Edge-exchangeable Random Graph Models

Candidate Number: 1023122

Supervisor: Prof. Francois Caron

Word Count: 5956

A thesis submitted for the degree of

MMath Mathematics and Statistics

Trinity 2021

Abstract

In the existing literature regarding random partition models, the Chinese Restaurant Process (CRP) arises as a popular generative process of producing exchangeable partitions of the natural numbers, with a broad range of applications in machine learning, biological modeling, and network analysis. This dissertation provides an introductory overview and a brief analysis of the classical one-parameter CRP and its extended two-parameter model, which contributes significantly to the understandings of modern interaction data, such as email exchanges and phone calls. To investigate these interactions via edge-labeled networks, we present the well-known Hollywood model using the two-parameter CRP, which generates edge-exchangeable hypergraphs with satisfying statistical features. As the binary edge-exchangeable graphs are more prevalent and are of great research interest, we introduce the binary Hollywood model and perform its Frequentist inference in terms of maximum likelihood estimates (MLEs) of parameters as well as its Bayesian posterior inference via a Metropolis-Hastings algorithm. Provided the strong implications of edge-exchangeability examined in the previous literature, we finally propose and investigate a more structured random graph model based on the binary Hollywood process, which relaxes the exchangeability assumption and evolves over time through an additional parameter.

Contents

Acronyms

CRP Chinese Restaurant Process
HP Hollywood Process
BHP Binary Hollywood Process
NHP New Hollywood Process
MLE Maximum Likelihood Estimate

Chapter 1

Introduction

An extensive and growing body of literature has investigated the topic of random partitions with a wide range of applications, including well-known findings in population genetics provided by Kingman (1978) and clustering algorithms given by Lau and Green (2007). For these various applications, a significant avenue is to examine the asymptotic behaviors of quantities in random partition models, such as the number of clusters and the distributions of clusters of different sizes. Among all partition models of the natural numbers, the Chinese Restaurant Process (CRP) is a well-known simple generative model, that produces exchangeable partitions and has basic structural features such as sparsity, as demonstrated by Cai et al. (2016), and logarithmic growth of the number of clusters, as demonstrated by Korwar and Hollander (1973). This one-parameter model has its name inspired by the process that customers enter in order and sit at some tables in a restaurant. Much of the recent literature pays particular attention to the more complex two-parameter Chinese Restaurant process, which is another exchangeable random partition model. The extensions of two-parameter CRP are also of great interest, including the utilization of the spatial distance-dependent CRP in image segmentation researched by Ghosh et al. (2011) and the spectral CRP using nonparametric clustering researched by Socher et al. (2011). While in this paper, we only present the one-parameter CRP and two-parameter CRP, give relevant examples to aid comprehension, and analyze their asymptotic properties.

Having described the famous exchangeable partition model CRP, we proceed to its application in network analysis. Nowadays, many social science network datasets fields are composed of interactions between individuals in a population, such as email exchanges, collaborations, and citations. To gain a better understanding, we convert a sequence of interaction data into an edge-labeled network with vertices representing individuals and edges

representing interactions. Additionally, many scholars studying interaction data, such as Crane and Dempsey (2018), attach great importance to the edge-exchangeable network models because edge-exchangeability is essential in sampling mechanisms and ensures that observed interactions are representative of the interactions in a population. In this paper, we introduce the popular Hollywood model from the class of edge-exchangeable network models, whose generative process is analogous to the two-parameter CRP. Then we focus on the binary Hollywood model and provide both Frequentist and Bayesian inferences, including computing maximum likelihood estimates and simulating the posterior distribution using the Metropolis-Hastings algorithm.

The final objective of this article is to establish a more structured Bayesian model based on the Hollywood Process. We remove the exchangeability assumption by having the probabilities in the selection process of edges time-dependent, which allows the model to evolve over time. To illustrate this more flexible model, we use examples and examine the asymptotic performance of the number of clusters.

The following is a description of the dissertation's structure. In Section 2, we provide background on exchangeable random partitions and present the Chinese Restaurant Process. In Section 3, we introduce the edge-labeled networks and the concept of edge-exchangeability. In Section 4, we describe the general Hollywood model and a special case binary Hollywood model. In Section 5, we apply the Bayesian inference to the binary Hollywood model and compare the MLEs of parameters to their true values. In Section 6, we perform the Bayesian posterior inference, specifically by simulating the posterior distribution of parameters via a machine learning algorithm called Metropolis-Hastings. In Section 7, we establish a non-edge-exchangeable edge-labeled network model based on the binary Hollywood model and examine the effects of the additional parameter. In Section 8, we provide a summary of this dissertation, list major findings, and offer some recommendations for potential research on related topics.

Chapter 2

Chinese Restaurant Process

Definition 2.1 (Partition of $[n]$)**.** *For a finite set* $[n] \coloneqq \{1, 2, \ldots, n\}$*, its partition is defined as a set* $\pi_n = \{A_{n,1}, \ldots, A_{n,K_n}\}$ *where* $A_{n,j}$*,* $j = 1, \ldots K_n$*, are non-empty and non-overlapping subsets of* $[n]$ *with* $\bigcup_{i=1}^{K_n} A_{n,j} = [n]$*.*

According to definition 2.1 of the term partition provided by Di Benedetto et al. (2017), the terms $A_{n,j}(\ j = 1, \ldots K_n)$ and K_n refer to the partition's clusters and the number of clusters, respectively. The cardinality of a set is imposed by this definition, while this definition also applies to the countably infinite set $\mathbb{N} \coloneqq \{1, 2, \ldots\}$. As an illustration, $\pi_8 = \{\{1, 7\}, \{2, 3, 5\}, \{4, 6, 8\}\}$. There are three clusters here: $A_{8,1} = \{1, 7\}$ $A_{8,2} = \{2, 3, 5\}$ $A_{8,3} = \{4, 6, 8\}$. It is worth noting that different orders of subsets reflect the same partition.

Definition 2.2 (Exchangeable Partition)**.** *A random partition* π_n *is said to be* exchangeable *if its distribution is invariant under permutations of* $[n]$*;* $\pi \coloneqq (\pi_n)_{n \geq 1}$ *is an exchangeable partition of* $\mathbb{N}$ *if* π_n *is exchangeable for every n.*

Exchangeability assigns the same probability to partitions that contain the same number of elements with equal sizes, according to definition 2.2 provided by Di Benedetto et al. (2017). The probability of a partition is unaffected by element relabeling. As an example, for an exchangeable partition $\pi_4 = \{\{1, 2\}, \{3, 4\}\}$ of $[4]$, it satisfies $\mathbb{P}(\{\{1, 2\}, \{3, 4\}\}) = \mathbb{P}(\{\{1, 3\}, \{2, 4\}\}) = \mathbb{P}(\{\{2, 3\}, \{1, 4\}\})$.

2.1 One-parameter Chinese Restaurant Process

We now present a generative method for a particular class of exchangeable partition models, that generates a distribution on the partitions of $[n]$ or $\mathbb{N}$.

The following is a summary of the *Chinese Restaurant Process(CRP)* with just one parameter ϑ. Consider a Chinese restaurant with an unlimited number of tables, each labeled with $1, 2 \ldots$. Customers enter the restaurant in turn and sit at a table according to a random process:

- The first customer sits at the first table.
- The $(n+1)_{\text{th}}$ customer sits at a new table with probability $\frac{\vartheta}{n+\vartheta}$ or joins an existing table $j = 1, \ldots K_n$ with probability $\frac{m_{n,j}}{n+\vartheta}$, where $m_{n,j}$ is the number of customers on the j_{th} table and K_n is the number of occupied tables after n customers have walked in.

This procedure is denoted as $\text{CRP}(\vartheta, n)$. The total probability is checked to be $\frac{\vartheta}{n+\vartheta} + \sum_{j=1}^{k_n} \frac{m_{n,j}}{n+\vartheta} = \frac{\vartheta + \sum_{j=1}^{k_n} m_{n,j}}{n+\vartheta} = \frac{\vartheta+n}{n+\vartheta} = 1$. To ensure that probabilities are positive, the parameter ϑ takes only positive values. Since the probability of attracting a new customer to a table is proportional to the number of customers seated at that table, new customers are more likely to select tables with more customers, and therefore popular tables become more popular. From real-life observations, customers have a propensity to congregate, so the setup of the model is consistent with this assumption.

Fig 1 shows a seating arrangement of eight customers in a restaurant as a simple example of the one-parameter $\text{CRP}(\vartheta, 8)$ described above. The partition $\pi_8 = \{\{1,7\}, \{2,3,5\}, \{4,6,8\}\}$ with three clusters $A_{8,1} = \{1,7\}$, $A_{8,2} = \{2,3,5\}$, $A_{8,3} = \{4,6,8\}$ indicate the labels of customers seated at the first three tables.

Let us call the table where the j_{th} customer is seated as c_j. The probability of the seating arrangement in Fig 1 can be determined as:

$$\begin{aligned} &\mathbb{P}(c_1, c_2, \ldots, c_8) = \mathbb{P}(c_1) * \mathbb{P}(c_2|c_1) * \ldots \mathbb{P}(c_8|c_1, \ldots c_7) \\ &= \frac{\vartheta}{\vartheta} * \frac{\vartheta}{1+\vartheta} * \frac{1}{2+\vartheta} * \frac{\vartheta}{3+\vartheta} * \frac{2}{4+\vartheta} * \frac{1}{5+\vartheta} * \frac{1}{6+\vartheta} * \frac{2}{7+\vartheta} \end{aligned}$$

The probability of this seating arrangement is invariant under permutations of customers' labels: the denominator is an unchanged product $\vartheta(\vartheta+1)\ldots(\vartheta+n-1)$, and permutations

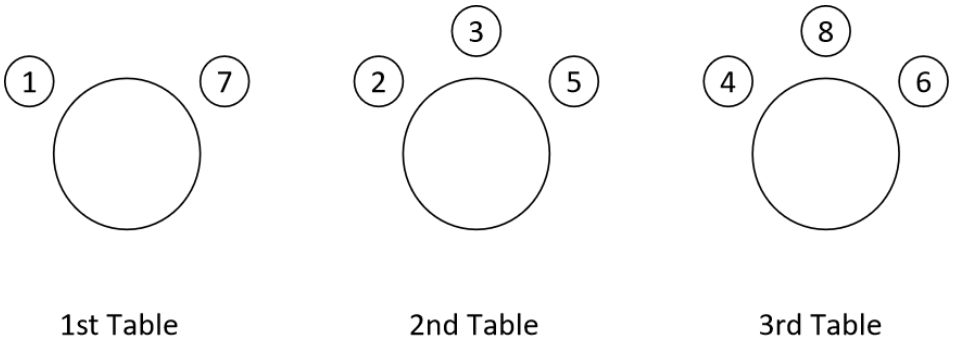

Figure 1: Chinese Restaurant Process with a seating plan for eight customers.

only permute factors in the numerator without changing its value. The same reasoning applies to the probability of an arbitrary seating plan for an infinite number of customers. Therefore, one-parameter CRP satisfies the exchangeability property.

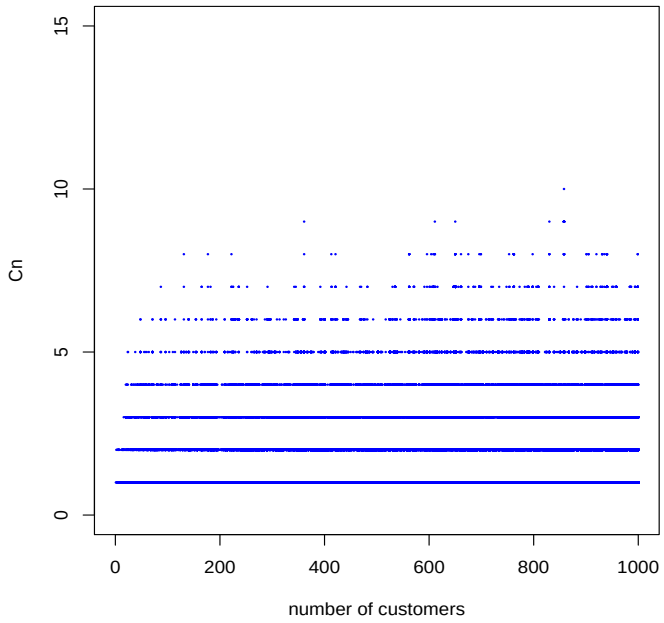

Figure 2: $C_n = (c_1, c_2, \ldots, c_n)$ versus the number of customers n in a Chinese Restaurant Process with $\vartheta = 0.5$.

We describe $C_n := (c_1, c_2, \ldots, c_n)$ as a vector with the j_{th} entry c_j defined above. Fig 2 depicts all entries of the vector C_n for the number of customers n ranging from 0 to 1000 in CRP$(0.5, n)$. In this plot generated via simulated data, there are fewer customers at tables with higher labels for a given n. The graph also reveals that the number of total tables $K_n = \max(c_1, c_2, \ldots, c_n)$ is a non-decreasing function of n with a steep increase for $n \leq 200$ and a slight increase for large n.

In addition, we analyze the statistical properties of K_n to better understand the growing trend of the random variable K_n with n.

$$K_n = \sum_{i=1}^{n} I\{c_i = new\}$$

As $I\{c_i = new\}, i = 1, \ldots, n$ are identity functions indicating whether the i_{th} customer sits at a new table with success probabilities $\frac{\vartheta}{i-1+\vartheta}$ depending only on i and known parameter ϑ, K_n is a sum of independent Bernoulli variables. Then

$$\begin{aligned}
E(K_n) &= \sum_{i=1}^{n} \mathbb{P}\{c_i = new\} = \sum_{i=1}^{n} \frac{\vartheta}{i-1+\vartheta} = \sum_{i=0}^{n-1} \frac{\vartheta}{i+\vartheta}, \\
\operatorname{Var}(K_n) &= \sum_{i=1}^{n} \operatorname{Var}(I\{c_i = new\}) = \sum_{i=1}^{n} \frac{\vartheta}{i-1+\vartheta}\left(1 - \frac{\vartheta}{i-1+\vartheta}\right) \\
&= \sum_{i=1}^{n} \frac{\vartheta(i-1)}{(i-1+\vartheta)^2} = \sum_{i=0}^{n-1} \frac{\vartheta i}{(i+\vartheta)^2}.
\end{aligned}$$

To understand the asymptotic behavior of the expectation, we approximate the exact sum by an integral:

$$\sum_{i=0}^{n-1} \frac{\vartheta}{i+\vartheta} \approx \int_0^{n-1} \frac{\vartheta}{x+\vartheta} dx = \vartheta\left[\log(x+\vartheta)\right]_0^{n-1} = \vartheta\left[\log(n+\vartheta-1) - \log(\vartheta)\right].$$

Therefore, for a sufficiently large number of edges n, $E(K_n)$ behaves asymptotically as $\vartheta \log(n)$, written as $E(K_n) \sim \vartheta \log(n)$.

Due to analogous reasoning, as $\frac{i-1}{i-1+\vartheta} \to 1$ as $i \to \infty$, we approximate the variance of K_n by the same integral:

$$\operatorname{Var}(K_n) \approx \sum_{i=0}^{n-1} \frac{\vartheta}{i+\vartheta} \sim \vartheta \log(n).$$

Then we further deduce that

$$\begin{aligned}
\operatorname{Var}\left(\frac{K_n}{\vartheta \log(n)}\right) &\sim \frac{1}{\vartheta \log(n)} \to 0 \text{ as } n \to \infty, \\
\frac{K_n}{\vartheta \log(n)} &\to 1 \text{ almost surely as } n \to \infty.
\end{aligned}$$

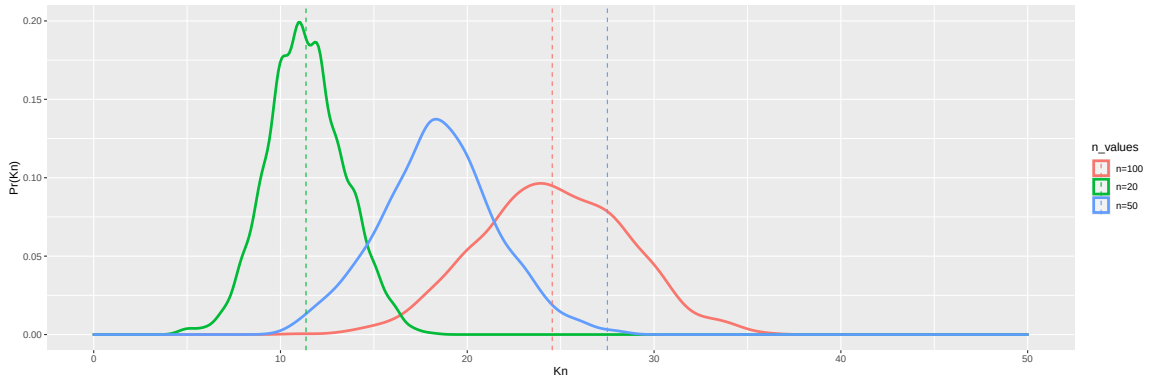

Figure 3: Density plots of K_n for $n = 20,\ n = 50,\ n = 100$; dashed vertical lines reflect the corresponding means of K_n

This result confirms that the number of clusters K_n grows in a logarithmic nature with n, consistent with the shape in Fig 2. We arrive at a stronger conclusion using Lindeberg Feller Central Limit Theorem:

$$\frac{K_n - \vartheta log(n)}{\sqrt{\vartheta log(n)}} \rightarrow N(0,\ 1) \text{ as } n \rightarrow \infty,$$

where $N(0,\ 1)$ is a standard normal distribution.

Three continuous density plots of K_n are shown in Fig 3, each corresponding to a different n value. It is worth noting that K_n is a discrete random variable which only takes positive integer values; however, we use the continuous distribution graphs for better observation and interpretation. The ranges of possible values of K_n widen as n increases from 20 to 50 and 100, and the relevant curves shift to the right and become flatter. For n in increasing order, K_n has expectations as 11.37, 24.55 and 27.50 respectively. In general, this evidence suggests that K_n might perform as a Gaussian distribution when n tends to infinity, and $E(K_n)$ rises at a decreasing speed.

2.2 Two-parameter Chinese Restaurant Process

Now we introduce a new parameter α and the two-parameter Chinese Restaurant Process is described in the same setup with probabilities now determined by two parameters α and ϑ. Customers walk in sequentially and sit at a random table as the following:

- The first customer sits at the first table.

- The $(n+1)_{\text{th}}$ customer sits at a new table with probability $\frac{K_n\alpha+\vartheta}{n+\vartheta}$ and joins an existing table $j = 1, \ldots K_n$ with probability $\frac{m_{n,j}-\alpha}{n+\vartheta}$, where $m_{n,j}$ is the number of customers seated at the j_{th} table and K_n is the number of occupied tables after n customers have entered.

This process is denoted as a CRP(α, ϑ, n). Analogous to the one-parameter CRP scenario, this two-parameter model possesses the exchangeability property which can be easily deduced by expanding the probability of any seating plan. The only modification in this two-parameter case is to include an additional parameter α to further specify the probabilities. When $\alpha = 0$, it becomes exactly a one-parameter CRP. From the possible ranges of variables $m_{n,j}$, K_n and constraints on positive probabilities, we derive the parameter space for the infinite numbers of customers case:

$$\begin{cases} 1 \leq m_{n,j} \leq n \\ 0 < \frac{m_{n,j}-\alpha}{n+\vartheta} < 1 \end{cases} \Rightarrow -\vartheta < \alpha < 1$$

$$\begin{cases} 1 \leq K_n \leq n \\ 0 < \frac{K_n\alpha+\vartheta}{n+\vartheta} < 1 \end{cases} \Rightarrow \alpha \geq 0$$

As a result, the two-parameter CRP with an infinite number of customers has a constrained parameter space:

$$H_2 = \left\{(\alpha, \vartheta) \in \mathbb{R}^2 : \ 0 \leq \alpha < 1, \ \vartheta > -\alpha\right\}.$$

Additionally, there is a situation when there are a limited number of customers, where

$$\mathbb{P}\left\{c_{n+1} = new\right\} = \frac{K_n\alpha + \vartheta}{n + \vartheta} = 0.$$

Therefore, $\vartheta = -k\alpha$ for some positive integer k and $\alpha < 0$, and

$$H_2{}^* = \left\{(\alpha, \vartheta) \in \mathbb{R}^2 : \ \alpha < 0, \ \vartheta = -k\alpha \ for \ some \ k \epsilon \mathbb{N}^+\right\}.$$

A result about the asymptotic performance of K_n can also be easily inferred: $K_n \rightarrow k \ a.s. \ as \ n \rightarrow \infty$.

The infinite population case with parameter space H_2 is the subject of this section. To examine the effects of parameters on K_n, we generate the graphs below to compare the means of K_n at various parameter values.

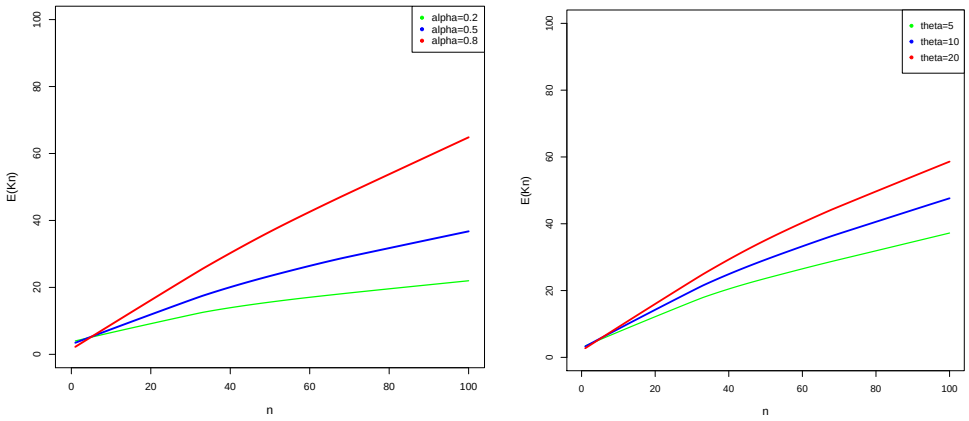

Figure 4: Plots of $E(K_n)$ versus n; Left: $\vartheta = 5$ and $\alpha = 0.2,\ 0.5\ 0.8$, Right: $\alpha = 0.5$ and $\vartheta = 5,\ 10,\ 20$.

The curves representing $E(K_n)$ with larger values of α lie above the curves representing lower α, as shown in the left graph of Fig 4. And the curves representing $E(K_n)$ with larger values of ϑ also lie above the curves representing lower ϑ in the right graph. This implies that $E(K_n)$ has a steady growth as any parameter increases for a fixed n. This outcome is in line with our calculations:

$$E\left(K_n\right) = \sum_{i=1}^{n} \mathbb{P}\left\{c_i = new\right\} = \sum_{i=1}^{n} \frac{K_i\alpha + \vartheta}{i - 1 + \vartheta} \tag{1}$$

$$= \sum_{i=1}^{n} 1 - \frac{i - 1 - K_i\alpha}{i - 1 + \vartheta} \tag{2}$$

From (1):

$$\forall \alpha > 0,\ \alpha \uparrow \Rightarrow\ E\left(K_n\right) \uparrow$$

From (2):

$$\vartheta \uparrow \Rightarrow\ \frac{i - 1 - K_i\alpha}{i - 1 + \vartheta} \downarrow \Rightarrow 1 - \frac{i - 1 - K_i\alpha}{i - 1 + \vartheta} \uparrow \Rightarrow E\left(K_n\right) \uparrow$$

As a result, we conclude that $E\left(K_n\right)$ is dependent on parameters: $E\left(K_n\right)$ has strictly positive relationships with α or ϑ. Notice that we are only investigating the actions of $E\left(K_n\right)$ when one parameter is fixed and the alternative is changeable; the case where two parameters vary simultaneously is not covered.

Chapter 3

Edge-exchangeable Networks

3.1 Interaction data

Definition 3.1 (Interaction data)**.** *An interaction process for a population P is a function $F : S \to fin(P)$ where S is a set indexing interactions, and $fin(P)$ is the set containing all finite multisets of P.*

In Def 3.1 provided by Crane and Dempsey (2018), interactions can be labeled using different methods, such as in an alphabetical order $\{A, B, C, \ldots\}$, thus the indexing set S can contain terms of any form. For simplicity, it suffices to consider $S = [n]$ containing a finite number of elements or $S = \mathbb{N}$ containing countably infinite elements. $F : S \to fin(P)$ is an injective function such that, for any given number $m \in S$, $F(m)$ refers to a set containing all the individuals who take part in the m_{th} interaction. This definition is prevalently used in the analyses of various network problems: phone calls and email exchanges are typical examples of directed interaction processes in which interactions initiate from senders to recipients; undirected interaction processes also exist in occasions where directions of interactions cannot determined, such as making friends in a club. In a directed interaction process $F : S \to fin(P)$, the direction of the m_{th} interaction is assumed to be represented by the canonical order of elements in $F(m)$.

As an example, $F : I \to fin(P)$ is an interaction process with $I = \{1, 2, 3, 4, 5, 6, 7\}$, $P = \{A, B, C, D, E\}$. And $F : I \to fin(P)$ is defined specifically as

$$
\begin{aligned}
F(1) &= (A, B) \\
F(2) &= (B, D) \\
F(3) &= (A, E) \\
F(4) &= (C, B) \\
F(5) &= (E, C) \\
F(6) &= (E, B) \\
F(7) &= (D, C)
\end{aligned} \tag{3}
$$

To help illustrate this example, the interactions are depicted in the graph below. In Fig 5, individuals in population P are represented by vertices $\{A, B, C, D, E\}$, interactions are represented by edges between vertices, and arrows show the directions of interactions. $F(3) = (A, E)$ indicates that the 3_{rd} interaction takes place between individuals A and E with a direction from A to E. The graph without labels on edges reflects the corresponding undirected example. All interactions are two-dimensional vectors, so the graph involves only binary edges. While in general cases, interactions can be of any finite sizes, reflected by the real-life situation that an email may have several recipients.

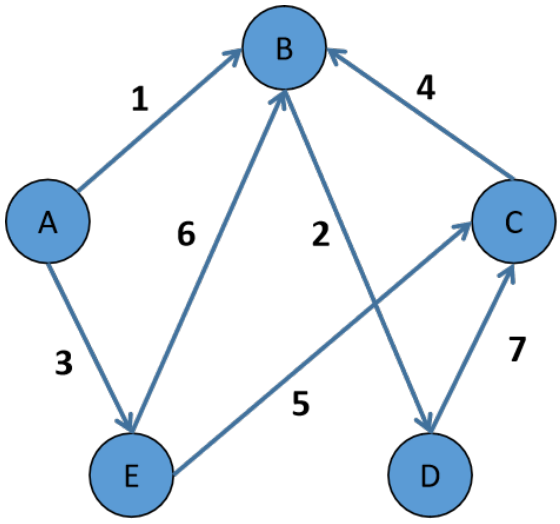

Figure 5: Depiction of an interaction process

3.2 Special Statistical Units in Networks

Conventional network data is represented as a graph with labeled vertices, with the goal of sampling a subset from a population. By marking the vertices in the observed results,

we obtain all possible edges between the sampled vertices. Since this completely captures the relationships between certain vertices while failing to reveal information about other vertices, which causes a great loss of information, hence interactions are not sampled sufficiently. For this reason, the edges are used as statistical units. To describe this in detail, we construct a subsample of data in an interaction process $F : \mathbb{N} \to \text{fin}(P)$ by constraining the network on chosen interactions via $F|_S : S \to fin(P),\ s \mapsto F(s)$ for some finite $S \in \mathbb{N}$. As an illustration, graphs from two sampling methods are shown in Fig 6 with statistical units as vertices and edges, respectively.

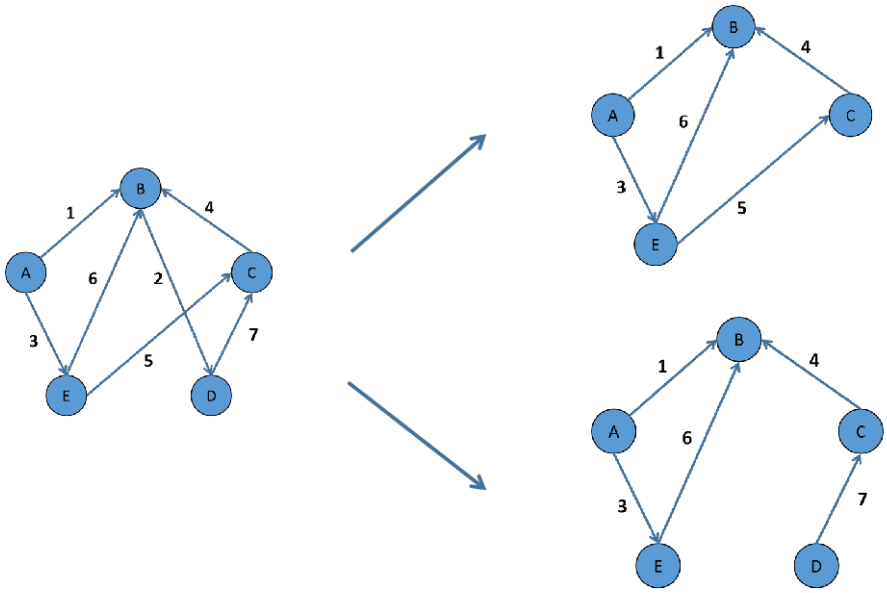

Figure 6: Left: network representation of an interaction process of a population (same as Fig 5). Upper right: observed network by sampling vertices. Lower right: observed network by sampling edges

The sampled individuals $\{A, B, C, E\}$ in the upper right network in Fig 6 inherit all interactions $\{1, 3, 4, 5, 6\}$ between each two of them in the original data. Compared to the upper right one, the sampled interactions $\{1, 3, 4, 6, 7\}$ and the connected vertices $\{A, B, C, D, E\}$ in the lower right graph are a better representation of the structure of interactions among the population.

3.3 Edge-labeled networks

Definition 3.2 (Edge-labeled network)**.** *An edge-labeled network induced by an interaction process $F : S \to fin(P)$ is a set containing all network representations of the interaction*

sequence.

According to a formal definition provided by Crane and Dempsey (2018), "an edge-labeled network induced by an interaction process $I : S \to fin(P)$ is an equivalence class

$\varepsilon_I = \bigcup_{P':\#P'=\#P} \left\{ I' : S \to fin\left(P'\right) : \ \rho I = I' \text{ for some bijection } \rho : P \to P' \right\}$, where $I' = \rho I$ is defined as $\forall i \in S : \ I'(i) = (\rho I)(i) = \rho\left(I(i)\right)$."

Proof of equivalence in definitions:

Step 1. We verify that an interaction-induced edge-labeled network is an equivalence class. It suffices to show that if $I : S \to fin(P)$, $I' : S \to fin\left(P'\right)$, and $I \sim I' \Leftrightarrow \exists$ bijection $\rho : P \to P'$ satisfies $\rho I = I'$, then $\sim$ is an equivalence relation satisfying three important properties.

(1) Reflexivity: trivial case in which ρ is the identity permutation

(2) Symmetry:

$$\exists\, bijection\ \rho : P \to P' \text{ s.t. } \rho I = I' \Rightarrow \exists \rho^{-1} : P' \to P \ : \ \rho^{-1} I' = I$$

(3) Transitivity:

$$I \sim I',\ I' \sim I'' \Leftrightarrow \exists\, bijections\ \rho_1, \rho_2 \text{ s.t. } \rho_1 I = I',\ \rho_2 I' = I''$$

$$\Rightarrow \rho := \rho_2 \circ \rho_1 \ \ s.t. \ \ \rho I = \rho_2\left(\rho_1(I)\right) = \rho_2\left(I'\right) = I \Rightarrow I \sim I''$$

Step 2.

All network representations of an interaction sequence are within this equivalence class as they are isomorphic up to permutations of vertices. And this equivalence class contains all edge-labeled networks that are identical after removing vertex labels, thus two definitions given above are equivalent.

As an example, three vertex-labeled, edge-labeled graphs on the left of Fig 7 are representations of a single interaction sequence, which has an interaction structure shown on the right. This result again supports our conclusion that vertex-labeled networks whose visual representations are identical after removing vertex labels refer to the same edge-labeled network.

To better analyze network-related quantities, we write them as symbols shown in Table 1. In Table 1, isolated vertices are not taken into account because they do not contain any details about interactions and therefore are not of interest. The definition of $M_k(\varepsilon)$ indicates that

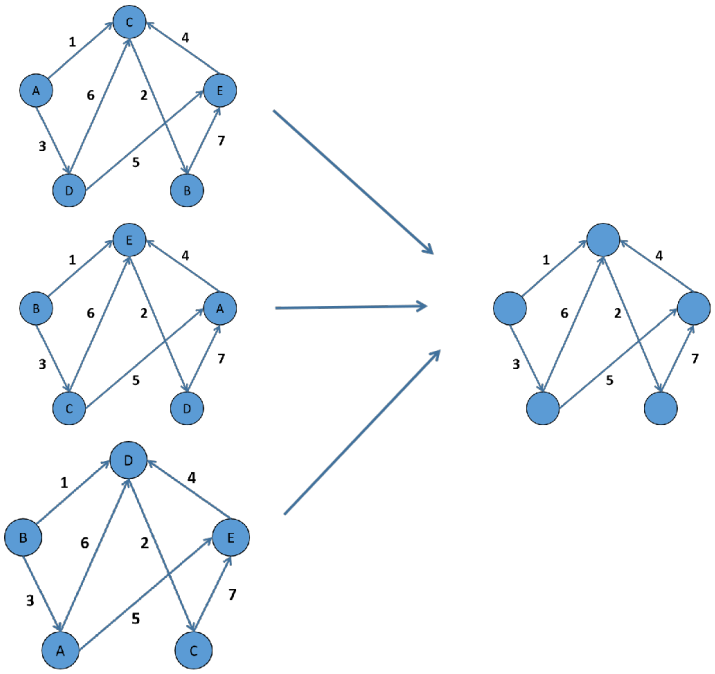

Figure 7: Left: graphical presentations of three interaction processes with an identical interaction structure. Right: edge-labeled graph corresponding to the class of all interaction processes with the same structure of those graphs on the left.

Table 1: An indication of meanings of multiple quantities of a network

Symbols	Definitions
$V(\varepsilon)$	The set of non-isolated vertices in ε
$E(\varepsilon)$	The set of edges in ε
$v(\varepsilon)$	Number of non-isolated vertices in ε
$e(\varepsilon)$	Number of edges in ε
$M_k(\varepsilon)$	Number of $k-ary$ edges in ε
$N_k(\varepsilon)$	Number of vertices which appear k times in ε
$d(\varepsilon)$	Degree distribution $d(\varepsilon) = \left(\frac{N_k(\varepsilon)}{v(\varepsilon)}\right)_{k\geq 1}$
$m(i)$	Degree of vertex i

our discussion is not limited to networks with merely binary edges; for $k \geq 3$, the graphs are referred to as multigraphs (defined in section 4). $d(\varepsilon)$ is the degree distribution in which the k_{th} entry is the proportion of vertices with degree k. $m(i)$ is the degree of vertex i which is the number of vertices adjacent to it. The edge-labeled network in Fig 5, for example, has the values of these quantities shown below.

(1) $V(\varepsilon) = \{A, B, C, D, E\}$
(2) $E(\varepsilon) = \{1, 2, 3, 4, 5, 6, 7\}$

(3) $v(\varepsilon) = 5$

(4) $e(\varepsilon) = 7$

(5) $M_k(\varepsilon) = \begin{cases} 7 \; for \; k = 2 \\ 0 \; otherwise \end{cases}$

(6) $N_k(\varepsilon) = \begin{cases} 2 \; for \; k = 2 \\ 2 \; for \; k = 3 \\ 1 \; for \; k = 4 \end{cases}$

(7) $d(\varepsilon) = \left(\frac{2}{5}, \; \frac{2}{5}, \; \frac{1}{5}\right)$

(8) $m(A) = 2, \; m(D) = 2$

3.4 Edge-Exchangeable Networks

Given an edge-labeled network ε induced by interaction data $F : S \rightarrow \text{fin}(P)$ and a permutation of interactions $\sigma : S \rightarrow S$, we denote the edge-labeled network obtained by relabeling the edges of ε through a permutation σ as ε^σ. Accordingly, ε^σ is the edge-labeled network induced by $F^\sigma : S \rightarrow \text{fin}(P)$, $F^\sigma(i) = F\left(\sigma^{-1}(i)\right)$. We consider the finite or countably infinite indexing set S as $[n]$ or $\mathbb{N}$ without loss of generality, thus a sequence of interaction data induced by $F : S \rightarrow \text{fin}(P)$ has an edge set $\{F(1), \; F(2), \ldots \}$. We are now able to introduce the concept of an edge-exchangeable network.

Definition 3.3 (Edge exchangeable network)**.** *A sequence of interaction data is edge-exchangeable if*

$\forall \; n \leq |S|, \; n < \infty \; : \; (F(1), \; F(2), \ldots F(n)) =_d (F^\sigma(1), \; F^\sigma(2), \ldots F^\sigma(n))$ *for any permutations* $\sigma : [n] \rightarrow [n]$.

Accordingly, an edge-labeled network ε *is edge-exchangeable if* $\varepsilon^\sigma =_d \varepsilon$ *for all permutations* $\sigma : S \rightarrow S$.

In an edge-exchangeable network model, edge-labeled networks that are isomorphic up to relabeling of edges have equal probabilities, according to this definition 3.3 given by Crane and Dempsey (2018). As an example, two graphs in Fig 8 have the same probability as there exists a permutation $\sigma : [7] \rightarrow [7]$ where $\sigma\left((1, 2, 3, 4, 5, 6, 7)\right) = (7, 1, 4, 3, 6, 5, 2)$, thus $\varepsilon^\sigma =_d \varepsilon$. The edge-exchangeability assumption is essential for random sampling of edges within a large network since the ordering of interactions has no impact on the models, which is compatible with many real-life network problems. The discrepancy between vertex-exchangeability and edge-exchangeability can be illustrated via a simple example:

in interaction data, the probabilities of seeing an email from individual A to individual B and from individual A to individual C usually differ in sampled data observation by random edge-sampling, but two probabilities are equal when A to B and A to C refer to the same edge in two graphs from a vertex-exchangeable model.

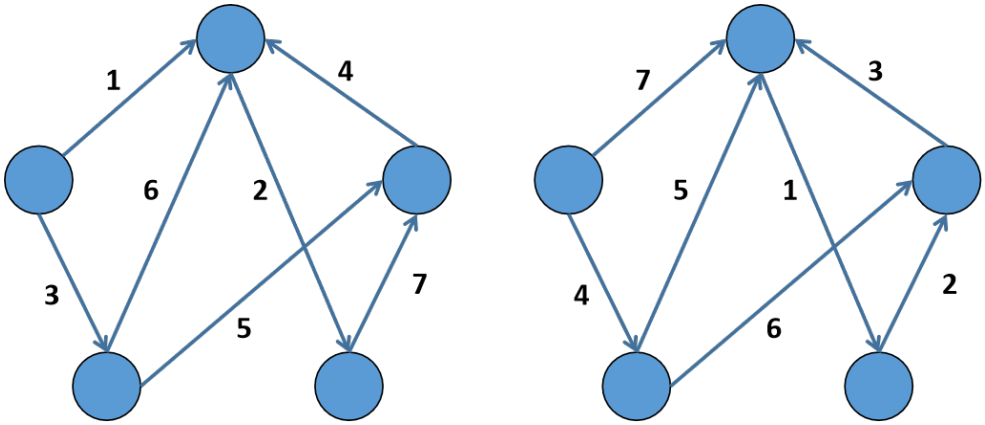

Figure 8: Two edge-labeled networks with the same probability in an edge-exchangeable model

Chapter 4

The Hollywood Model

4.1 The Hollywood Model

In this section, we present a canonical class of edge-exchangeable graph models based on the two-parameter Chinese Restaurant Process. As inferred by its name, the Hollywood model is described in terms of actors' collaboration network, with actors corresponding to vertices, films corresponding to edges, and each edge involving the collection of actors who participate in this film. To better understand the graphical properties of this model, we introduce the terms "hypergraph" and "multigraph" as below.

Definition 4.1 (Hypergraph)**.** *A hypergraph is a graph in which generalized edges (called hyperedges) may connect more than two nodes.*

According to definition 4.1 given by Berge (1989), a hyperedge connects several nodes. And a hypergraph can be represented graphically in a plane by identifying the hyperedges between vertices as domains that contain vertices connected by these edges. Fig 9 shows a hypergraph H with set of vertices $V(H) = \{A, B, C, D, E\}$ and set of edges $E(H) = \{E_1 = \{A, B, C\}, E_2 = \{B, C, D\}, E_3 = \{D, E\}\}$ in a plane.

Definition 4.2 (Multigraph)**.** *A multigraph is a graph in which multiple edges between nodes are permitted.*

This definition of multigraph is described by Harary (1994). Since the same collection of actors can collaborate in different movies and there are typically more than two actors in a movie, the Hollywood process aims to generate edge-exchangeable networks that are

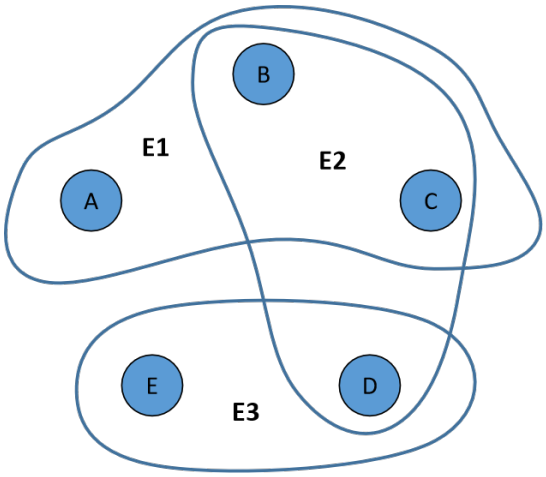

Figure 9: A plane representation of a hypergraph

multigraphs and hypergraphs to simulate the realistic model of actors' collaboration. Since an actor may play several roles in a movie, self-loops are permitted in this setup.

Suppose $p = (p_k)_{k\geq 1}$ is a distribution on $\mathbb{N}$ and the parameters α, ϑ satisfies either of the followings:

- $0 \leq \alpha < 1,\ \vartheta > -\alpha$ (infinite population)
- $\alpha < 0,\ \vartheta = -k\alpha$ for some positive integer k (finite population)

The method of producing a series of interactions in a Hollywood process is described as follows in terms of its graphical representation. Given $E_1, E_2, \ldots E_{n-1}$, the first $(n-1)$ edges, we first determine the number of vertices incident with the next edge using $A_n \sim p$. Provided $A_n = k$, we start the procedure of selecting k vertices $E_{n,1}, E_{n,2}, \ldots E_{n,k}$ in sequence, from the first node to the final one. Denote the total number of unique vertices contained in the first $(n-1)$ edges as well as the first $(j-1)$ vertices connected to the n_{th} edge as $V_n(j)$. For each $i = 1, 2, \ldots,\ V_n(j)$, let $D_{n,j}(i)$ be the number of the times that the vertex labeled i appear in the vertex selection process up to and including the $(j-1)_{\text{th}}$ vertex of the n_{th} edge. The vertex $E_{n,j}$ is chosen randomly from vertices labeled $1, 2, \ldots V_n(j)$ or a newly introduced vertex labeled $V_n(j)+1$ according to the following probabilities:

$$\mathbb{P}\left(E_{n,j} = i \middle| E_1, \ldots E_{n-1}, E_{n,1}, \ldots, E_{n,j-1}\right) \propto \begin{cases} D_{n,j}(i) - \alpha, & i = 1, 2, \ldots V_n(j) \\ \alpha V_n(j) + \vartheta, & i = V_n(j)+1 \end{cases}$$

By iterating this process, we record all the vertices incident with the n_{th} edge. Given all the edges $E_1, E_2, \ldots E_n$, we define $F_n : [n] \to fin(P)$ as $F_n(i) = E_i$ for $i = 1, 2, \ldots n$. Then we derive the edge-labeled network induced by F_n as Y_n, which is also denoted as

ξ_{F_n}. This process $(Y_n)_{n\geq 1}$ refers to the ***"Hollywood Process"*** with parameter (α, ϑ, p) , written as HP(α, ϑ, p). And a family of distributions on edge-exchangeable graphs induced by $F_{\mathbb{N}} : [\mathbb{N}] \to fin(P)$ is known as the ***"Hollywood model"*** emerges.

4.2 The Binary Hollywood Model

Starting with this section, we investigate the binary Hollywood model, in which each movie is performed by only two actors. Equivalently, each edge involves only two vertices (self-loops are allowed) in the generated network. In this case, the binary Hollywood Model becomes a method of sampling endpoints for binary edges, which is an application of the two-parameter Chinese Restaurant Process by identifying tables/clusters as vertices and identifying customers as half-edges.

As all edges are binary, we set $A_n = 2$ for all $n \geq 1$. To build a network with n edges $\{E_1, E_2, \ldots E_n\}$, we follow a Chinese Restaurant Process in $2n$ steps to assign $2n$ half-edges to vertices in sequence: $E_1 = \left(E_{1,1}, E_{1,2}\right), E_2 = \left(E_{2,1}, E_{2,2}\right), \ldots E_n = \left(E_{n,1}, E_{n,2}\right)$, where $E_{k,j}$ represents the starting node$(j = 1)$ or ending node$(j = 2)$ of the k_{th} edge for any finite k. The quantities $V_n(j)$ and $D_{n,j}(i)$ are defined identically in section 4.1.

The generative process of interaction sequences can be illustrated as follows. Provided $E_1, E_2, \ldots, E_{n-1}, E_{n,j-1}$, we choose a vertex to become $E_{n,1}(j = 1)$ as the starting node or $E_{n,2}(j = 2)$ as the ending node of the n_{th} edge with the following probabilities:

$$\mathbb{P}\left(E_{n,j} = i \middle| E_1, E_2, \ldots, E_{n-1},\ E_{n,j-1}\right) \propto \begin{cases} D_{n,j}(i) - \alpha, & i = 1, 2, \ldots V_n(j) \\ \alpha V_n(j) + \vartheta, & i = V_n(j) + 1 \end{cases}$$
$$= \begin{cases} \frac{D_{n,j}(i)-\alpha}{\vartheta+2n+j-3}, & i = 1, 2, \ldots V_n(j) \\ \frac{\alpha V_n(j)+\vartheta}{\vartheta+2n+j-3}, & i = V_n(j) + 1 \end{cases},$$

where we get the denominator by $\alpha V_n(j) + \vartheta + \sum_{i=1}^{V_n(j)} \left(D_{n,j}(i) - \alpha\right) = \vartheta + \sum_{i=1}^{V_n(j)} D_{n,j}(i) = \vartheta + 2(n-1) + (j-1) = \vartheta + 2n + j - 3$. Continue with this iterative process, we obtain a sequence of vertices and construct a network. This entire process is defined as the ***"binary Hollywood process"*** with parameter (α, ϑ), written as BHP(α, ϑ).

For any random edge-labeled network with a finite number of interactions $n \geq 1$ in a

BHP(α, ϑ) where $\alpha \neq 0$, the likelihood of Y_n has a closed-form expression as the following:

$$\mathbb{P}\left(Y_n = \varepsilon; \alpha, \vartheta\right) = \alpha^{v(\varepsilon)} \frac{\left(\frac{\vartheta}{\alpha}\right)^{\uparrow v(\varepsilon)}}{\vartheta^{\uparrow m(\varepsilon)}} \prod_{k=2}^{\infty} \exp\left\{N_k(\varepsilon) \log(1-\alpha)^{\uparrow(k-1)}\right\}, \tag{4}$$

where $\varepsilon \in \xi_{[n]}$, $v(\varepsilon)$ is the number of non-isolated vertices in ε, $N_k(\varepsilon)$ is the number of vertices with degree $k \geq 1$, $e(\varepsilon)$ is the number of binary edges in ε, $m(\varepsilon) = 2e(\varepsilon) = \sum_{k=1}^{\infty} kN_k(\varepsilon)$ is the total degree of ε, and $x^{\uparrow i} = x(x+1)\ldots(x+i-1)$.

Proof:

Suppose the number of vertices in ε is $N = v(\varepsilon)$ and the number of edges in ε is $n = e(\varepsilon)$. A directed network ε in a binary Hollywood process is characterized by the resulting sequence of vertices $\{E_{1,1}, E_{1,2}, E_{2,1}, \ldots, E_{n,2}\}$, rewritten as $\{V_1, V_2, V_3, \ldots, V_{2n}\}$ correspondingly. Then we write $v_1, v_2, \ldots, v_{2n}$ as labels of the vertices $V_1, V_2, \ldots, V_{2n}$, then $\{v_1, v_2, \ldots, v_{2n}\} = [N]$. Finally, we assume that $m\left(v_i\right)$ is the number of edges formed during the first $(i-1)$ steps that contain vertex v_i.

$$\mathbb{P}\left(Y_n = \varepsilon; \alpha, \vartheta\right) \tag{5}$$

$$= \mathbb{P}\left(E_1, E_2, \ldots, E_n; \alpha, \vartheta\right) \tag{6}$$

$$= \mathbb{P}\left(V_1 = v_1\right) \mathbb{P}\left(V_2 = v_2 | v_1\right) \ldots \mathbb{P}(V_{2n} = v_{2n} | v_1, \ldots, v_{2n-1}) \tag{7}$$

$$= \prod_{i=2}^{2n} \frac{\left(m\left(v_i\right)\right) I\left\{v_i \in \left\{v_{1,\ldots,}v_{i-1}\right\}\right\} + [\alpha(k-1)+\vartheta]\, I\left\{v_i \notin \left\{v_{1,\ldots,}v_{i-1}\right\}\right\}}{\vartheta + \sum_{j=1}^{i-1} m(v_j)} \tag{8}$$

$$= \frac{\prod_{i=2}^{2n} \left(m\left(v_i\right) - \alpha\right) I\left\{v_i \in \left\{v_{1,\ldots,}v_{i-1}\right\}\right\} + [\alpha(k-1)+\vartheta]\, I\left\{v_i \notin \left\{v_{1,\ldots,}v_{i-1}\right\}\right\}}{\vartheta(\vartheta+1)\ldots(\vartheta+m(\varepsilon)-1)} \tag{9}$$

$$= \frac{\left\{\prod_{k\geq 2} \left((1-\alpha)(2-\alpha)\ldots(k-1-\alpha)\right)^{N_k(\varepsilon)}\right\} * [\vartheta(\vartheta+\alpha)(\vartheta+2\alpha)\ldots(\vartheta+\alpha(N-1))]}{\vartheta(\vartheta+1)\ldots(\vartheta+m(\varepsilon)-1)} \tag{10}$$

$$= \frac{\left\{\prod_{k\geq 2} \left((1-\alpha)^{\uparrow(k-1)}\right)^{N_k(\varepsilon)}\right\} * \left[\frac{\vartheta(\vartheta+\alpha)(\vartheta+2\alpha)\ldots(\vartheta+\alpha(N-1))}{\alpha^N}\right] * \alpha^N}{\vartheta^{\uparrow m(\varepsilon)}} \tag{11}$$

$$= \frac{\left\{\prod_{k\geq 2} \left((1-\alpha)^{\uparrow(k-1)}\right)^{N_k(\varepsilon)}\right\} * \left[\left(\frac{\vartheta}{\alpha}\right)\left(\frac{\vartheta}{\alpha}+1\right)\ldots\left(\frac{\vartheta}{\alpha}+N-1\right)\right] * \alpha^N}{\vartheta^{\uparrow m(\varepsilon)}} \tag{12}$$

$$= \alpha^{v(\varepsilon)} \frac{\left(\frac{\vartheta}{\alpha}\right)^{\uparrow v(\varepsilon)}}{\vartheta^{\uparrow m(\varepsilon)}} \prod_{k=2}^{\infty} \exp\left\{N_k(\varepsilon) \log(1-\alpha)^{\uparrow(k-1)}\right\} \tag{13}$$

(8): This line is the product of all conditional probabilities and is simplified by writing probabilities in two ranges into one expression using identity functions. The expression $I\{A\}$ means the identity function of event A.

(8)⇒(9): Denominators in these two lines are equivalent because as j increases by one, one additional vertex and edge are generated and the sum of degrees of formed vertices increases by one. Therefore, the denominator becomes a product of factors starting from ϑ to $\vartheta + m(\varepsilon) - 1$.

(9)⇒(10): The numerator in (9) can be divided into two categories: probabilities of selecting new nodes and probabilities of selecting established nodes. There are N unique vertices in the generated network, implying that we have chosen N new nodes (including the first vertex), and thus the product is $\vartheta(\vartheta + \alpha)(\vartheta + 2\alpha)\ldots(\vartheta + \alpha(N-1))$. In the other case, a node has degree k has been chosen as a new node for the first time and chosen for an additional $(k-1)$ times as an established node, and thus the product is $(1-\alpha)(2-\alpha)\ldots(k-1-\alpha)$. As there is $N_k(\varepsilon)$ number of nodes of degree k, the probability for each $k \geq 2$ is $((1-\alpha)(2-\alpha)\ldots(k-1-\alpha))^{N_k(\varepsilon)}$, and the total product of probabilities is as shown.

(10)⇒(13): These three steps use the ascending factorial function $x^{\uparrow i}$ and $\alpha \neq 0$ to simplify the expressions and obtain the result as shown.

A special case of the binary Hollywood model is when $\alpha = 0$, corresponding to a one-parameter Chinese Restaurant Process with an expression of likelihood as the following:

$$\mathbb{P}\left(Y_n = \varepsilon; \vartheta\right) = \frac{\{\prod_{k\geq 2}(1 * 2\ldots(k-1))^{N_k(\varepsilon)}\} * \vartheta^{v(\varepsilon)}}{\vartheta(\vartheta+1)\ldots(\vartheta+m(\varepsilon)-1)}$$

$$= \frac{\{\prod_{k\geq 2}(k!)^{N_k(\varepsilon)}\} * \vartheta^{v(\varepsilon)}}{\vartheta^{\uparrow m(\varepsilon)}}$$

With slight changes to the equation (4), the probability of undirected networks can be estimated using the likelihood formula calculated for directed edge-labeled networks. Consider an undirected network $\varepsilon^{'}$, its probability is given as the expression in (4) multiplied by the total number of directed networks corresponding to $\varepsilon^{'}$. We write this quantity as $S(\varepsilon^{'})$, which unfortunately has no closed-form expression. However, since $S\left(\varepsilon^{'}\right)$ is independent of parameter (α, ϑ) and can be counted directly, whether a network is directed or not makes no difference to our inferences. Utilizing this result, we can further corroborate that edge-labeled networks generated by the binary Hollywood process satisfy edge-exchangeability.

Theorem 4.1 (Edge-exchangeability of the binary Hollywood model)**.** *The binary Hollywood model determines an edge-exchangeable probability distribution on ξ for all possible values of parameter (α, ϑ) in the parameter space.*

Proof. Let Y_n $(n \in \mathbb{N}^+)$ be a random network created by BHP(α, ϑ). Then the distribution of Y_n is given by (4) which is dependent on Y_n via $v(Y_n)$ and N_k (Y_n) , $(m(\varepsilon) = \sum_{k=1}^{\infty} kN_k(\varepsilon)$ is excluded). By the definition of edge-exchangeability, Y_n is an edge-exchangeable network since the number of non-isolated vertices and the number of vertices with degree k are invariant under relabeling edges. □

4.3 Interpretation of parameters

In this section, we examine parameter (α, ϑ) in two split parameter spaces of the binary Hollywood model, which correspond to finite and infinite population sizes of networks.

- $0 \leq \alpha < 1$, $\vartheta > -\alpha$ (infinite size): $\alpha > 0$ increases the probability of observing an unseen node and decreases the probability of observing an existing node in the network. So when α approaches one, it is more likely to see new vertices and $v(Y_n) \to \infty$ as $n \to \infty$. And ϑ has similar effects as α, though only having a minor impact on the probabilities when $n \to \infty$.
- $\alpha < 0$, $\vartheta = -k\alpha$ for some positive integer k (finite size): $\alpha < 0$ increases the probability that a newly added edge contains existing vertices while decreasing the probability of observing a new node, and similarly for ϑ. As a result, $v(Y_n) \to k$ as $n \to \infty$ with a proof parallel to the CRP case. To verify this theoretical result, here is a graph generated by simulating the Hollywood process.

In Fig 10, two curves are drawn as fitted lines of simulated data, where $v(Y_n)$ is the number of vertices in a random network generated with n edges in a binary Hollywood model. We observe that the red curve and the blue curve representing $v(Y_n)$ with parameter $(\alpha, \vartheta) = (-1, 10)$ and $(\alpha, \vartheta) = (-1, 20)$ both converge rapidly to their k values 10 and 20 when n increases. Although the range of n is limited as $(0,\ 500)$, $v(Y_n)$ in this finite population case has a nice asymptotic performance, which is consistent to our theoretical speculation above.

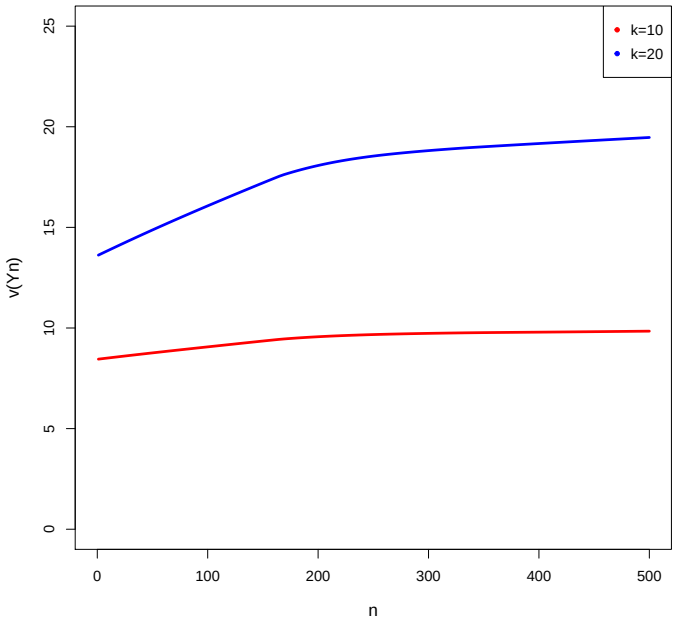

Figure 10: A plot of $v(Y_n)$ versus n: blue curve has parameter $(\alpha, \vartheta) = (-1, 10)$; red curve has parameter $(\alpha, \vartheta) = (-1, 20)$

Chapter 5

Frequentist inference of the binary Hollywood model

The Frequentist inference of the binary Hollywood Model is investigated in terms of maximum likelihood estimates in this section. Given the probability of observing a random edge-labeled network Y_n with parameter (α, ϑ) and $\alpha \neq 0$ in (4), the log-likelihood is

$$\begin{aligned}
\ell(\alpha, \vartheta; Y_n) &= \log\left(\mathbb{P}(Y_n; \alpha, \vartheta)\right) \\
&= \log\left(\alpha^{v(Y_n)} \frac{\left(\frac{\vartheta}{\alpha}\right)^{\uparrow v(Y_n)}}{\vartheta^{\uparrow m(Y_n)}} \prod_{k=2}^{\infty} \exp\left\{N_k(Y_n) \log(1-\alpha)^{\uparrow(k-1)}\right\}\right) \\
&= v(Y_n) \log(\alpha) + \sum_{j=0}^{v(Y_n)-1} \log\left(\frac{\vartheta}{\alpha} + j\right) \\
&\quad - \sum_{j=0}^{m(Y_n)-1} \log(\vartheta + j) + \sum_{k=2}^{\infty} N_k(Y_n) \sum_{j=0}^{k-2} \log(1 - \alpha + j)
\end{aligned}$$

where quantities are defined as before.

Setting the partial derivatives of log-likelihood to zero yields the maximum likelihood estimates (MLE) of parameter (α, ϑ), denoted as $\left(\widehat{\alpha}, \widehat{\vartheta}\right)$. Calculate the partial derivatives as follows.

$$\frac{(\alpha, \vartheta; Y_n)}{\alpha} = \frac{v(Y_n)}{\alpha} + \sum_{j=0}^{v(Y_n)-1} \frac{-\vartheta/\alpha^2}{\vartheta/\alpha + j} - \sum_{k=2}^{\infty} \sum_{j=0}^{k-2} \frac{N_k(Y_n)}{1 - \alpha + j}$$

$$\frac{(\alpha, \vartheta; Y_n)}{\vartheta} = \sum_{j=0}^{v(Y_n)-1} \frac{1/\alpha}{\vartheta/\alpha + j} - \sum_{j=0}^{m(Y_n)-1} \frac{1}{\vartheta + j}$$

We solve this system of equations and estimate $\widehat{\alpha}, \widehat{\vartheta}$ using the L-BFGS-B method, which is a modification of the quasi-Newton method allowing box-constraints, since we cannot give closed-form expressions for them. There is a condition $\alpha \neq 0$ which can trigger issues during the solving process; however, these estimates can be applied separately to the parameter spaces $\alpha < 0$ (finite population) and $0 < \alpha < 1$ (infinite population) to solve these problems. We then verify that the second partial derivatives at these points are negative, therefore these estimates are MLEs. The simulation results are obtained using the R programming, as shown below.

Table 2: Maximum likelihood estimates (MLEs) for (α, ϑ) based on network data with 2, 000 edges and 100 iterations.

MLE(α)	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$
$\vartheta = 5$	0.1989191	0.4946729	0.8002839
$\vartheta = 10$	0.1952891	0.4990428	0.8012615
$\vartheta = 20$	0.1971488	0.4992367	0.8003941

MLE(ϑ)	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$
$\vartheta = 5$	5.361953	5.53215	5.411126
$\vartheta = 10$	10.38161	10.61045	10.69782
$\vartheta = 20$	20.27777	19.93874	19.35484

The asymptotic consistencies of MLEs of parameters in the binary Hollywood model are checked in Table 2. While simulated networks have a relatively small size of 2000 edges, the MLE of α is found to be quite accurate, with an inaccuracy level at 0.001. Compared to the MLE of α, the MLE of ϑ is far less accurate. It is still a fair outcome as ϑ has a much wider range of possible values than α, and hence contains more randomness.

For the special case $\alpha = 0$, the log-likelihood satisfies

$$\ell\left(\vartheta; Y_n\right) = \log\left(\frac{\{\prod_{k\geq 2}\left(k!\right)^{N_k(Y_n)}\} * \vartheta^{v(Y_n)}}{\vartheta^{\uparrow m(Y_n)}}\right)$$

$$= \sum_{k=2}^{\infty} N_k\left(Y_n\right)\log(k!) + v\left(Y_n\right)\log(\vartheta) - \sum_{j=0}^{m(Y_n)-1}\log(\vartheta + j)$$

Then we differentiate the log-likelihood with respect to ϑ and get

$\frac{d(\vartheta;Y_n)}{d\vartheta} = \frac{v(Y_n)}{\vartheta} - \sum_{j=0}^{m(Y_n)-1}\frac{1}{\vartheta+j}$, where the last term can be approximated by $log(m\left(Y_n\right))$. Setting this derivative to zero, we obtain the result $\widehat{\vartheta} = \frac{v(Y_n)}{\log(m(Y_n))}$.

The likelihood of an unoriented network $\varepsilon^{'}$ created by the binary Hollywood process is the probability of a corresponding directed network multiplied by a factor $S(\varepsilon^{'})$, which is the count of the number of ways to transform $\varepsilon^{'}$ into a directed network. Then we have

$$\mathbb{P}\left(Y_n' = \varepsilon'; \alpha, \vartheta\right) = S\left(\varepsilon^{'}\right) \alpha^{v(\varepsilon')} \frac{\left(\frac{\vartheta}{\alpha}\right)^{\uparrow v(\varepsilon')}}{\vartheta^{\uparrow m(\varepsilon')}} \prod_{k=2}^{\infty} \exp\left\{N_k(\varepsilon') \log(1-\alpha)^{\uparrow(k-1)}\right\}$$

The log-likelihood of a network data Y_n' is

$$\ell'\left(\alpha, \vartheta; Y_n'\right) = \ell\left(\alpha, \vartheta; Y_n\right) + \log(S(Y_n'))$$

where $\ell\left(\alpha, \vartheta; Y_n\right)$ is the log-likelihood for any directed network Y_n which shares identical edges with Y_n'.

Chapter 6

Bayesian inference via Metropolis-Hastings

In the Bayesian approach, the unknown parameter (α, ϑ) is given a prior distribution $\pi(\alpha, \vartheta)$ reflecting subjective beliefs about it. Given an edge-labeled networks Y_n in the binary Hollywood model and its associated likelihood $\mathbb{P}(Y_n; \alpha, \vartheta)$, we compute the posterior distribution using the Bayes theorem:

$$\begin{aligned}\pi(\alpha, \vartheta | Y_n) &\propto \pi(\alpha, \vartheta) * \mathbb{P}(Y_n; \alpha, \vartheta) \\ &= \frac{\pi(\alpha, \vartheta)\mathbb{P}(Y_n; \alpha, \vartheta)}{\iint \pi(\alpha, \vartheta)\mathbb{P}(Y_n; \alpha, \vartheta)\, d\alpha d\vartheta}\end{aligned}$$

The aim is to reveal posterior information of parameters (α, ϑ) in the space $H = \{(\alpha, \vartheta) \in \mathbb{R}^2 : 0 < \alpha < 1,\ \vartheta > -\alpha\}$ for the infinite population case, in which we exclude the simpler case $\alpha = 0$. We utilize a machine learning algorithm called Metropolis-Hastings to sample from the posterior distribution since the posterior is intractable.

Priors: Since parameters α, ϑ are independent, we have $\pi(\alpha, \vartheta) \propto \pi(\alpha)\pi(\vartheta)$. The prior typically contains existing knowledge or expert recommendations for parameters. As an example, a common prior for α is $\pi(\alpha) \sim U(0, 1)$ as $0 < \alpha < 1$, indicating that there are no prior beliefs about α. We also select a prior $\pi(\vartheta) \sim \text{Gamma}(2, 2)$ for ϑ because ϑ has a distribution on almost all positive real numbers. Then the following information is provided:

Prior distribution $\pi(\alpha) \sim U(0, 1); \pi(\vartheta) \sim \text{Gamma}(2, 2)$ independent

Likelihood $\mathbb{P}(Y_n; \alpha, \vartheta)$

Posterior distribution $Q(\alpha, \vartheta) \coloneqq \pi\left(\alpha, \vartheta | Y_n\right) \propto \pi(\alpha, \vartheta) \mathbb{P}\left(Y_n; \alpha, \vartheta\right)$,

where $Q(\alpha, \vartheta)$ is the target distribution in the Metropolis-Hastings algorithm.

On the parameter space of (α, ϑ), we introduce a proposal distribution $q\left(\alpha, \vartheta | \, \alpha^{'}, \vartheta^{'}\right)$. It must satisfy $q\left(\alpha, \vartheta | \, \alpha^{'}, \vartheta^{'}\right) \geq 0$ and $\iint q\left(\alpha, \vartheta | \, \alpha^{'}, \vartheta^{'}\right) \, d\alpha d\vartheta = 1$ for any $\left(\alpha^{'}, \vartheta^{'}\right) \in H$, and does not depend on $\alpha^{'}, \vartheta^{'}$. To generate a sequence of Markov chain samples $\left(\alpha^{(t)}, \vartheta^{(t)}\right)_{t \geq 1}$ in H, the Metropolis-Hastings algorithm proceeds as the following:

Start with arbitrary $\alpha^{(0)}, \vartheta^{(0)}$, for $t = 1, 2, 3$:
1. Sample $(\alpha^*, \vartheta^*) \sim q(\bullet | \alpha^{t-1}, \vartheta^{t-1})$
2. Compute acceptance probability:

$$\mathbb{P}\left(\alpha^*, \vartheta^* \middle| \alpha^{t-1}, \vartheta^{t-1}\right) = \min\left(1, \, \frac{Q\left(\alpha^*, \vartheta^*\right) q(\alpha^{t-1}, \vartheta^{t-1} | \alpha^*, \vartheta^*)}{Q\left(\alpha^{t-1}, \vartheta^{t-1}\right) q(\alpha^*, \vartheta^* | \alpha^{t-1}, \vartheta^{t-1})}\right)$$

3. Sample $U \sim U[0, 1]$. If $U \leq \mathbb{P}\left(\alpha^*, \vartheta^* \middle| \alpha^{t-1}, \vartheta^{t-1}\right)$, set $(\alpha^t, \vartheta^t) = (\alpha^*, \vartheta^*)$; otherwise set $(\alpha^t, \vartheta^t) = (\alpha^{t-1}, \vartheta^{t-1})$.

This algorithm produces samples distributed according to $\pi\left(\alpha, \vartheta | Y_n\right)$. Given these posterior samples, we evaluate the posterior mean as the Bayes estimator, which minimizes the mean square error (MSE). The posterior means of α, ϑ are 0.9963147 and 2.15318, respectively. The posterior density functions are plotted below.

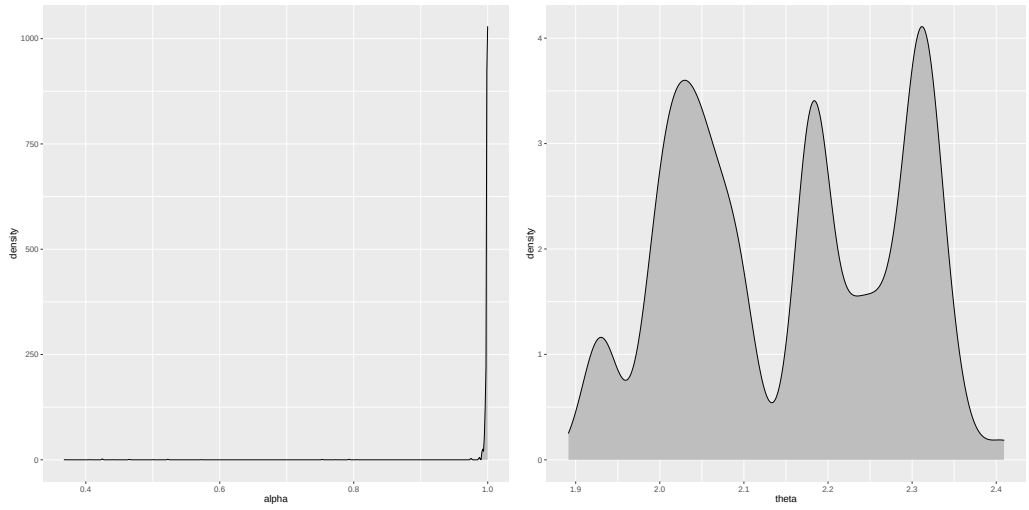

Figure 11: Density plots of the posterior distributions of parameters; Left: α, Right: ϑ

In Fig 11, two density plots are results from 5,000 posterior samples of parameters. The density of α is observed to be around one and ϑ has a density primarily concentrating in the

range $(1.9, 2.4)$. For a relatively small sample size, the choice of priors and the initial values of parameters have significant impacts on the derived distributions, so it is also reasonable to assume other more insightful priors to reduce randomness.

Chapter 7

Time-dependent binary Hollywood model

In this section, our goal is to investigate a more structured binary model which relaxes the exchangeability assumption. As there are diverse methods to break the exchangeability property of the Hollywood model, we use a setup that enables the interaction model to evolve. Specifically, the probability of a vertex joining a formed edge or creating a new edge depends on the increase of its degree over the last t time units.

With the same setup, the generative process of interaction sequences in this ameliorated model can be demonstrated as follows. Given $V_1 = v_1, V_2 = v_2, \ldots, V_{2n-1} = v_{2n-1}$, where v_{2k-1}, v_{2k} are labels of the vertices that form the starting and ending nodes of the edge k for any $k \geq 1$ and can be repetitive. Then we can select a vertex to become V_i as the starting/ending node with the following probabilities:

(1) $1 \leq i \leq t, i \in \mathbb{N}$

$$\mathbb{P}\left(V_i = v_i | v_1, \ldots, v_{i-1}\right) \propto \begin{cases} m\left(v_i\right) - \alpha, v_i \in \{v_1, \ldots, v_{i-1}\} \\ \alpha(k-1) + \vartheta, v_i \notin \{v_1, \ldots, v_{i-1}\} \, v_i \end{cases}$$
$$= \begin{cases} \frac{m(v_i)-\alpha}{\vartheta+i-1}, & v_i \in \{v_1, \ldots, v_{i-1}\} \\ \frac{\alpha(i-1)+\vartheta}{\vartheta+i-1}, v_i = max\left\{v_1, \ldots, v_{i-1}\right\} + 1 \end{cases},$$

where $m\left(v_i\right)$ is the degree of vertex v_i in the first $(i-1)$ steps. As $\sum_{j=1}^{i-1} m\left(v_j\right) = i-1$, the denominator is as shown.

(2) $i > t,\ i \in \mathbb{N}$

$$\mathbb{P}\left(V_i = v_i | v_1, \ldots, v_{i-1}\right) \propto \begin{cases} G_t\left(v_i\right) - \alpha, & v_i \in \{v_{i-1-t}, \ldots, v_{i-1}\} \\ \alpha t + \vartheta, & v_i = max\left\{v_1, \ldots, v_{i-1}\right\} + 1 \end{cases}$$

$$= \begin{cases} \frac{G_t(v_i)-\alpha}{\vartheta+t}, v_i \in \{v_{i-t}, \ldots, v_{i-1}\} \\ \frac{\alpha t+\vartheta}{\vartheta+t}, v_i = max\{v_1, \ldots, v_{i-1}\}+1 \end{cases},$$

where $G_t(x) := \sum_{j=i-1-t}^{i-1} I\{x = v_j\}$ is the number of times that v_j is the vertex chosen in the previous t steps (time units). This process is denoted as $NHP(\alpha, \vartheta, t)$. To aid comprehension of this complex model, we use a simple example to describe the generative process. The following Table 3. shows the chosen vertices in the first nine steps of this model.

Table 3: The table indicating the labels of the first 9 vertices $\{V_1, \ldots, V_9\}$

(Vn)$n >= 1$	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
(vn)$n >= 1$	1	1	2	2	3	1	2	4	3	?

As the vertex label of each element in the first row is shown in the same position on the second row of Table 3, it is observed that four vertices are chosen in the first 9 time units with vertex 1 three times, vertex 2 three times, vertex 3 twice and vertex 4 once. Provided that $t = 5$, α, ϑ are known parameters, we may determine the vertex label v_{10} with the following probabilities:

$$\mathbb{P}(V_{10} = v_{10} | v_1, \ldots, v_9) = \begin{cases} \frac{G_5(v_{10})-\alpha}{\vartheta+\sum_{j=10-5}^{10-1} G_5(v_j)}, v_{10} \in \{v_5, v_{6,} \ldots, v_9\} \\ \frac{5\alpha+\vartheta}{\vartheta+\sum_{j=5}^{9} G_5(v_j)}, v_{10} = 5 \end{cases}$$

$$= \begin{cases} \frac{1-\alpha}{\vartheta+5}, v_{10} = 1 \\ \frac{1-\alpha}{\vartheta+5}, v_{10} = 2 \\ \frac{2-\alpha}{\vartheta+5}, v_{10} = 3 \\ \frac{1-\alpha}{\vartheta+5}, v_{10} = 4 \\ \frac{5\alpha+\vartheta}{\vartheta+5}, v_{10} = 5 \end{cases}$$

This time-dependent model produces edge-labeled networks that are not edge-exchangeable, as seen in the following example. Using the data in Table 2 and adding one more vertex V_{10} with $v_{10} = 4$, then we can get a graphical representation of this network containing five edges on the left-hand side in Fig 12.

In Fig 12, ε_2 is a graph obtained from ε_1 by permuting the edges E_4 and E_5 in ε_1, which become E_4 and E_5 in ε_2. According to the graphical representation of ε_2 on the right, we can reproduce a sequence of vertices in ε_2 generated in the first 10 time units. The results are shown in the following Table 3. with vertex labels of ε_1 on the second row and vertex labels of ε_2 on the third row, which makes the comparison between two networks easier.

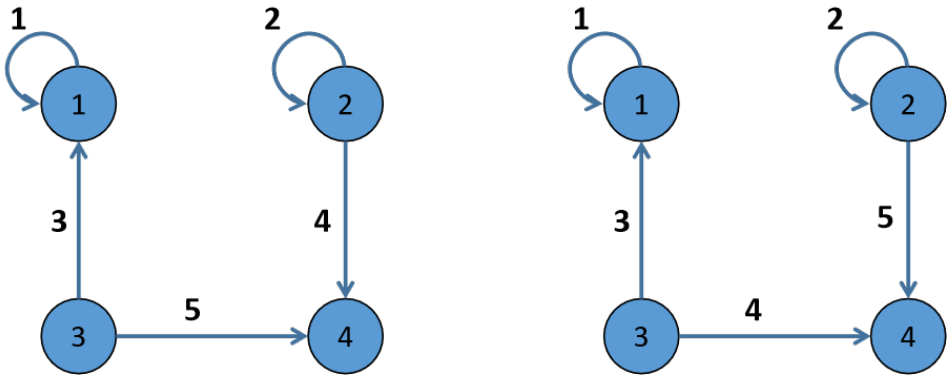

Figure 12: Graphical depictions of two networks from this model; Left: ε_1, Right: ε_2, where ε_1 and ε_2 are isomorphic up to relabeling edges

Table 4: Vertex labels for networks ε_1 and ε_2

(Vn)n >= 1	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
(vn)n >= 1 for ε1	1	1	2	2	3	1	2	4	3	4
(vn)n >= 1 for ε2	1	1	2	2	3	1	3	4	2	4

Using the results in Table 3, we can compute the probabilities of observing two networks as follows.

$$\mathbb{P}\left(Y_5 = \varepsilon_1\right) = \mathbb{P}\left(V_1 = 1\right)\mathbb{P}\left(V_2 = 1|v_1\right)\ldots\mathbb{P}\left(V_{10} = 4|v_5, \ldots, v_9\right),$$
$$\mathbb{P}\left(Y_5 = \varepsilon_2\right) = \mathbb{P}\left(V_1 = 1\right)\mathbb{P}\left(V_2 = 1|v_1\right)\ldots\mathbb{P}\left(V_{10} = 4|v_5, \ldots, v_9\right),$$

where Y_5 is a random edge-labeled network with five edges. Since the sequences of selected endpoints of the first three edges in ε_1 and ε_2 are identical, these two probabilities can only differ through the products of the last four terms. The product of probabilities for ε_1 is $\mathbb{P}(V_7 = 2|v_2, \ldots, v_6)\mathbb{P}(V_8 = 4|v_3, \ldots, v_7)\mathbb{P}(V_9 = 3|v_4, \ldots, v_8)\mathbb{P}(V_{10} = 4|v_5, \ldots, v_9) = \frac{2-\alpha}{\vartheta+5} * \frac{5\alpha+\vartheta}{\vartheta+5} * \frac{1-\alpha}{\vartheta+5} * \frac{1-\alpha}{\vartheta+5}$. And the product of probabilities for ε_2 is $\mathbb{P}(V_7 = 3|v_2, \ldots, v_6)\mathbb{P}(V_8 = 4|v_3, \ldots, v_7)\mathbb{P}(V_9 = 2|v_4, \ldots, v_8)\mathbb{P}(V_{10} = 4|v_5, \ldots, v_9) = \frac{1-\alpha}{\vartheta+5} * \frac{5\alpha+\vartheta}{\vartheta+5} * \frac{1-\alpha}{\vartheta+5} * \frac{1-\alpha}{\vartheta+5}$. Since these two probabilities are different, the networks generated in NHP$(\alpha, \vartheta, t = 5)$ in general do not satisfy the edge-exchangeability property that the typical Hollywood models possess.

Here is an outcome of a simulation of $MHP(\alpha = 0.5, \vartheta = 1.3, t = 6)$ with 20 edges. This generated network has 23 vertices with degrees of vertices

$(d(V_1), d(V_2), \ldots, d(V_{23})) = (2, 2, 1, 1, 4, 1, 6, 3, 1, 1, 1, 1\ ,1, 2\ ,1, 2, 1, 1, 1, 2, 1, 1, 1)$. To learn more about the properties of this model, we examine its asymptotic behaviors for a sufficiently large number of edges.

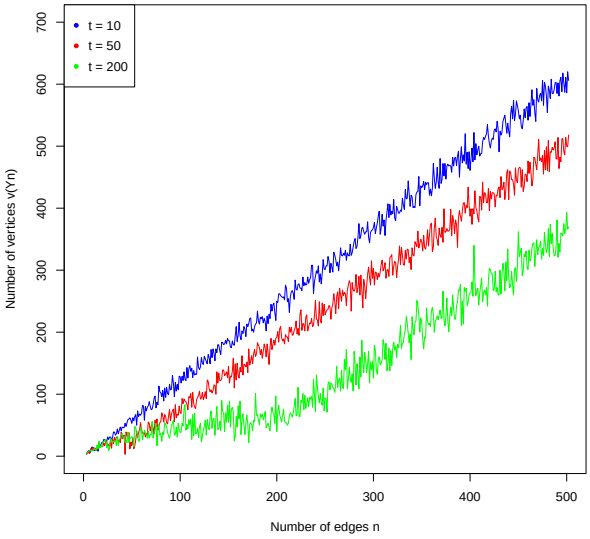

Figure 13: Three curves with colors blue, red, and green representing the number of vertices in networks from $MHP(\alpha = 0.5, \vartheta = 3, t)$ with $t = 10,\ 50,\ 200$, respectively

In Fig 13, the corresponding blue and red curves for $t = 10$ and $t = 50$ are approximately linear with the gradient of $t = 10$ case higher than that of $t = 50$ case. For small t, we can infer that the number of vertices $v\ (Y_n)$ is approximately a linear function of the number of edges n in the network, with its gradient decreasing as t increases. Different from the other two curves, the green curve describing the network in $MHP(\alpha = 0.5, \vartheta = 3, t = 200)$ is made up of two lines with different gradients, with the turning point at $n = 200$. This is a reasonable result since the network with the number of edges n lower than t is generated via the typical $HP(\alpha = 0.5, \vartheta = 3)$ represented by the first part of the curve. As n increases and exceeds t, the curve showing the number of created vertices is comparable to the red and blue curve, but this linear green curve has a lower gradient than those of the red and blue curves. For this special $t = 200$ case, the gradient of the first half is lower than that of the second half, indicating that the number of vertices grows at a faster rate in this modified model than in the canonical Hollywood model.

Chapter 8

Conclusion

This dissertation is summarized in terms of three parts. In the first part, we provide a systematic and detailed introduction to the standard CRP and two-parameter CRP by introducing the concepts of random partitions and exchangeable partitions. We further examine the effects of parameters in separate constrained parameter spaces and corroborate the theoretical findings with simulation results in CRPs. We show that the expectation of the number of clusters in CRP has a logarithmic growth pattern and asymptotically behaves like a normal distribution. And we reach the conclusion that both parameters in the two-parameter CRP are positively related to the number of clusters.

In the second part, we describe the generative process of the Hollywood model with a major emphasis on the binary case after clarifying the definitions of edge-labeled graphs and edge-exchangeable networks. Furthermore, we utilize the quasi-Newton method to calculate MLEs and use the simulation results to check their consistencies. A machine learning algorithm is implemented in the Bayesian inference, but it yields unsatisfactory results for posterior distributions.

In the final part, we present a more structured model that relaxes the exchangeability assumption in the Hollywood model by adding a parameter, that enables the model to be time-dependent. We show with a simple example why edge-exchangeability does not apply to this model and clarify how this new parameter affects the number of clusters.

Further study can be directed, and three possible directions are mentioned below. Firstly, to get a deeper understanding of the posterior distributions of parameters, it may be useful to assume different priors and increase the number of iterations to achieve higher precision. To reduce randomness, we can also use other distribution sampling techniques besides

Metropolis-Hastings, such as Gibbs Sampling and Hamiltonian Monte-Carlo methods. Secondly, it may be insightful to fit some real-life datasets with the non-edge-exchangeable model. Finally, it is worth discussing whether certain statistical properties of the Hollywood model such as sparsity are still valid in the new model. As if the new model has these features, it may be a strong alternative to the current Hollywood model when analyzing time-dependent edge-labeled network problems.

Bibliography

Berge, C. (1989). *Hypergraphs: The Theory of Finite Set*. North-Holland Publishing, Amsterdam, The Netherlands.

Blei, D. M. and Frazier, P. I. (2011). Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(8).

Cai, D., Campbell, T., and Broderick, T. (2016). Edge-exchangeable graphs and sparsity. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4249–4257, Red Hook, NY, USA. Curran Associates Inc.

Caron, F., Davy, M., and Doucet, A. (2012). Generalized polya urn for time-varying dirichlet process mixtures. *arXiv preprint arXiv:1206.5254*.

Crane, H. and Dempsey, W. (2018). Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113(523):1311–1326.

Di Benedetto, G., Caron, F., and Teh, Y. W. (2017). Non-exchangeable random partition models for microclustering. *arXiv preprint arXiv:1711.07287*.

Ghosh, S., Ungureanu, A., Sudderth, E., and Blei, D. (2011). Spatial distance dependent chinese restaurant processes for image segmentation. *Advances in Neural Information Processing Systems*, 24:1476–1484.

Gnedin, A., Haulk, C., and Pitman, J. (2009). Characterizations of exchangeable partitions and random discrete distributions by deletion properties. *arXiv preprint arXiv:0909.3642*.

Harary, F. (1994). *Graph Theory*. Addison-Wesley, Reading, MA.

Kingman, J. F. C. (1978). Random partitions in population genetics. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 361(1704):1–20.

Korwar, R. M. and Hollander, M. (1973). Contributions to the theory of dirichlet processes. *The Annals of Probability*, 1(4).

Lau, J. W. and Green, P. J. (2007). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics*, 16(3):526–558.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158.

Socher, R., Maas, A., and Manning, C. (2011). Spectral chinese restaurant processes: Non-parametric clustering based on similarities. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 698–706, Fort Lauderdale, FL, USA. PMLR.

Appendix

```
## One-parameter CRP with one parameter theta and nDnr customers
CRP <- function(theta, nDnr){
  vDnrTbl <- rep(0, nDnr)  # table label for each customer
  vDnrTbl[1] <- 1
  if (nDnr == 1){
    return (c(1))
  }
  else{
    for (dnr in 2:length(vDnrTbl)) {
      vOcc <- table(vDnrTbl[1:(dnr-1)])
      vProb <- c(vOcc, theta) / (dnr - 1 + theta)
      nTbl <- as.numeric(names(vOcc)[length(vOcc)])
      vDnrTbl[dnr] <- sample.int(nTbl+1, size=1, prob=vProb)
    }
    return (vDnrTbl)
  }
}


## Plot of Cn versus n
theta <- 0.5
n1 <- 1000
vDnrTbl.func <- function(theta, nDnr){
  return (CRP(theta,nDnr))
}
a <- numeric(n1)
res1 <- 0
for (i in 1:n1){
  res <- vDnrTbl.func(theta, i)
  a[i] <- length(res)
  res1 <- c(res1,res)
}
x <- rep(c(1:n1), a)
y <- res1[-1]
plot(x,y,pch=16,cex=0.3,cex.lab=1,col="blue",xlab="number of customers",ylab="
    Cn",xlim=c(0,1000),ylim=c(0,15))



## Density Plots of Kn
theta <- 10
```

```
N <- 1000
Kn.n <- function(n){
  Kn <- numeric(N)
  for (i in 1:N){
    Kn[i]=max(CRP(theta,n))
  }
  return(Kn)
}
A = c(rep(c("n=20","n=50","n=100"),each=N))
B = c(Kn.n(20),Kn.n(50),Kn.n(100))
C = c(mean(B[1:N]),mean(B[N+1:2*N]),mean(B[(2*N+1):(3*N)]))
D = c("n=20","n=50","n=100")
aa <- list(n_values = A,Kn = B)
mm <- list(n_values = D,mean = C)
data <- data.frame(lapply(aa, "length<-", max(lengths(aa))))
data.mean <- data.frame(lapply(mm, "length<-", max(lengths(mm))))
p <-ggplot(data,aes(x = Kn))
p+geom_density(aes(color = n_values),lwd=1.2)+xlim(c(0,50))+ylim(c(0,0.2)) +
    geom_vline(data=data.mean,aes(xintercept=mean,color=n_values),linetype="
    dashed")+ylab("Pr(Kn)")



## Two-parameter CRP with parameters theta,alpha and nDnr customers
CRP2 <- function(theta, alpha,nDnr){
  vDnrTbl <- rep(0,nDnr)
  if (nDnr == 1){
    return (c(1))
  }
  else{
    vDnrTbl[1] <- 1
    for (dnr in 2:length(vDnrTbl)) {
      vOcc <- table(vDnrTbl[1:(dnr-1)])
      prob <- c(vOcc - alpha) / (dnr - 1 + theta)
      vProb <- c(prob, 1-sum(prob))
      nTbl <- as.numeric(names(vOcc)[length(vOcc)])
      vDnrTbl[dnr] <- sample.int(nTbl+1, size=1, prob=vProb)
    }
  }
  return (vDnrTbl)
}


## Plot of E(Kn) versus n with theta=5 and alpha=0.2/0.5/0.8
Mean_Kn <- function(theta, alpha,nDnr){
  Kn <- numeric(20)
  for (i in 1:20){
    Kn[i]= max(CRP2(theta, alpha, nDnr))
  }
  res = mean(Kn)
  return (res)
}
theta = 5
x = c(1:100)
```

```
y1 = numeric(100)
y2 = numeric(100)
y3 = numeric(100)
for (i in 1:100){
  y1[i]= Mean_Kn(theta,0.2,x[i])
}
for (i in 1:100){
  y2[i]= Mean_Kn(theta,0.5,x[i])
}
for (i in 1:100){
  y3[i]= Mean_Kn(theta,0.8,x[i])
}
plot(lowess(x,y1),type="l",col="green",xlab="n",ylab="E(Kn)",cex.lab=1.2,lwd
    =2,xlim=c(0,100),ylim=c(0,100))
lines(lowess(x,y2),type="l",col="blue",lwd=3)
lines(lowess(x,y3),type="l",col="red",lwd=3)
legend("topright",border = "black",pch=c(20,20),legend=c("alpha=0.2","alpha
    =0.5","alpha=0.8"),col=c("green","blue","red"),bty="o")


## Plot of E(Kn) versus n with alpha=0.5 and theta=5/10/20
alpha = 0.5
x = c(1:100)
y1 = numeric(100)
y2 = numeric(100)
y3 = numeric(100)
for (i in 1:100){
  y1[i]= Mean_Kn(5,alpha,x[i])
}
for (i in 1:100){
  y2[i]= Mean_Kn(10,alpha,x[i])
}
for (i in 1:100){
  y3[i]= Mean_Kn(20,alpha,x[i])
}
plot(lowess(x,y1),type="l",col="green",xlab="n",ylab="E(Kn)",cex.lab=1.2,lwd
    =2,xlim=c(0,100),ylim=c(0,100))
lines(lowess(x,y2),type="l",col="blue",lwd=3)
lines(lowess(x,y3),type="l",col="red",lwd=3)
legend("topright",border = "black",pch=c(20,20),legend=c("theta=5","theta
    =10","theta=20"),col=c("green","blue","red"),bty="o")


## Plot of Kn in Hollywood process with theta = -1
alpha = -1
x = c(1:500)
z1 = numeric(500)
z2 = numeric(500)
Kn <- function(theta,alpha,nDnR){
  return(max(CRP2(theta,alpha,nDnR)))
}
for (i in 1:500){
  z1[i]= Kn(10,alpha,x[i])
}
```

```
for (i in 1:500){
  z2[i]= Kn(20,alpha,x[i])
}
plot(lowess(x,z1),type="l",col="red",xlab="n",ylab="v(Yn)",cex.lab=1.2,lwd=3,
    xlim=c(0,500),ylim=c(0,25))
lines(lowess(x,z2),type="l",col="blue",lwd=3)
legend("topright",border = "black",pch=c(20,20),legend=c("k=10","k=20"),col=c
    ("red","blue"),bty="o")



## Binary Hollywood process with parameters alpha,theta
dir.proc <- function(num.edges,alpha,theta) {
  n = 1
  deg.vertices = c(1)
  if(runif(1) < (theta+length(deg.vertices)*alpha)/(n+theta)){ #join a new
    table
    deg.vertices[length(deg.vertices)+1] = 1
    init.edge = c(1,2)
  } else{
    deg.vertices = c(2)
    init.edge = c(1,1)
  }

  edge.set = init.edge
  for (n in 2:(num.edges)) {
    new.edge = vector(length = 2)
    for(i in 1:2) { #each edge connects two vertices
      new.node.prob = (theta+length(deg.vertices)*alpha)/(2*(n-1)+i-1+theta)
      if(runif(1) < new.node.prob){
        deg.vertices[length(deg.vertices)+1] = 1
        new.edge[i] = length(deg.vertices)
      } else{
        temp.node = sample(x=1:length(deg.vertices),size=1,prob= (deg.vertices
     - alpha)/(2*(n-1)+i-1+theta) / (1-new.node.prob))
        deg.vertices[temp.node] = deg.vertices[temp.node]+1
        new.edge[i] = temp.node
      }
    }
    edge.set = rbind(edge.set, new.edge)
  }
  return(list("edge.set" = edge.set, "deg.vertices" = deg.vertices))
}



## Log-likelihood function calculated as a sum of four parts
log.lik <- function(deg.vertices,num.edges,alpha,theta) {
  num.vertices = length(deg.vertices)
  part1 = num.vertices*log(alpha)
  part2 = lgamma(theta/alpha+num.vertices)-lgamma(theta/alpha)
  m = 2*num.edges
  part3 = lgamma(theta+m)-lgamma(theta)
  part4 = sum(lgamma(deg.vertices[deg.vertices>1]-alpha)-lgamma(1-alpha))
```

```
  return(part1+part2-part3+part4)
}


## MLE of alpha
alpha.hat <- function(num.edges,alpha,theta){
  deg.vertices = dir.proc(num.edges,alpha,theta)$deg.vertices
  opt_alpha <- function(alpha){
    return(-1*log.lik(deg.vertices,num.edges,alpha,theta))
  }
  optimal = numeric(100)
  for (i in 1:100) {
    optimal[i] = optim(par=0.25,fn=opt_alpha,method="L-BFGS-B",lower=0.001,
    upper=0.999)$par
  }
  return(mean(optimal))
}
alpha_hat <- function(num.edges,alpha,theta){
  res = numeric(100)
  for (i in 1:100){
    res[i]=alpha.hat(num.edges,alpha,theta)
  }
  return(mean(res))
}


## MLE of theta
theta.hat <- function(num.edges,alpha,theta){
  deg.vertices = dir.proc(num.edges,alpha,theta)$deg.vertices
  opt_theta <- function(theta){
    return(-1*log.lik(deg.vertices,num.edges,alpha,theta))
  }
  optimal = numeric(100)
  for (i in 1:100) {
    optimal[i] = optim(par=5,fn=opt_theta,method="L-BFGS-B",lower=0.5,upper
    =25)$par
  }
  return(mean(optimal))
}
theta_hat <- function(num.edges,alpha,theta){
  res = numeric(100)
  for (i in 1:100){
    res[i]=theta.hat(num.edges,alpha,theta)
  }
  return(mean(res))
}


## Metropolis-Hastings Algorithm
log_lik <- function(param){
  num.edges = 2000
  deg.vertices = dir.proc(num.edges,param[1],param[2])$deg.vertices
  return(log.lik(deg.vertices,num.edges,param[1],param[2]))
}
```

```
## Define priors
prior <- function(param){
  alpha = param[1]
  theta = param[2]
  aprior = dunif(alpha, min=0, max=1, log=T)
  tprior = dgamma(theta, shape=2, rate=2,log=T)
  return(aprior+tprior)
}

## Define posteriors
posterior <- function(param){
  return(log_lik(param)+prior(param))
}

## Define proposal distributions
proposalfunction <- function(param){
  return(rnorm(2, mean=param, sd=c(0.1,0.1)))
}

run_metropolis_MCMC <- function(startvalue, iterations){
  chain = array(dim = c(iterations+1,2))
  chain[1,] = startvalue
  for (i in 1:iterations){
    proposal = proposalfunction(chain[i,])
    probab = exp(posterior(proposal) - posterior(chain[i,]))
    if (runif(1) < probab){
      chain[i+1,] = proposal
    }else{
      chain[i+1,] = chain[i,]
    }
  }
  return(chain)
}
chain = run_metropolis_MCMC(c(0.5,1.5),5000)
palpha = chain[,1]
ptheta = chain[,2]
pmean_alpha = mean(chain[,2]) # posterior mean of alpha
pmean_theta = mean(chain[,2]) # posterior mean of theta
burnIn = 2500
acceptance = 1-mean(duplicated(chain[-(1:burnIn),])) # acceptance probability
    of M-H algorithm

## Density plots of posterior distributions of alpha,theta
pmean_alpha = mean(chain[,1])
pmean_theta = mean(chain[,2])
aa <- list(alpha=palpha)
palpha_data <- data.frame(lapply(aa, "length<-", max(length(palpha))))
p<-ggplot(palpha_data, aes(x = alpha))
p + geom_density(color = "black", fill = "gray",cex.lab=2)
bb <- list(theta=ptheta)
ptheta_data <- data.frame(lapply(bb, "length<-", max(length(ptheta))))
p<-ggplot(ptheta_data, aes(x = theta))
p + geom_density(color = "black", fill = "gray")
```

```
## Extended Non-edge-exchangeable Hollywood model
# Probabilities related to edges formed in the last 2t time units/ las t
    entire edges
dir.proc_ex <- function(num.edges,alpha,theta,t) {
  if(num.edges<=t){
    return(dir.proc(num.edges,alpha,theta))
  }
  else{
    n = 1
    deg.vertices = c(1)
    deg.vertices.array = array(0,dim=c(5000,5000))
    deg.vertices.array[1,1]= 1

    if(runif(1) < (theta+length(deg.vertices)*alpha)/(n+theta)){ #join a new
    table
      deg.vertices[length(deg.vertices)+1] = 1
      init.edge = c(1,2)
    } else{
      deg.vertices = c(2)
      init.edge = c(1,1)
    }
    edge.set = init.edge
    deg.vertices.array[2,1:length(deg.vertices)]=deg.vertices

    for (n in 3:t) {
      new.edge = vector(length = 2)
      for(i in 1:2) {
        new.node.prob = (theta+length(deg.vertices)*alpha)/(2*(n-1)+i-1+theta)
        if(runif(1) < new.node.prob){
          deg.vertices[length(deg.vertices)+1] = 1
          new.edge[i] = length(deg.vertices)
        } else{
          temp.node = sample(x=1:length(deg.vertices),size=1,prob= (deg.
    vertices - alpha)/(2*(n-1)+i-1+theta) / (1-new.node.prob))
          deg.vertices[temp.node] = deg.vertices[temp.node]+1
          new.edge[i] = temp.node
        }
      }
      edge.set = rbind(edge.set, new.edge)
      deg.vertices.array[n,1:length(deg.vertices)]=deg.vertices
    }

    for (n in (t+1):(num.edges)){
      new.edge = vector(length = 2)
      for(i in 1:2) {
        new.node.prob = (theta+t*alpha)/(t+theta)
        if(runif(1) < new.node.prob){
          deg.vertices[length(deg.vertices)+1] = 1
          new.edge[i] = length(deg.vertices)
        } else{
          deg.vertices.in.t = (deg.vertices.array[n-1,]-deg.vertices.array[n-t
```

```
    ,])
          y = which(deg.vertices.in.t>0)[]
          temp.node = sample(x=y,size=1,prob= (deg.vertices.in.t[y] - alpha)/(
    t + theta)/ (1-new.node.prob))
          deg.vertices[temp.node] = deg.vertices[temp.node]+1
          new.edge[i] = temp.node
        }
      }
      deg.vertices.array[n,1:length(deg.vertices)] = deg.vertices
      edge.set = rbind(edge.set, new.edge)
    }
    return(list("edge.set" = edge.set, "deg.vertices" = deg.vertices))
  }
}


## Density plots of v(Yn) in the extended model
alpha <- 0.5
theta <- 3
t1 <- 10
t2 <- 50
t3 <- 200
x = c(3:502)
y1 = numeric(500)
y2 = numeric(500)
y3 = numeric(500)
Kn <- function(num.edges,alpha,theta,t){
  return(length(dir.proc_ex(num.edges,alpha,theta,t)$deg.vertices))
}
for (i in 1:500){
  y1[i]= Kn(x[i],alpha,theta,t1)
}
for (i in 1:500){
  y2[i]= Kn(x[i],alpha,theta,t2)
}
for (i in 1:500){
  y3[i]= Kn(x[i],alpha,theta,t3)
}
plot(y1~x,type="l",col="blue",xlab="Number of edges n",ylab="Number of
    vertices v(Yn)",cex.lab=1,lwd=1,xlim=c(0,500),ylim=c(0,700))
lines(y2~x,type="l",col="red",lwd=1)
lines(y3~x,type="l",col="green",lwd=1)
legend("topleft",border = "black",pch=c(20,20),legend=c("t = 10","t = 50","t =
     200"),col=c("blue","red","green"),bty="o")
```