ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Department of Statistical Sciences "Paolo Fortunati"

Second Cycle Degree/Master in Statistical Sciences

Curriculum Data Science

Bayesian Nonparametric Models for Sparse Bipartite Networks

Supervisor:

Prof. Saverio Ranciati

Presented by:

Andrea Ferrero

Co-supervisors:

Julyan Arbel

Tâm Le Minh

Session II

Academic Year 2024/2025

Acknowledgments

I would like to express my gratitude to the people who made this work possible and supported me throughout my academic journey.

First and foremost, to Julyan Arbel and Tâm Le Minh, my internship supervisors from the Inria Center at the University of Grenoble Alpes. I warmly thank them, as well as the rest of the Statify team, for the priceless professional and personal development they offered me (and the great coffee breaks!). I was given the immense opportunity of experiencing first-hand the world of research in a wonderful and positive environment.

To my supervisor at the University of Bologna, Prof. Saverio Ranciati, for his guidance and academic support.

To my partner Diana, whose essential presence has always helped me grow as a human being, a quality I value infinitely more than any degree or certification.

To my great friends, the new and the old ones, for the great moments we have shared and hopefully will keep on sharing.

Last but not least, to my family, for the constant support I was lucky to receive.

Abstract

Many real world networks are sparse, containing relatively few connections compared to the number of possible ones. Recent developments in network theory have introduced edge exchangeable models (Cai et al., 2016; Crane and Dempsey, 2018), which are closely connected to Bayesian Nonparametrics and have been extensively studied in the unipartite setting, in which edges may connect any pair of nodes. A key advantage of these models is their ability to capture sparsity while preserving a simple yet flexible structure.

This thesis extends the edge exchangeability framework to the bipartite setting, where edges occur only between nodes belonging to two distinct groups. We propose a Bayesian Nonparametric model based on the Pitman-Yor process, and prove that, under specific conditions, it can achieve sparsity similarly to its unipartite version. Theoretical results are validated through simulation studies.

We further develop a fully Bayesian estimation procedure for the model parameters, supported by additional simulations assessing its asymptotic properties. Finally, we apply the method to a real ecological network, demonstrating how to evaluate model fit and draw conclusions from the estimation process.

Keywords: Sparse Networks, Edge Exchangeability, Bayesian Nonparametrics, Pitman-Yor Process, Bipartite Networks, Bayesian Inference

Contents

Acknowledgments **iii**

Abstract **v**

Introduction **1**

1 Theoretical Framework **3**
1.1 Bayesian Nonparametrics . . . 3
1.1.1 Dirichlet Process . . . 3
1.1.2 Pitman-Yor Process . . . 7
1.2 Network Analysis . . . 9
1.2.1 Fundamentals of Graph Theory . . . 9
1.2.2 Network Models and Exchangeability . . . 10
1.2.3 Network Models and Sparsity . . . 12
1.3 Bayesian Nonparametric Network Models . . . 14
1.3.1 Unipartite Pitman-Yor Product Model . . . 14

2 Bayesian Nonparametric Model for Bipartite Networks **17**
2.1 Bipartite Pitman-Yor Product Model . . . 17
2.2 Sparsity Property . . . 18
2.3 Simulation studies . . . 22

3 Bayesian Model Estimation and Evaluation **27**
3.1 Inferential Framework and Methodologies . . . 27
3.1.1 Model Likelihood . . . 27
3.1.2 Bayesian Model Specification . . . 28
3.1.3 Prior Choice . . . 29
3.1.4 Bayesian Computation . . . 29
3.1.5 Posterior Predictive Check . . . 30
3.1.6 Hypothesis Testing . . . 31
3.2 Simulation Studies: Posterior Convergence . . . 33
3.3 Application to Real Data . . . 33

Conclusions **39**

Bibliography **41**

Appendix A: Supplementary Material **43**

Appendix B: Code **47**

Introduction

Networks provide a useful framework for representing and modelling complex relational data, with applications ranging from social interactions to biological systems. A fundamental characteristic of many real world networks is their sparsity: despite potentially containing many nodes, the observed number of edges connecting them is relatively small. Classical network models based on the assumption of node exchangeability are inadequate to capture this phenomenon (Orbanz and Roy, 2013), motivating the development of new probabilistic approaches that can represent sparse structures while retaining flexibility.

Progress in network theory has introduced edge exchangeable models (Cai et al., 2016; Crane and Dempsey, 2018). These models are closely related to the theory of Bayesian Nonparametrics (BNP), particularly through connections with the Pitman-Yor (PY) process and the Dirichlet Process (DP). In the unipartite setting, where edges may link any pair of nodes, edge exchangeable models using the PY process are capable of generating sparse networks.

The primary aim of this thesis is to extend the edge exchangeable framework to the bipartite case, where edges can only connect nodes belonging to two distinct groups. The bipartite structure is widely applicable, such as in ecological networks, recommender systems, and affiliation networks. Despite their prevalence, theoretical developments for sparse bipartite models remain limited. We contribute to this gap by proposing a Bayesian Nonparametric model for bipartite networks based on the PY process, and we prove that, under suitable conditions, the model retains the sparsity property established in the unipartite case. These theoretical results are validated through simulation studies, where we also examine the special case of the model using the DP.

Beyond the theoretical extension, we develop a fully Bayesian estimation procedure for the proposed model. Our methodology allows to perform inference on model parameters, and we investigate its performance in controlled simulations, showing that posterior distributions converge toward the true parameter values as the amount of data increases. We then demonstrate the applicability of the model through the analysis of an ecological network of plants and pollinators, illustrating how to evaluate model fit and how to extract insights on the sparsity mechanism.

The remainder of this thesis is organized as follows. In Chapter 1, we introduce the necessary concepts from BNP and network theory, and we present the PY-based model in the unipartite setting. In Chapter 2, we extend this model to bipartite networks, prove its sparsity property, and provide supporting simulations. In Chapter 3, we describe the inferential framework, assess model convergence with simulations, and apply it to real ecological data. Finally, we conclude by discussing implications of our findings and potential directions for future research.

Chapter 1

Theoretical Framework

The goal of this chapter is to review the theoretical background. The first section presents the main concepts from the field of Bayesian Nonparametrics, which constitute the foundations of the models treated in this thesis. The second section introduces network analysis, its graph theoretic foundations and the main network modelling framework. Particular attention will be given to the various notions of exchangeability applicable to random graphs, and how model assumptions affect their properties. In conclusion, a link with Bayesian Nonparametrics will be established by presenting existing models and results, which served as main motivation for this work.

1.1 Bayesian Nonparametrics

The field of Bayesian Nonparametrics (BNP), as the name suggests, can be viewed as the Bayesian counterpart of classical nonparametric statistics. Its core idea is to rely on less stringent assumptions about the data generating process, to increase flexibility and allow complexity to grow with the data.

This generally requires more advanced mathematical formulations. The original difficulties stemmed from the infinite dimensional parameter space that characterises nonparametric models, and the related necessity to construct probability measures to express prior beliefs on such spaces.

Since the pioneering work of Ferguson (1973) introducing the cornerstone of Bayesian nonparametrics, the Dirichlet Process, the research on the subject has grown extensively, to the point where the theoretical and computational challenges have become tractable. Nowadays, performing Bayesian Nonparametric inference in practice is doable, and a considerable amount of applications have been proposed.

The field extends much further than the following presentation. For an extensive review of the subject, we refer to Hjort et al. (2010) and Müller et al. (2015).

1.1.1 Dirichlet Process

The Dirichlet Process (DP) represents one of the simplest, yet widely used concepts from BNP.

Introduced by Ferguson (1973), the DP defines a prior on the space of probability measures. By specifying

its parameters, one can express different prior beliefs about the data generating process.

In this section, we assume to observe a sample $x_1, \ldots, x_n$ from a sequence of random variables $X_1, \ldots, X_n$, each taking values in a measurable space $(\Theta, \mathcal{B})$. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be the underlying probability space and $\mathcal{P}_\Theta$ denote the set of all probability measures defined on $(\Theta, \mathcal{B})$.

Definition 1 (Dirichlet Process, (Ferguson, 1973))**.** Let $\alpha > 0$ and G_0 be a probability measure on $(\Theta, \mathcal{B})$.

A $\mathrm{DP}(\alpha, G_0)$ is a random probability measure G such that for every finite measurable partition $\{B_1, \ldots, B_m\}$ of Θ, the vector of probabilities

$$(G(B_1), \ldots, G(B_m)),$$

has a Dirichlet distribution with parameters $(\alpha G_0(B_1), \ldots, \alpha G_0(B_m))$.

To stress how the DP can be seen as a prior on the distribution of the data, we can view the data generating process in the following hierarchical formulation:

$$\begin{aligned} G &\sim \mathrm{DP}(\alpha, G_0), \\ X_1, \ldots, X_n \mid G &\overset{\text{i.i.d.}}{\sim} G. \end{aligned}$$

The variables are conditionally independent and identically distributed given the random measure G. Equivalently, they can be seen as an exchangeable sequence, where G acts as the corresponding de Finetti measure, which in turn is random and drawn from the Dirichlet Process.

Following the Bayesian paradigm, the primary object of interest when estimating the distribution is the posterior $G \mid X_1, \ldots, X_n$. It turns out that the Dirichlet Process is conjugate with respect to i.i.d. sampling, meaning that the posterior distribution is again a Dirichlet Process.

Theorem 1 (Posterior of a DP, Ferguson (1973))**.** *Consider* $G \sim \mathrm{DP}(\alpha, G_0)$ *and* $X_1, \ldots, X_n \mid G \overset{\text{i.i.d.}}{\sim} G$*. We write* δ_{X_i} *for the Dirac measure, for* $i = 1, \ldots, n$*. Then,*

$$G \mid X_1, \ldots, X_n \sim \mathrm{DP}\left(\alpha + n, \frac{\alpha G_0 + n\mathbb{P}_n}{\alpha + n}\right),$$

where $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$

Considering the partition $\{B, B^c\}$, the expected value of $G(B)$ under a Dirichlet Process is $\mathrm{E}[G(B)] = G_0(B)$, while the variance is $\mathrm{Var}(G(B)) = \frac{G_0(B)G_0(B^c)}{\alpha+1}$. This shows that α plays the role of concentration parameter and G_0 of prior mean. The higher the concentration parameter, the more the prior distribution is concentrated around the prior mean, as shown in Figure 1.1.

From this it follows that the posterior mean of $G(B)$ is $E[G(B) \mid X_1, \ldots, X_n] = \frac{\alpha G_0 + n\mathbb{P}_n}{\alpha+n}$, a weighted average of the prior mean G_0 and $\mathbb{P}_n$, the Empirical Cumulative Distribution Function (CDF), the frequentist nonparametric estimator for CDFs. Each element is weighted by the prior and data sample sizes, resembling standard results with conjugate models.

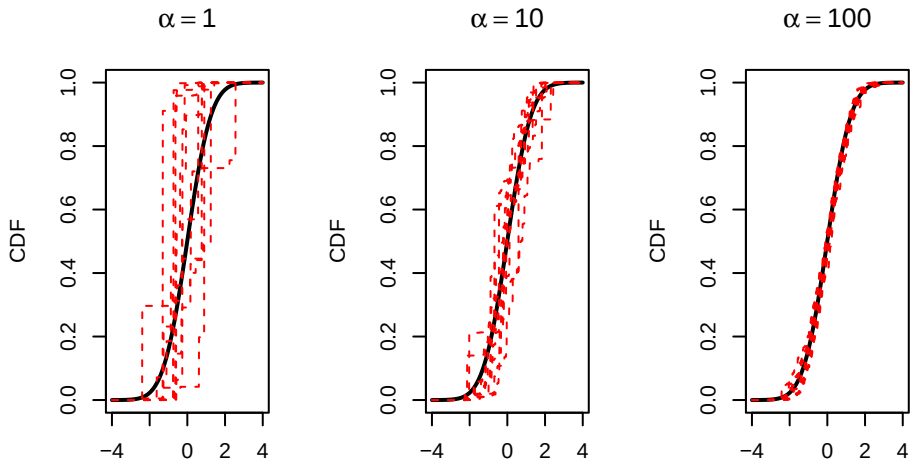


Figure 1.1: Three draws from a DP(α, G_0) (in red), with increasing α and $G_0 = N(0,1)$ (in black). Approximate sample paths are sampled from the Dirichlet Distribution implied by Definition 1.

The crucial property of the DP is its almost sure discreteness. This property has a number of implications, the immediate one being that the DP alone is not ideal for the estimation of continuous distributions. However, one could resort to mixture models with DP as prior on the mixing measure. More on this can be found in Lo (1984).

Moreover, since the measure is discrete, we can always write it as the infinite sum:

$$G(\cdot) = \sum_{k=1}^{\infty} w_k \delta_{\theta_k}(\cdot),$$

where $(\theta_k)_{k\in\mathbb{N}}$ is an infinite sequence of atoms in the space Θ and $(w_k)_{k\in\mathbb{N}}$ is an infinite sequence of non-negative weights summing to 1, associated to the atoms.

Another consequence of a.s. discreteness is that drawing a finite number of values from a DP will result in ties with positive probability, resulting in K_n unique values, which we denote as $x_1^*, \ldots, x_{K_n}^*$. Each observed value x_i can thus be written as

$$x_i = x_{c_i}^*,$$

where $c_i \in \{1, \ldots, K_n\}$ is an integer-valued indicator denoting the atom (or cluster) to which the observation x_i belongs.

The collection $\{c_1, \ldots, c_n\}$ represents a realisation of the underlying random partition $\{C_1, \ldots, C_n\}$ of the data induced by the DP, and often receives most of the attention in many applications. Its predictive law is known and takes the name of the Chinese Restaurant Process (CRP). Figure 1.2 provides a graphical illustration.

Definition 2 (DP CRP)**.** Given n draws from a Dirichlet Process, with K_n unique values of frequencies $n_1, \ldots, n_{K_n}$, the predictive distribution of C_{n+1} is the following:

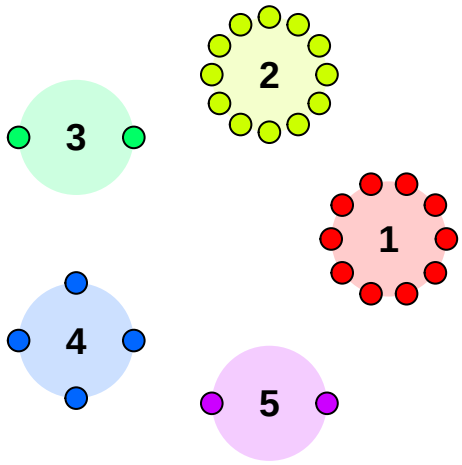


Figure 1.2: Chinese Restaurant Process representation. Different colors (the tables) represent different observed unique values. Customers sitting at tables denote the observed data belonging to the underlying group, represented by the unique value.

$$\Pr(C_{n+1} = k \mid C_1, \ldots, C_n) = \begin{cases} \frac{n_k}{\alpha+n}, & \text{for } k = 1, \ldots, K_n \\ \frac{\alpha}{\alpha+n}, & \text{for } k = K_n + 1 \end{cases}$$

The process is designed to promote a reinforcement mechanism, meaning that the probability of belonging to cluster k is proportional to the cluster size. On the contrary, higher α increases the chance of creating a new cluster at every draw.

The predictive structure is not only useful to understand the mechanism behind the process, but it also constitutes a valuable tool, from an algorithmic point of view, to generate samples from the random partition induced by the DP.

Another well known fact, shown in Korwar and Hollander (1973), is that the number of unique values in a DP sample grows logarithmically with the sample size. More precisely,

$$K_n \sim \alpha \log(n) \quad \text{a.s. as } n \to \infty.$$

If the partition structure is the main object of interest, another essential tool is the Exchangeable Partition Probability Function (EPPF). This function represents the probability of observing a specific partition of the data.

Definition 3 (DP EPPF)**.** Consider n draws from a Dirichlet Process, with K_n unique values of frequencies $n_1, \ldots, n_{K_n}$. The probability of observing such partition is given by the Exchangeable Partition Probability Function (EPPF):

$$p_n(n_1, \ldots, n_{K_n}) = \frac{\alpha^{K_n}\,\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{k=1}^{K_n} \Gamma(n_k),$$

where $\Gamma(\cdot)$ denotes the Gamma function.

Note that the EPPF depends only on the the parameter α and the partition structure.

Another common representation of the DP is through the stick-breaking construction. It gives a clear picture of the underlying generating process, using the analogy of a stick of unit length breaking into smaller pieces (the weights) every time a new atom is drawn.

Definition 4 (DP Stick-Breaking, (Sethuraman, 1994))**.** Let $\alpha > 0$ and G_0 a probability measure on a measurable space $(\Theta, \mathcal{B})$. Then a Dirichlet Process $G \sim \mathrm{DP}(\alpha, G_0)$ can be represented as:

$$\begin{aligned}
\beta_k &\sim \mathrm{Beta}(1, \alpha), \quad \text{for } k = 1, 2, \ldots, \infty, \\
w_k &= \beta_k \prod_{i=1}^{k-1} (1 - \beta_i), \\
\theta_k &\overset{\text{i.i.d.}}{\sim} G_0, \\
G(\cdot) &= \sum_{k=1}^{\infty} w_k \delta_{\theta_k}(\cdot).
\end{aligned}$$

The distribution on the weights w_k implied by the stick-breaking construction is frequently termed the GEM(α) distribution, after Griffiths-Engen-McCloskey (Pitman, 2006).

We can then write the representation more compactly as:

$$\begin{aligned}
w_k &\overset{\text{i.i.d.}}{\sim} \mathrm{GEM}(\alpha), \\
\theta &\overset{\text{i.i.d.}}{\sim} G_0, \\
G(\cdot) &= \sum_{k=1}^{\infty} w_k \delta_{\theta_k}(\cdot).
\end{aligned}$$

The stick-breaking construction also serves as a commonly used tool to generate values from a Dirichlet Process, by choosing a truncation level on the number of stick breaking steps.

1.1.2 Pitman-Yor Process

Introduced by Pitman and Yor (1997), the Pitman-Yor (PY) process can be seen as an extension of the Dirichlet Process.

We will denote the Pitman-Yor process as $\mathrm{PY}(\alpha, \sigma, G_0)$, where α is the concentration parameter as before, and σ is the discount parameter.

The major motivation to choose a PY process over a DP is to increase the flexibility of the prior and accommodate a power-law growth rate in the number of unique values, which is a behaviour often encountered in practice, instead of the logarithmic growth.

In particular, if $\sigma > 0$, the number of unique values K_n from a PY satisfies:

$$\frac{K_n}{n^\sigma} \xrightarrow{\text{a.s.}} S_\sigma \qquad \Longleftrightarrow \qquad K_n \sim S_\sigma \, n^\sigma \quad \text{a.s. as } n \to \infty.$$

where the r.v. S_σ is called sigma diversity. (See Pitman (2006))

For $\sigma = 0$, the PY process coincides with the DP. This can be seen by comparing the various representations also available for the PY process.

Definition 5 (PY Stick-Breaking, (Pitman and Yor, 1997))**.** Let $\alpha > -\sigma$, $\sigma \in [0, 1)$, and G_0 a probability measure on a measurable space $(\Theta, \mathcal{B})$. Then a random probability measure $G \sim \text{PY}(\alpha, \sigma, G_0)$ admits the following constructive representation:

$$\begin{aligned}
\beta_k &\sim \text{Beta}(1 - \sigma, \alpha + k\sigma) \quad \text{for } k = 1, 2, \ldots, \infty, \\
w_k &= \beta_k \prod_{i=1}^{k-1} (1 - \beta_i), \\
\theta_k &\overset{\text{i.i.d.}}{\sim} G_0, \\
G(\cdot) &= \sum_{k=1}^{\infty} w_k \delta_{\theta_k}(\cdot).
\end{aligned}$$

With $\sigma = 0$, one recovers the DP stick-breaking construction (See Definition 4).

Equivalently, we can denote the weights distribution as the $\text{GEM}(\alpha, \sigma)$ and write the stick breaking representation as follows:

$$\begin{aligned}
w_k &\overset{\text{i.i.d.}}{\sim} \text{GEM}(\alpha, \sigma), \\
\theta &\overset{\text{i.i.d.}}{\sim} G_0, \\
G(\cdot) &= \sum_{k=1}^{\infty} w_k \delta_{\theta_k}(\cdot).
\end{aligned}$$

The Pitman-Yor process also admits a Chinese Restaurant Process representation, which also gives further intuition on its connections with the Dirichlet Process.

Definition 6 (PY CRP, (Pitman and Yor, 1997))**.** Given n observations from a Pitman-Yor process, with K_n unique values of frequencies $n_1, \ldots, n_{K_n}$, the predictive distribution of C_{n+1} is given by:

$$\Pr(C_{n+1} = k \mid C_1, \ldots, C_n) = \begin{cases} \frac{n_k - \sigma}{\alpha + n}, & \text{for } k = 1, \ldots, K_n \\ \frac{\alpha + \sigma K_n}{\alpha + n}, & \text{for } k = K_n + 1 \end{cases}$$

The Pitman–Yor CRP with $\sigma > 0$ shows where the power-law effect takes place thanks to a bigger-gets-bigger mechanism: only a few clusters will be big, having overcome the effect of the discount parameter, while most will remain small. However, with $\sigma = 0$ one recovers exactly the Dirichlet Process CRP (See Definition 2).

An efficient and vectorised R function to sample the cluster allocations using the CRP representation can be found in Script 1.

In conclusion, also samples from the PY process can be characterised by its EPPF.

Definition 7 (PY EPPF, (Pitman and Yor, 1997))**.** Consider n draws from a Pitman-Yor Process, with K_n unique values of frequencies $n_1, \ldots, n_{K_n}$. The probability of observing such partition is given by the Exchangeable Partition Probability Function (EPPF):

$$p_n(n_1, \ldots, n_{K_n}) = \frac{\prod_{i=1}^{K_n - 1}(\alpha + i\sigma)}{(\alpha + 1)_{n-1}} \prod_{k=1}^{K_n} (1 - \sigma)_{n_k - 1},$$

where $(a)_n = a(a+1)\cdots(a+n-1)$ denotes the rising factorial (Pochhammer symbol).

Again, the EPPF depends only on the model parameters and the partition structure.

1.2 Network Analysis

Networks are data structures used to represent relationships between entities. The goal of network analysis is to model and infer the mechanisms that generate observed relational structures, enabling prediction, pattern discovery, and causal interpretation. Many real world networks exhibit distinctive features, like the aforementioned sparsity, and it is crucial to develop models able to capture such properties.

In what follows, the minimal background is given, but for a more extensive review of the statistical analysis of networks we refer to Crane (2018).

1.2.1 Fundamentals of Graph Theory

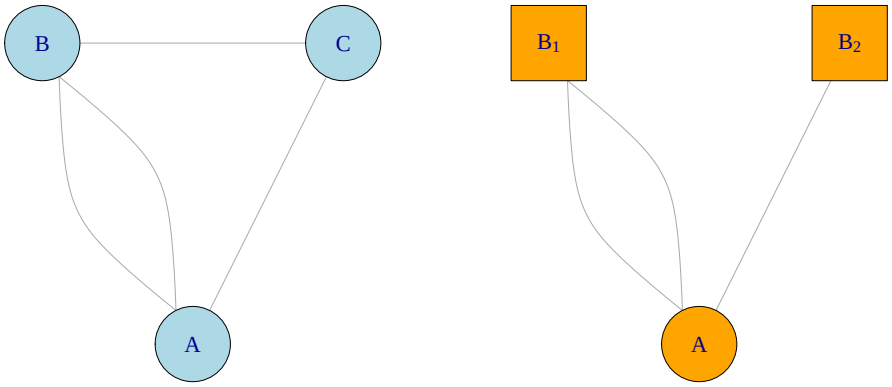
Graphs are the common choice when it comes to representing network structures.

Formally, a graph $\mathcal{Y} = (\mathcal{V}, \mathcal{E})$ consists of a set of *vertices* or *nodes* $\mathcal{V}$, which represent the entities, and a set of *edges* $\mathcal{E}$, which represent the relations between entities. The number of nodes of the graph will be indicated as $v(\mathcal{Y})$, while the number of edges $e(\mathcal{Y})$.

Graphs can be distinguished in different classes based on edge types. First, a graph can be either **directed** or **undirected**. In the former case, edges have an orientation, indicating a sender and a receiver, and are formally represented by the set of ordered pairs $\mathcal{E} := \mathcal{V} \times \mathcal{V}$. An example is email interactions. In the other case, edges simply represent mutual relationships, and are indicated by the set of unordered pairs $\mathcal{E} := \{\{i, j\} \ : i, j \in \mathcal{V},\, i \neq j\}$. This will be the assumed type, unless otherwise stated.

Furthermore, graphs can also be **weighted** or **unweighted**. In weighted graphs, edges are linked to an integer or real value y_{ij} representing the strength or frequency of the association. For instance, in plant-pollinator networks, the number of visits of a pollinator to a specific plant might be more than one, carrying therefore higher weight. In unweighted graphs, edges simply indicate the presence of a relationship and $y_{ij} \in \{0, 1\}$.

Another distinction to be made is between **unipartite** and **multipartite** graphs. The unipartite case is when all nodes belong to the same group and any pair of nodes might form an edge. On the contrary, multipartite graphs have multiple classes of nodes, and edges are allowed only between nodes of different groups. An example of a **bipartite** structure is given again by plant-pollinator networks, where edges are visits from pollinators to plants.



(a) Unipartite Graph (b) Bipartite Graph

Figure 1.3: Examples of unipartite and bipartite graph structures. Nodes B_1 and B_2 belong to a different group than node A.

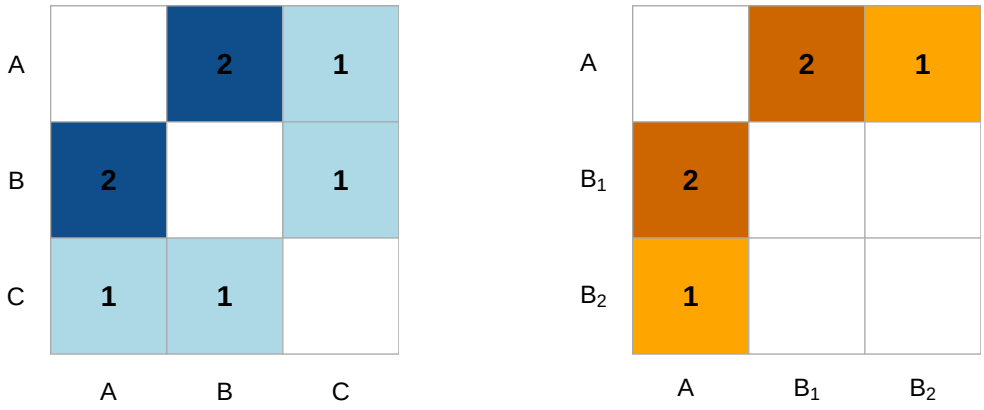
A common way to represent graphs is via *adjacency* matrices. The adjacency matrix $\mathbf{Y} = (Y_{ij})_{i,j \in \mathcal{V}}$ of a graph $\mathcal{Y}$ is a matrix of dimensions $v(\mathcal{Y}) \times v(\mathcal{Y})$, with each entry $Y_{ij} = y_{ij}$ being the weight of the specific edge between node i and j.

Adjacency matrices of unipartite undirected graphs are symmetric, therefore $Y_{ij} = Y_{ji}$, while for directed graphs this need not be the case. Moreover, if *self loops* are allowed, $Y_{ii} \neq 0$, meaning a node can be connected to itself. For bipartite graphs, adjacency matrices can be viewed as rectangular with dimensions $v_A(\mathcal{Y}) \times v_B(\mathcal{Y})$, where $v_A(\mathcal{Y})$ and $v_B(\mathcal{Y})$ are the number of nodes in the 2 groups.

Another frequent concept discussed in network analysis is **node degree**, which is the number of edges connected to a specific node. Often times, it is of interest to study the degree distribution of a network, both on its own and by comparing it with other networks to characterise potential differences.

1.2.2 Network Models and Exchangeability

From a statistical perspective, an observed network can be viewed as a realisation from a generative stochastic process. This process, in turn, defines a random variable $\mathcal{Y}$ taking values in the space of possible graphs, following a specific probability distribution. We call this random variable a **random graph**.



(a) Adjacency matrix of unipartite graph.

(b) Adjacency matrix of bipartite graph.

Figure 1.4: Examples of adjacency matrices. The adjacency matrix in (a) represents the unipartite graph in Figure 1.3, while the matrix in (b) the bipartite graph in the same figure. Empty cells are entries of weight zero, where edges are absent. The bipartite adjacency matrix could be reduced to a matrix of dimensions 1×2.

To model observed data, one can work with network models, which simply specify the distribution of such random variable. After observing a specific network, one can proceed to infer the parameters of the model, if present, and perform standard statistical analysis thereafter.

However, it is crucial to understand what are the typical structures observed under a given network model, as this provides valuable insights on its properties, and therefore when its use is most appropriate or when the model is actually misspecified.

In general, properties of statistical models are tied to their assumptions. Network models are no exception, and different assumptions lead to properties which might be more or less desired. In what follows, we review some of the possible modelling assumptions.

Node Exchangeability

Historically, network models have mainly been developed assuming **node exchangeability**, where nodes are treated as the fundamental units of observation. The node exchangeability assumption implies that the probability distribution of a random graph is invariant to permutation of its vertices, implicitly stating that node labels do not carry additional, relevant information. An illustration of node permutation can be found in Figure 1.5.

Definition 8 (Node exchangeable random graph)**.** Let $\mathcal{Y}$ be a random graph with a fixed set of nodes $\mathcal{V}$, and let $\mathbf{Y} = (Y_{ij})_{i,j\in\mathcal{V}}$ be its random adjacency matrix. The random graph is node exchangeable if, for any permutation π of the node set $\mathcal{V}$,

$$(Y_{ij})_{i,j\in\mathcal{V}} \overset{d}{=} (Y_{\pi(i)\,\pi(j)})_{i,j\in\mathcal{V}},$$

where $\overset{d}{=}$ denotes equality in distribution.

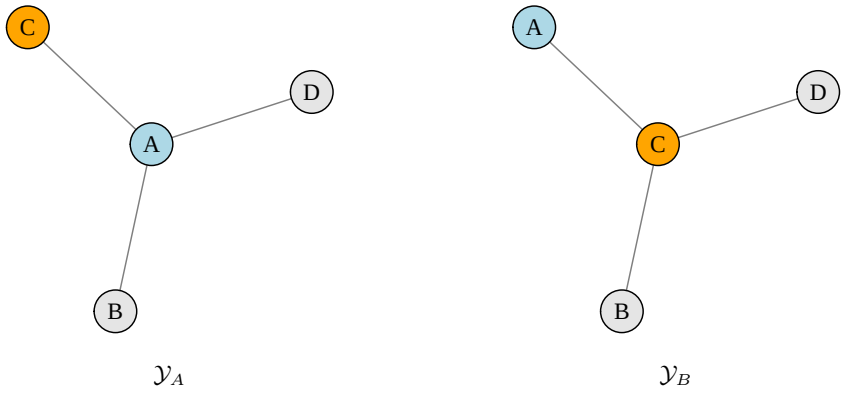


Figure 1.5: Example of node permutation. $\mathcal{Y}_B$ is obtained by permuting nodes A and C of $\mathcal{Y}_A$. Under node exchangeability, the two graphs have the same probability.

A notable example of node exchangeable models are the family of Stochastic Block Models (Holland et al., 1983; Snijders and Nowicki, 1997; Wang and Wong, 1987), which in the simplest formulation assume the existence of K latent groups of nodes, and cluster them based on their number of edges.

Edge Exchangeability

Another approach is to assume **edge exchangeability**, introduced simultaneously in the works of Crane and Dempsey (2018) and Cai et al. (2016).

In this framework, the graph is constructed from a sequence of n observed edges $Y_1, \ldots, Y_n$, which constitute the fundamental units of observation. The assumption states that the probability distribution of the random graph is invariant to permutation of the order of arrival of the edges, which is therefore assumed to not carry any additional information. An illustration of this idea can be found in Figure 1.6.

Definition 9 (Edge exchangeable random graph)**.** Let $\mathcal{Y} = (\mathcal{V}, \mathcal{E})$ be a random graph with sequence of random edges $Y_1, \ldots, Y_n$. The random graph is edge exchangeable if, given any permutation π of the indices $\{1, \ldots, n\}$,

$$(Y_1, \ldots, Y_n) \stackrel{d}{=} (Y_{\pi(1)}, \ldots, Y_{\pi(n)}).$$

1.2.3 Network Models and Sparsity

Sparsity is a property often encountered with real-life networks. Intuitively, sparse networks present relatively few edges and their nodes are less interconnected. An illustration comparing dense and sparse graphs can be found in Figure 1.7.

To be more precise, sparsity is an asymptotic property of graph sequences whose nodes and edges are assumed to be increasing. The attribute "sparse" is then used to refer to graphs assumed to belong to a sparse graph sequence, according to Definition 10.

Definition 10 (Sparsity)**.** Consider a unipartite graph sequence $(\mathcal{Y}_n)_{n\in\mathbb{N}}$, with each $\mathcal{Y}_n$ a random graph with finite vertex set $\mathcal{V}_n$ and finite edge set $\mathcal{E}_n$. Assume the sequence is *growing*, that is $\mathcal{V}_n \subseteq \mathcal{V}_{n+1}$ and

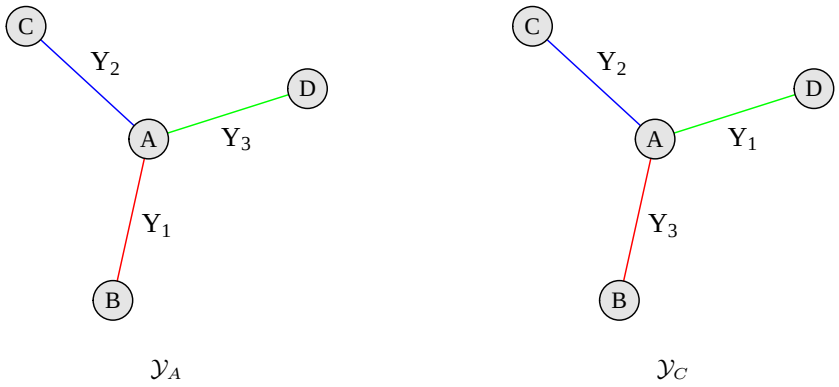
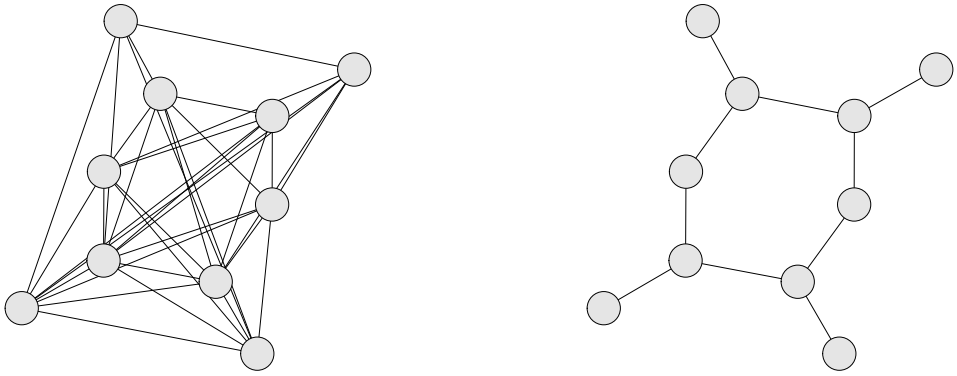


Figure 1.6: Example of edge permutation. $\mathcal{Y}_C$ is obtained by permuting the order of arrival of edges in $\mathcal{Y}_A$. Under edge exchangeability, the two graphs have the same probability.



(a) Dense Graph

(b) Sparse Graph

Figure 1.7: Example of Dense (a) and Sparse (b) graphs. Sparsity occurs when the number of edges is relatively low.

$\mathcal{E}_n \subseteq \mathcal{E}_{n+1}$.

Let $d(\mathcal{Y}_n) = \frac{e(\mathcal{Y}_n)}{v(\mathcal{Y}_n)^2}$ be the **density** of the graph.

The sequence is said to be **sparse** if:

$$d(\mathcal{Y}_n) \xrightarrow[n\to\infty]{\text{a.s.}} 0.$$

Equivalently, we say the sequence is sparse if $e(\mathcal{Y}_n) = o(v(\mathcal{Y}_n)^2)$, i.e. the number of edges grows **sub-quadratically** with the number of nodes.

Conversely, the graph sequence is **dense** if $e(\mathcal{Y}_n) = \Omega(v(\mathcal{Y}_n)^2)$, i.e. the number of edges grows **at least quadratically** with the number of nodes.

Given that many real networks exhibit sparsity, it is natural to look for models that inherently capture this characteristic. As we anticipated, such a property heavily depends on the assumptions made by the model. Indeed, following from *Aldous-Hoover theorem* (Aldous (1981), Hoover (1979)), it can be shown that *growing*

graph sequences generated under the assumption of node exchangeability are either empty or dense almost surely (Orbanz and Roy, 2013). This makes node exchangeable models unsuited to model sparse networks by design.

Such limitations of node exchangeability led to the development of different modelling assumptions. One such example can be found in Caron and Fox (2017), where they have developed sparse models under the assumption of Kallenberg exchangeability (Kallenberg, 2005).

However, also the **edge exchangeability** assumption allows for the construction of sparse network models, following from the work of Crane and Dempsey (2018), Cai et al. (2016). Due to its simplicity, it makes edge exchangeable models a particularly appealing tool.

1.3 Bayesian Nonparametric Network Models

Some of the proposed edge exchangeable models are strongly connected to the theory of Bayesian Nonparametrics, where BNP process priors define the distribution of the random edges, and therefore of the random graph. For example, the *Hollywood model* (Crane and Dempsey, 2018) can be seen as an extension to unipartite *hypergraphs* (graphs where weighted edges are allowed to connect more than two nodes) of a model using a Pitman-Yor process. The Dirichlet Process version for classic graphs (non hypergraphs) was treated in Williamson (2016), referred to in that work as the *Dirichlet Network Distribution*.

The main idea of how the PY process is used in the unipartite case is described as follows and further illustrated in Figure 1.8. We will refer to this construction as the Pitman-Yor Product Model (PYPM).

1.3.1 Unipartite Pitman-Yor Product Model

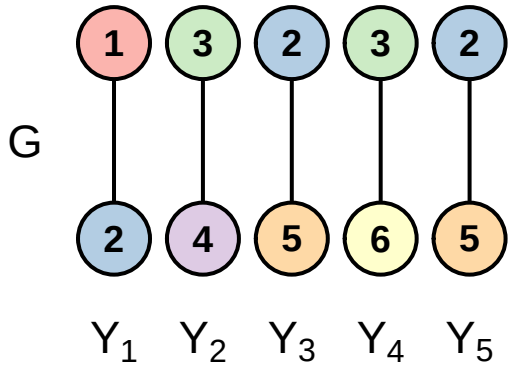


Figure 1.8: Illustration of the edge sampling process for the unipartite Pitman-Yor Product Model. Every edge connects pairs of values drawn from the Pitman-Yor process, labelled based on the order of arrival.

To generate a unipartite edge exchangeable network from a Pitman-Yor process, one can see each random edge Y_t as a pair of random nodes X_t and Z_t drawn from a product probability measure $G \times G$, and then put a PY prior on G. The hierarchical formulation is:

$$G \sim \mathrm{PY}(\alpha, \sigma, G_0),$$
$$Y_t = (X_t, Z_t) \mid G \overset{\text{i.i.d.}}{\sim} G \times G \quad \text{for } t = 1, \dots, n.$$

The generative process can be thought of as sampling the pairs from the CRP construction of the PY process (see Definition 6). At step t, the edge Y_t will be either a pair of already observed nodes or new ones, with probabilities depending on the state of the network. For simplicity, one can ignore the real valued labels of the nodes coming from G_0, and instead use their order of arrival to identify them.

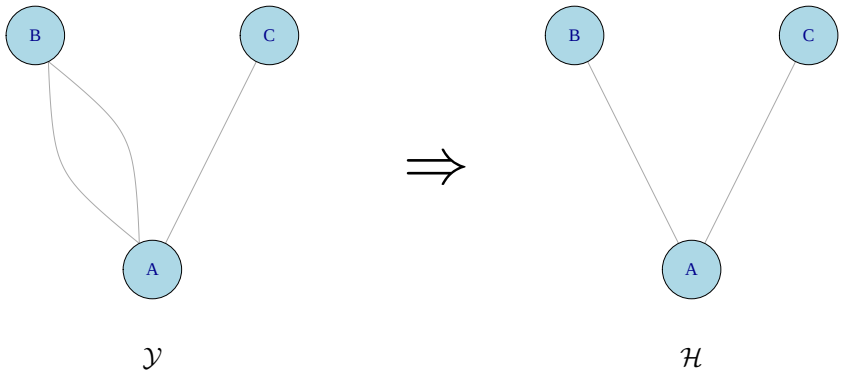


Figure 1.9: Example of unweighted graphs. $\mathcal{H}$ is obtained by removing the multiple edges of $\mathcal{Y}$.

Sparsity properties of the PYPM in the unipartite case have been studied by Crane and Dempsey (2018). The authors investigated both weighted networks and their **unweighted** version, where we discard multiple edges and retain only the topological structure. An example of this process is illustrated in Figure 1.9. Their work shows that the PYPM produces sparse graph sequences $(\mathcal{Y}_n)_{n\in\mathbb{N}}$ for $0.5 < \sigma < 1$. Moreover, it produces sparse unweighted graph sequences $(\mathcal{H}_n)_{n\in\mathbb{N}}$ for $0 < \sigma < 1$.

This model will serve as foundation for the bipartite extension proposed in the next chapter.

Chapter 2

Bayesian Nonparametric Model for Bipartite Networks

The goal of this chapter is to extend the previously developed theory from the unipartite to the bipartite case. More specifically, an extension of the Pitman-Yor Product Model to deal with bipartite networks will be presented and studied. Particular attention will be given to the sparsity property for unweighted networks generated by such model, of which we provide the proof. To conclude, simulation studies will complement the theory providing further insights on the model properties.

Throughout the chapter, we consider the bipartite graph $\mathcal{Y} = (\mathcal{V}_A, \mathcal{V}_B, \mathcal{E})$, with node sets of the two groups $\mathcal{V}_A$ and $\mathcal{V}_B$. We recall edges in $\mathcal{E}$ for bipartite graphs are only allowed to connect nodes of different groups.

2.1 Bipartite Pitman-Yor Product Model

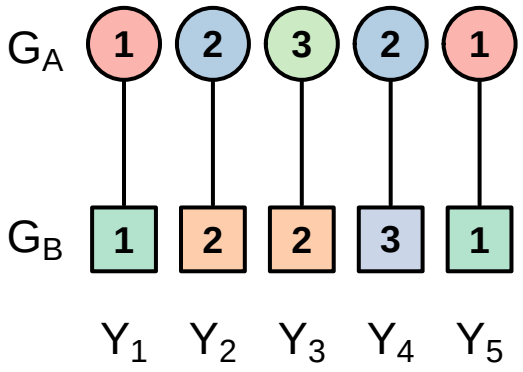


Figure 2.1: Illustration of the edge sampling process for the bipartite Pitman-Yor Product Model. Every edge contains pairs of unique values from the two Pitman-Yor processes, labelled based on the order of arrival. Different shapes indicate different groups.

The Pitman-Yor Product Model described in subsection 1.3.1 can be easily extended to generate bipartite networks. A similar illustration to the one already provided can be found in Figure 2.1.

The simplest approach is to view edges Y_t as pairs of nodes X_t^A and X_t^B drawn from the product probability measure $G_A \times G_B$, where each measure follows its own PY process whose parameters are potentially different. Below, we see the hierarchical formulation of the model:

$$\begin{aligned} &G_A \sim \mathrm{PY}(\alpha_A, \sigma_A, G_0)\,,\, G_B \sim \mathrm{PY}(\alpha_B, \sigma_B, G_0), \\ &Y_t = (X_t^A, X_t^B) \mid G_A, G_B \overset{\text{i.i.d.}}{\sim} G_A \times G_B \quad \text{for } t = 1, \ldots, n. \end{aligned}$$

As before, real valued node labels from G_0 can be ignored, setting them to be the order of arrival of the unique value. The sampling process can still be seen as drawing from the CRP construction, but this time done separately for the two groups.

A crucial assumption of the model is the independence between nodes of different groups, since the product of the probability measures was taken. This implies that for future edges, the presence or absence of specific nodes of group B in the data does not influence the probability of observing a given node of group A, and vice versa.

2.2 Sparsity Property

The main question of interest would be to verify if the Pitman-Yor Product Model is able to produce sparse bipartite graph sequences. In what follows, we will specifically treat unweighted graphs $\mathcal{H}_n$, ignoring weights and focusing only on the topological structure, similarly to the idea shown in Figure 1.9.

To extend the notion of sparsity in unweighted bipartite graphs we define the density statistics, also known in the ecological literature as *connectance*, to be the following quantity:

$$d(\mathcal{H}_n) = \frac{e(\mathcal{H}_n)}{v_A(\mathcal{H}_n)\, v_B(\mathcal{H}_n)}.$$

Therefore, density in unweighted bipartite graphs reflects the same idea as in the unipartite case, that is to divide the number of (unique) edges by the maximum possible number of connections in the graph.

Definition 11 (Sparsity in unweighted bipartite graphs)**.** Let $d(\mathcal{H}_n)$ be the density of an unweighted bipartite graph. The growing graph sequence $(\mathcal{H}_n)_{n\in\mathbb{N}}$, where $n = e(\mathcal{H}_n)$, is said to be sparse if:

$$d(\mathcal{H}_n) \xrightarrow[n\to\infty]{\text{a.s.}} 0.$$

The goal is to show that unweighted bipartite graphs generated by the PYPM also exhibit sparsity in the sense of the previous definition. This is what is claimed in the following proposition.

Proposition 1 (Sparsity for the Bipartite Pitman-Yor Product Model)**.** Let $(\mathcal{Y}_n)_{n\in\mathbb{N}}$ be the bipartite graph sequence generated by the Pitman-Yor Product Model of section 2.1. The induced unweighted bipartite sequence $(\mathcal{H}_n)_{n\geq 1}$ is sparse for either σ_A or σ_B in $(0, 1)$.

This result can be considered an extension of the sparsity property already proved by Crane and Dempsey (2018). The proof itself overlaps substantially, requiring minor adjustments at the beginning to deal with the bipartite structure.

Proof. We want to show:

$$\limsup_{n\to\infty} d(\mathcal{H}_n) = \limsup_{n\to\infty} \frac{e(\mathcal{H}_n)}{v_A(\mathcal{H}_n)v_B(\mathcal{H}_n)} \overset{\text{a.s.}}{=} 0.$$

Consider the number of edges $e(\mathcal{H}_n)$. We will denote the minimum operator as $a \wedge b = \min(a, b)$.

Let $N_{k,A}(\mathcal{Y}_n)$ be the number of nodes in group A with degree k in the original graph. $N_{k,B}(\mathcal{Y}_n)$ is the analogous for group B.

Note that the number of vertices in each group remains unchanged after removing edge multiplicities:

$$v_A(\mathcal{H}_n) = v_A(\mathcal{Y}_n), \quad v_B(\mathcal{H}_n) = v_B(\mathcal{Y}_n).$$

Moreover, recall that the degree of a node, let's say in group A for the original graph, is the number of edges containing that node. We will call it $\deg_{\mathcal{Y}_n}(a)$.

Since in $\mathcal{H}_n$ the degree of a node in one group cannot exceed the number of nodes in the other, and at the same time cannot be bigger than the degree of the same node in $\mathcal{Y}_n$, we can state that:

$$\begin{aligned} e(\mathcal{H}_n) = \sum_{a\in\mathcal{V}_A} \deg_{\mathcal{H}_n}(a) &\leq \sum_{a\in\mathcal{V}_A} (\deg_{\mathcal{Y}_n}(a) \wedge v_B(\mathcal{H}_n)) \\ &= \sum_{k=1}^{\infty} \left(\sum_{a:\deg_{\mathcal{Y}_n}(a)=k} (k \wedge v_B(\mathcal{H}_n)) \right) \\ &= \sum_{k=1}^{\infty} (k \wedge v_B(\mathcal{H}_n)) N_{k,A}(\mathcal{Y}_n). \end{aligned}$$

Symmetrically, the same can be said with group B:

$$e(\mathcal{H}_n) = \sum_{b\in\mathcal{V}_B} \deg_{\mathcal{H}_n}(b) \leq \sum_{k=1}^{\infty} (k \wedge v_A(\mathcal{H}_n))\ N_{k,B}(\mathcal{Y}_n).$$

Without loss of generality, consider the first inequality for group A.

For any $K \geq 1$ we can show that:

$$\begin{aligned}
\limsup_{n\to\infty} d(\mathcal{H}_n) &\leq \limsup_{n\to\infty} \sum_{k=1}^{K} \frac{(k \wedge v_B(\mathcal{H}_n))\, N_{k,A}(\mathcal{Y}_n)}{v_B(\mathcal{H}_n)\, v_A(\mathcal{H}_n)} + \limsup_{n\to\infty} \sum_{k=K+1}^{\infty} \frac{(k \wedge v_B(\mathcal{H}_n))\, N_{k,A}(\mathcal{Y}_n)}{v_B(\mathcal{H}_n)\, v_A(\mathcal{H}_n)} \\
\text{(Step 2)} &\leq \sum_{k=1}^{K} \limsup_{n\to\infty} \frac{k \wedge v_B(\mathcal{H}_n)}{v_B(\mathcal{H}_n)} + \limsup_{n\to\infty} \sum_{k=K+1}^{\infty} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n)} \\
&\leq \limsup_{n\to\infty} \sum_{k=K+1}^{\infty} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n)}.
\end{aligned}$$

Step 2 comes from $\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n)} \leq 1$ and $\frac{k \wedge v_B(\mathcal{H}_n)}{v_B(\mathcal{H}_n)} \leq 1$.

Moreover, since $v_B(\mathcal{H}_n) \xrightarrow{a.s.} \infty$ (Pitman, 2006), we can also state:

$$\limsup_{n\to\infty} \frac{k \wedge v_B(\mathcal{H}_n)}{v_B(\mathcal{H}_n)} = 0.$$

In conclusion, we have the following upper bound:

$$\limsup_{n\to\infty} d(\mathcal{H}_n) \leq \limsup_{n\to\infty} \sum_{k=K+1}^{\infty} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n)}. \tag{2.1}$$

The last upper bound is the same as the one in equation (27) of Crane and Dempsey (2018).

In general, by Pitman (2006) we have that under any PY(α_A, σ_A) with $\sigma_A \in (0,1)$, for every $k \geq 1$:

$$\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} \xrightarrow[n\to\infty]{\text{a.s.}} p_{\sigma_A}(k) := \sigma_A \frac{\Gamma(k - \sigma_A)}{k!\,\Gamma(1 - \sigma_A)}.$$

This implies that for every $\epsilon, \delta > 0$ and fixed $k \geq 1$, there exists $R = R(\epsilon, \delta, k)$ such that:

$$\Pr\left(\left| \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - p_{\sigma_A}(k) \right| < \epsilon \,, \forall n > R \right) \geq 1 - \delta.$$

Now fix $K \geq 1$. For each $1 \leq k \leq K$, let $R_k = R(\epsilon/K, \delta/K, k)$ be the index after which we have:

$$\Pr(A_k) = \Pr\left(\left| \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - p_{\sigma_A}(k) \right| < \frac{\epsilon}{K} \,, \forall n > R_k \right) \geq 1 - \frac{\delta}{K}.$$

We want to consider the probability of the event A_k being true for every k, that is:

$$\begin{aligned}
\Pr\left(\bigcap_{k=1}^{K} A_k\right) &= 1 - \Pr\left(\left(\bigcap_{k=1}^{K} A_k\right)^c\right) \\
&= 1 - \Pr\left(\bigcup_{k=1}^{K} A_k^c\right) \quad \text{(by De Morgan's laws)} \\
&\geq 1 - \sum_{k=1}^{K} \Pr(A_k^c) \quad \text{(by Boole's inequality)} \\
&\geq 1 - \sum_{k=1}^{K} \frac{\delta}{K} = 1 - \delta.
\end{aligned}$$

Where De Morgan's laws and Boole's inequality are standard results in probability theory (see Billingsley (1979)).

Defining $R^* = \max_{1 \leq k \leq K} R_k$, the previous statement can be made more explicit by saying:

$$\Pr(\bigcap_{k=1}^{K} A_k) = \Pr\left(\left|\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - p_{\sigma_A}(k)\right| < \frac{\epsilon}{K}, \forall n > R^*, \forall k = 1, \ldots, K\right) \geq 1 - \delta.$$

Since every difference in absolute value is bounded by $\frac{\epsilon}{K}$:

$$\begin{aligned}
\left|\sum_{k=1}^{K} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - \sum_{k=1}^{K} p_{\sigma_A}(k)\right| &\leq \sum_{k=1}^{K} \left|\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - p_{\sigma_A}(k)\right| \quad \text{(by triangle inequality)} \\
&< K \cdot \frac{\epsilon}{K} = \epsilon.
\end{aligned}$$

Moreover, note that:

$$\sum_{k=K+1}^{\infty} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} = 1 - \sum_{k=1}^{K} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))}.$$

The last statement means that the differences in absolute values of the tail sums can also be bounded by the same quantity:

$$\begin{aligned}
\left|\sum_{k=K+1}^{\infty} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - \sum_{k=K+1}^{\infty} p_{\sigma_A}(k)\right| &= \left|1 - \sum_{k=1}^{K} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - \left(1 - \sum_{k=1}^{K} p_{\sigma_A}(k)\right)\right| \\
&= \left|\sum_{k=1}^{K} \frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))} - \sum_{k=1}^{K} p_{\sigma_A}(k)\right| \\
&< \epsilon.
\end{aligned}$$

Therefore we also have the following probability bound:

$$\Pr\left(\left|\sum_{k=K+1}^{\infty}\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))}-\sum_{k=K+1}^{\infty}p_{\sigma_A}(k)\right|<\epsilon\,,\forall n>R^*\right)\geq 1-\delta. \tag{2.2}$$

Let $\bar{p}_{\sigma_A,K}=\sum_{k=K+1}^{\infty}p_{\sigma_A}(k)$ and combine upper bound 2.1 with the previous statements to conclude that, for every $K\geq 1$:

$$\begin{aligned}
&\Pr\left(\limsup_{n\to\infty} d(\mathcal{H}_n)<\bar{p}_{\sigma_A,K}+\epsilon\right)\\
\text{(by upper bound 2.1)}\geq\ &\Pr\left(\limsup_{n\to\infty}\sum_{k=K+1}^{\infty}\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))}<\bar{p}_{\sigma_A,K}+\epsilon\right)\\
\text{(Inclusion step)}\geq\ &\Pr\left(\left|\sum_{k=K+1}^{\infty}\frac{N_{k,A}(\mathcal{Y}_n)}{v_A(\mathcal{H}_n))}-\bar{p}_{\sigma_A,K}\right|<\epsilon\,,\,\forall n>R^*\right)\\
\text{(by bound 2.2)}\geq\ &1-\delta.
\end{aligned}$$

The inclusion step is true because if the sequence stays within $\pm\epsilon$ of $\bar{p}_{\sigma_A,K}$ (Event A), the limsup will necessarily be below $\bar{p}_{\sigma_A,K}$ (Event B). Since $A\subseteq B$, then $\Pr(B)\geq\Pr(A)$.

Since the probability bound just derived holds not only for every $K\geq 1$ but also for all $\epsilon,\delta>0$, and since:

$$\bar{p}_{\sigma_A,K}=\frac{\Gamma(K+1-\sigma_A)}{\Gamma(K+1)\Gamma(1-\sigma_A)}\xrightarrow{K\to\infty}0,$$

we can conclude:

$$\limsup_{n\to\infty} d(\mathcal{H}_n)\overset{\text{a.s.}}{=}0.$$

Hence, the sequence $(\mathcal{H}_n)_{n\in\mathbb{N}}$ is sparse almost surely. □

Remark 1. It is not necessary to have both groups modelled by a PY for the resulting bipartite graph sequence $(\mathcal{H}_n)_{n\in\mathbb{N}}$ to be sparse. As long as both processes produce an infinite number of unique values in the limit (true also for the DP), and at least one is a PY process, the proof holds thanks to the symmetry of the first argument.

Remark 2. Having both groups modelled by a DP process does not guarantee the resulting bipartite graph sequence $(\mathcal{H}_n)_{n\in\mathbb{N}}$ will be sparse. This follows from the crucial step using the limit $p_{\sigma_A}(k)$ valid for $\sigma_A\in(0,1)$.

2.3 Simulation studies

In this section we present simulation studies to support the previously made claims. The goal is to provide empirical evidence showing how (and when) the average density of unweighted networks simulated from the model tends to zero with increasing number of edges.

We first analysed the model in two configurations: the first is the symmetric case with $G_A\sim\text{PY}(5,\sigma,G_0)$ and $G_B\sim\text{PY}(5,\sigma,G_0)$, where $\sigma_A=\sigma_B=\sigma$, and the second is the asymmetric case with $G_A\sim\text{DP}(5,G_0)$

and $G_B \sim \text{PY}(5, \sigma, G_0)$, where the first group has $\sigma_A = 0$. For both configurations, we let the discount parameter vary and kept $\alpha_A = \alpha_B = 5$. This follows from the fact that the discount parameter is the one shown to induce sparsity.

We examined the four cases with $\sigma = \{0, 0.25, 0.5, 0.75\}$, and for each parameter choice we simulated networks of increasing number of edges $n = e(\mathcal{Y}_n)$. More specifically, at every n we drew 100 independent networks with n edges from the given model, and computed the average density $d(\mathcal{H}_n)$ alongside its 95% confidence intervals. The results are shown in Figure 2.2.

In the left plot of Figure 2.2a, the average density with the first model clearly goes to zero with $\sigma > 0$, while with $\sigma = 0$ the density appears to be increasing, providing initial evidence in favour of Remark 2. Moreover, it appears that increasing values of the discount parameter (darker colours) lead to faster convergence. The right plot shows the same curves in log-log scale, where the strong linear behaviour might suggest that the density statistics follows a power-law relationship of the type:

$$d(\mathcal{H}_n) = \gamma\, n^{\beta}.$$

This claim follows from the intuition that, assuming such a functional form, the expected behaviour taking logs on both sides is precisely linear:

$$\log d(\mathcal{H}_n) = \log \gamma \, + \beta \log n$$

The slopes of the lines, the β coefficients, are estimates of the rate of change of the density statistics w.r.t. the number of edges. Therefore, computed slopes using OLS reported on top of the figure, confirm the visual intuition suggesting faster convergence to zero for higher discount parameters.

In Figure 2.2b, we observe the same behaviour in the asymmetric model, reinforcing the claim made in Remark 1. However, there are some differences compared to the previous result. In particular, the density statistics converges to zero much slower than before, and the variability seems higher due to wider confidence bands. Moreover, the linear approximation in the log-log scale seems poorer for lower sizes.

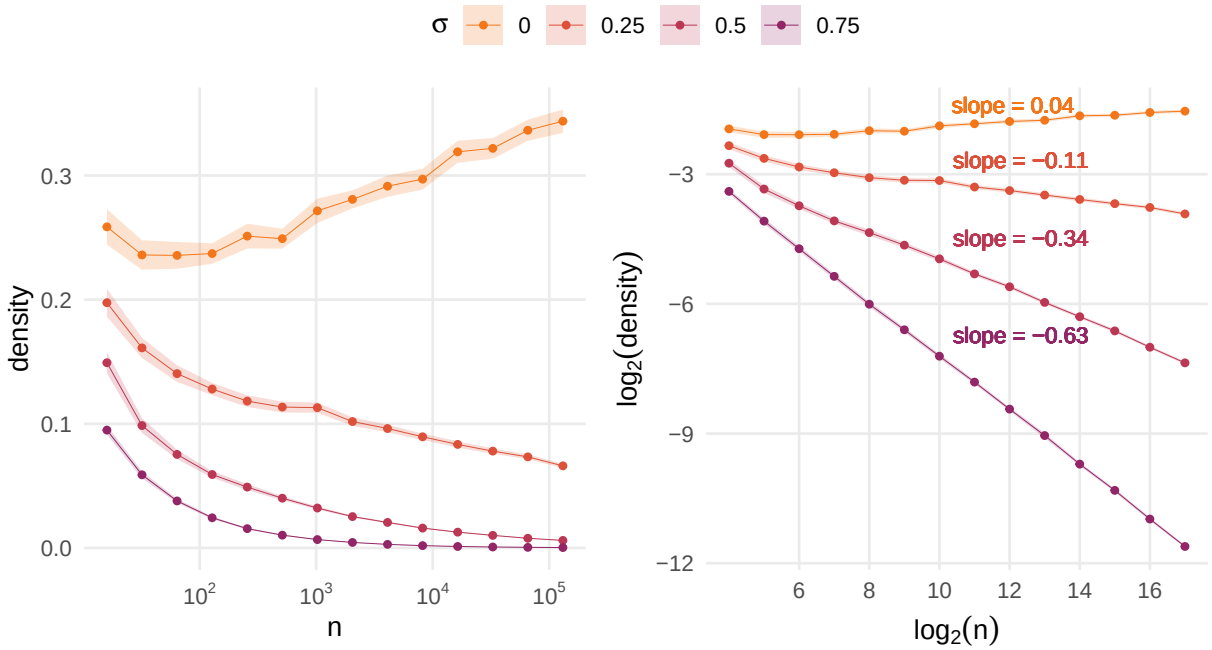
As mentioned before, the simulations suggest that when we model both groups with only the Dirichlet Process, instead of the Pitman-Yor process, we do not reach sparsity. To investigate further this specific situation, we carried out additional simulation studies. In particular, we chose again identical $\alpha_A = \alpha_B = \alpha$ for both groups, but this time we let it vary. This was to ensure the specific choice of $\alpha = 5$ taken before was not just a special, unlucky case. The goal of the simulation is therefore to empirically study the behaviour of the density statistics for different α values, as the number of edges increases. The setup is similar, but now we explore much bigger networks. To reduce the computational cost, instead of sampling new 100 networks at every n, we simply sample once 100 networks with number of edges $n = 10^7$, and retrieve the average density statistics at specific stages of the sampling process coinciding with the number of edges we want to study. This makes the results slightly less reliable than before, as the network densities at each size are dependent on the results computed before. However, having sampled many networks guarantees a sufficiently high degree of variability to gain enough intuition on the underlying behaviour.

In Figure 2.3, we see the results of this simulation study. The plot is organised with the α values on the x

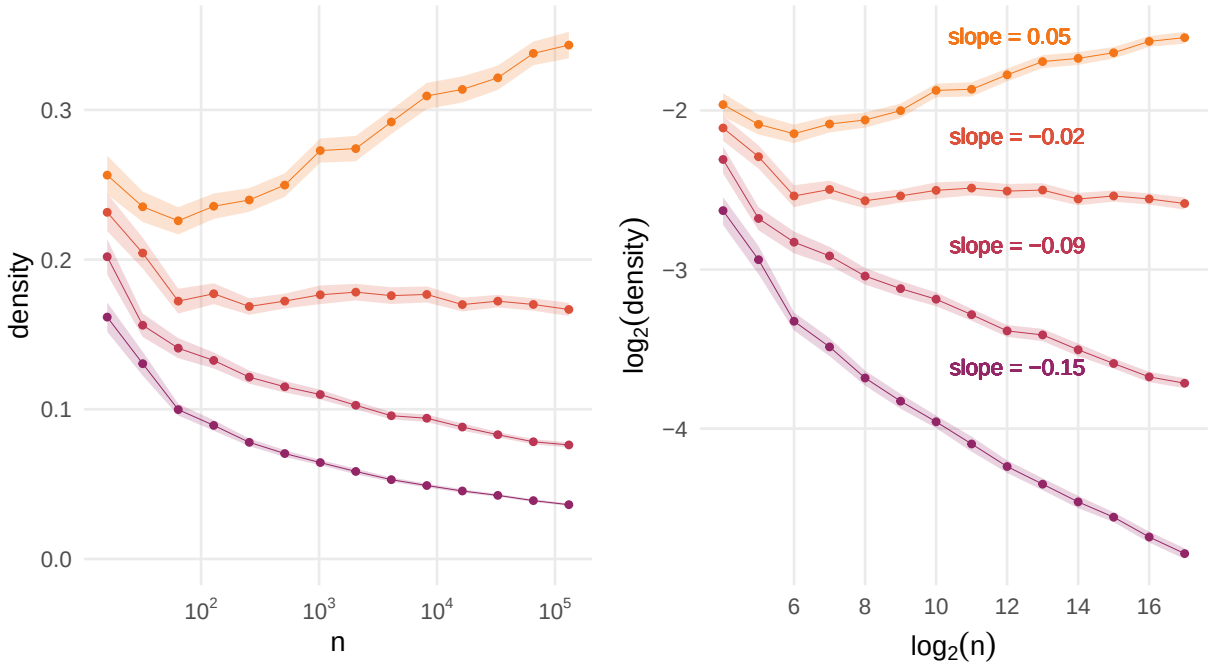
axis, and different colours now represent different sample sizes.

We see that for $\alpha < 1.5$, the density starts from high values with few edges and decreases with more and more connections. However, it does not reach zero. After $\alpha > 1.5$, the behaviour of the density statistics changes, observing essentially what we observed in the previous simulations with $\alpha = 5$: the density statistics starts at low values and increases with more and more connections entering the networks. This inconsistent behaviour based on the α parameter chosen shows the relevance of the simulation study with different concentration parameters.

To conclude, the insights we gained from the simulation studies confirm Proposition 1 and Remark 1. Moreover, as suggested by Remark 2, the DP alone does not guarantee sparsity.



(a) Simulation with $G_A \sim \text{PY}(5, \sigma, G_0), G_B \sim \text{PY}(5, \sigma, G_0)$.



(b) Simulation with $G_A \sim \text{DP}(5, G_0), G_B \sim \text{PY}(5, \sigma, G_0)$.

Figure 2.2: Average estimated density and 95% confidence intervals over different sizes. Natural (left) and log-log scale (right). Slope in log-log scale estimated with OLS.

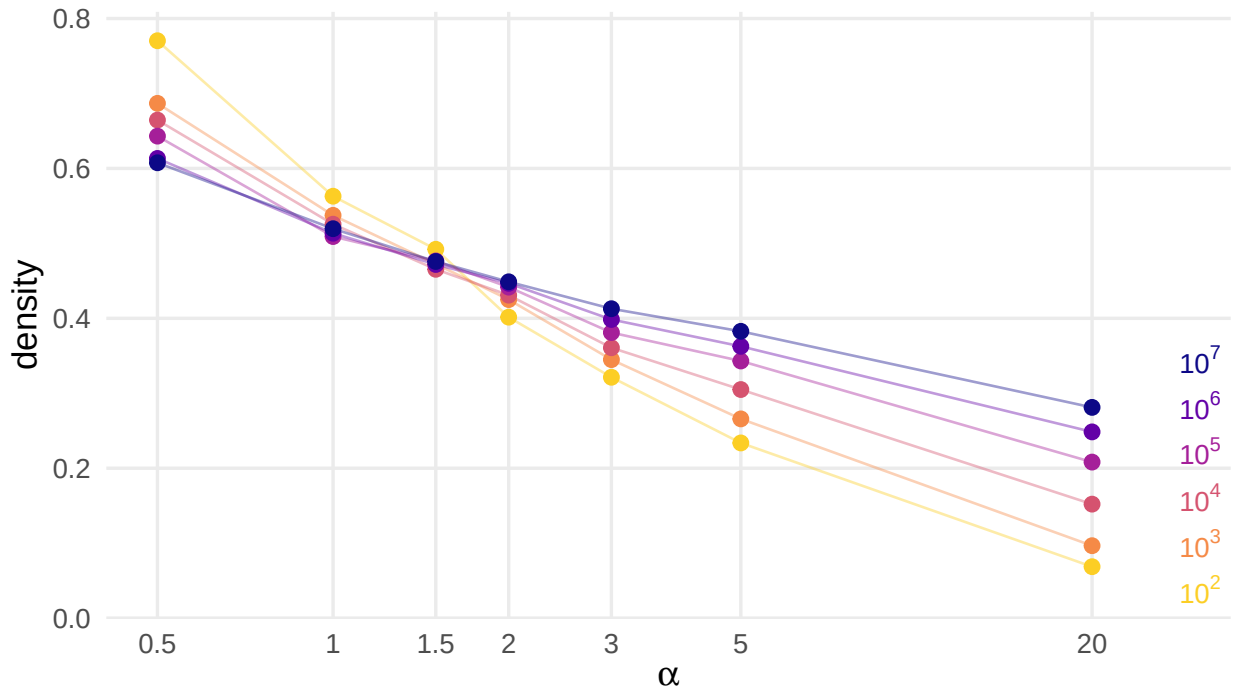


Figure 2.3: Average estimated density over varying α values with increasing networks size, represented by different colours. Simulation results for 100 networks with $G_A \sim \mathrm{DP}(\alpha, G_0), G_B \sim \mathrm{DP}(\alpha, G_0)$.

Chapter 3

Bayesian Model Estimation and Evaluation

The goal of this chapter is to show how to perform Bayesian estimation of the Pitman-Yor Product Model on bipartite networks. In the first section, we define the inferential framework and review related methodologies. The second section compares estimation results on artificial data under two different simulation setups, with the goal of showing convergence of the posterior distributions to the true parameter values. To conclude, an application to real network data is showcased in the last section.

3.1 Inferential Framework and Methodologies

3.1.1 Model Likelihood

Recall that we are considering bipartite networks $\mathcal{Y} = (\mathcal{V}_A, \mathcal{V}_B, \mathcal{E})$. For convenience, we also recall the model formulation, where we assume an exchangeable sequence of edges coming from the product probability measure $G_A \times G_B$ forms the bipartite network $\mathcal{Y}$:

$$
\begin{aligned}
&G_A \sim \mathrm{PY}(\alpha_A, \sigma_A, G_0)\,, \quad G_B \sim \mathrm{PY}(\alpha_B, \sigma_B, G_0), \\
&Y_t = (X_t^A, X_t^B) \mid G_A, G_B \overset{\text{i.i.d.}}{\sim} G_A \times G_B \quad \text{for } t = 1, \ldots, n.
\end{aligned}
$$

Inference on the Pitman-Yor Product Model for bipartite networks requires estimation of the parameters of the independent probability measures assumed for the groups.

For the set of parameters, sometimes it will be simpler to refer to $\theta = (\alpha_A, \sigma_A, \alpha_B, \sigma_B)$. For a sequence of random variables, we will use the compact notation $Y_{1:n}$.

The first step is to define the joint likelihood function. Thanks to the independence assumption between the groups of nodes, the joint likelihood can be decomposed into the following product:

$$L(Y_{1:n};\theta) = L(X^A_{1:n};\alpha_A,\sigma_A) \times L(X^B_{1:n};\alpha_B,\sigma_B).$$

The likelihood of observations (the nodes X^A_t and X^B_t) coming from a $\mathrm{PY}(\alpha,\sigma,G_0)$ is completely determined by their partition structure. This means the total sample size n and the frequency n_k of every unique value, indexed by $k = 1,\dots,K_n$, constitute sufficient statistics for the parameters of the process.

Therefore, the likelihood can be expressed through the EPPF (see Definition 7):

$$p_n(n_1,\dots,n_{K_n}) = \frac{\prod_{k=1}^{K_n-1}(\alpha+k\sigma)}{(\alpha+1)_{n-1}} \prod_{k=1}^{K_n}(1-\sigma)_{n_k-1}.$$

To facilitate numerical computations, one can consider the logarithm of the EPPF. Using the connection between Gamma functions and rising factorials $(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)}$, the log EPPF is:

$$\log p_n(n_1,\dots,n_{K_n}) = \log\frac{\Gamma(\alpha+1)}{\Gamma(\alpha+n)} - K_n\log\Gamma(1-\sigma) + \sum_{k=1}^{K_n-1}\log(\alpha+k\sigma) + \sum_{k=1}^{K_n}\log\Gamma(n_k-\sigma) \tag{3.1}$$

Therefore, the joint log likelihood of a sequence of edges coming from the PYPM turns out to be:

$$\log L(Y_{1:n};\theta) = \log L(X^A_{1:n};\alpha_A,\sigma_A) + \log L(X^B_{1:n};\alpha_B,\sigma_B), \tag{3.2}$$

where each addend has the form of Equation 3.1.

3.1.2 Bayesian Model Specification

Given the likelihood structure, estimation of the parameters can be performed with either an Empirical Bayes (EB) or a fully Bayesian approach. The former would amount to maximise numerically the joint log likelihood in Equation 3.2, while the latter focuses on the joint posterior distribution of the parameters in the following hierarchical model:

$$\begin{aligned}
&\alpha_A \sim \Pi_{\alpha_A}\,, \quad \sigma_A \sim \Pi_{\sigma_A}\,, \quad \alpha_B \sim \Pi_{\alpha_B}\,, \quad \sigma_B \sim \Pi_{\sigma_B},\\
&G_A \mid \alpha_A,\sigma_A \sim \mathrm{PY}(\alpha_A,\sigma_A,G_0)\,, \quad G_B \mid \alpha_B,\sigma_B \sim \mathrm{PY}(\alpha_B,\sigma_B,G_0),\\
&Y_t = (X^A_t, X^B_t) \mid G_A, G_B \overset{\text{i.i.d.}}{\sim} G_A \times G_B \quad \text{for } t = 1,\dots,n,
\end{aligned}$$

where appropriate prior distributions are chosen satisfying the constraints on the parameters.

Theoretical guarantees of both approaches have been investigated in Franssen and van der Vaart (2022). In particular, for the discount parameter of a $\mathrm{PY}(\alpha,\sigma,G_0)$ the EB approach yields an asymptotically normal estimator, while following a fully Bayesian approach its posterior distribution converges to a Gaussian centred around the EB estimator, satisfying a Bernstein-Von-Mises type of theorem.

As mentioned in the paper, despite estimation of the α parameter being doable, it is in general of less interest

than σ, which has the key role of determining the process type. Focus on the discount parameter is further motivated by our work of the previous chapter, showing how it is the sparsity inducing parameter.

In the following discussion, we take a fully Bayesian approach, which has the advantage of allowing for automatic uncertainty quantification via inspection of the posterior distributions.

3.1.3 Prior Choice

A crucial part in Bayesian modelling is the choice of the prior distributions. In the absence of information on the parameters, uninformative or flat priors can be specified. However, one can also define weakly informative priors (Gelman et al., 2013). The idea is to have priors that are reasonably wide but not too flat, so that most of the prior mass lies within reasonable values. By doing so, the estimation process usually becomes less problematic, especially with complex models. For the estimation of the PYPM, we have opted for the following prior distributions:

$$\alpha_A \sim \text{Gamma}(3, 0.4)\,,\ \alpha_B \sim \text{Gamma}(3, 0.4),$$
$$\sigma_A \sim \text{Beta}(1, 1)\,,\ \sigma_B \sim \text{Beta}(1, 1).$$

Given this choice, the discount parameters have a uniform prior on the unit interval. On the other hand, the concentration parameters are set to have weakly informative priors on $\mathbb{R}^+$. Table A.1 contains further descriptive information on the prior distributions.

3.1.4 Bayesian Computation

Usually, the posterior distributions are not accessible in closed form, which makes them hard to analyse directly. To overcome this issue, the standard approach is to use Markov Chain Monte Carlo algorithms. The key idea is to sample from a Markov chain that has reached a stationary distribution coinciding with the joint posterior. Such samples can then be considered as valid draws from the distribution of interest and can be used to compute descriptive statistics, such as posterior mean, median and even credible intervals, by computing the quantiles within which we have the desired posterior probability.

An efficient way to carry out MCMC sampling is by using the software Stan (Stan Development Team (2024)). Stan relies on variants of Hamiltonian Monte Carlo (HMC), and its sampling efficiency makes it a well established tool. For further details, see Gelman et al. (2013). The only requirements are differentiable priors and likelihood, which makes it suited to perform inference for the PYPM. Stan can be used by writing the model in its native language, and then by compiling it using built-in packages such as *rstan* for the R programming language. Models with non standard likelihoods, such as for the PYPM, can be coded up with custom functions. An example of how to do that in our case is provided in Script 2 and Script 3.

A crucial step when using MCMC methods is to ensure the target distribution is reached. To this end, a variety of diagnostic tools and methodologies have been proposed. A popular approach, explained in Gelman and Rubin (1992), is to rely on samples coming from multiple chains instantiated in different points of the parameter space, usually at least four. This is to ensure a bad starting value does not lead to incomplete exploration of the posterior distribution. Then, the next step is to examine the different chains to assess if

there are any departures from stationarity, which can be detected visually in *trace plots* if sudden changes in behaviour or clear trends appear. Moreover, one wants to check if they have "mixed" well, which means they overlap significantly, as this would imply that a common distribution was reached. A well known metric to check for mixing is the $\hat{R}$ statistics, defined as:

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}},$$

where $\hat{V} = \frac{N-1}{N}W + \frac{1}{N}B$, with W the variance within the chains and B the variance between the chains. Values of $\hat{R}$ close to one indicate the variability between chains is relatively low, suggesting good mixing.

Another important diagnostic check is to ensure the information contained in the dependent samples is sufficiently high, ideally close to the information we would have with independent draws. To this mean, effective sample size (ESS) is usually reported, which decreases with higher autocorrelation in the chains:

$$ESS = \frac{N}{1 + 2\sum_{t=1}^{\infty} \rho_t}.$$

Moreover, autocorrelation plots can be shown to check if after a few lags the correlation decreases fast enough. In case the chains are strongly autocorrelated, standard practice is to perform *thinning*, which consists in keeping only every k-th sample and discarding the others, in order to reduce autocorrelation between consecutive samples.

As mentioned before, Stan relies on HMC to perform sampling. Without entering in the details, the idea is to explore the posterior distribution using Hamiltonian dynamics. At each iteration, the sampler approximates the path it should follow to reach the desired point in the posterior space. Stan uses the *leapfrog* algorithm and automatically tunes most of the parameters that are related to the sampler details. However, in some situations of pathological posterior geometry, for example regions of high-curvature, standard default settings might fail to approximate the path correctly. Such a problem is termed a *divergent transition*. Stan warns that results may be unreliable if divergent transitions occur. This is because a badly approximated path produces draws that are not valid samples from the target distribution, thereby biasing inference. If the number of divergent transitions are not too many, it might be enough to increase the approximation precision with the appropriate parameter (adapt delta), at the cost of longer computational time. However, if the problem persists, model reparametrisation might be required.

3.1.5 Posterior Predictive Check

An important part when evaluating the goodness of fit of a Bayesian model is to perform a Posterior Predictive Check (PPC) (Gelman et al., 1996). A PPC consists in ensuring the model is capable of reproducing characteristics of interest, usually a statistics t of the observed data $Y_{1:n}$, when generating new samples $\tilde{Y}_{1:n}$.

To perform a PPC in general, the first step is to simulate from the posterior predictive distribution

$$p(\tilde{Y}_{1:n} \mid Y_{1:n}) = \int p(\tilde{Y}_{1:n}|\theta)p(\theta|Y_{1:n})d\theta,$$

where θ is the generic parameter vector.

To do so, one can draw a dataset $\tilde{Y}_{1:n}^{(s)}$ from $p(\tilde{Y}_{1:n}|\theta^{(s)})$, where $\theta^{(s)}$ is the last sampled value of the parameter vector from the posterior distribution $p(\theta|Y_{1:n})$. Repeating this step for every iteration $s = 1, \ldots, S$ in the MCMC chains, one obtains S datasets of size n from the posterior predictive distribution.

Computing the statistics $t(\tilde{Y}_{1:n}^{(s)})$ for every s yields draws from $p(t(\tilde{Y}_{1:n})|Y_{1:n})$. One can then compare $t(Y_{1:n})$ with this distribution and check if it is well represented. In case it is not, it would be an indication that the model is not capable of capturing the desired property. A common metric used in this case is the so called posterior predictive "p-value" (Gelman et al., 2013):

$$\Pr(t(\tilde{Y}_{1:n}) \geq t(Y_{1:n})|Y_{1:n}).$$

The posterior predictive p-value is the probability that predicted quantities are equal to or more extreme than observed ones. Values close to 1 or 0 indicate most of the posterior predictive mass is away from the observed value, indicating a bad fit. A value close to 0.5 would generally indicate good fit.

Usually, the statistics t are either the mean, the variance or the quantiles. However, it is possible to choose any statistics of interest that fits the desired application. One such quantity in the case of this work was the density of unweighted bipartite networks $d(\mathcal{H}_n)$.

An efficient way to perform a PPC in Stan is to use the *generated quantities* block, where one specifies the distribution from which the new data should be sampled. In the context of the PYPM, this amounts to translate in Stan the PY sampling algorithm in Script 1 to create a bipartite network of the same number of edges as the observed data, and after that store its density statistics. A snippet showing the Stan implementation can be found in Script 4.

3.1.6 Hypothesis Testing

In the Bayesian framework, hypothesis testing can be performed in various ways. A common choice is to use the Bayes Factor (BF), which is the quantity that summarises how strong is the updating process when passing from prior to posterior beliefs, after having observed the data.

More precisely, considering a generic parameter $\theta \in \Theta$ and observed data $Y_{1:n}$, after partitioning $\Theta = \Theta_0 \cup \Theta_1$ with $\Theta_0 \cap \Theta_1 = \varnothing$ and defining the general hypothesis:

$$H_0 : \theta \in \Theta_0\,, \quad H_1 : \theta \in \Theta_1,$$

we have the following identity:

$$\underbrace{\frac{p(\theta \in \Theta_0 \mid Y_{1:n})}{p(\theta \in \Theta_1 \mid Y_{1:n})}}_{\text{Posterior Odds}} = \underbrace{\frac{p(Y_{1:n} \mid \theta \in \Theta_0)}{p(Y_{1:n} \mid \theta \in \Theta_1)}}_{\text{Bayes Factor}} \times \underbrace{\frac{p(\theta \in \Theta_0)}{p(\theta \in \Theta_1)}}_{\text{Prior Odds}}.$$

Posterior and prior odds are ratios that indicate if the null hypothesis is more or less likely (being greater or smaller than one) after and before having observed the data, respectively. The Bayes Factor, on the other hand, is the multiplicative factor that determines the magnitude of the updating step from prior to posterior.

There are historically well known threshold values to interpret the BF to state how strong is the evidence for H_0. The values come from Harold Jeffreys, and we report them in Table 3.1.

Interpretation	BF	$\log_{10}$ BF
Negative (Evidence for H_1)	< 1	< 0
Barely worth mentioning	1 to 3.2	0 to 0.5
Substantial evidence	3.2 to 10	0.5 to 1
Strong evidence	10 to 32	1 to 1.5
Very strong evidence	32 to 100	1.5 to 2
Decisive evidence	> 100	> 2

Table 3.1: Evidence in favour of H_0. Jeffreys' scale to interpret the Bayes Factor.

The Bayes Factor can be computed easily if the likelihood is known and both hypothesis are simple. When at least one of the two is composite, a simple way to compute it is given by the following ratio:

$$\text{BF} = \frac{\text{Posterior Odds}}{\text{Prior Odds}} = \frac{p(\theta \in \Theta_0 \mid Y_{1:n})}{p(\theta \in \Theta_1 \mid Y_{1:n})} \times \frac{p(\theta \in \Theta_1)}{p(\theta \in \Theta_0)}.$$

Note that if the prior on the parameter is putting equal mass on the null and alternative hypothesis, the Bayes Factor will simply coincide with the Posterior Odds.

In practice, when the posterior is complex and MCMC methods are used, the BF is computed by estimating the Posterior odds with the values from the MCMC chains satisfying the null and alternative hypothesis. Prior odds, depending on the tractability of the prior used, are usually directly accessible.

In the context of the PYPM, it could be of interest to check if the data supports the hypothesis that the generating processes for the groups are $\text{DP}(\alpha, G_0)$, instead of $\text{PY}(\alpha, \sigma, G_0)$. The utility stems in being able to determine if the group is contributing or not to the sparsity of the network. This would amount to test the following hypothesis for both groups:

$$H_0 : \sigma = 0\,, \quad H_1 : \sigma > 0.$$

However, as Stan requires continuous (and differentiable) distributions as priors, the posterior distributions will also have the same properties. Therefore, we will never observe samples for the discount parameter exactly equal to zero. The easiest way to perform an approximate test is to consider as null hypothesis the discount parameter being in a small enough interval $(0, \delta)$, with $\delta > 0$. Being the length of the interval arbitrary, it is best to check how the choice affects the decision. The final hypothesis which will be tested is:

$$H_0 : \sigma < \delta\,, \quad H_1 : \sigma > \delta.$$

An alternative method could use a *spike-and-slab* prior. Originally introduced by Mitchell and Beauchamp (1988) for variable selection in regression by inducing sparsity (forcing coefficients to be close to zero), it consists of mixing a Dirac mass at zero (the "spike") with a relatively uniform mass on the remaining parameter space (the "slab"). For a review of Bayesian variable selection methods, see O'Hara and Sillanpää

(2009). We decided to stick to the previous approximation for simplicity, but consider the spike-and-slab approach a possible refinement worth further investigation.

3.2 Simulation Studies: Posterior Convergence

We have performed simulation studies with the goal of showing if the model is capable of recovering the true parameter values with more and more data. From the Bayesian point of view, if the posterior distributions concentrate around these values, it means we are more and more confident about the correct data generating process.

	Setup 1		Setup 2	
	Group A	Group B	Group A	Group B
Process	DP	PY	DP	PY
α	5	5	5	5
σ	0	0.2	0	0.7

Table 3.2: Summary of the simulation setups: process type and parameters by group.

Two setups were chosen: in the first, group A comes from a Dirichlet Process with $\alpha_A = 5$, therefore $\sigma_A = 0$, while group B comes from a Pitman-Yor with $\alpha_B = \alpha_A = 5$ but $\sigma_B = 0.2$. The second setup is similar, but $\sigma_B = 0.7$. Given the setups, we are able to check if the model correctly captures both a PY effect and a DP effect, by inspecting the posteriors on the discount parameters. The study was carried out by simulating a network of 10^5 edges for each setup, and for the full network as well as for different subsets of $10^2, 10^3, 10^4$ edges, the model was estimated. Then, for every sample size the posterior distributions were analysed simultaneously to check convergence towards the true parameters.

The simulation results are displayed in Figure 3.1. We generally see good concentration of the posterior distributions toward the true values, indicated by the red, dashed lines. It seems that strong discount parameters make it easier for the model to recover the correct parameter values. This is evident by comparing σ_B in both simulations, since in the second one the posterior is much tighter.

The DP effect for group A is also captured, since the posterior distributions on the discount parameter σ_A move towards zero. However, the concentration strength depends on how informative is the dataset, as σ_A in simulation two seems to concentrate less on zero and more on neighbouring values.

In Table A.2, we report the Stan setup used for the simulations, while in Figure A.1 the convergence diagnostics. We observe good mixing and high enough effective sample size with the default sampler parameters. No divergent transitions occurred.

3.3 Application to Real Data

In this section, the bipartite PYPM was estimated on real plants and pollinators network data from North Carolina, gathered by Motten (1986). The dataset is publicly available on the website Web of Life. In Figure 3.2, we present a visualisation of the network, where the bipartite structure is evident. The plot shows the unique edges connecting plants and pollinators, the former marked by different colours. Weights for each edge were recorded but not shown to avoid clutter. In particular, the network contains 143 unique

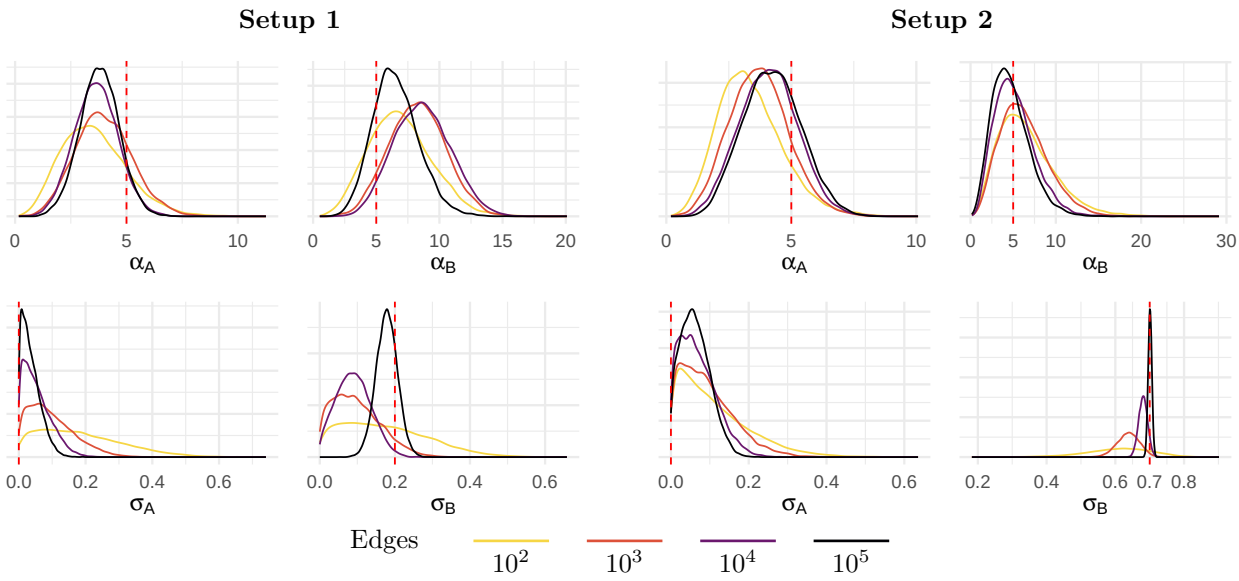


Figure 3.1: Evolution of the posterior distributions in the two simulation setups. With more edges, the posterior distributions converge to the true generating values (red, dashed lines).

edges and 2225 total connections. Being the number of plants and pollinators 13 and 44, respectively, the statistics $d(\mathcal{H}_n)$ turns out to be 0.25. The group of plants will be referred to group A, while the pollinators as group B.

Estimation of the Bayesian model with Stan was carried out using the settings specified in Table A.2. Higher thinning and adapt delta were necessary to reach acceptable results from the sampling process. The diagnostics are available in Figure A.2, where we see good mixing and convergence properties as well as very low autocorrelation. We also mention good $\widehat{R}$ and ESS values for all the parameters.

H_0 :	$\sigma_A < \delta$		$\sigma_B < \delta$	
δ	BF	$\log_{10}$BF	BF	$\log_{10}$BF
0.01	26.69	1.43	3.55	0.55
0.05	47.54	1.68	5.33	0.73
0.10	117.90	2.07	8.85	0.95

Table 3.3: Bayes Factors for the discount parameters relative to threshold δ.

In Figure 3.3, we show the posterior analysis with estimated posterior densities, medians and 95% credible intervals. For group A, we see the posterior distribution of the discount parameter concentrates towards zero, suggesting a possible DP effect for the group of plants. For the group of pollinators, we observe higher uncertainty concerning such parameter. The 95% credible interval starts from essentially zero up to 0.24, with a posterior median of 0.1, suggesting a weak and uncertain PY effect. The Posterior Predictive Check suggests a sufficiently good fit in terms of the density statistics of the unweighted network, as the observed value is not in the tails of the posterior predictive distribution (confirmed by a posterior p-value far from zero or one).

In Table 3.3, we show the Bayes Factors for three different δ values, with the goal of testing the hypothesis that the discount parameter for the two groups is zero. The interpretation refers to the thresholds mentioned

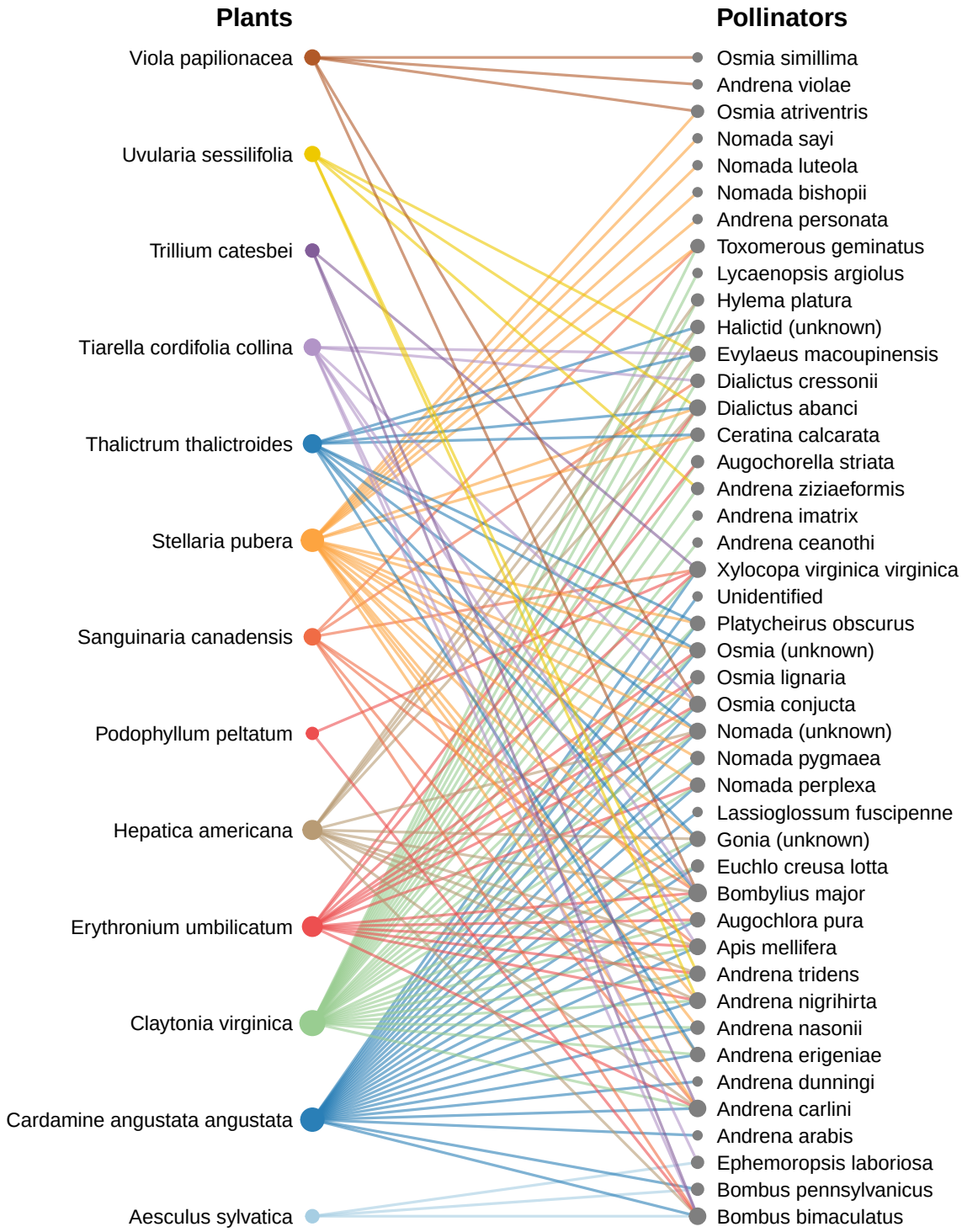
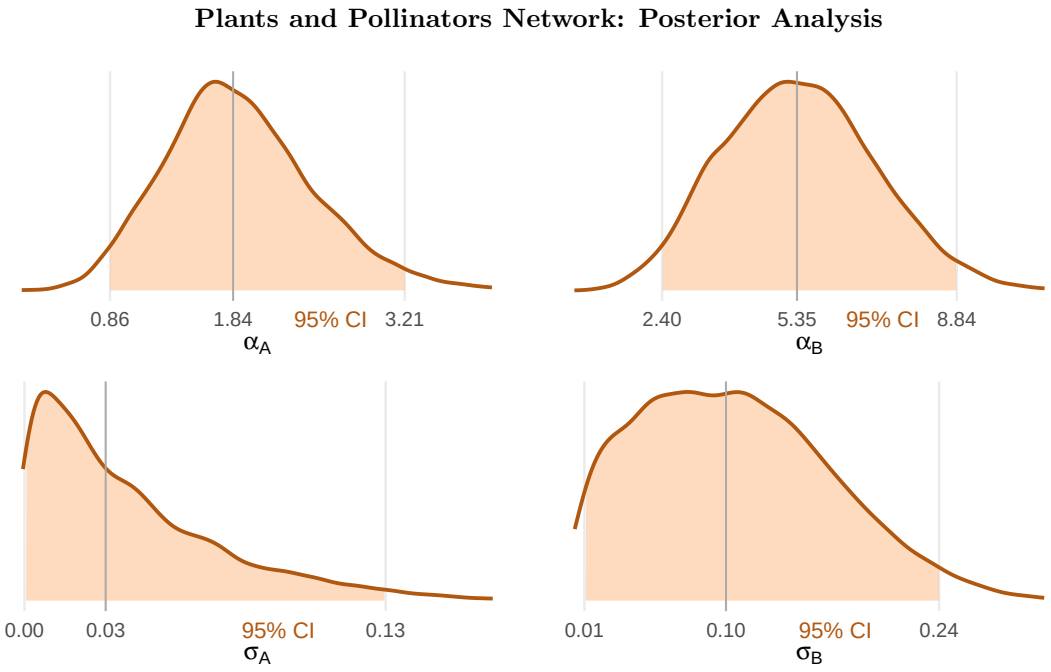


Figure 3.2: Plants and pollinators network in the North Carolina dataset (Network ID: 025). Different node sizes indicate different degrees. Edges are coloured based on plant species. Personal elaboration of the data, available at: https://www.web-of-life.es/.

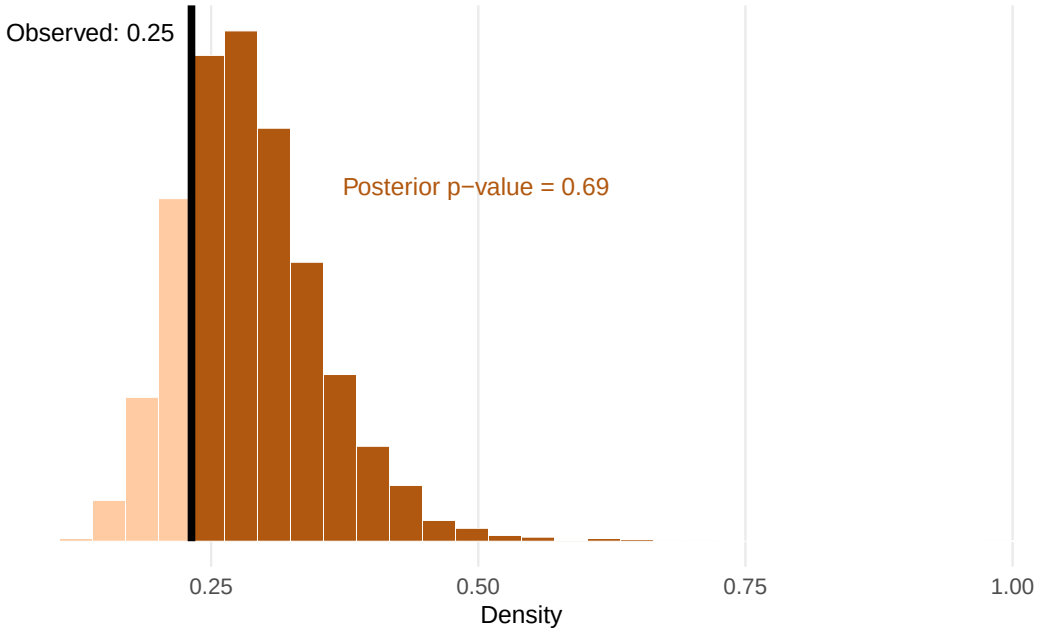
.

in Table 3.1. Even with the most conservative δ, the Bayes Factor suggests strong evidence for a DP effect for the group of plants. Moreover, for the group of pollinators, we should interpret the results as showing substantial evidence in favour of a DP effect. This complements the information we had on σ_B coming from simple inspection of the posterior summaries.

To conclude, the model captures adequately the density statistics of the unweighted network, even though there is margin for improvement. Moreover, it suggests the data does not contain a strong sparsity inducing mechanism, as in both groups there is at least some evidence for a DP process.



(a) Posterior estimates.



(b) Posterior predictive check.

Figure 3.3: Posterior analysis of plants and pollinators network: (a) posterior distributions, posterior medians (grey vertical line) and 95% CI (shaded areas). (b) posterior predictive check on the density statistics.

Conclusions

In this thesis, we extended the framework of edge exchangeability, previously developed for unipartite networks, to the bipartite setting. In particular, we demonstrated how the Pitman–Yor process, a fundamental tool in Bayesian Nonparametrics, can be employed to generate bipartite edge exchangeable networks by modelling each edge as a pair of values drawn independently from distinct Pitman–Yor processes.

We further established that the sparsity property proven by Crane and Dempsey (2018) for unipartite unweighted networks also holds in the bipartite case, requiring only minor adjustments to account for the bipartite structure. Simulation studies supported this theoretical result, showing that network density tends to zero on average as the number of edges grows. Additional insights were gained by examining the special case of the Dirichlet process, which was found empirically not to induce sparsity in bipartite networks.

Furthermore, we illustrated a fully Bayesian approach to perform inference on the parameters using the software Stan, which allows for efficient MCMC sampling from the posterior distributions. We assessed convergence of the posteriors to true parameter values under additional simulation studies. We also discussed the matters of prior choice, posterior predictive checking for model evaluation and hypothesis testing, in particular through the practical application to real data. Inference on the discount parameters is particularly useful for identifying which group of the bipartite network drives the sparsity mechanism. This can be assessed by testing whether the group specific Pitman–Yor process reduces to the Dirichlet process, which we showed does not induce sparsity.

Clear limitation of the proposed model is the independence assumption between the two groups. Allowing for more complex dependence structures would certainly increase the applicability of the model, which as of now remains rather restricted.

Moreover, there remains to prove if the model creates sparse graph sequences even when retaining multiple edges, in accordance to the unipartite version of the model. Furthermore, there were hints suggesting the existence of a relationship between the discount parameters and the speed of convergence towards zero of the density statistics. Finding a closed form for such relationship would shed light on the precise effect of the discount parameters in inducing sparsity.

We would like to mention that experiments presented in this thesis were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

Bibliography

Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598.

Billingsley, P. (1979). *Probability and Measure*. Wiley series in probability and mathematical statistics. Wiley, New York.

Cai, D., Campbell, T., and Broderick, T. (2016). Edge-exchangeable graphs and sparsity. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4242–4250.

Caron, F. and Fox, E. B. (2017). Sparse Graphs Using Exchangeable Random Measures. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(5):1295–1366.

Crane, H. (2018). *Probabilistic foundations of statistical network analysis*. Monographs on statistics and applied probability ; 157. CRC Press, Taylor & Francis Group, Boca Raton, FL.

Crane, H. and Dempsey, W. (2018). Edge Exchangeable Models for Interaction Networks. *Journal of the American Statistical Association*, 113(523):1311–1326.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

Franssen, S. E. M. P. and van der Vaart, A. W. (2022). Empirical and full bayes estimation of the type of a pitman-yor process. *arXiv preprint arXiv:2208.14255*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science Series. CRC, Boca Raton, Florida, third edition.

Gelman, A., Meng, X.-L., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6(4):733–760.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010). *Bayesian Nonparametrics*. Cambridge Series. Cambridge University Press.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Hoover, D. N. (1979). Relations on Probability Spaces and Arrays of Random Variables. *Institute for Advanced Study, Princeton, NJ*.

Kallenberg, O. (2005). *Probabilistic Symmetries and Invariance Principles*. Probability and its Applications. Springer, New York.

Korwar, R. M. and Hollander, M. (1973). Contributions to the Theory of Dirichlet Processes. *The Annals of Probability*, 1(4):705–711.

Lo, A. Y. (1984). On a Class of Bayesian Nonparametric Estimates: I. Density Estimates. *The Annals of Statistics*, 12(1):351–357.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian Variable Selection in Linear Regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

Motten, A. F. (1986). Pollination Ecology of the Spring Wildflower Community of a Spring Wildflower Community. *Ecological Monographs*, 56(1):21–42.

Müller, P., Quintana, F. A., Jara, A., and Hanson, T. (2015). *Bayesian Nonparametric Data Analysis*. Springer Series in Statistics. Springer, New York, NY.

O'Hara, R. B. and Sillanpää, M. J. (2009). A review of Bayesian variable selection methods: what, how and which. *Bayesian Analysis*, 4(1):85–118.

Orbanz, P. and Roy, D. (2013). Bayesian Models of Graphs, Arrays and Other Exchangeable Random Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37.

Pitman, J. (2006). *Combinatorial stochastic processes*, volume 1875 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650.

Snijders, T. A. B. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100.

Stan Development Team (2024). *Stan Modeling Language Users Guide and Reference Manual*. Stan Development Team. `https://mc-stan.org`.

Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.

Williamson, S. A. (2016). Nonparametric Network Models for Link Prediction. *Journal of Machine Learning Research*, 17(202):1–21.

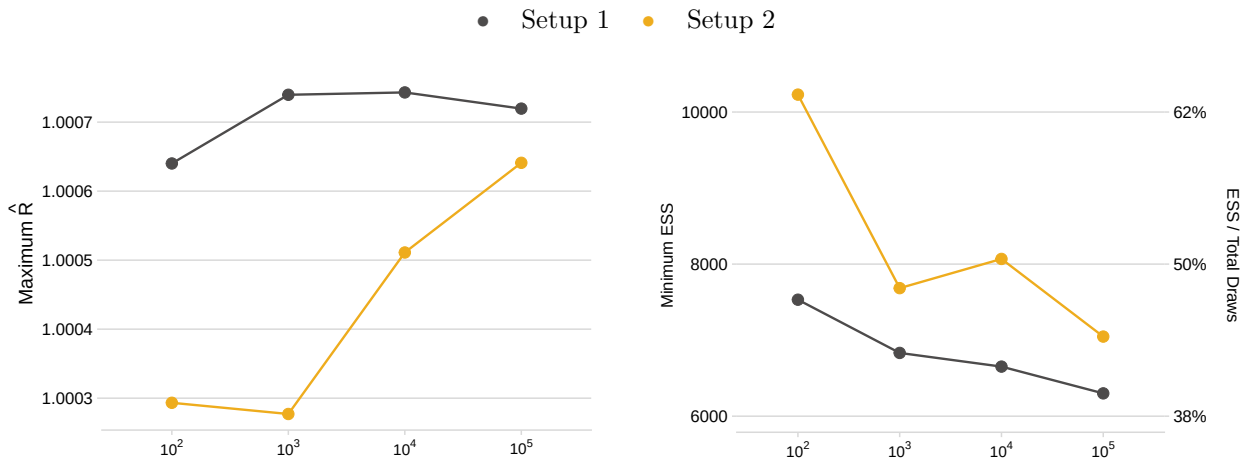
Appendix A: Supplementary Material

Property	Prior for α	Prior for σ
	Gamma(shape $= 3$, rate $= 0.4$)	$U(0, 1)$
Expected Value	7.5	0.5
Variance	18.75	0.0833
Standard Deviation	4.33	0.2887
95% Central Interval	(1.55, 18.06)	(0.025, 0.975)

Table A.1: Prior distributions for the model parameters.

Setting	Simulation Studies	Real Data
Number of Chains	4	4
Total Iterations	5000 per chain	20000 per chain
Warmup Iterations	1000 per chain	1000 per chain
Post-warmup Iterations	4000 per chain	19000 per chain
Thinning Interval	1	10
Retained Iterations	4000 per chain	1900 per chain
Seed	42	42
Adapt Delta	0.95 (default)	0.999
Maximum Tree Depth	10 (default)	10 (default)

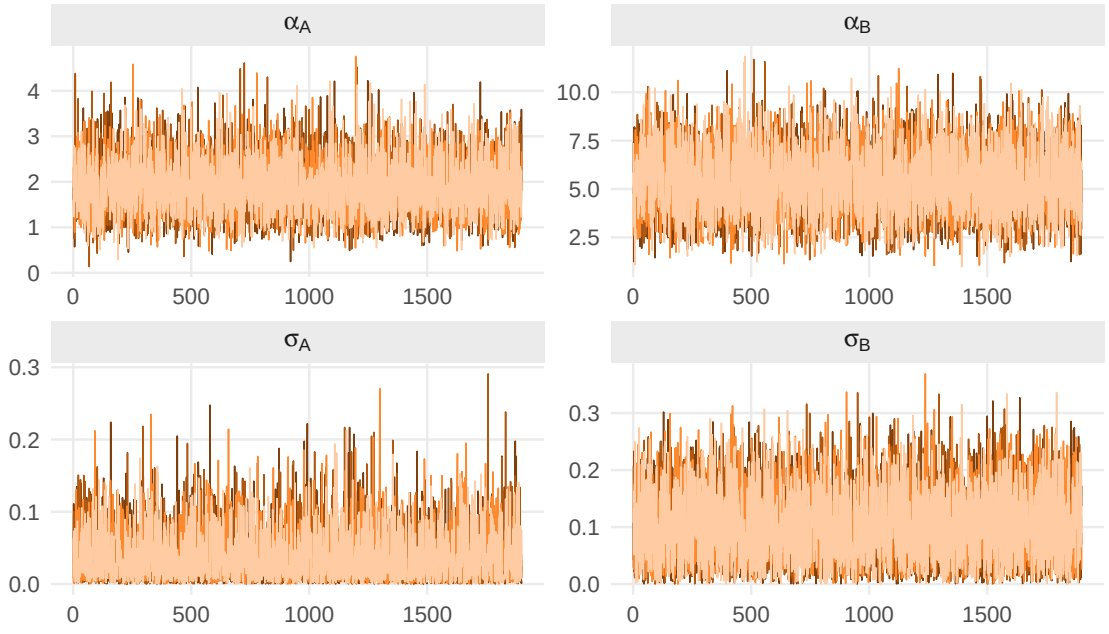
Table A.2: Stan settings used to estimate the model.



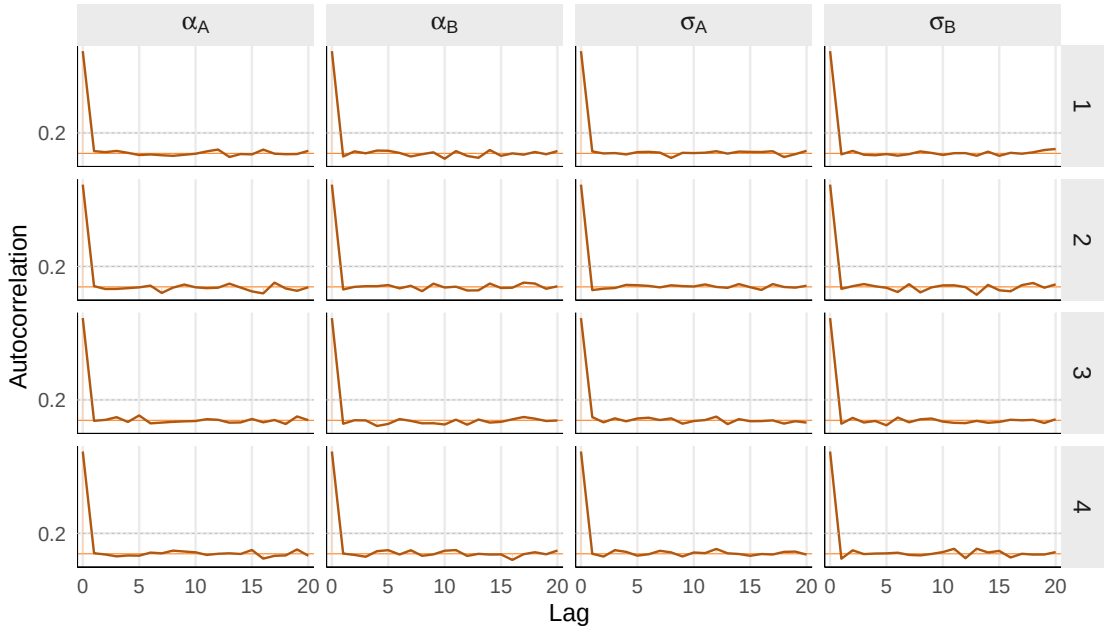
(a) Maximum $\hat{R}$

(b) Minimum ESS

Figure A.1: Convergence diagnostics for the simulation studies over sample sizes. (a) Maximum $\hat{R}$ and (b) minimum effective sample size of the 4 parameters of the model. No divergent transitions were detected.



(a) Trace plots.



(b) ACF plots.

Figure A.2: MCMC diagnostics for the estimation results on real data. Trace plots (a) and autocorrelation plots (b) indicate good convergence and mixing. On top of that, $\widehat{R}$ was one for all parameters, and the effective sample sizes were $ESS = 7495$ (α_A), 7935 (α_B), 7536 (σ_A), and 7702 (σ_B). No divergent transitions occurred.

Appendix B: Code

```r
py_sampler <- function(N, alpha, sigma) {
  x <- numeric(N)
  x[1] <- 1

  counts <- numeric(N)
  counts[1] <- 1

  active_nodes <- c(1)

  for (n in 2:N) {
    k <- length(active_nodes)
    prob <- c(counts[active_nodes] - sigma, alpha + sigma * k)

    # Draw cluster index (either existing or new)
    x[n] <- sample(
      c(active_nodes, k + 1),
      size = 1,
      prob = prob
    )

    counts[x[n]] <- counts[x[n]] + 1

    # If it is a new cluster, add to active_nodes
    if (counts[x[n]] == 1) {
      active_nodes <- c(active_nodes, x[n])
    }
  }

  return(list(
    x = x,
    counts = counts,
    active_nodes = active_nodes,
    active_counts = counts[active_nodes],
    K = length(active_nodes)
  ))
}
```

Script 1: R function to draw a sample from the Pitman-Yor process, with unique values (the nodes) labelled based on the order of arrival. The function uses the CRP construction to determine the probabilities of observing something new or old. It takes as input the desired sample size and the parameters of the process. The variable *active nodes* stores which nodes have been sampled to easily extract their frequencies by accessing the *counts* variable. This avoids unnecessary computations at every iteration of the loop, increasing efficiency when N is large. Along with the sample, it returns other information stored during the sampling process.

```
functions {
  real eppf_lp(array[] n_k, int K, real alpha, real sigma) {
    int n = sum(n_k);

    // Term 1
    vector[K - 1] k_vec = linspaced_vector(K - 1, 1, K - 1);
    real term1 = sum(log(alpha + sigma * k_vec));

    // Term 2
    real term2 = lgamma(alpha + n) - lgamma(alpha + 1);

    // Term 3
    vector[K] n_k_vec = to_vector(n_k);
    vector[K] gamma_terms = lgamma(n_k_vec - sigma);
    real term3 = sum(gamma_terms) - K * lgamma(1 - sigma);

    return term1 - term2 + term3;
  }
}
```

Script 2: Custom Stan function to compute the log EPPF for one Pitman-Yor sample. The function takes as inputs the array of frequencies of each unique value, the number of unique values and the process parameters.

```
data {
... // Node data for both groups and prior parameters
}

parameters {
  real<lower = 0> alpha_A;
  real<lower = 0, upper = 1> sigma_A;
  ... // same for B
}

model {
... // priors with user defined parameters

  target += eppf_lp(n_A, K_A, alpha_A, sigma_A) +
  eppf_lp(n_B, K_B, alpha_B, sigma_B);
}
```

Script 3: Sketch of Pitman-Yor Product Model estimation in Stan. The **data block** contains information on nodes (frequencies and number of unique values) and on parameters of the prior distributions, based on the user preference. The **parameters block** defines the model parameters and their constraints. The **model block** specifies the prior distributions (using the parameters eventually passed in the data block) and then updates the model likelihood at every iteration with the *target* argument.

```
generated quantities {
  real density_ppc; // save only density
  {
    // local scope to discard the sampled network
    int maxN = e_obs; // upper limit on number of draws
    array[maxN, maxN] int seen;
    int unique_edges;
    array[maxN] int countsA;
    int KactA;
    int currA; // repeat for B last 3 objects

    for (i in 1 : maxN) { // Create zero valued objects
      countsA[i] = 0;
      countsB[i] = 0;
      for (j in 1 : maxN)
        seen[i, j] = 0;
    }

    seen[1, 1] = 1; // Initialize first edge
    unique_edges = 1;
    countsA[1] = 1;
    KactA = 1; // repeat for B last 2 objects

    for (n in 2 : e_obs) {// A side
        real denomA = alpha_A + n - 1;
        vector[KactA] cA = to_vector(countsA)[1 : KactA];
        vector[KactA + 1] pA;
        pA[1 : KactA] = (cA - sigma_A) / denomA;
        pA[KactA + 1] = (alpha_A + sigma_A * KactA)/denomA;
        currA = categorical_rng(pA);
        countsA[currA] += 1;
        if (currA == KactA + 1)
          KactA += 1;
        // repeat for B side

      if (seen[currA, currB] == 0) { // Is new edge unique
        seen[currA, currB] = 1;
        unique_edges += 1;
      }
    }

    density_ppc = unique_edges / (1.0 * KactA * KactB);
  }
}
```

Script 4: The optional **generated quantities block** is used to create additional variables at every MCMC iteration, after parameters values have been sampled from the posterior. This allows to perform a posterior predictive check by first creating a network from the Pitman-Yor Product Model given the sampled parameters, and after that by storing its density. The code above follows the same logic as Script 1 to generate a network. Only the density is stored to avoid overpassing memory limits.