begin looking at some of the movies with the highest degrees. The highest ten are shown in Table 2.3. To see how does the degree of a movie grows, we sort the ratings chronologically and see how the degree of the movies grow in time. To make a correct assessment of how this growth is, we must work within the context of how fast is the network growing and put our measurements in this scales. Let l_t be the number of links that entered the network on day t, $k_{i,t}$ the degree of the movie node m_i on day t and $l_{i,t}$ the number of links from l_t that belong to movie m_i. We define the relative popularity of a film on a day t as $l_{i,t}/l_t$, and the degree of a movie relative to the growth of the network as $k_{i;t}/l_t$. For example, the film *Pirates of the Caribbean: The Curse of the Black Pearl*, which is one of the most rated movies in the database, has its release date in the during the dataset's time-span. This allows us to see its degree evolution from the first rating it received. In Figure 2.12 are the plots of the relative popularity and growth of the film throughout its history in the dataset. We can see in the plot on the left if the figure how the film had a sudden rise of popularity between

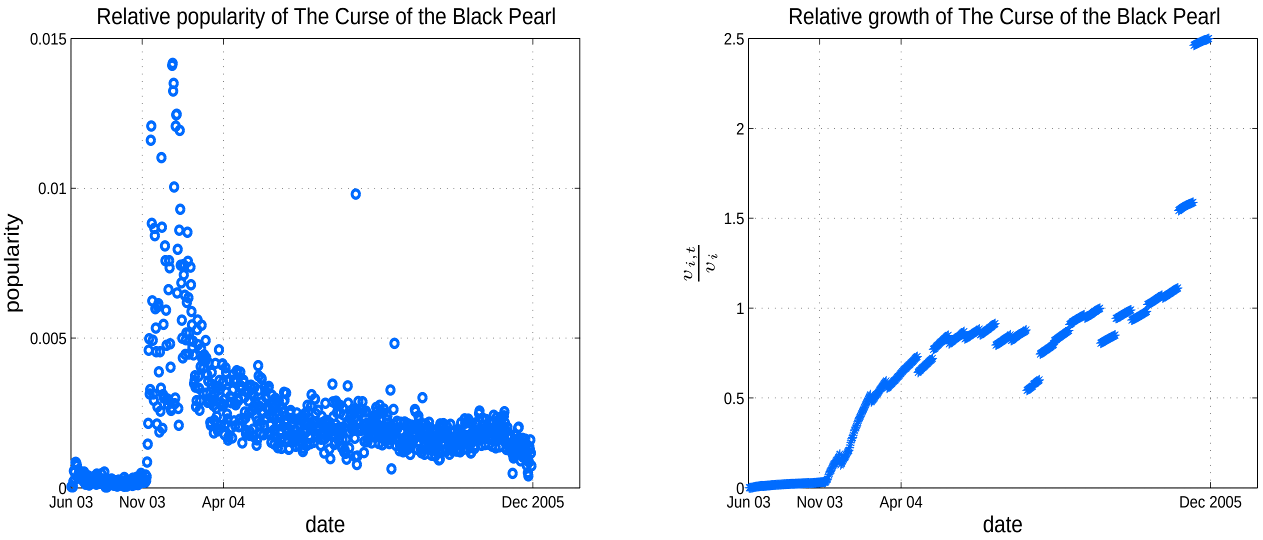

Figure 2.12: Relative popularity (left) and relative growth (right) of the film *Pirates of the Caribbean: The Curse of the Black Pearl* throughout its history in the Netflix dataset.

November 2003 and April 2004. After this period it remains a popular film with an essentially steady influx of ratings. In the plot on the right we see its relative growth. Between November 2003 and April 2004, just after its much anticipated release on DVD, the degree of the movie grows dramatically, then it continues to grow but at a slower rate. On the other hand, *Pretty Woman* was released in 1990, nine years before the start of Netflix. It also is one of the most-rated films but does not have a breakthrough like in the case of *The Curse of the Black Pearl*. This is a movie that has been a constant popular choice of the costumers and in Figure 2.13 we see that, although there are some oscillations, the relative popularity of the film remains. This could imply that once the degree of a film has reached a the status of a dominant film, it does not acquire every time a bigger share of the new edges but grows steady pace which can help to explain the saturation seed in the degree distributions. This tells us that although the degree of a node plays an important role on the number of new edges that it gets, there are other processes influencing the attachment of the edges. As we know, in the film industry there are very intense promotional campaigns of movies intended to introduce them into the market. This, from the networks point of view, can be understood as making a movie node artificially more attractive.

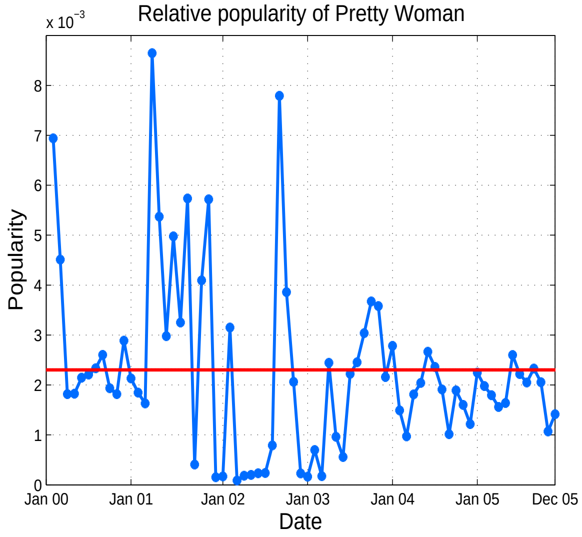

Figure 2.13: Relative popularity the film *Pretty Woman* throughout its history in the Netflix dataset. The red line indicates the mean.

Values of the ratings

During the course of this project a paper was posted on the arXiv[3] pre-print server by Lorenz [18] in which he showed that the histograms of film ratings from the Internet Movie Database[4] (`imdb`) have two or three peaks. Moreover, he showed that these characteristics can be approximated by Lévy skew α-stable distributions. In the Netflix dataset the ratings are single peaked. This can be because the values that a rating can assume are only integers from 1 to 5, while in `imdb` the ratings are also integers but go from 1 to 10, which allows users to be more specific when they enter a rating for a film. In Figure 2.14 we show a few individual-movie histograms and the histogram of all the ratings in the dataset.

[3] *http:/.arxiv.org*

[4] *http://www.imdb.com*

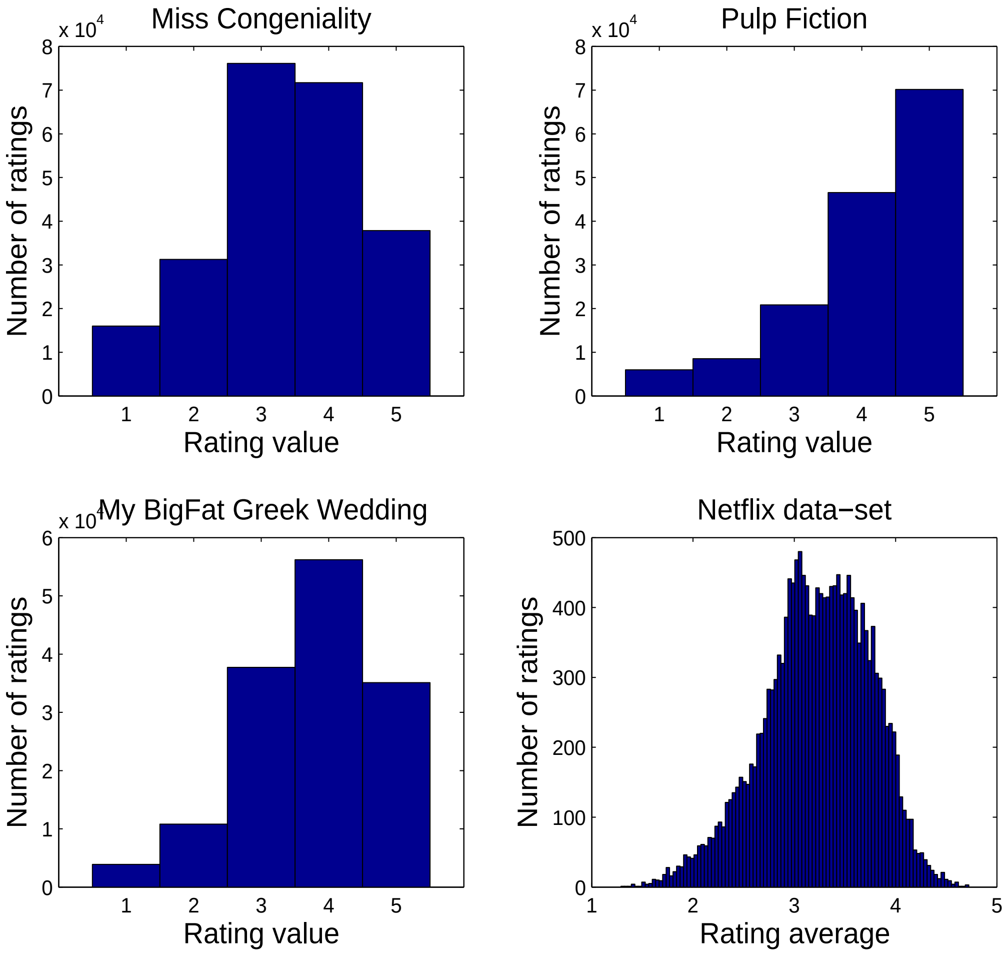

Figure 2.14: Histograms of the ratings of some of the movies from all dates in the database (top and bottom left) and of the average ratings of all films in the Netflix dataset (bottom right).

Chapter 3

Catalogue and logistic network growth

3.1 Logistic attachment

In the Netflix that we discussed in Chapter 2, we saw that the degree distribution of the movies exhibits exponential decay in its tails. This can indicate saturation in the degrees of the nodes. For example, a movie that is a big hit acquires many ratings at the time of its release (i.e. when it is added to the network), but once most people who have interest have seen it, it acquires ratings (edges) at a slower pace. In this section we will explore some growth mechanism ideas that could reproduce these observations. We borrow the concept of *carrying capacity* from population dynamics and use it to describe the network's "bound" on the attractiveness of a film.

Logistic growth

There are many examples of saturation in nature. One of the best known examples comes from population dynamics where the population of a species grows for as long as the resources allow it. If $y(t)$ is the population of a species at a time t, Verlhulst suggested that its change was described by the equation [20, 14]:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = ry\left(1 - \frac{y}{c}\right), \qquad y(t_0) = y_0. \tag{3.1}$$

The constant $c > 0$ is the carrying capacity of the environment and $r > 0$ is the reproductive parameter. The idea behind this equation is that a population will grow as long as the environment can sustain it. We can see from the negative term of equation (3.1) that the change in the population limited by its size relative to the capacity of the environment. When y is close to zero, we can neglect the quadratic term in the equation and we will observe exponential growth of the population. When

$y = c$ then $\frac{\mathrm{d}y}{\mathrm{d}t} = 0$ and it is a steady equilibrium, which means that the population will settle [20]. The logistic equation has solution:

$$y(t) = \frac{y_0 c e^{rt}}{c + y_0 \left(e^{rt} - 1\right)}. \tag{3.2}$$

This function is plotted in Figure 3.1. We can see how it grows exponentially for early values of t and it grows every time slower as $y(t)$ approaches c, which in this case is 1.

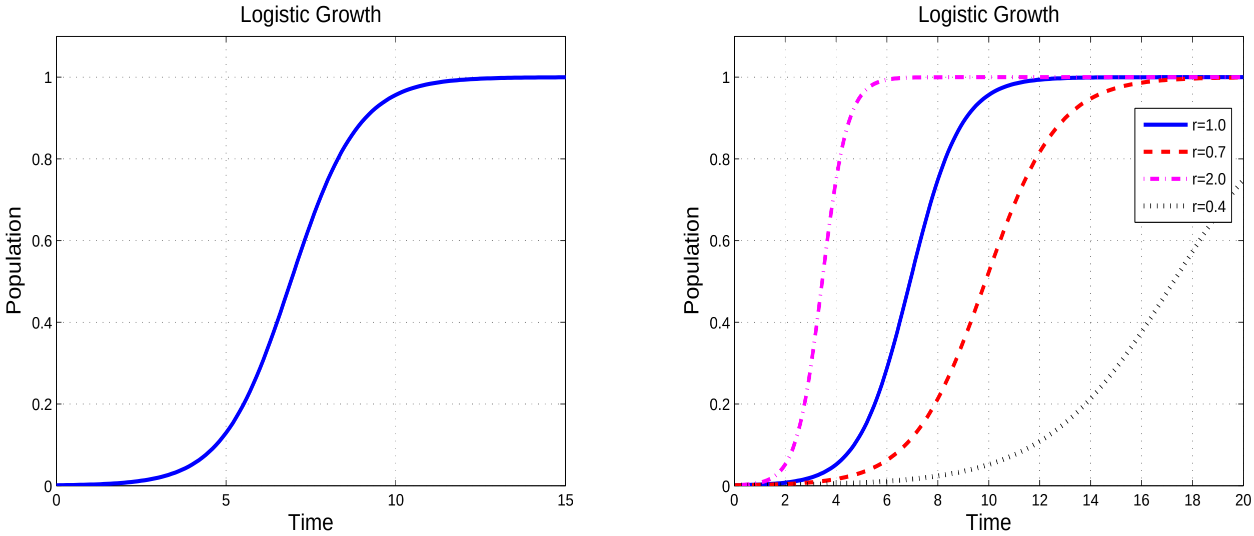

Figure 3.1: Example of logistic growth. Different values of the reproductive parameter r in equation (3.2).

Logistic attachment mechanism

Self-limiting behaviour is not alien to networks science, it has been observed empirically by Newman in networks of collaborations in scientific journals [21]. Scientists are connected if they have collaborated in at least one paper. Newman asserted that the probability of collaboration between two scientists P_m is a function of the number of common co-authors m and has the form:

$$P_m = A - Be^{-m/m_0},$$

where A, B and m_0 are constants [21], and displays saturation as m grows.
As it was mentioned before, in the Netflix network we saw some nodes that appeared to be saturated with edges. This motivates us to develop an attachment mechanism which combines preferential attachment with the ideas behind self-limiting growth of

the logistic equation. Let $\mathcal{G}$ be a unipartite network with M nodes. Every time-step we add a new node that will have an edge connecting it to one of the pre-existing nodes in the graph. We define the probability P_{v_i} that a new edge in the network connects to a node v_i to be logistic function y in equation (3.2), evaluated at its degree k_i:

$$P_{v_i} = \frac{f(k_i)}{\sum_{j=1}^{M} f(k_j)}. \tag{3.3}$$

The idea behind this attachment mechanism is to have the attractiveness of a node be the result of applying the solution to the logistic equation to its degree. This way, the attractiveness will behave like the function on Figure 3.1, where it grows exponentially for the first few edges that the node receives. After some more edges the attractiveness does not grow as fast, and eventually stops growing. The carrying capacity of the network c will determine the maximum attractiveness of the nodes. The reproductive parameter r determines how fast will the nodes get to their maximum attractiveness. Higher values of r will mean faster ascent (see the image on the right of Figure 3.1). The initial condition y_0 is also important. A small value of y_0 will make the function grow very fast initially. A large value of y_0 (closer to c) will bring slower growth. A value of y_0 greater than c is unphysical because it would imply that a node loses edges which doesn't happen in our model.

To get the solution of the logistic equation into a more suitable form, it is best to nondimensionalise equation (3.1). Take $y = c\hat{y}$ and $t = \hat{t}$. In this particular case we wish to keep the time-scale (or more properly, the degree-scale), so we leave r as a parameter whose value we can change to see its effect on the system. Equation (3.1) now becomes:

$$\frac{\mathrm{d}\hat{y}}{\mathrm{d}\hat{t}} = r\hat{y}(1-\hat{y}), \qquad \hat{y}(0) = \hat{y}_0, \tag{3.4}$$

which has solution (dropping hats):

$$y(t) = \frac{y_0 e^{rt}}{1 + y_0(e^{rt} - 1)}. \tag{3.5}$$

Now we can rewrite equation (3.3) as:

$$P_{v_i} = \frac{y_0 e^{rk_i}}{1 + y_0(e^{rk_i} - 1)} \left[\sum_{j=1}^{M} \frac{y_0 e^{rk_j}}{1 + y_0(e^{rk_j} - 1)} \right]^{-1}. \tag{3.6}$$

The carrying capacity used, without loss of generality, for our tests and simulations was $c = 1$. This is because any value of c can be rescaled back into 1. The initial condition y_0 is also kept as a parameter of the model. On the left of Figure 3.2

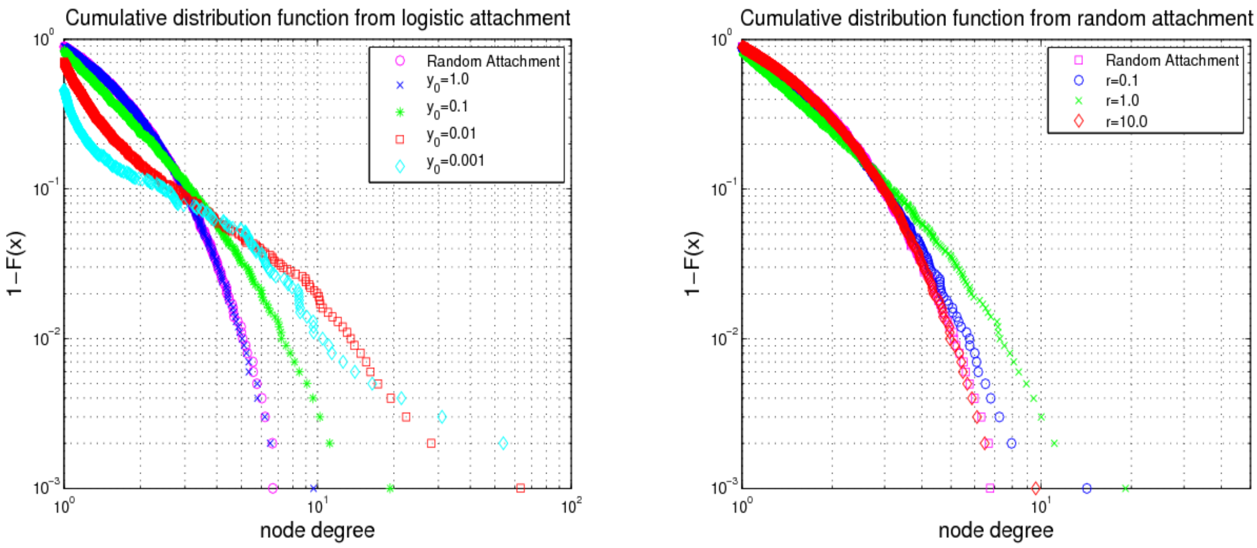

Figure 3.2: Comparison of CDFs of simulated networks using different values of y_0 (left) and r (right) in the logistic attachment model.

we see the cumulative distribution functions of the mean degree distribution of 1000 simulations of a network. Each simulated network contained 1000 nodes, and each node had one edge to attach to the pre-existing nodes. Different values of the initial condition y_0 were used. One network was generated through uniform attachment (i.e. $P_{v_i} = 1/M$) for the sake of comparison. The degree distribution for $y_0 = 1$ (blue crosses) was the same as the uniform attachment (magenta circles). This not surprising as equation (3.5) is always 1 in this case. Networks generated through uniform attachment have exponential degree distributions [4]. For the other values of the initial condition we observed different results. When $0 < y_0 \ll 1$ the first nodes that got edges took off very quickly and got a high degree very fast, but a dominant node did not appear here as it does in networks with superlinear attachment. This is because as a node gains edges the change in its attractiveness is every time less. So a node with a high degree might be much more attractive compared to those nodes with degree 1 but not too different from those who have, say 10 edges more. For example, when $y_0 = 0.01$ and $r = 1$, we have $y(1) = 0.0267$ and $y(5) = 0.5998$. This means that a node with degree 5 is 22 times more attractive than a node with degree 1. But $y(10) = 0.9955$, so a node with degree 10 is only 1.65 times more attractive than one with degree 5. This is why when y_0 is very low we get the fat tails we see on the plot. The image on the right of Figure 3.2 shows a similar experiment, using logistic attachment with an initial value $y_0 = 0.1$ and several values of r we simulated 1000 networks of 1000 nodes each. None of the networks showed a degree distribution

very different from the exponential of the uniform attachment model.
The aim of logistic attachment was to reproduce the saturation effect observed in the degree distributions of some networks. As we saw from the results, logistic attachment produces networks that are similar to the ones produced by uniform attachment or with unusually fat tails that, however, if the network is let to grow for a very long time will eventually converge to a uniform network. This is in part due to the bound that the carrying capacity imposes on the attractiveness of the nodes. After some time, most nodes eventually "catch up" and become equally attractive. Alternative attempts to reproduce self-limiting behaviour might include the use of other functions that, while expressing saturation, do not bound the attractiveness like the logarithm of the degree. This is a topic that could yield interesting results in future research.

3.2 Catalogue growth networks

The Netflix dataset described in Chapter 2 motivated the development of growth and evolution mechanisms of bipartite networks in which the nodes come from predefined lists or *catalogues*.

3.2.1 Decreasing node-fitness

Suppose we have two sets of nodes, *individuals* $\mathcal{I} = \{u_1, u_2, \ldots, u_U\}$, *artifacts* $\mathcal{A} = \{m_1, m_2, \ldots m_M\}$, an empty graph $\mathcal{G}$ and two constants $a, b > 0$. We define the vectors D_u and D_m as:

$$D_u = \begin{bmatrix} a + k_{u_1} \\ a + k_{u_2} \\ \vdots \\ a + k_{u_U} \end{bmatrix}, \qquad D_m = \begin{bmatrix} b + k_{m_1} \\ b + k_{m_2} \\ \vdots \\ b + k_{m_M} \end{bmatrix}. \tag{3.7}$$

Each entry in the vectors is the degree of the node plus a constant. The mechanism starts with an empty network at $t = 0$. Every time-step two nodes, one individual and one artifact, are chosen from the catalogues and a binary edge is placed between them. Each individual and artifact are chosen with probabilities

$$P_{u_i} = \frac{a + k_{u_i}}{||D_u||_1}, \qquad P_{m_j} = \frac{b + k_{m_j}}{||D_m||_1}. \tag{3.8}$$

The constants a and b are the *initial attractiveness* of the individuals and artifacts as defined in Price's model and in the BA with shifted linear kernel [2, 22]. In this section when we talk about the degree k_i of any node, we will be referring to the degree of the node at the time t, or $k_i(t)$ to simplify notation. This is why at any time t the norms of the catalogue vectors (3.7) are:

$$||D_u||_1 = \sum_{i=1}^{U} [a + k_{u_i}] = aU + t,$$

$$||D_m||_1 = \sum_{i=1}^{M} [b + k_{m_j}] = bM + t,$$

because only one edge is added at every time step and there are no duplicate edges. If $N_k(t)$ denotes the number of individuals with degree k, then the probability of choosing at random an individual with degree k is

$$P_k(t) = \frac{a+k}{aU+t} N_k(t). \tag{3.9}$$

All the equations and calculations that follow will be done on the nodes of individuals unless stated otherwise. The results for the artifacts are analogous, one has just got to change a for b, U for M and vice-versa.

A master equation approach to this mechanism is similar to the one employed by Krapivsky and Redner to study the growth of unipartite networks with a shifted linear kernel [15].

$$\begin{aligned}
\frac{\mathrm{d}N_0}{\mathrm{d}t} &= -\left[\frac{a}{aU+t}\right] N_0, && N_0(0) = U,\\
\frac{\mathrm{d}N_k}{\mathrm{d}t} &= \left[\frac{a+(k-1)}{aU+t}\right] N_{k-1} - \left[\frac{a+k}{aU+t}\right] N_k && N_k(0) = 0,\\
& && k = 1, \dots, M-1,\\
\frac{\mathrm{d}N_M}{\mathrm{d}t} &= \left[\frac{a+(M-1)}{aU+t}\right] N_{M-1} && N_M(0) = 0.
\end{aligned} \tag{3.10}$$

The negative terms in the equations account for the loss of nodes that acquire an edge and stop having degree 0 or k. There is no loss term for $\frac{\mathrm{d}N_M}{\mathrm{d}t}$ because M is the maximum degree that a node can attain (i.e. there only exist M artifacts). The positive terms represent the gain of nodes. There is no gain term in $\frac{\mathrm{d}N_0}{\mathrm{d}t}$ because a node cannot lose edges and therefore its degree cannot decrease.

The set of individuals active in the network at a time t is $\mathcal{I}_{>0} = \{u \in \mathcal{I} : k_u > 0\}$, and has size

$$|\mathcal{I}_{>0}| = \sum_{k=1}^{M} N_k(t) = U - N_0(t). \tag{3.11}$$

There is a fixed number of individuals and artifacts available from the catalogues, so at time $t = UM$ the network becomes fully connected, *i.e.* $N_M(MU) = U$.

Nondimensionalisation

To nondimensionalise the model, let $N_k = A\hat{N}_k$ for $k = 0, 1, \ldots, M$ and $t = B\hat{t}$. Substituting in the master equations, for example when $k = 0$ we get :

$$\frac{\mathrm{d}\hat{N}_0}{\mathrm{d}\hat{t}} = -\left[\frac{a}{1+B\hat{t}}\right] B\hat{N}_0,$$

which suggests that we use $A = U$, although it gets canceled but seems reasonable to have $\hat{N}_k(t)$ go from 0 to 1, and $B = aU$. Now equations (3.10) become:

$$\frac{\mathrm{d}\hat{N}_0}{\mathrm{d}\hat{t}} = -\left[\frac{a}{1+\hat{t}}\right] \hat{N}_0, \qquad \hat{N}_0(0) = 1.$$

$$\frac{\mathrm{d}\hat{N}_k}{\mathrm{d}\hat{t}} = \left[\frac{a+(k-1)}{1+\hat{t}}\right] \hat{N}_{k-1} - \left[\frac{a+k}{1+\hat{t}}\right] \hat{N}_k, \quad \hat{N}_k(0) = 0, \quad k = 1, \ldots, M-1 \quad (3.12)$$

$$\frac{\mathrm{d}\hat{N}_M}{\mathrm{d}\hat{t}} = \left[\frac{a+(M-1)}{1+\hat{t}}\right] \hat{N}_{M-1}, \qquad \hat{N}_M(0) = 0.$$

Solution

The model can be solved exactly for all t and k. The solutions are (dropping hats from the nondimensional form):

$$N_0(t) = \frac{1}{(1+t)^a},$$

$$N_k(t) = \frac{(a)_k}{k!(1+t)^{a+k}} t^k, \quad k = 1, 2, \ldots, M-1, \quad (3.13)$$

$$N_M(t) = \frac{(a)_M}{M!} t^M {}_2F_1(M, a+M; M+1; -t),$$

where Pochhammer's symbol $(a)_n$ is defined as

$$(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)} = \prod_{i=0}^{n-1}(a+i),$$

and ${}_2F_1(M, a+M; M+1; -t)$ is the hypergeometric function

$$\begin{aligned} {}_2F_1(a, b; c; z) &= \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!} \\ &= \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_{n=0}^{\infty} \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)} \frac{z^n}{n!}. \end{aligned} \quad (3.14)$$

the functions $N_i(t)$ $i = 0, 1, \ldots, M$ can be very difficult to compute [11], but some properties of Pochhammer's symbol and the hypergeometric function as well as other mathematical techniques like asymptotic expansions may yield simpler expressions for certain values of a. For example, if $a = 1$, Pochhammer's symbol is $(1)_n = n!$, and properties of the hypergeometric function (see equation (A.8) on Appendix A) The equations in (3.13) now become

$$\begin{aligned} N_0(t) &= \frac{1}{(1+t)}, \\ N_k(t) &= \frac{t^k}{(1+t)^{k+1}} \qquad\qquad k = 1, 2, \ldots, M-1, \\ N_M(t) &= \left(\frac{t}{1+t}\right)^M \end{aligned} \tag{3.15}$$

Figures 3.3 and 3.4 show numerical simulations of a network plotted along with

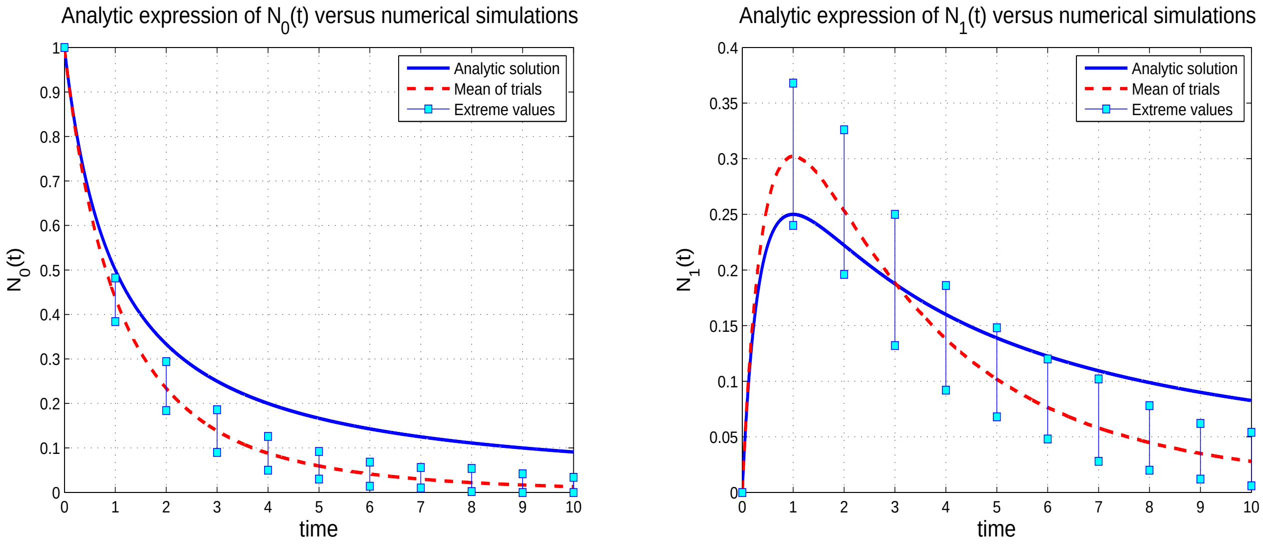

Figure 3.3: Plots of $N_0(t)$ (left) and $N_1(t)$ (right) from 1000 simulations and analytics of a bipartite network in which $U = 500$ and $a = 1$.

evaluations of the analytic solutions outlined by the equations in (3.15), the vertical bars in the plots are the maximum and minimum values observed in the simulations.

Asymptotics

If $0 < a \ll 1$, we can use the expansion $N_k(t) = N_{k_0}(t) + aN_{k_1}(t) + a^2N_{k_2}(t) + \ldots$ to get approximate solutions to the differential equations (3.12) which can be rewritten

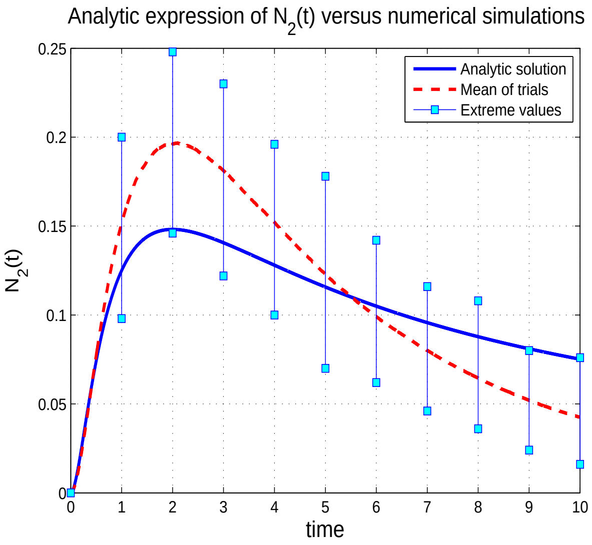

Figure 3.4: Plot of $N_2(t)$ from 1000 simulations and the analytic solution from equation (3.15) of a bipartite network in which $U = 500$ and $a = 1$.

using these expansions as:

$$\frac{\mathrm{d}(N_{0_0}(t) + aN_{0_1}(t) + \ldots)}{\mathrm{d}t} = -\left[\frac{a}{1+t}\right](N_{0_0}(t) + aN_{0_1}(t) + \ldots),$$
$$N_{0_0}(0) + aN_{0_1}(0) + \ldots = 1. \qquad (3.16)$$

$$\begin{aligned}\frac{\mathrm{d}(N_{k_0}(t) + aN_{k_1}(t) + \ldots)}{\mathrm{d}t} &= \left[\frac{a+(k-1)}{1+t}\right](N_{k-1_0}(t) + aN_{k-1_1}(t) + \ldots) \\ &\quad - \left[\frac{a+k}{1+t}\right](N_{k_0}(t) + aN_{k_1}(t) + \ldots),\end{aligned}$$
$$N_{k_0}(0) + aN_{k_1}(0) + \ldots = 0. \qquad (3.17)$$

$$\frac{\mathrm{d}(N_{M_0}(t) + aN_{M_1}(t) + \ldots)}{\mathrm{d}t} = \left[\frac{a+(M-1)}{1+t}\right](N_{M-1_0}(t) + aN_{M-1_1}(t) + \ldots),$$
$$N_{M_0}(0) + aN_{M_1}(0) + \ldots = 0. \qquad (3.18)$$

To solve this equations for terms with the same power of a. The solutions are

$$N_0(t) \approx \sum_{r=0}^{\infty} \frac{[-a \log{(1+t)}]^r}{r!} \tag{3.19}$$

$$N_1(t) \approx a\frac{t}{1+t} + a^2\frac{\log{(1+t)} - t\log{(t)}}{1+t} + \dots \tag{3.20}$$

$$\begin{aligned} N_2(t) \approx a\frac{t^2}{2(1+t)^2} + a^2 \left(\frac{\log{(1+t)}}{1+t} - \frac{t}{1+t} - \frac{1}{4}t^2(2\log{(t)} - 1) \right. \\ \left. - \frac{(t-2)t}{2(1+t)^2} - \frac{\log{(1+t)}}{(1+t)^2} \right) + \dots \end{aligned} \tag{3.21}$$

$$\vdots$$

$$N_M(t) \approx a\Gamma(M)t^M {}_2\tilde{F}_1(M, M; M+1; -t) + ... \tag{3.22}$$

In equation (3.22) the term ${}_2\tilde{F}_1(a, b; c; z)$ is the regularized hypergeometric function defined in equation A.18 in Appendix A (see also equation (C.2) in Appendix C), and $\Gamma(x)$ is the gamma function. The order of each approximation is the highest power of a that is used, for example equation (3.20) shows an order 2 approximation. The purpose of asymptotic expansions is to take advantage of small parameters in the models to get approximate analytic solutions. In this particular case it is only useful for the lower degrees (*i.e.* $k = 0, 1$). Computing the regularised hypergeometric function is more or less the same work than computing the normal one. In this case an approximation to $N_M(t)$ does not simplify calculations and is therefore of little practical use. We can see some comparisons between the analytic solutions of the model with approximations and numerical simulations of the network in Figure 3.5. The model outlined in equations (3.12) seems to capture some of the behaviour of the evolution of the network. However it is evident from Figures 3.3, 3.4 and 3.5 that it does not agree well enough with the simulations, in particular for small values of a. When a is larger, the analytical solutions are much more accurate, as shown in Figure 3.6. This model is called a decreasing node-fitness model. This is because nodes start with an initial attractiveness a, and its relative value to the network catalogues decreases as the system evolves. Individuals that do not receive edges soon enough will have their attractiveness decreasing as $a/(aU + t)$. This is particularly evident when a is small. When a is larger, say $a \approx \mathcal{O}(1)$, the network grows at a faster rate because nodes with no edges have comparable probabilities of receiving edges to those with nonzero degree, at least in the early stages of the process. Figure 3.7

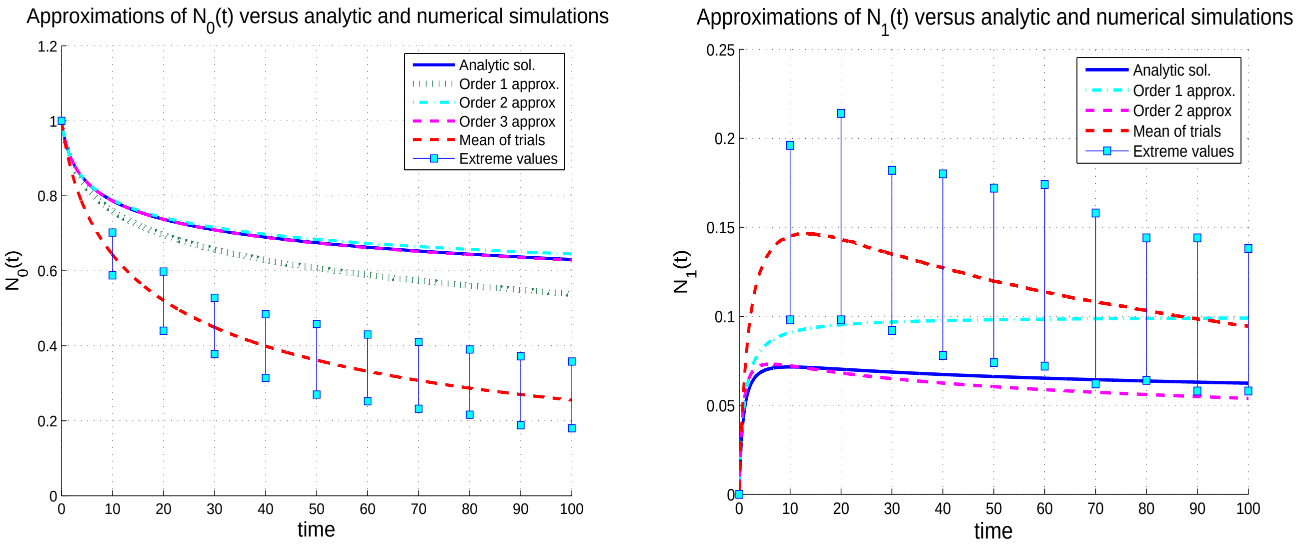

Figure 3.5: Plot of $N_0(t)$ (left) and $N_1(t)$ (right) from 1000 simulations, analytical solution and asymptotic approximations when $U = 500$ and $a = 0.1$.

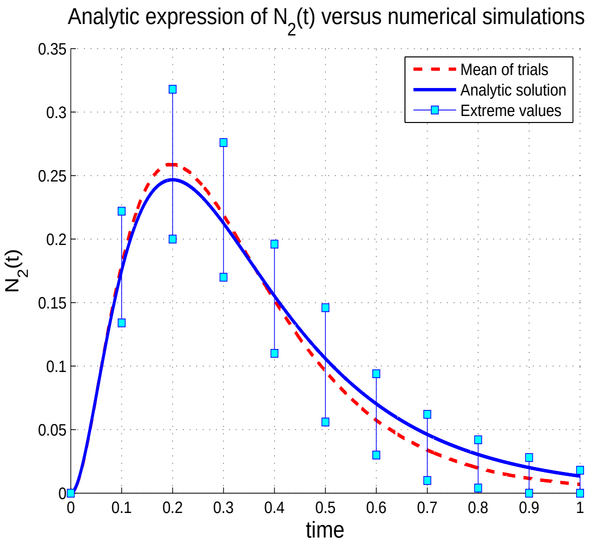

Figure 3.6: Plot of $N_2(t)$ from equation (3.13) when $k = 2$ to 1000 simulations of a network using $U = 500$ and $a = 10$.

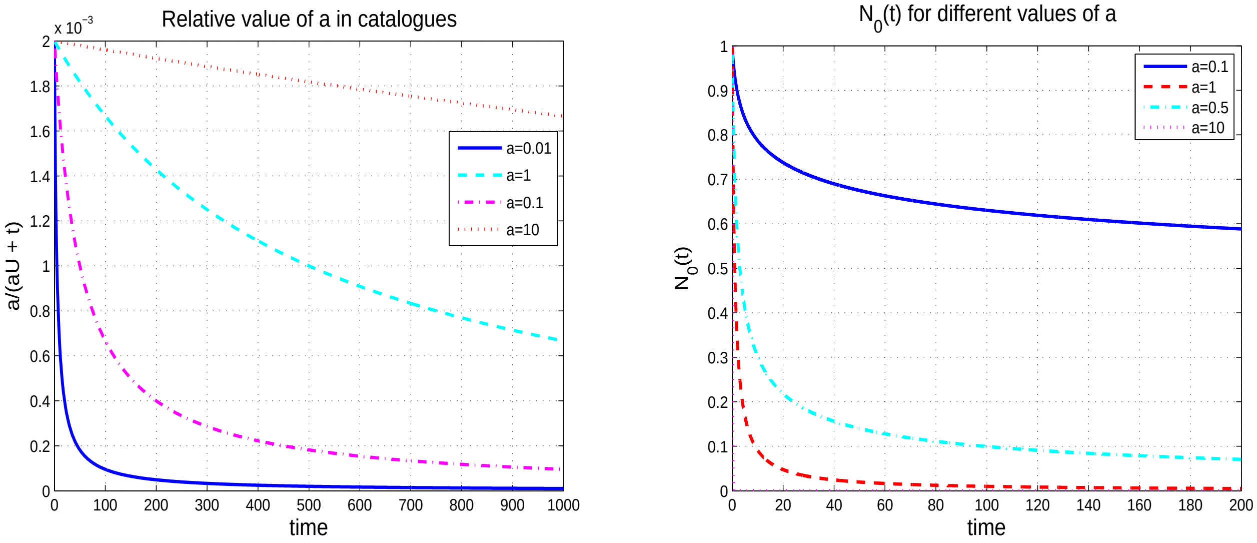

Figure 3.7: (left) Decreasing fitness of nodes according to the value of parameter a. (right) Different realisations of $N_0(t)$ for values of a.

shows the relative value of the initial fitness a of the nodes as the network grows and how it influences this growth. Another way to understand this is to think about a as a nominal value that stays constant and due to some phenomenon (*e.g.* inflation), its real value decreases in time.

3.2.2 Constant node-fitness

In the previous model, we saw that the initial attractiveness of individuals and artifacts decrease in time. While it is an interesting mathematical model in its own right, it fails to capture the true behaviour of the system from which it was motivated. The model was inspired by the Netflix system, where a list of all movies is available to the users at all times. It still sounds like a reasonable assumption that more popular movies get chosen more often, but there is no reason to believe that movies that have not been rated at a time t have an ever-decreasing chance of being chosen. Users can browse through online catalogues and rent whatever movie they find interesting, regardless of its popularity. With this in mind, in the following model we let all nodes have some probability of receiving an edge that is independent of their degree. We use a similar approach to the one by Evans and Plato in their network rewiring model [9]. In the model, an edge is assigned using preferential attachment with probability p and using uniform attachment with probability $1 - p$.

We have the two sets of nodes, individuals $\mathcal{I}$ and artifacts $\mathcal{A}$, an empty graph $\mathcal{G}$ and two constants $p, q \in [0, 1]$. The catalogue vectors D_u and D_m are now simply the degree vectors of the nodes

$$D_u = \begin{bmatrix} k_{u_1} \\ k_{u_2} \\ \vdots \\ k_{u_U} \end{bmatrix}, \qquad D_m = \begin{bmatrix} k_{m_1} \\ k_{m_2} \\ \vdots \\ k_{m_M} \end{bmatrix}. \tag{3.23}$$

The initial conditions of the model must be different as in the previous model. If we started with a completely empty graph at $t = 0$, we would have $||D_m||_1 = ||D_u||_1 = 0$, which would result in division by zero in the probabilities and subsequent equations, and no nodes ever being able to be chosen for connections. To overcome this problem, we let the system begin with two randomly chosen nodes connected as a seed. This is not uncommon in other growth models such as Price's and the BA model which must also start from a non-empty seed network [8, 22, 25]. This is equivalent to a shift in the time variable and a change in the initial conditions so we can have $||D_m||_1 = ||D_u||_1 = 1$ at $t = 0$. Every time step a new edge is added to the network. The individual and artifact nodes are chosen with probabilities

$$\begin{aligned} P_{u_i} &= \left[\frac{1-p}{U} + \frac{pk_{u_i}}{1+t}\right], \\ P_{m_j} &= \left[\frac{1-q}{M} + \frac{qk_{m_j}}{1+t}\right]. \end{aligned} \tag{3.24}$$

Now there is no danger of division by zero and we can move on to the model of the system. The rate equations for the individuals in the network are

$$\begin{aligned} \frac{\mathrm{d}N_0}{\mathrm{d}t} &= -\left[\frac{1-p}{U}\right] N_0, && N_0(0) = U - 1, \\ \frac{\mathrm{d}N_k}{\mathrm{d}t} &= \left[\frac{1-p}{U} + \frac{p(k-1)}{1+t}\right] N_{k-1} && N_1(0) = 1 \\ &\quad - \left[\frac{1-p}{U} + \frac{pk}{1+t}\right] N_k, && N_k(0) = 0 \quad k > 1, \\ & && k = 1, \dots, M-1 \\ \frac{\mathrm{d}N_M}{\mathrm{d}t} &= \left[\frac{1-p}{U} + \frac{p(M-1)}{1+t}\right] N_{M-1} && N_M(0) = 0. \end{aligned} \tag{3.25}$$

Again, the negative terms account for the loss of nodes and the positive terms, for the gain of nodes. The number of individuals in the network is still given by equation (3.11). Let us not forget that the calculations for the artifact nodes are analogous with the adequate change of terms.

Nondimesionalisation

To nondimensionalise equations (3.25) we take $N_k = U\hat{N}_k$ and $t+1 = U\hat{t}$. The model then becomes:

$$
\begin{aligned}
\frac{\mathrm{d}\hat{N}_0}{\mathrm{d}\hat{t}} &= -(1+p)\hat{N}_0, & \hat{N}_0\left(\frac{1}{U}\right) &= 1 - \frac{1}{U},\\
\frac{\mathrm{d}\hat{N}_k}{\mathrm{d}\hat{t}} &= \left[(1-p) + \frac{p(k-1)}{\hat{t}}\right]\hat{N}_{k-1} & \hat{N}_1\left(\frac{1}{U}\right) &= \frac{1}{U}\\
&\quad - \left[(1-p) + \frac{pk}{\hat{t}}\right]\hat{N}_k, & \hat{N}_k\left(\frac{1}{U}\right) &= 0 \quad k > 1,\\
& & k &= 1, \dots, M-1\\
\frac{\mathrm{d}\hat{N}_M}{\mathrm{d}\hat{t}} &= \left[(1-p) + \frac{p(M-1)}{\hat{t}}\right]\hat{N}_{M-1}, & \hat{N}_M\left(\frac{1}{U}\right) &= 0.
\end{aligned}
\tag{3.26}
$$

Solution

The solutions for the nondimensional model are (dropping hats)

$$N_0(t) = a\exp\left\{-(1-p)\left(t - \frac{1}{U}\right)\right\}, \tag{3.27}$$

$$N_1(t) = \left[b + U^{-(p+1)}(1-b)t^{-(p+1)}\right] t\exp\left\{-(1-p)t + \frac{1-p}{U}\right\}. \tag{3.28}$$

$$\vdots$$

The values of a and b are

$$a = \left(1 - \frac{1}{U}\right), \qquad b = a\frac{1-p}{1+p}.$$

One can obtain analytical expressions for $N_k(t)$ when $k > 1$, however they become increasingly tedious to calculate and, the expressions become very large. Asymptotic expansions or a numerical scheme may be a better way to solve this model.

Asymptotics

When p has values close to its extremes, we can obtain approximate solutions to the model in equations (3.26) using asymptotic expansions. If $0 < p \ll 1$ we use the expansion $N_k(t) = N_{k_0}(t) + pN_{k_1}(t) + p^2N_{k_2}(t) + \dots$ and rewrite equations (3.26) to obtain:

$$\begin{aligned} \frac{\mathrm{d}(N_{0_0} + pN_{0_1} + \dots)}{\mathrm{d}t} &= -(1+p)(N_{0_0} + pN_{0_1} + \dots), \\ N_0\left(\frac{1}{U}\right) &= 1 - \frac{1}{U}, \end{aligned} \tag{3.29}$$

$$\begin{aligned} \frac{\mathrm{d}(N_{k_0} + pN_{k_1} + \dots)}{\mathrm{d}t} &= \left[(1-p) + \frac{p(k-1)}{t}\right](N_{(k-1)_0} + pN_{(k-1)_1} + \dots) \\ &\quad - \left[(1-p) + \frac{pk}{t}\right](N_{k_0} + pN_{k_1} + \dots), \\ N_1\left(\frac{1}{U}\right) &= \frac{1}{U}, \quad N_k\left(\frac{1}{U}\right) = 0, \quad k > 1, \quad k = 1, \dots, M-1, \end{aligned} \tag{3.30}$$

$$\begin{aligned} \frac{\mathrm{d}(N_{M_0} + pN_{M_1} + \dots)}{\mathrm{d}\hat{t}} &= \left[(1-p) + \frac{p(M-1)}{t}\right](N_{(M-1)_0} + pN_{(M-1)_1} + \dots), \\ N_M\left(\frac{1}{U}\right) &= 0. \end{aligned} \tag{3.31}$$

After solving for the terms that have with the same power of p we get

$$N_0(t) = e^{-\left(t - \frac{1}{U}\right)}\left(1 - \frac{1}{U}\right)\sum_{k=0}^{\infty}\frac{p^k\left(t - \frac{1}{U}\right)^k}{k!}. \tag{3.32}$$

$$\begin{aligned} N_1(t) \approx e^{-\left(t-\frac{1}{U}\right)}\Bigg\{&\left(1 - \frac{1}{U}\right)t + \frac{1}{U^2} \\ &+p\left[\left(1 - \frac{1}{U}\right)t^2 + \left(\frac{2}{U^2} + \frac{1}{U} - 2\right)t + \frac{\log t}{U^2} + \frac{2}{U} - \frac{1}{U^3} + \frac{\log U}{U^2}\right]\Bigg\}, \end{aligned} \tag{3.33}$$

$$N_2(t) \approx e^{-\left(t-\frac{1}{U}\right)}\left[\left(1 - \frac{1}{U}\right)\frac{t^2}{2} + \frac{t}{U^2}\left(1 + \frac{1}{U}\right)\frac{1}{2U^2}\right]. \tag{3.34}$$

$$\vdots$$

Note that equation (3.32) is equal to equation (3.27) which is why the former is written not as an approximation, but as an equality.

If $0 < (1-p) \ll 1$, we make a change of variables $w = (1-p)$ and use the expansion $N_k(t) = N_{k_0}(t) + wN_{k_1}(t) + w^2N_{k_2}(t) + \dots$. As before, we have to express equations (3.26) in terms of w (see equations (C.3), (C.4) and (C.5) in Appendix C). The approximate solutions are

$$N_0(t) = \sum_{k=0}^{\infty} \frac{(-1)^k \left(1 - \frac{1}{U}\right) \left(t - \frac{1}{U}\right)^k}{k!}, \tag{3.35}$$

$$N_1(t) \approx \frac{1}{U^2 t} + w \left\{ \left(1 - \frac{1}{U}\right) \frac{t}{2} - \frac{1}{U^2} + \frac{1}{U^2 t} \left[\log t - \frac{1}{2} - \frac{3}{2U} - \log U \right] \right\}, \tag{3.36}$$

$$N_2(t) \approx \frac{-1}{2U^4 t^2} + \frac{1}{2U^2}, \tag{3.37}$$

$$N_3(t) \approx \frac{-1}{4U^2 t} + \frac{t}{8U^2} + \frac{2U^3 - 1}{8U^6 t^3}. \tag{3.38}$$

$$\vdots$$

These solutions are complicated to obtain and not very informative. For example the expression for $N_1(t)$ in equation (3.36) is already difficult to read and is only an order 1 approximation of $N_1(t)$. Perhaps only the approximation to $N_0(t)$ is of use. For example, when an approximate size of the network is needed.

Numerical solutions

Given the difficulty of obtaining analytical solutions for all degree values in our model, we must rely on numerical methods to compute solutions. In this work we used the standard four-step Runge-Kutta method [27]. Let $\mathbf{y}(t) = [N_0, \ldots, N_M]^T$, and $f(t, \mathbf{y})$ be the right-hand side of equations (3.26):

$$f(t, \mathbf{y}) = \begin{bmatrix} -(1-p)N_0 \\ \vdots \\ \left((1-p) + \frac{p(k-1)}{t}\right) N_{k-1} - \left((1-p) + \frac{pk}{t}\right) N_k \\ \vdots \\ \left((1-p) + \frac{p(M-1)}{t}\right) N_{M-1} \end{bmatrix}. \tag{3.39}$$

At the initial time $t_0 = \frac{1}{U}$, the initial conditions are $\mathbf{y}_0 = \mathbf{y}(t_0) = \left[1 - \frac{1}{U}, \frac{1}{U}, 0, \ldots, 0\right]^T$. We calculate $\mathbf{y}_{n+1}$ in the following way

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{1}{6} \left[\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right], \tag{3.40}$$

where:

$$\begin{aligned}
\mathbf{k}_1 &= f(t_n, \mathbf{y}_n), \\
\mathbf{k}_2 &= f(t_n, +\tfrac{1}{2}h, \mathbf{y}_n + \tfrac{1}{2}h\mathbf{k}_1), \\
\mathbf{k}_3 &= f(t_n, +\tfrac{1}{2}h, \mathbf{y}_n + \tfrac{1}{2}h\mathbf{k}_2), \\
\mathbf{k}_4 &= f(t_n + h, \mathbf{y}_n + h\mathbf{k}_3).
\end{aligned} \tag{3.41}$$

Figure 3.8 shows values of $\log(N_k(t))$ for all t and k obtained using the model and the values obtained by the simulation of 1000 networks. We have taken the logarithm of the solution so we have a clearer picture of the solutions because all of its values lie between 0 and 1. The plot has time in the x-axis and node-degree in the y-axis, the colour indicates the value of $N_k(t)$. Red indicates areas where $N_k(t)$ is very close to one and dark blue, where $N_k(t) = 0$. This comparison of the solution of the model with the simulations of the network suggests that there is something missing in the model as it fails to reproduce the behaviour observed in the simulated networks, especially as t (dimensional) approaches $UM = 3000$. The network that we are trying to model becomes fully connected at $t = UM$, and in the solution of the model on the left image of Figure 3.8 we see that at the final time there still are plenty of nodes whose degree is lower than 100. In the simulated networks we can see clearly that this is not the case, all nodes are fully connected at the final time as expected. We need to improve this model so the values of $N_k(t)$ go to zero as they should and $N_M(t) = 1$ at the final time.

3.2.3 Constant node-fitness with catalogue update

In networks from catalogues, once a node is fully connected it cannot receive any new edges. In particular, if an artifact is connected to every single individual, it means that there is one less artifact to choose from. This was not accounted for in the previous models and led to serious flaws. To incorporate this behaviour into the model, we must reconsider the probabilities that the individuals and artifacts have of receiving an edge. When an edge is added to the network, its probability of being attached to an individual of degree k using uniform attachment is $N_k/(U - N_M)$, because there are N_M individuals incapable of receiving new edges so it must choose from the rest. The probability of the edge to connect to an individual of degree k using preferential attachment is $kN_k/(1 + t - MN_M)$. The denominator accounts for the number of edges (the sum of all degrees) of the nodes in the catalogue $1 + t$, as

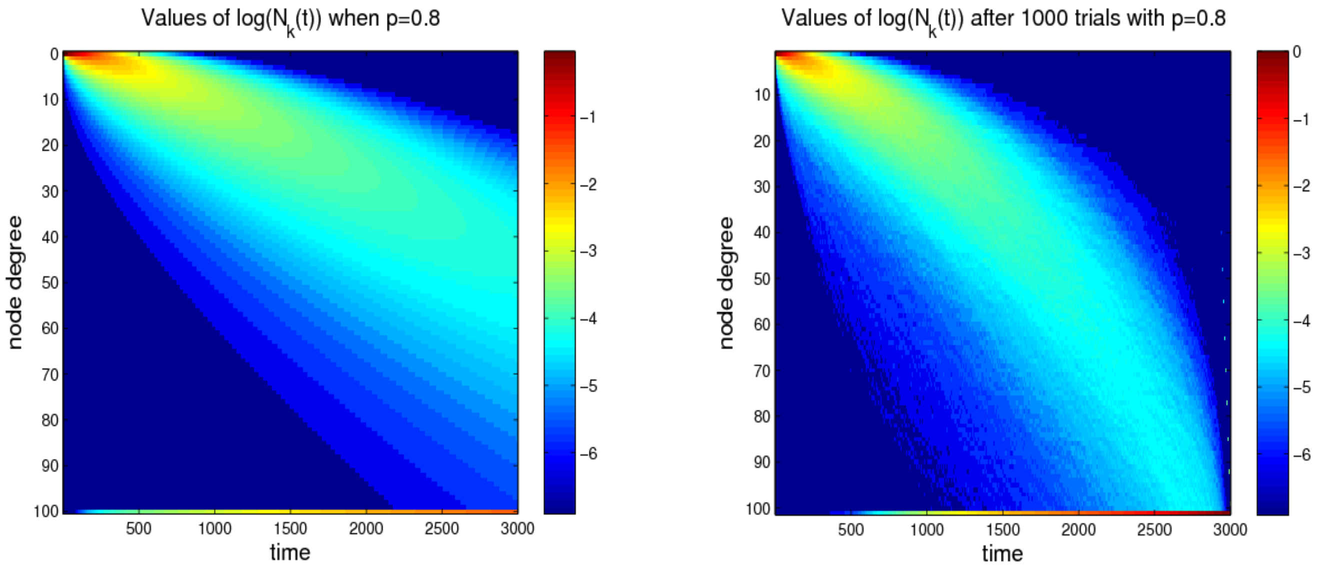

Figure 3.8: Comparison of results obtained by the model of equations (3.26) (left) and the mean values of $N_k(t)$ after 1000 simulations (right) when $M = 100$, $U = 30$ and $p = 0.8$.

in the previous minus the edges of the nodes that are now fully connected $MN_m(t)$. The analog also holds for the probabilities that an edge has of connecting to artifact nodes, which is obtained changing M for U and vice-versa. With these adjustments in the probabilities, we now reformulate the model of catalogue growth:

$$
\begin{aligned}
\frac{\mathrm{d}N_0}{\mathrm{d}t} &= -\left[\frac{1-p}{U-N_M}\right]N_0, && N_0(0) = U-1,\\
\frac{\mathrm{d}N_k}{\mathrm{d}t} &= \left[\frac{1-p}{U-N_M}+\frac{p(k-1)}{(1+t)-MN_M}\right]N_{k-1} && N_1(0) = 1,\\
&\quad-\left[\frac{1-p}{U-N_M}+\frac{pk}{(1+t)-MN_M}\right]N_k, && N_k(0) = 0 \quad k>1,\\
& && k = 1,\dots,M-1,\\
\frac{\mathrm{d}N_M}{\mathrm{d}t} &= \left[\frac{1-p}{U-N_M}+\frac{p(M-1)}{(1+t)-MN_M}\right]N_{M-1}, && N_M(0) = 0.
\end{aligned}
\tag{3.42}
$$

Nondimensionalisation

To nondimensionalise equations (3.42) we scale the degree distribution (as before) $N_k = U\hat{N}_k$ and unlike previous occasions we shift and rescale the time variable

$(1+t) = U\hat{t}$. Now the model becomes:

$$\begin{aligned}
\frac{\mathrm{d}\hat{N}_0}{\mathrm{d}\hat{t}} &= -\left[\frac{1-p}{1-\hat{N}_M}\right]\hat{N}_0, & \hat{N}_0\left(\frac{1}{U}\right) &= 1-\frac{1}{U}, \\
\frac{\mathrm{d}\hat{N}_k}{\mathrm{d}\hat{t}} &= \left[\frac{1-p}{1-\hat{N}_M} + \frac{p(k-1)}{\hat{t}-M\hat{N}_M}\right]\hat{N}_{k-1} & \hat{N}_1\left(\frac{1}{U}\right) &= \frac{1}{U}, \\
&\quad - \left[\frac{1-p}{1-\hat{N}_M} + \frac{pk}{\hat{t}-M\hat{N}_M}\right]\hat{N}_k, & \hat{N}_k\left(\frac{1}{U}\right) &= 0 \quad k>1, \\
& & k &= 1,\dots,M-1, \\
\frac{\mathrm{d}\hat{N}_M}{\mathrm{d}\hat{t}} &= \left[\frac{1-p}{1-\hat{N}_M} + \frac{p(M-1)}{\hat{t}-M\hat{N}_M}\right]\hat{N}_{M-1}, & \hat{N}_M\left(\frac{1}{U}\right) &= 0.
\end{aligned} \tag{3.43}$$

Solution

This model consists of $M+1$ coupled nonlinear differential equations. This makes an analytical solution very difficult to obtain specially because all equations depend on $\hat{N}_M(t)$. A numerical method is more appropriate to find the solutions of the model, but we cannot use and explicit finite differences scheme because it is unsuitable for a problem like this. The denominators in equations (3.43) will become very small as $\hat{t} \to M$, so the values of the equations, specially for high k as we will later see, will have abrupt changes in very small timescales. Instead of the explicit Runge-Kutta method we used in the previous section, we will use the implicit Euler which is more adequate for a stiff problem like this [28].

Let $\mathbf{y}(t) = [N_0,\dots,N_M]^T$, and $f(t,\mathbf{y})$ be the right hand side of equations (3.43) so that:

$$\dot{\mathbf{y}} = f(t,\mathbf{y}), \tag{3.44}$$

where:

$$f(t,\mathbf{y}) = \begin{bmatrix} -\frac{(1-p)}{1-N_M}N_0 \\ \vdots \\ \left(\frac{(1-p)}{1-N_M} + \frac{p(k-1)}{t-MN_M}\right)N_{k-1} - \left(\frac{(1-p)}{1-N_M} + \frac{pk}{t-MN_M}\right)N_k \\ \vdots \\ \left(\frac{(1-p)}{1-N_M} + \frac{p(M-1)}{t-MN_M}\right)N_{M-1} \end{bmatrix}, \tag{3.45}$$

and initial conditions:

$$\mathbf{y_0} = \begin{bmatrix} 1 - \frac{1}{U} \\ \frac{1}{U} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The implicit Euler finite differences scheme is [28]:

$$\mathbf{y_{n+1}} - hf(t, \mathbf{y_{n+1}}) - \mathbf{y_n} = 0. \tag{3.46}$$

At every iteration we must solve a system of $M + 1$ equations. We will use Newton's method to find the value of $\mathbf{y_{n+1}}$, for it we will need the Jacobian of equation (3.46):

$$\mathbf{J}(t, \mathbf{y_{n+1}}) = \begin{bmatrix} 1 - h\frac{\partial f_0}{\partial N_0} & 1 - h\frac{\partial f_0}{\partial N_1} & \dots & 1 - h\frac{\partial f_0}{\partial N_M} \\ \vdots & \vdots & \vdots & \vdots \\ 1 - h\frac{\partial f_k}{\partial N_0} & 1 - h\frac{\partial f_k}{\partial N_1} & \dots & 1 - h\frac{\partial f_k}{\partial N_M} \\ \vdots & \vdots & \vdots & \vdots \\ 1 - h\frac{\partial f_M}{\partial N_0} & 1 - h\frac{\partial f_M}{\partial N_1} & \dots & 1 - h\frac{\partial f_M}{\partial N_M} \end{bmatrix}. \tag{3.47}$$

The Jacobian has a bidiagonal structure were the $(M + 1)th$ column can be nonzero. Figure 3.9 shows the structure of the Jacobian of equation (3.46). As we mentioned, the stiff problem yields solutions that change abruptly in short periods of time. One example is shown on the right of Figure 3.9, where $N_{90}(t)$ is plotted, this solution corresponds to a network where $U = 30$, $M = 100$ and $p = 0.3$. The higher the value

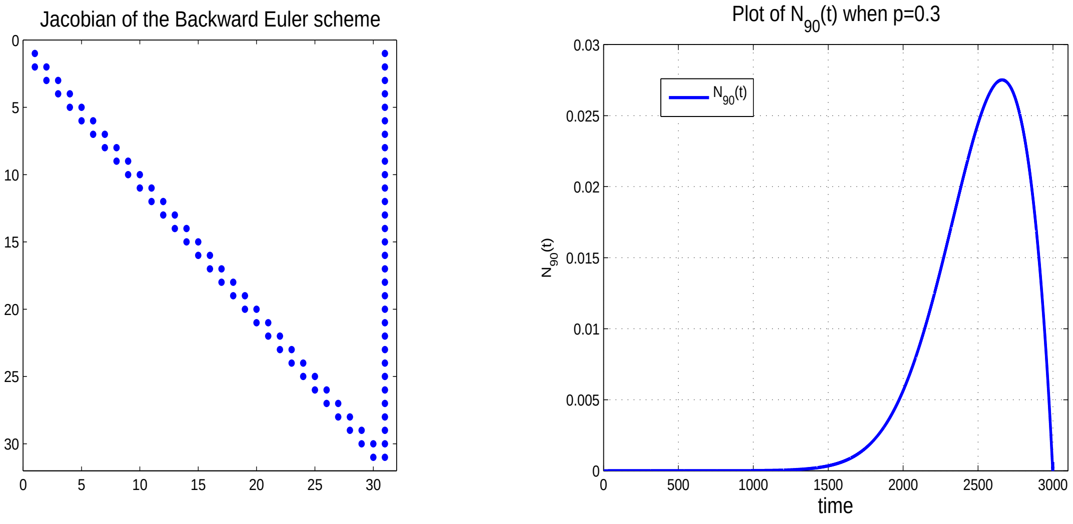

Figure 3.9: Structure of the Jacobian matrix (left) of the backward Euler scheme. Plot of $N_{90}(t)$ (right) computed numerically.

of k, the later $N_k(t) = 0$ reaches its maximum and the faster it must decrease to get to zero at the final time.

To show the solutions for all $N_k(t)$ we use the same colour plots where red indicates high values of $N_k(t)$ and dark blue indicates zero. Figure 3.10 shows solutions of equations (3.43). Each image has a different value for parameter p, the one on the left corresponds to $p = 0.2$, where uniform attachment is dominant over preferential attachment. We can see in this case that there are no nodes whose degree grows much faster than the rest. This is because the probability of preferential attachment is small enough to have most nodes getting edges at a similar rate. This contrasts with the image on the right which corresponds to $p = 0.8$, where preferential attachment is dominant. Here we can see how a few nodes get edges much faster than the rest (the lines that go down very quickly), and the majority of the nodes remain with low degrees for a longer time until finally (when the dominant nodes are fully connected) they also begin to receive edges and become ultimately connected at the final time. We can see how the plot of a simulation with $p = 0.8$ on the right of Figure 3.8 displays the same behaviour that we see in our solution. Dominant preferential attachment leads to a larger number of fully connected nodes earlier than dominant uniform attachment. Both images of Figure 3.10 show the row corresponding to $k = M$ that does not follow the same colouring patterns as the rest of the image. In Figure 3.11 we show a closer look at this row. The reason for this line is that the maximum degree M can be understood like an *attractor state* in which nodes upon

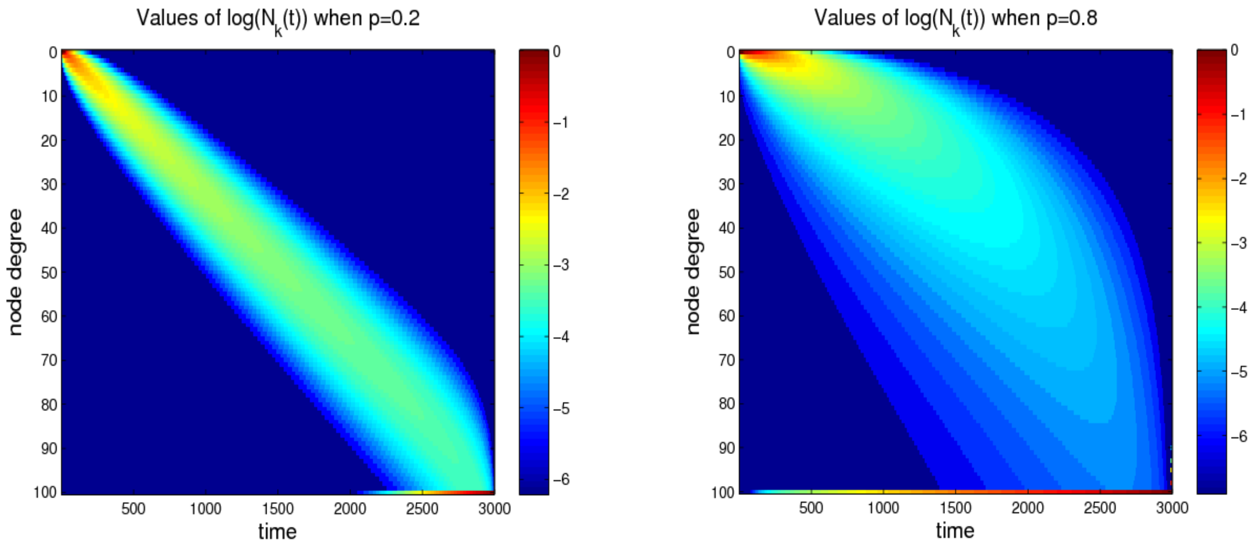

Figure 3.10: Solutions of $N_k(t)$ with $p = 0.2$ (left) and $p = 0.8$ (right) where $U = 30$ and $M = 100$.

arrival do not leave. We can see in the images how N_M attracts all the nodes, because as t approaches M all other the $N_k(t)$ must become empty really fast, which acts like a *de facto* implicit boundary condition.

The results given by this model are much more accurate than the ones given by the previous models. We saw it for $p = 0.8$, and we can see it in Figure 3.12that shows the results for $p = 0.6$ compared to the simulated networks, which also show good agreement. More images with solutions of different values of p are shown on Figures C.1-C.3 in Appendix C. A different comparison of how accurate the model is for different values of p is shown in Figure 3.13, where we see two plots of $N_{45}(t)$ for $p = 0.6$ and $p = 0.2$. The images on the left are a close-up of the images on the right. The model showed better accuracy for smaller values of p than for larger values, which can also be appreciated from Table 3.1, where we show the mean and maximum error of $N_k(t)$ calculated by the model for $k = 15, 45, 90$, against the result observed from 1000 simulations of networks.

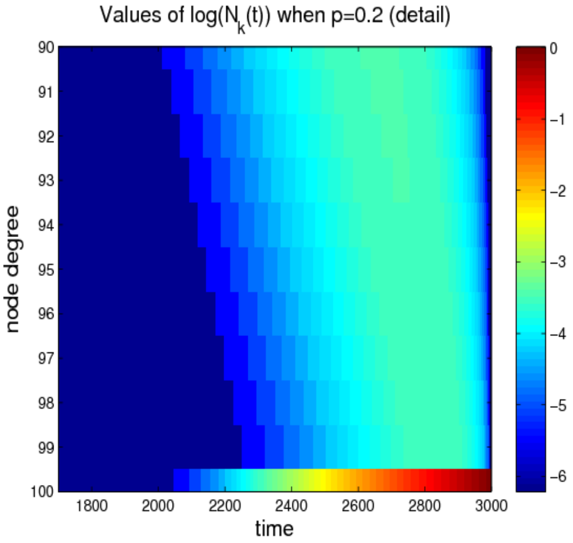

Figure 3.11: Detail of rows 90 to 100 and times 1800 to 3000 from the solution of $N_k(t)$ when $p = 0.2$ in Figure 3.10.

The model described in equations (3.42) captures the behaviour of the time-dependent degree-distribution $N_k(t)$ as seen from the images and plots. The update of the denominator in the master equations was what made the $N_k(t)$ go to zero as $t \to MU$ like it happens on the simulated networks. However, it made all equations $\frac{\mathrm{d}N_k}{\mathrm{d}t}$ depend not only on N_{k-1} and N_k, but also on N_M, which depends on N_{M-1} and so on. All equations depend on each other thus making it very difficult to obtain an analytical expression for the solutions. The numerical computations were also affected by the

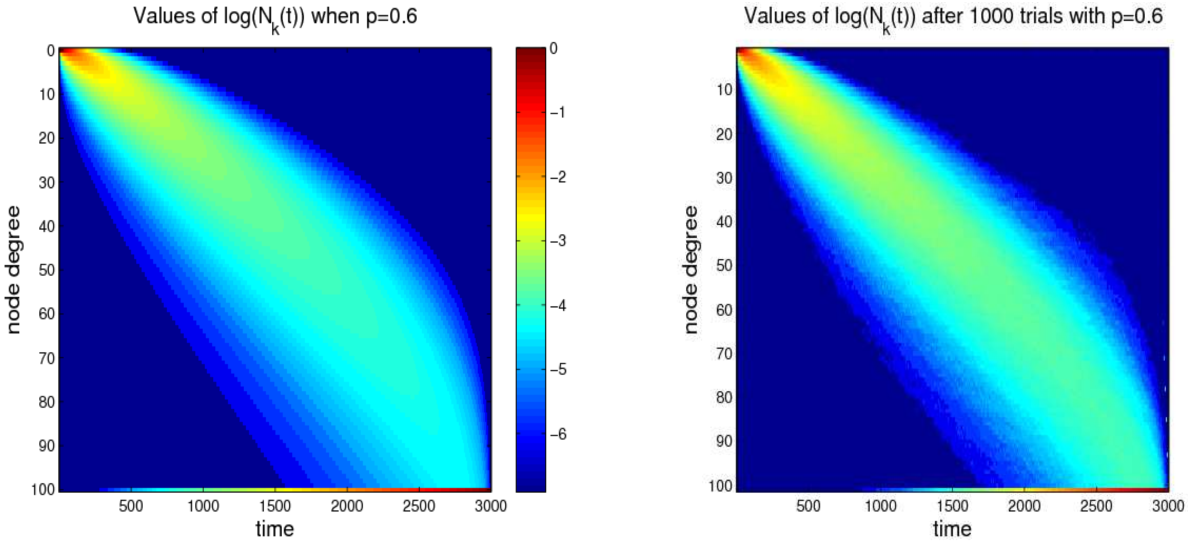

Figure 3.12: Comparison of results obtained by the model (left) and the mean values of $N_k(t)$ after 1000 simulations (right) when $M = 100$, $U = 30$ and $p = 0.6$.

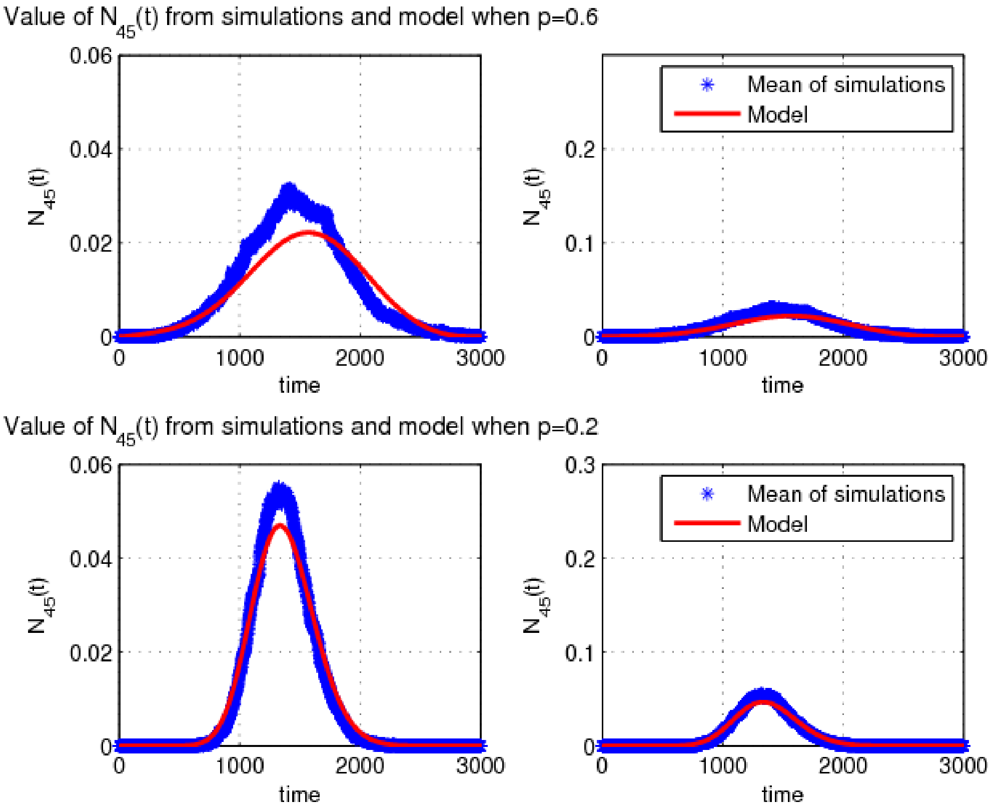

Figure 3.13: Comparisons of $N_{45}(t)$ from the model and the mean from 1000 simulated Networks when $p = 0.6$ (top) and $p = 0.2$ (bottom).

	$N_{15}(t)$		$N_{45}(t)$		$N_{90}(t)$	
p	max error	mean error	max error	mean error	max error	mean error
0.2	0.0014	0.0137	0.0013	0.0083	0.000983	0.0070
0.4	0.0026	0.0202	0.0025	0.0115	0.0015	0.0102
0.6	0.00035	0.0163	0.0025	0.017	0.0016	0.0217
0.8	0.0037	0.0124	0.0018	0.0064	0.0015	0.030

Table 3.1: Mean and maximum error of solutions to the model and realisations of the network for values of N_k and p.

update in the denominator because it made equations very stiff which makes solutions more computationally expensive to obtain, this is also true for large U and M.

3.2.4 Comparison to the Netflix dataset

In this section we compare the catalogue growth model with the Netflix dataset to see how well it reproduces its main features. Although we cannot visualise the complete process in the Netflix dataset as we did on the example networks of the previous sections (i.e. the data available to us is not from a fully connected network), we can see how it evolves until all the nodes have at least degree one. On the left of Figure 3.14 we show the values of the users' $N_k(t)$ from September 12, 2004. On this day there were 16,164 users, 7,546 movies and 82,460 ratings. The highest user degree was 593 and we can see that there are already some nodes that get edges at a faster

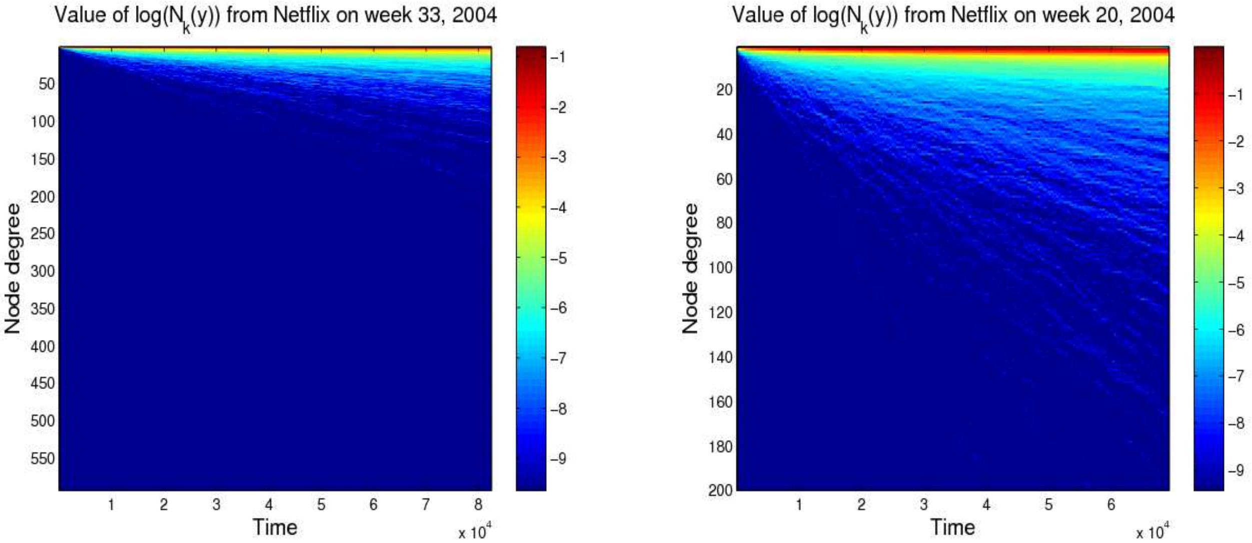

Figure 3.14: Values of $N_k(t)$ from the Netflix dataset on Sept 12, 2004 (left) and May 13, 2004 (right).

rate than the majority. This can be appreciated in greater detail on the image in the right of the Figure. Here we display $N_k(t)$ from May 13, 2004. On that day there were 12,996 users, 6,938 movies and 69,288 ratings, the highest degree of the users was 767. The image shows a closeup of the evolution of $N_k(t)$ from $k = 0$ to 200 and all values of t. On Figure 3.15 we show on the left image the values of $N_k(t)$ obtained

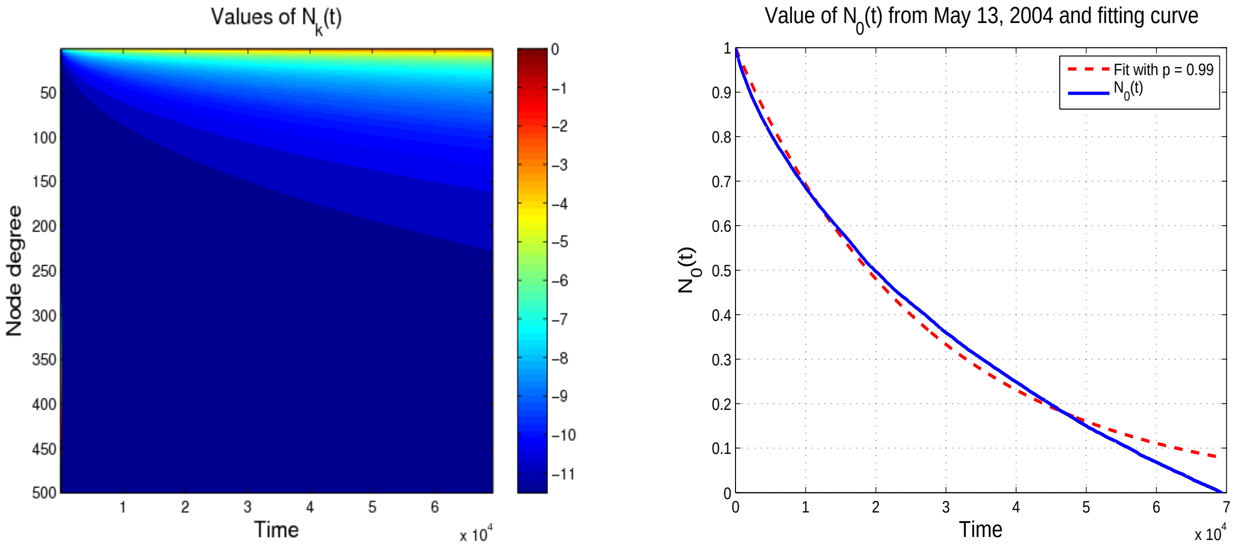

Figure 3.15: Values of $N_k(t)$ obtained using the constant node-fitness model (left) and a fit of $N_0(t)$ (right) with parameters $M = 1300$, $U = 500$ and $p = 0.99$

using the model of growth with constant node-fitness *without* catalogue update of equations (3.42) with parameters $M = 1300$, $U = 500$, $p = 0.99$ and t from 0 to 80,000. The reason why we used this model and not the one with catalogue update is because there are no fully-connected nodes up to $t = 80,000$ in the examples, so there is no need to update the denominator in the equations which means that in this regime both models are equivalent. The result obtained by the model does look like it could represent what goes on in the Netflix dataset and the fit for $N_0(t)$ on the image of the right confirms it, but the fit for the rest of the solutions is considerably less accurate. This suggests that the catalogue growth model does not explain completely the structure of the Netflix dataset.

Chapter 4

Conclusions

In this project we worked with a large dataset of ratings of movies entered into the Netflix database by its costumers in a period of over five years. We analysed the data to try to understand its main characteristics, and get an understanding of the mechanisms that produce such data structures. The ultimate goal was to make a mathematical model that describes the processes that shape the network.

The analysis in the Netflix dataset showed us that the degree distributions of the nodes followed power-law with exponential cutoff in the tails. This indicated that processes other than preferential attachment were also present in the development of the network. The cutoff in the tails hinted that nodes could get saturated with links and their attractiveness. Calculations on the number of ratings that some of the very popular films receive per day seem to confirm this findings. These movies, when they first become very popular receive a lot of links in short periods of time. After this breakthrough their income of ratings settle at a more or less stable level, which means that although the degree keeps growing, the fraction of new links that the movie gets, does not. This can be a combined effect of competition, another film taking over as dominant, and saturation, e.g. the public has seen the movie enough times.

We also showed through the clustering coefficient C_3 of the projected networks of users and movies, that there were strongly connected neighbourhoods of users who had rated the same movies, as well as in the projected network of films. It was also seen that individual nodes can have significant effects on the cliquishness of the projected networks. As we saw in the example of the projected network of users, one very prominent film can have very noticeable effects on the clustering coefficient of the complete projected network. We also calculated a clustering coefficient C_4 suitable for bipartite graphs that is based on the number of squares that include a node. The value of C_4 was found to be much lower than C_3 in the projected networks. Lower values of C_4 are a direct consequence of the diverse preferences of films that the users showed in the dataset in which two users, in many cases, had only one movie in common, which reduced the number of squares. In the Netflix dataset we observed power-law degree distributions with exponential tails, which were an indication that the attractiveness of the nodes, saturated. A first attempt to develop a model that could reproduce this saturation and yield a similar network structure, borrowed the concept of carrying capacity from population dynamics. The carrying capacity of an environment was interpreted in a network as the maximum attractiveness that a node could attain. The model we proposed had the attractiveness of the node grow logistically, that is, exponential growth in the beginning and zero growth as $k \to \infty$. Networks grown using this mechanism showed, depending on the initial condition which is a parameter of the model, exponential degree distributions just as networks grown using uniform attachment. In some cases when the initial value was very small, the CDF showed a few nodes with exceptionally large degrees in early stages of the growth of the network. As more nodes are added to the network, the distribution slowly approached an exponential. This attachment mechanism did not reflect any of the characteristics observed in the Netflix dataset.

A different type of model was proposed in which the nodes that were to be joined by an edge were chosen from predefined catalogues or lists of fixed size. The probability by which a node was chosen would depend on its degree. If allowed to evolve for a long time, this type of networks eventually became fully connected, i.e. all users have rated all movies. The first attempt of this type of models used shifted preferential attachment in the probabilities of the nodes. It was found that the value of the initial attractiveness parameter of the nodes was decreasing relative to the growth of the network. While it may be true for other type of networks [26], it was not the case in the Netflix dataset. The attractiveness of an un-rated film by this model, goes to zero as the network evolves. The master equations of this model were solved for all times and degree values, the resulting functions were expressed in terms of the Pochhammer's symbol and the hypergeometric function.

A second mechanism was proposed in which the fitness of the nodes was non-decreasing. To achieve this we used similar attachment probabilities as in the network rewiring model shown in Chapter 1 [9, 10]. With a fixed probability p we would choose the node using preferential attachment and with probability $(1-p)$, with uniform attachment. With this mechanism we assured that all nodes in the catalogues would have a non-decreasing probability to receive an edge. Analytical solutions, both exact and approximate, can be calculated for this model. However, their expressions were very long and complicated to obtain, so a numerical approach was favoured. As we mentioned, catalogue networks become fully connected in a finite time. This model did not account for fully-connected nodes that were no longer eligible for new edges, and as a result we got some serious inaccuracies for large values of t and k. When the size of the catalogues is very large and there is no need to compute $N_k(t)$ for high k, these models can be a useful tool. Compared to the Netflix data captured some of the behaviour of the dataset at early times. This model was improved so it could show the behaviour of $N_k(t)$ for high k and t. When a node becomes fully connected, the probabilities of attachment change for the rest of the nodes, because there is one less node to compete against. Incorporating this into the model makes the solutions reproduce what we observed in the simulations, and at the final time all N_k were correctly zero except N_M, which was 1. This improvement came at a cost, for all equations in the model were now coupled to the others, which makes an analytical solution very difficult to compute. The correction in the probabilities created a de-facto boundary condition which made the equations very stiff, requiring us to resort to more expensive numerical algorithms to find a solution. This method compared to the Netflix data did just as well as the model without catalogue update, because the difference between the two models is only evident as nodes start to be completely connected, which was not the case in the Netflix dataset.

In this model we have assumed fixed-sized catalogues, whereas in reality networks of this type may have variable-size catalogues. Further developments of this model could include a catalogue-size function $M(t)$ so that the model goes from having the form :

$$\frac{\mathrm{d}N_k}{\mathrm{d}t} = f(N_{k-1}, N_k, N_M, M, t),$$

to:

$$\frac{\mathrm{d}N_k}{\mathrm{d}t} = f(N_{k-1}, N_k, N_M, M, t),$$

$$\frac{\mathrm{d}M}{\mathrm{d}t} = g(M, t).$$

Where the size of the catalogues could be chosen according to the models studied.

In conclusion, our analysis of the Netflix dataset provided us with the motivation to develop network growth and evolution models in which characteristics of real networks such as saturation of edges and emergence of temporal dominant nodes would be explained. In doing so, the concept of a catalogue network was developed.

Catalogue networks, because of their characteristics, can benefit from different mathematical techniques such as differential equations, numerical analysis and asymptotics. We believe that the models we developed in this work to study both real and simulated networks, can be studied and further improved to be useful in a wide range of problems and applications.

Appendix A

The hypergeometric function

In this appendix we show some of the properties of the hypergeometric function that were used throughout this work. Most of the results shown here can be found in Abramowitz-Stegun [1], Forrey [11], and Olver [24].

The *hypergeometric equation* is the second order differential equation:

$$z(1-z)\frac{\mathrm{d}^2 w}{\mathrm{d}z^2} + \left[c - (a+b+1)z\right]\frac{\mathrm{d}w}{\mathrm{d}z} - abw = 0, \tag{A.1}$$

where $a, n, c, z \in \mathbb{C}$, and has regular singular points at $0, 1, \infty$. This equation is very important as any second order homogeneous ODE whose singularities are regular and no more than three can be transformed into it [24]. The solution to equation (A.1) is the ${}_2F_1$ *hypergeometric function*:

$${}_2F_1(a,b;c;z) = \sum_{n=0}^{\infty} \frac{a(a+1)\dots(a+n-1)b(b+1)\dots(b+n-1)}{c(c+1)\dots(c+n-1)}\frac{z^n}{n!}, \tag{A.2}$$

$$= \sum_{n=0}^{\infty} \frac{(a)_n(b)_n}{(c)_n}\frac{z^n}{n!}, \tag{A.3}$$

$$= \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)}\sum_{n=0}^{\infty} \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)}\frac{z^n}{n!}. \tag{A.4}$$

The function has the notation ${}_2F_1$, because it has two parameters in the numerator and one in the denominator. The symbol $(a)_n$ in equation (A.3), denotes Pochhammer's symbol:

$$(x)_n = \prod_{i=0}^{n-1}(x+i), \tag{A.5}$$

$$= \frac{\Gamma(x+n)}{\Gamma(x)}, \tag{A.6}$$

and $\Gamma(x)$ is the gamma function:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t. \tag{A.7}$$

From equation A.2 it is clear that ${}_2F_1(a,b;c;z) = {}_2F_1(b,a;c;z)$. The ${}_2F_1$ hypergeometric function has some special cases depending on the value of its parameters and argument. On case specially useful for this work is [1]:

$${}_2F_1(a,b;b;z) = (1-z)^{-a}. \tag{A.8}$$

The functions ${}_2F_1(a \pm 1, b; c; z)$, ${}_2F_1(a, b \pm 1; c; z)$ and ${}_2F_1(a, b; c \pm 1; z)$ are called the *contiguous* functions of ${}_2F_1(a,b;c;z)$, and are all related to it through Gauss' relations for contiguous functions [24]. Some of them are:

$$\begin{aligned} &c(c-1)(z-1){}_2F_1(a,b-1;c;z) \\ &\qquad + (2b-b-bz+az){}_2F_1(a,b;c;z) \\ &\qquad\qquad + b(z-1){}_2F_1(a,b+1;c;z) = 0, \end{aligned} \tag{A.9}$$

$$\begin{aligned} &c(c-1)(z-1){}_2F_1(a,b;c-1;z) \\ &\qquad + c[c-1-(2c-a-b-1)z]{}_2F_1(a,b;c;z) \\ &\qquad\qquad + (c-a)(c-b)z{}_2F_1(a,b;c+1;z) = 0, \end{aligned} \tag{A.10}$$

$$\begin{aligned} &[c-2a-(b-a)z]{}_2F_1(a,b;c;z) \\ &\qquad + a(1-z){}_2F_1(a+1,b;c;z) \\ &\qquad\qquad - (c-a){}_2F_1(a,b+1;c;z) = 0. \end{aligned} \tag{A.11}$$

These properties can be used to show some other properties:

$${}_2F_1(b,a+b;b+1;z) = U \sum_{n=0}^{\infty} \frac{(b+a)_n}{b+n} \frac{z^n}{n!}, \tag{A.12}$$

$${}_2F_1(a+1,b;c;z) = {}_2F_1(a,b;c;z) + \frac{\Gamma(c)}{a\Gamma(a)\Gamma(b)} z \sum_{n=1}^{\infty} \left[\frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)(n-1)!} z^{n-1} \right]. \tag{A.13}$$

$$ {}_2F_1(b, b+a; b+1; z) = b \sum_{n=0}^{\infty} \left[\frac{(b+a)_n}{b+n} \frac{z^n}{n!} \right] \tag{A.14} $$

$$ = b \sum_{n=0}^{\infty} \binom{b+a+n-1}{n} \frac{z^n}{b+n} \tag{A.15} $$

$$ {}_2F_1(a, b; a-1; z) = \frac{[a-1-(a-b-1)z]}{(a-z)^{b-1}}. \tag{A.16} $$

The general form of the hypergeometric function is:

$$ {}_pF_q(a_1, a_2, \ldots, a_p; b_1, \ldots b_q; z) = \sum_{n=0}^{\infty} \frac{(a_1)_n (a_2)_n \ldots (a_p)_n}{(b_1)_n \ldots (b_q)_n} \frac{z^n}{n!}. \tag{A.17} $$

The regularised ${}_p\tilde{F}_q$ hypergeometric function is:

$$ {}_p\tilde{F}_q(a_1, a_2, \ldots, a_p; b_1, \ldots b_q; z) = \frac{{}_pF_q(a_1, a_2, \ldots, a_p; b_1, \ldots b_q; z)}{\Gamma(b_1)\Gamma(b_2)\ldots\Gamma(b_q)}. \tag{A.18} $$

Appendix B

Power-law distributions

In this appendix we give a very brief description of power-law degree distributions. The content discussed here can be found in Clauset *et al* [7] and Newman [23].

When the probability of the measurement of a random event is inversely proportional to its value, we say that it follows a power-law distribution [23]. One of the first examples of a power law was given by linguist George Zipf [19], who observed that the frequency of any word from a large enough collection of text (*corpus*) was inversely proportional to its rank in a table of ordered frequencies of all words.

Power-laws tend to arise in data in which measurements do not fluctuate around a mean value, for example, the population of cities, the intensity of earthquakes and the distances travelled by humans in a day, seem to follow power-law behaviour [7, 12, 23].

The basic form of the degree distribution of a power-law is:

$$f(x) = Cx^{-\gamma}, \tag{B.1}$$

where $C \in \mathbb{R}$, $C > 0$. We can see from the form equation (B.1) one key property that will help us detect a power law. If we take the logarithm of both sides of the equation, we get:

$$\log f(x) = -\gamma \log x + \log C. \tag{B.2}$$

This indicates that in double logarithmic scales equation (B.1) must be a straight line with slope $-\gamma$. On the left of Figure B.1 we can see the histogram of measurements of a power-law distributed random variable with $\gamma = 3$, on the right we see the same measurements in a log-log scale. A straight line in a log-log scale does not assure a power law, it doesn't dismiss it and that is as far as it goes. There are other probability distributions that may look like a straight line on a log-log plot such as the Log-normal and exponential distributions. One way to determine if some data collection follows a power-law is by using maximum-likelihood estimators and the

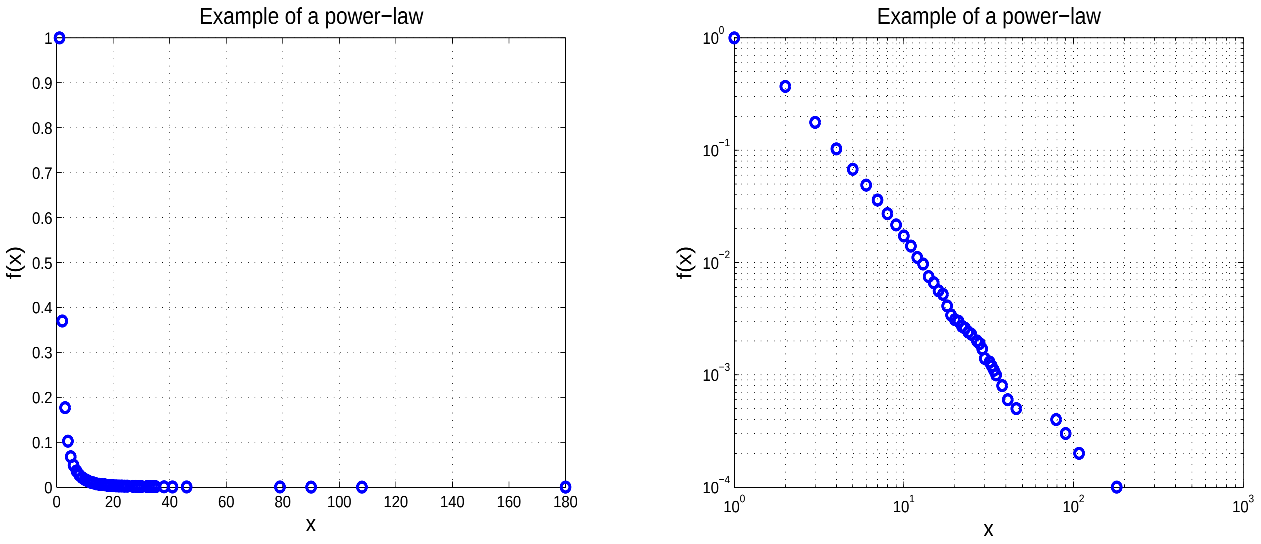

Figure B.1: Histogram of a power-law distributed variable with $\gamma = 3$ on normal scale (left) and log-log scale (right).

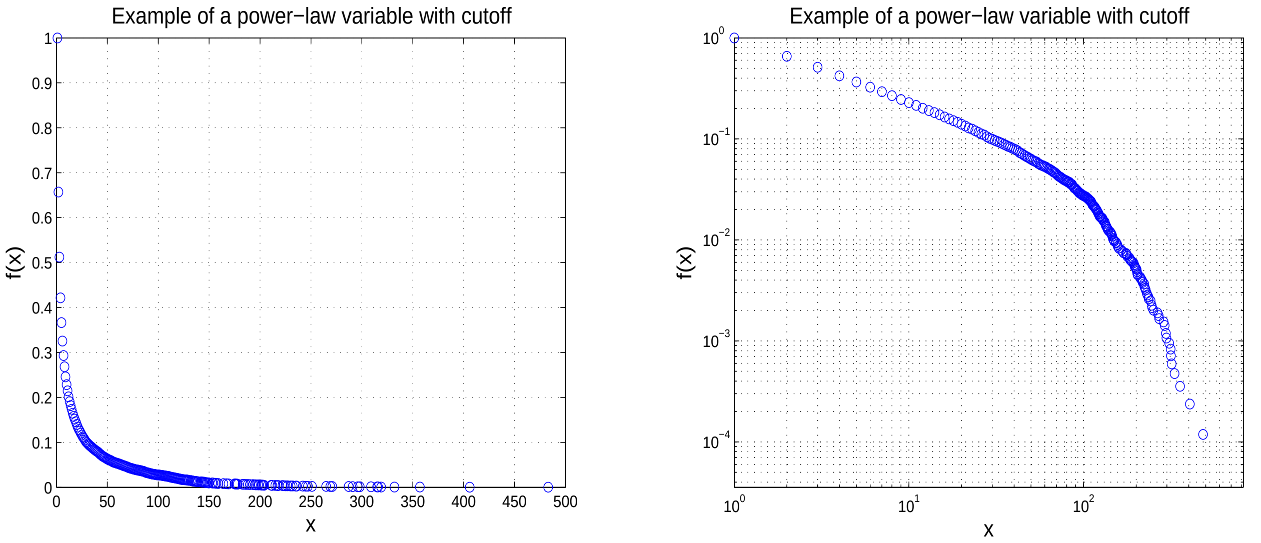

Figure B.2: Histogram of a power-law with cutoff distributed variable with $\gamma = 0.6$ and $\lambda = 0.0088$ on normal scale (left) and log-log scale (right).

Kolmogorov-Smirnoff statistic, described in [7].

There are other degree distributions that are often mistaken for a power-law, but are not quite so. A common example is the power-law with cutoff, which follows a distribution $g(x)$:

$$g(x) = Cx^{-\gamma}e^{-\lambda x}. \tag{B.3}$$

This type of distribution behaves like a power-law for early values of x because of the dominance of $x^{-\gamma}$, but as x gets larger, $e^{-\lambda x}$ dominates and brings down the tail, hence the cutoff. On Figure B.2 we see an example of a power-law with cutoff distributed variable with $\gamma = 0.6$ and $\lambda = 0.0088$. The image on the right was created using natural scales, the one on the left uses double-logarithmic scales.

Appendix C

Catalogue growth calculations

C.1 Decreasing node-fitness

C.1.1 Asymptotics

Equations (3.16) show the expansion of $N_k(t)$ in powers of a for $k = 0, \ldots, U$. Here we show with more detail the calculations we used to solve the differential equations:.

- $N_0(t)$:

$$\begin{aligned}
&\frac{\mathrm{d}(N_{0_0}(t) + aN_{0_1}(t) + \ldots)}{\mathrm{d}t} = -\left[\frac{a}{1+t}\right](N_{0_0}(t) + aN_{0_1}(t) + \ldots), \\
&N_{0_0}(0) + aN_{0_1}(0) + \ldots = 1.
\end{aligned}$$

  Compare terms with the same power of a to get

$$\begin{aligned}
&\frac{\mathrm{d}N_{0_0}}{\mathrm{d}t} = 0 && N_{0_0}(0) = 1 \Rightarrow && N_{0_0}(t) = 1, \\
&\frac{\mathrm{d}N_{0_1}}{\mathrm{d}t} = \frac{-N_{0_0}}{1+t} && N_{0_1}(0) = 0 \Rightarrow && N_{0_1}(t) = -\log(1+t), \\
&\frac{\mathrm{d}N_{0_2}}{\mathrm{d}t} = \frac{-N_{0_1}}{1+t} && N_{0_2}(0) = 0 \Rightarrow && N_{0_2}(t) = \frac{\log(1+t)^2}{2}, \\
&\quad\vdots \\
&\frac{\mathrm{d}N_{0_r}}{\mathrm{d}t} = \frac{-N_{0_r}}{1+t} && N_{0_r}(0) = 0 \Rightarrow && N_{0_r}(t) = \frac{(-1)^r \log(1+t)^r}{r!}.
\end{aligned}$$

- $N_k(t)$:

$$\begin{aligned}\frac{\mathrm{d}(N_{k_0}+aN_{k_1}+a^2N_{k_2}+\ldots)}{\mathrm{d}t} =&\frac{a+k+1}{1+t}(N_{(k-1)_0}+aN_{(k-1)_1}+\ldots)\\ &-\frac{a+k}{1+t}(N_{k_0}+aN_{k_1}+\ldots),\\ N_{k_0}(0)+aN_{k_1}(0)+\ldots=0.&\end{aligned}$$

Comparing terms with the same power of a we get

$$\frac{\mathrm{d}N_{k_0}}{\mathrm{d}t}=\frac{k-1}{1+t}N_{(k-1)_0}-\frac{k}{1+t}N_{k_0} \qquad N_{k_0}(0)=0\Rightarrow \qquad N_{k_0}(t)=0,$$

$$\frac{\mathrm{d}N_{k_1}}{\mathrm{d}t}=\frac{t^{(}k-1)}{(1+t)^k}-\frac{k}{(1+t)}N_{k_1} \qquad N_{k_1}(0)=0\Rightarrow \quad N_{k_1}(t)=\frac{t^k}{k(1+t)^k}.$$

$$\vdots$$

For greater powers of a, analytic expressions are possible to obtain in a mechanical but rather tedious way. As an example, here are some examples of the results that can be obtained

$$\begin{aligned}N_1(t)=&\,a\frac{t}{1+t}+a^2\frac{\log(1+t)-t\log(t)}{1+t}\\ &+a^3\left((\log(1+t)-2)\log(1+t)+2-\frac{\log(1+t)}{2(1+t)}+\frac{t(\log(t)-1)}{1+t}\right.\\ &\left.-\frac{\log(t)\log(1+t)}{1+t}-\frac{\mathrm{Li}_2(-t)}{1+t}\right)+\ldots.\end{aligned}$$

$$\begin{aligned}N_2(t)=&\,a\frac{t^2}{2(1+t)^2}+a^2\left(\frac{\log(1+t)}{1+t}-\frac{t}{1+t}-\frac{1}{4}t^2(2\log(t)-1)\right.\\ &\left.-\frac{(t-2)t}{2(1+t)^2}-\frac{\log(1+t)}{(1+t)^2}\right)+\ldots\end{aligned}$$

$$N_3(t)=a\frac{t^3}{3(1+t)^3}+\ldots,$$

Where $\mathrm{Li}_2(t)$ is the polylogarithm function

$$\mathrm{Li}_n(z)=\sum_{k=1}^{\infty}\frac{z^k}{k^n}. \tag{C.1}$$

- $N_M(t)$:

$$\begin{aligned}\frac{\mathrm{d}(N_{M_0}+aN_{M_1}+\ldots)}{\mathrm{d}t}&=\frac{a+M-1}{1+t}(N_{(M-1)_0}+aN_{(M-1)_1}+\ldots),\\ N_{M_0}(0)+aN_{M_1}(0)+\ldots=0.&\end{aligned}$$

As before, compare terms with the same power of a and solve

$$\begin{array}{lll}
\frac{\mathrm{d}N_{M_0}}{\mathrm{d}t} = 0 & N_{M_0}(0) = 0 \Rightarrow & N_{M_0}(t) = 0 \\[2em]
\frac{\mathrm{d}N_{M_1}}{\mathrm{d}t} = \frac{t^{M-1}}{(1+t)^M} & N_{M_1}(0) = 0 \Rightarrow & N_{M_1}(t) = \Gamma(M) t^M {}_2\tilde{F}_1(M, M; M+1; -t) \\
\quad\vdots & &
\end{array}$$

Where ${}_2\tilde{F}_1(a, b; c; z)$ is the regularized hypergeometric function

$$ {}_2\tilde{F}_1(a, b; c; z) = \frac{{}_2F_1(a, b; c; z)}{\Gamma(c)}. \tag{C.2}$$

And ${}_2F_1$ is the hypergeometric function defined in equation (A.2).

C.2 Constant node-fitness

C.2.1 Asymptotics

When $0 < p \ll 1$, we expand $N_k(t)$ as shown in equations (3.31). The solutions are calculated as follows:

- $N_0(t)$:

$$\begin{aligned}
\frac{\mathrm{d}(N_{0_0} + pN_{0_1} + \dots)}{\mathrm{d}t} &= -(1+p)(N_{0_0} + pN_{0_1} + \dots), \\
N_{0_0}\left(\frac{1}{U}\right) + pN_{0_1}\left(\frac{1}{U}\right) + \dots &= 1 - \frac{1}{U}.
\end{aligned}$$

Compare same powers of p and solve

$$\begin{array}{lll}
\frac{\mathrm{d}N_{0_0}}{\mathrm{d}t} = -N_{0_0}, & N_{0_0}\left(\frac{1}{U}\right) = 1 - \frac{1}{U} & \Rightarrow N_{0_0}(t) = \left(1 - \frac{1}{U}\right) e^{-\left(t - \frac{1}{U}\right)}, \\[2em]
\frac{\mathrm{d}N_{0_1}}{\mathrm{d}t} = -N_{0_1} + N_{0_0}, & N_{0_1}\left(\frac{1}{U}\right) = 0 & \Rightarrow N_{0_1}(t) = \left(1 - \frac{1}{U}\right)\left(t - \frac{1}{U}\right) e^{-\left(t - \frac{1}{U}\right)}, \\[2em]
\frac{\mathrm{d}N_{0_2}}{\mathrm{d}t} = -N_{0_2} + N_{0_2}, & N_{0_2}\left(\frac{1}{U}\right) = 0 & \Rightarrow N_{0_2}(t) = \left(1 - \frac{1}{U}\right) \frac{\left(t - \frac{1}{U}\right)^2}{2} e^{-\left(t - \frac{1}{U}\right)}, \\
\quad\vdots & & \\
\frac{\mathrm{d}N_{0_k}}{\mathrm{d}t} = -N_{0_k} + N_{0_{(k-1)}}, & N_{0_k}\left(\frac{1}{U}\right) = 0 & \Rightarrow N_{0_k}(t) = \left(1 - \frac{1}{U}\right) \frac{\left(t - \frac{1}{U}\right)^k}{k!} e^{-\left(t - \frac{1}{U}\right)},
\end{array}$$

- $N_1(t)$:

$$\frac{\mathrm{d}(N_{1_0} + pN_{1_1} + \dots)}{\mathrm{d}t} = (1-p)[N_{0_0} + pN_{0_1} + \dots] - \left[(1-p) + \frac{p}{t}\right][N_{1_0} + pN_{1_1} + \dots],$$
$$N_{1_0}\left(\frac{1}{U}\right) + pN_{1_1}\left(\frac{1}{U}\right) + \dots = \frac{1}{U}.$$

Compare powers of p and solve

$$\frac{\mathrm{d}N_{1_0}}{\mathrm{d}t} = N_{0_0} - N_{1_0} \qquad N_{1_0}\left(\frac{1}{U}\right) = \frac{1}{U} \quad \Rightarrow N_{1_0}(t) = \left(1 - \frac{1}{U}\right) te^{-\left(t-\frac{1}{U}\right)} + \frac{e^{-\left(t-\frac{1}{U}\right)}}{U^2}.$$

$$\frac{\mathrm{d}N_{1_1}}{\mathrm{d}t} = N_{0_1} - N_{0_0} - \left[N_{1_1} - \left(1 - \frac{1}{t}\right)N_{1_0}\right], \quad N_{1_1}\left(\frac{1}{U}\right) = 0 \quad \Rightarrow N_{1_1}(t) = \left[\left(1 - \frac{1}{U}\right)t^2 + \left(\frac{2}{U^2} + \frac{1}{U} - 2\right)t + \frac{\log t}{U^2} + \frac{2}{U} - \frac{1}{U^3} + \frac{\log U}{U^2}\right].$$

- $N_2(t)$:

$$\frac{\mathrm{d}(N_{2_0} + pN_{2_1} + \dots)}{\mathrm{d}t} = \left[(1-p)\frac{p}{t}\right][N_{1_0} + pN_{1_1} + \dots] - \left[(1-p) + \frac{2p}{t}\right][N_{2_0} + \dots],$$
$$N_{2_0}\left(\frac{1}{U}\right) + pN_{2_1}\left(\frac{1}{U}\right) + \dots = 0.$$

Compare the powers of p and solve:

$$\frac{\mathrm{d}N_{2_0}}{\mathrm{d}t} = N_{1_0} - N_{2_0}, \quad N_{2_0}\left(\frac{1}{U}\right) = 0 \quad \Rightarrow N_{2_0}(t) = \left[\left(1 - \frac{1}{U}\right)\frac{t^2}{2} + \frac{t}{U^2} - \left(1 + \frac{1}{U}\right)\frac{1}{2U^2}\right]e^{-\left(t-\frac{1}{U}\right)}.$$

$$\vdots$$

When $0 < (1-p) \ll 1$, an asymptotic expansion is still possible. However, we must change variables $w = (1-p)$ and expand in powers of w, $N_k(t) = N_{k_0}(t) + wN_{k_1}(t) +$

$w^2 N_{k_2}(t) + \dots$. The resulting equations are

$$\frac{\mathrm{d}(N_{0_0} + wN_{0_1} + \dots)}{\mathrm{d}t} = -w(N_{0_0} + pN_{0_1} + \dots),$$
$$N_0\left(\frac{1}{U}\right) = 1 - \frac{1}{U}. \tag{C.3}$$

$$\begin{aligned}\frac{\mathrm{d}(N_{k_0} + wN_{k_1} + \dots)}{\mathrm{d}t} &= \left[w + \frac{(1-w)(k-1)}{t}\right](N_{(k-1)_0} + wN_{(k-1)_1} + \dots) \\ &\quad - \left[w + \frac{(1-w)k}{t}\right](N_{k_0} + wN_{k_1} + \dots), \end{aligned}\tag{C.4}$$
$$N_1\left(\frac{1}{U}\right) = \frac{1}{U}, \quad N_k\left(\frac{1}{U}\right) = 0, \quad k > 1, \quad k = 1, \dots, M-1.$$

$$\frac{\mathrm{d}(N_{M_0} + wN_{M_1} + \dots)}{\mathrm{d}\hat{t}} = \left[w + \frac{(1-w)(M-1)}{t}\right](N_{(M-1)_0} + wN_{(M-1)_1} + \dots), \tag{C.5}$$
$$N_M\left(\frac{1}{U}\right) = 0.$$

The solutions are:

- $N_0(t)$:

$$\begin{aligned}&\frac{\mathrm{d}N_{0_0}}{\mathrm{d}t} = 0 && N_{0_0}\left(\frac{1}{U}\right) = 1 - \frac{1}{U} && \Rightarrow N_{0_0}(t) = \left(1 - \frac{1}{U}\right)\\
&\frac{\mathrm{d}N_{0_1}}{\mathrm{d}t} = -N_{0_0} && N_{0_1}\left(\frac{1}{U}\right) = 0 && \Rightarrow N_{0_1}(t) = -\left(1 - \frac{1}{U}\right)\left(t - \frac{1}{U}\right),\\
&\frac{\mathrm{d}N_{0_2}}{\mathrm{d}t} = -N_{0_1} && N_{0_2}\left(\frac{1}{U}\right) = 0 && \Rightarrow N_{0_2}(t) = \frac{\left(1 - \frac{1}{U}\right)\left(t - \frac{1}{U}\right)^2}{2},\\
&\vdots\\
&\frac{\mathrm{d}N_{0_k}}{\mathrm{d}t} = -N_{0_{(k-1)}} && N_{0_k}\left(\frac{1}{U}\right) = 0 && \Rightarrow N_{0_k}(t) = \frac{(-1)^k\left(1 - \frac{1}{U}\right)\left(t - \frac{1}{U}\right)^k}{k!}.\end{aligned}$$

- $N_1(t)$:

$$\begin{aligned}&\frac{\mathrm{d}N_{1_0}}{\mathrm{d}t} = -\frac{N_{1_0}}{t} && N_{1_0}\left(\frac{1}{U}\right) = \frac{1}{U} \Rightarrow \quad N_{1_0}(t) = \frac{1}{U^2 t},\\
&\frac{\mathrm{d}N_{1_1}}{\mathrm{d}t} = N_{0_0} - \left[\frac{N_{11}}{t} + \left(1 - \frac{1}{t}\right)N_{1_0}\right] && N_{1_1}\left(\frac{1}{U}\right) = 0\end{aligned}$$

$$\Rightarrow N_{1_1}(t) = \left(1 - \frac{1}{U}\right)\frac{t}{2} - \frac{1}{U^2} + \frac{1}{U^2 t}\left[\log t - \frac{1}{2} + \frac{3}{2U} + \log U\right].$$

- $N_2(t)$:

$$\frac{\mathrm{d}N_{2_0}}{\mathrm{d}t} = N_{1_0} - \frac{2N_{2_0}}{t} \qquad N_{2_0}\left(\frac{1}{U}\right) = 0 \Rightarrow \qquad N_{2_0}(t) = \frac{-1}{2U^4t^2} + \frac{1}{2U^2}.$$

- $N_3(t)$:

$$\frac{\mathrm{d}N_{3_0}}{\mathrm{d}t} = N_{2_0} - \frac{3N_{3_0}}{t} \quad N_{3_0}\left(\frac{1}{U}\right) = 0 \Rightarrow \quad N_{3_0}(t) = \frac{-1}{4U^2t} + \frac{t}{8U^2} + \frac{2U^3-1}{8U^6t^3}.$$

$$\vdots$$

C.3 Constant node-fitness with catalogue update

C.3.1 Solutions

Some comparisons of simulations of networks with the solutions of the model are shown in Figures C.1, C.2, 3.12 and C.3. The results are compared to 1000 simulated networks with $M = 100$, $U = 30$ and different values of p.

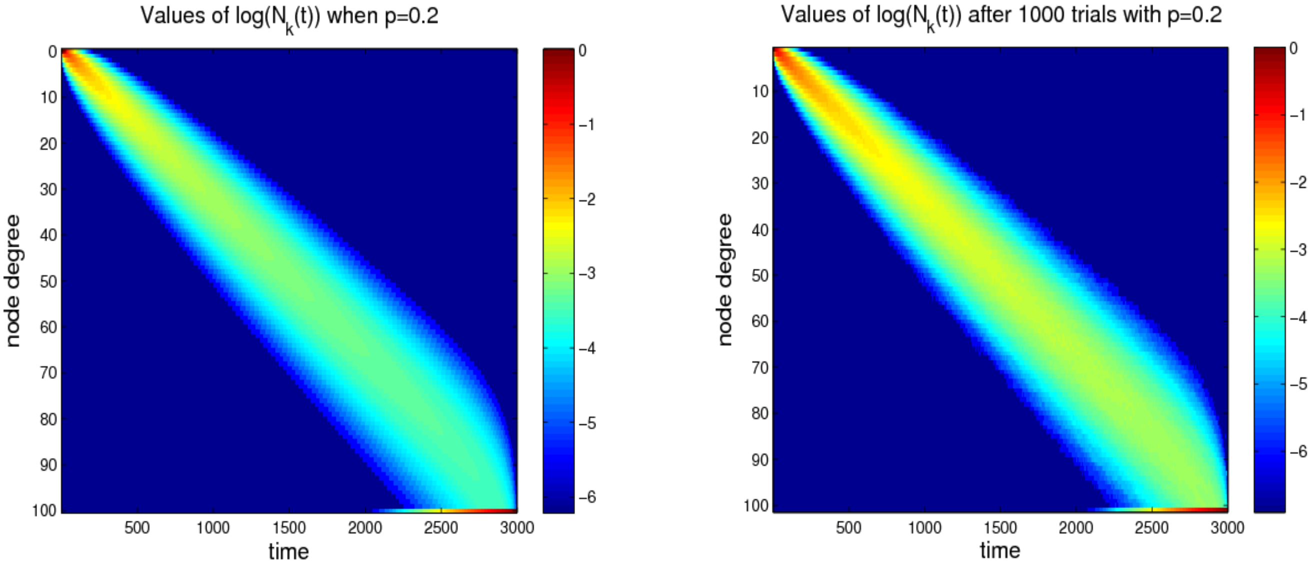

Figure C.1: Solutions of the model and simulations when $p = 0.2$.

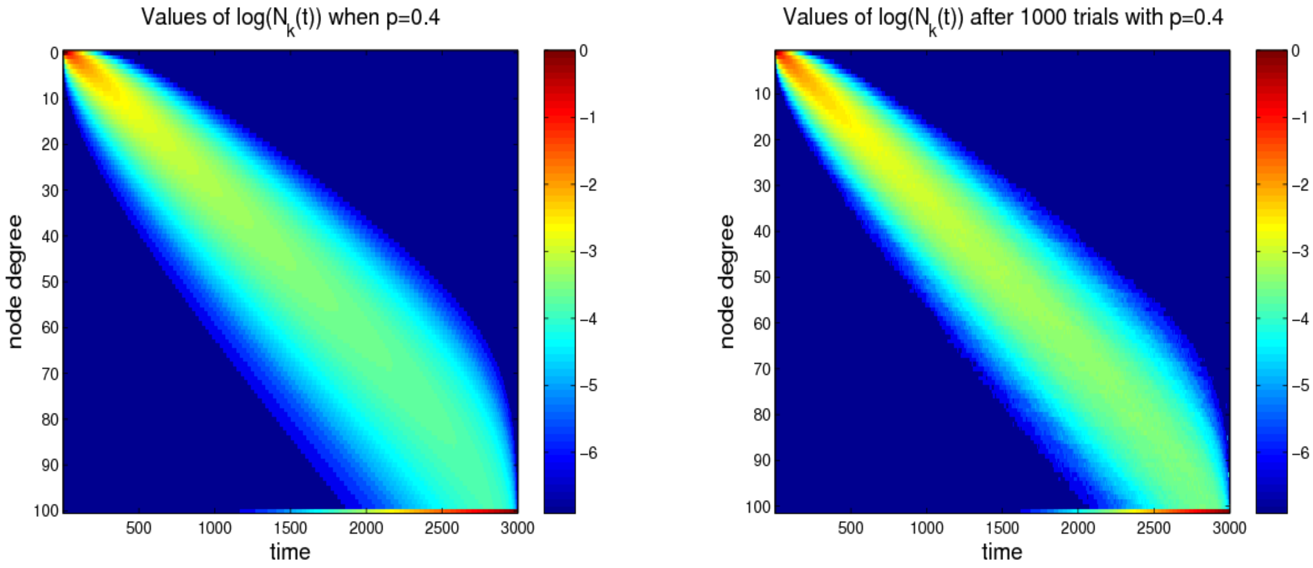

Figure C.2: Solutions of the model and simulations when $p = 0.4$.

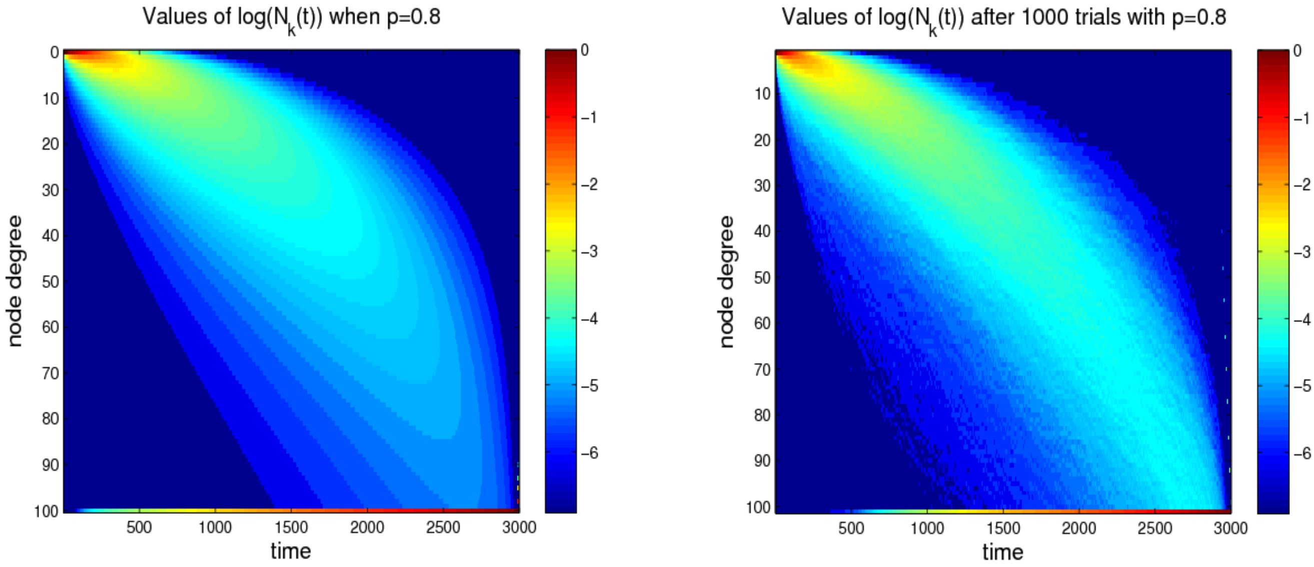

Figure C.3: Solutions of the model and simulations when $p = 0.8$.

Bibliography

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover, New York, 9 ed., 1964.

[2] R. Albert and A.-L. Barabási, *Statistical Mechanics of Complex Networks*, Reviews of Modern Physics, 74 (2002), p. 47.

[3] A.-L. Barabási and R. Albert, *Emergence of Scaling in Random Networks*, Science, 286 (1999), p. 509.

[4] A.-L. Barabási, R. Albert, and H. Jeong, *Mean-Field Theory for Scale-free Random Networks*, Physica A, 272 (1999), pp. 173–187.

[5] H. Bauke, *Parameter Estimation for Power-law Distributions by Maximum Likelihood Methods*, The European Physical Journal B, 58 (2007), pp. 167–173.

[6] J. Bennett and S. Lanning, *The Netflix Prize*, Proceedings of KDD Cup and Workshop 2007, (2007).

[7] A. Clauset, C. R. Shalizi, and M. E. J. Newman, *Power-law Distributions in Empirical Data*, arXiv:0706.1062v1, (2007).

[8] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas, *Characterization of Complex Networks: A Survey of Measurements*, Advances In Physics, 56 (2007), p. 167.

[9] T. S. Evans and A. D. K. Plato, *Exact Solution for the Time Evolution of Network Rewiring Models*, Physical Review E, 75-5 (2007).

[10] T. S. Evans and A. D. K. Plato, *Network Rewiring Models*, Networks and Heterogeneous Media, 3 (2008), p. 221.

[11] R. C. Forrey, *Computing the Hypergeometric Function*, J. Comput. Phys., 137 (1997), pp. 79–100.

[12] M. C. González, C. A. Hidalgo, and A.-L. Barabási, *Understanding Individual Human Mobility Patterns*, Nature, 453 (2008), p. 779.

[13] J. L. Gross and J. Yellen, eds., *Handbook of Graph Theory*, CRC Press, 2004.

[14] M. Kot, *Elements of Mathematical Ecology*, Cambridge University Press, 2003.

[15] P. L. Krapivsky and S. Redner, *Organization of Growing Random Networks*, Phys. Rev. E, 63 (2001).

[16] P. L. Krapivsky, S. Redner, and F. Leyvraz, *Connectivity of Growing Random Networks*, Phys. Rev. Lett., 85 (2000), pp. 4629–4632.

[17] P. G. Lind, M. C. González, and H. J. Herrmann, *Cycles and Clustering in Bipartite Networks*, Physical Review E, 72 (2005).

[18] J. Lorenz, *Universality of Movie Rating Distributions*, arXiv:0806.2305v1, physics.soc-ph (2008).

[19] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts, 1999.

[20] J. Murray, *Mathematical Biology: I. An Introduction*, Springer, 2005.

[21] M. E. J. Newman, *Clustering and Preferential Attachment in Growing Networks*, Phys. Rev. E, 64 (2001).

[22] M. E. J. Newman, *The Structure and Function of Complex Networks*, SIAM Review, 45 (2003), pp. 167–256.

[23] M. E. J. Newman, *Power laws, Pareto distributions and Zipf's law*, Contemporary Physics, 46 (2005), p. 323.

[24] F. W. J. Olver, *Asymptotics and Special Functions*, Academic Press, 1974.

[25] J. Paulo R. Guimaraes, M. A. M. de Aguiar, J. Bascompte, P. Jordano, and S. F. dos Reis, *Random Initial Condition in Small Barabasi-Albert Networks and Deviations from the Scale-free Behavior*, Physical Review E, 71 (2005).

[26] D. J. d. S. Price, *Networks of Scientific Papers*, Science, 149 (1965), pp. 510–515.

[27] E. Süli, *Numerical Solution of Differential Equations.*, Lecture Notes, Computing Laboratory, University of Oxford, October 2006. http://web.comlab.ox.ac.uk/oucl/work/endre.suli/nsodes.pdf.

[28] E. Süli and D. Mayers, *Introduction to Numerical Analysis*, Cambridge University Press, 2003.

[29] B. Waclaw and I. M. Sokolov, *Finite-size Effects in Barabási-Albert Growing Networks*, Physical Review E, 75 (2007).

[30] D. J. Watts and S. H. Strogatz, *Collective Dynamics of 'Small-World' Networks.*, Nature, 393 (1998), pp. 440–442.