DRAFT

LAPO SANTI

Contents

1. Introduction

Introduction:

The purpose of this paper is to introduce a novel methodology to infer the latent ordering of blocks that exhibit a transitivity property in directed networks. Specifically, we develop a method to infer the community structure of binary interactions, where the groups are ordered in a one-dimensional hierarchy that determines the preferred edge direction. The paper builds on a class of models closely related to stochastic blockmodels, where only non-negative entries are considered. However, in this model, we are ignoring the zero entries, which can be debatable whether we are still dealing with a network as we are ignoring the sparsity that is one of the staples of networks.

Research Question:

The research question of this paper is to infer the block structure underlying a set of binary interactions with transitivity properties in directed networks. Our approach advances the existing literature by considering a method to learn the ordering of the blocks that reflects better an ordering of the block themselves.

Importance of the Study:

This paper innovates the field of statistics by providing a novel methodology to infer the latent ordering of blocks, which can be very useful under certain circumstances. The proposed method has practical applications in analyzing tennis tournament data, where we can learn how competitive a tournament is by inferring the blocks and their ordering, from the strongest to the weakest.

Objectives:

The objective of this paper is to develop a method to infer the community structure of binary interactions where the groups are ordered in a one-dimensional hierarchy that determines the preferred edge direction. Specifically, we aim to learn the ordering of the blocks that exhibit a transitivity property in directed networks.

Methodology:

We start by introducing the theory behind Partially Ordered Markov Models (POMMs) and the definition of a Strong Stochastic Transitivity matrix (SST matrix). Then we combine the two ideas into one, showing that we can represent the SST matrix with a collection of level sets. We impose a very specific dependence structure between the blocks, a Markovian one. Next, we specify the full model, where outcomes of tennis matches are represented as a binomial distribution, the POMM ordering is imposed on the entries of the winning probability matrix, and a dirichlet multinomial distribution is imposed on the assignment of players to blocks. This distribution is expanded into a Chinese Restaurant Process (CRP), which can be used to avoid assumptions on the initial number of blocks. Initially, we assume that the number of blocks follows a Poisson distribution. Finally, we simulate data and test our methodology with simulated data, followed by applying it to a real tennis dataset.

Outline of the Paper:

This paper is organized as follows. Section 2 introduces the theory behind Partially Ordered Markov Models (POMMs) and the definition of a Strong Stochastic Transitivity

matrix (SST matrix). In Section 3, we combine the two ideas into one, showing that we can represent the SST matrix with a collection of level sets. Section 4 describes the dependence structure imposed on the blocks, a Markovian one, and how we specify the full model. In Section 5, we explain how we expand the dirichlet multinomial distribution into a Chinese Restaurant Process (CRP) and the assumptions we make on the number of blocks. Section 6 presents our simulation study, where we test our methodology with simulated data. Finally, in Section 7, we apply our methodology to a real tennis dataset. We conclude the paper with a discussion of our findings and potential future research directions.

2. Posets and Dags

Definition 1. *[Poset]To define poset, a partially ordered set we start from D, a set of elements. The binary relation $\prec$ on D is said to be a partial order if:*

(1) *For any $x \in D, x \prec x$(reflexivity)*

(2) *For any $x, y, z \in D, x \prec y$ and $y \prec z \implies x \prec z$(transitivity)*

(3) *For any $x, y \in D, x \prec y$ and $y \prec x \implies x = y$(antisymmetry).*

Then we call $(D, \prec)$ a partially ordered set, or a poset.

A *finite poset* $(D, \prec)$ is a poset where D has a finite number of distinct elements.

Example: let D be the finite set defined by representing the $M \times N$ array of probabilities. Let (u, v) and (q,r) be any two elements of D and define the binary relation on D by

$$(q, r) \prec (u, v) \iff q \prec u \text{ and } r \prec v \tag{4}$$

There exists a correspondence between posets and directed cyclic graphs.

Let (D, F) be a directed acyclic graph, where $D = \{y_1, \ldots, y_n\}$, a finite set. To construct a poset to which this digraph corresponds, we define the binary relation $\prec$ on D by

(5) $y_i \prec y_i$ for $i = 1 \ldots n$

(6) $y_i \prec y_j$ if there exists a directed path from y_i to $y_j \in (D, F)$

We saw above that the correspondence is many-to-one. Given a finite poset, one may construct a class of directed acyclic graphs; the correspondence described above is in a sense the minimal directed acyclic graph since it has the smallest possible directed edge set Pomms definitions

Definition 2 (Cone)**.** *For any $y \in D$, the cone of y is the set*

$$cone\ y = \{x \in D : x \prec y; x \neq y\}$$

Definition 3 (Adjl)**.** *For any $y \in D$ the adjacent lower neighbourhood of y is the set*

$$adjl\ y = \{x \in D : (x, y) is\ a\ directed\ edge\ in (D, F)\}$$

Definition 4 (Dilation)**.** *For any $y \in D$, the dilation of y is the set*

$$dil\ y = \bigcup \{\overline{adjl}\ x : y \in \overline{adjl}\ \}$$

Definition 5 (Excluded dilation)**.** *For any $y \in D$, the excluded dilation of y is the set*

$$dil\ ^* y = dil\ y \setminus \{y\} \tag{7}$$

Definition 6 (Minimal element)**.** *In general, an element $y \in D$ is called minimal element if there is no other element x satisfying $x \prec y$ where adjl s is the set of adjacent lower neighbors of $s \in D$.*

Definition 7 (Cover of a Subset)**.** *The cover of a subset B is a set of all elements x in D such that x is adjacent to an element in B and x is not in B. Formally, the cover of B is defined as follows:*

$$covr\ B = x \in D : adjl\ x \subset B \text{ and } x \notin B$$

where adjl x is the set of all adjacent elements of x in D.

Intuitively, the cover of a subset B represents all the elements in D that are outside of B but are adjacent to at least one element in B. In other words, the cover of B captures the neighborhood of B in D.

Definition 8 (Level Sets)**.** *The level sets of a poset D are a sequence of nonempty cover sets defined recursively as follows:*

$$L^0 = D_{min}; \quad L^i = covr\left(\bigcup_{k=0}^{i-1} L^k\right)$$

where D_{min} is the set of all minimal elements in D.

The first level set L^0 is simply the set of all minimal elements in D. The subsequent level sets are defined by taking the union of all the previous level sets and taking the cover of this union. Intuitively, each level set captures the neighborhood of the previous level sets in D.

2.1. **Partially ordered Markov models:** Consider a finite set of random variables $\{Z(s_1), \ldots, Z(s_n)\}$ indexed by location or "points"

$$D = \{s_1, \ldots, s_n\} : n \in \{1, 2, \ldots\}$$

That is, we assume the existence of a directed acyclic graph (D, F) and its corresponding poset $(D, \prec)$. Let (D, F) be a finite, directed acyclic graph and its corresponding poset $(D, \prec)$. Consider $s \in D$ and recall the definition of cone s. Also, let the quantity U_s denote any subsets of points not related to s. Formally:

$$U_s \subset \{u \in D : u \text{ and } s \text{ are not related } \}$$

Definition 9 (POMM)**.** *Then $\{Z(s) : s \in D\}$ is said to be a partially ordered Markov model (POMM) if , for all $s \in D$ and any U_s*

$$P(Z(s)|Z(\ cone\ s), Z(U_s)) = P(Z(s)|Z(\ adjl\ s) \tag{8}$$

Proposition 1. *[Joint Distribution] Let (D, F) be a directed acyclic graph with no singleton points and let $(D, \prec)$, be its associated poset. Suppose that $\{Z(s) : s \in D\}$ is a POMM. Then*

$$P(Z(D)) = P(Z(L^0)) \prod_{k=1}^{m} \prod \{P(Z(u))|Z(\ adjl\ u) : u \in L^k\} \tag{9}$$

$$= P(Z(L^0)) \prod \{P(Z(u))|Z(\ adjl\ u) : u \in D \setminus L^0\} \tag{10}$$

where $L^0, L^1, \ldots L^m$ are the level sets as defined previously.

Result 1 relates the probability of a random variable defined on a poset to the probabilities of its restrictions to the lower level sets of the poset.

The result states that the probability of Z on the entire poset D can be expressed as a product of the probabilities of Z restricted to the level sets $L^0, L^1, \ldots, L^m$ of the poset, where $L^0 = D_{\min}$ is the set of minimal elements of D, and L^k is the set of elements of D that are not in any of the previous level sets $L^0, L^1, \ldots, L^{k-1}$ and whose immediate predecessors are all in the union of the previous level sets $\bigcup_{i=0}^{k-1} L^i$.

The first part of the result states that the probability of Z on D is equal to the product of the probability of Z on L^0 and the conditional probabilities of Z on the elements of each subsequent level set L^k, given the values of Z on their immediate predecessors. This can be seen as a form of the chain rule of probability, where the joint probability of Z on D is decomposed into a product of conditional probabilities.

The second part of the result simplifies the product by noting that the conditional probabilities of Z on the elements of $D \setminus L^0$ are determined by the values of Z on their immediate predecessors, which are all in L^0 or $D \setminus L^0$. Therefore, the product can be simplified to the product of the probability of Z on L^0 and the conditional probabilities of Z on the elements of $D \setminus L^0$ given the values of Z on their immediate predecessors in $D \setminus L^0$. This simplification reduces the number of terms in the product and makes the computation of the joint probability of Z on D more efficient.

3. Application to the SST matrix

In this section, I will introduce a matrix that displays the Strong Stochastic Transitivity (SST) property. The aim of this section is to model the probability of a player winning a match in a tournament against another one. To achieve this goal, we need to ensure that the probabilities are arranged in a consistent manner. Specifically, we want to ensure that if Player A is stronger than Player B, and Player B is stronger than Player C, then Player A must be stronger than Player C. This is a well-known mathematical concept known as transitivity, and it is essential to impose it on the probabilities of victory between the players. For instance, if we know that Djoković is stronger than Medvedev and Medvedev is stronger than Kyrgios, then we can infer that Djoković must be stronger than Kyrgios. By using the SST property, we can guarantee that the probabilities of victory reflect this logical relationship, which enhances the clarity and coherence of our model. After having introduced the SST property, we want to build upon the definitions of Section(1), and re-express the SST matrix within the POMMs' framework. This new definition will allow us to have a coherent and tractable framework to express the joint probability distribution of such ordered probabilities and, ultimately, to perform inference.

3.1. **Defining matrix P (SST matrix).** The matrix under consideration, that is P, is a collection of victory probabilities among K entities, which could represent players or also groups of players. The matrix is denoted by $\underset{K\times K}{P}$, where K is the number of players/ group of players taken into consideration.

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,K} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & \cdots & p_{K,K} \end{pmatrix}$$

Each element $p_{i,j}$ in the matrix P represents the probability of player i winning over player j in a tennis match, where draws are not permitted. Therefore, it must be the case that $p_{i,j} + p_{j,i} = 1$ in order to satisfy the requirements for a valid probability. It follows that $p_{j,i}$ can be expressed as $1 - p_{i,j}$. The lower triangular entries of matrix P can be determined from the upper triangular entries. Consequently, our focus is on modelling the upper triangular part of P.

Without loss of generality, we can assume that player/team 1 is the strongest, and player/team K is the weakest.

From this assumption, it follows that the elements in the upper triangular part of the matrix must remain above 0.5 to maintain the assumption of monotonicity in the probabilities. Violation of this assumption could lead to a contradiction where a weaker player/team has a higher probability of winning than a stronger one. For instance, if $p_{1,2} = 0.4 \leq 0.5$, then $p_{2,1} = 0.6$ would imply that player 1 is weaker than player 2, which is contradictory with our baseline assumption. Furthermore, we set the main diagonal of matrix P to 0.5 for teams and 0 for individual players.

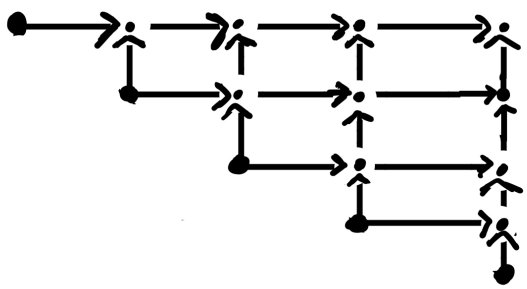


FIGURE 1. Dag representation of the Poset imposed onto the SST matrix

Therefore, we must constrain the probabilities as follows:

- The probabilities must increase monotonically as the index of the columns j increases;
- The probabilities must decrease monotonically as the index of the rows i increases.

The matrix P with the described modification will look like this:

$$\begin{pmatrix} 0.5 & \leq & p_{1,2} & \leq & ... & \leq & p_{1,K} \\ & & \vee| & & ... & & \vee| \\ 1-p_{1,2} & \leq & 0.5 & \leq & ... & \leq & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots & ... & \vdots & \vdots \\ 1-p_{1,K} & \leq & 1-p_{2,K} & \leq & ... & \leq & 0.5 \end{pmatrix}$$

3.2. **Defining a Poset over P.** Having defined P, now we want to re-define it within a finite Poset framework to obtain $(P, \leq)$. Let (i, j) and (p, q) be two elements of P and define the binary relation $\leq$ on P as

$$(p, q) \leq (i, j) \iff p \leq i \text{ and } q \leq j \tag{11}$$

Now, $(P, \leq)$ is clearly a Poset since it satisfies the three properties of Definition (**??**). We represent the corresponding directed acyclic graph (P, F), where F is the set of directed edges between vertices in Figure (**??**) by using the definition of Adjacent Lower Neighborhood that we introduced before.

Definition 10 (Cone over P)**.** *In this case, for any $(i, j) \in D$ the cone of (i, j) is the set:*

$$\textit{cone}\ (i, j) = \{(i, j-1), \dots (i, i), (i+1, j), \dots, (K, j)\} \tag{12}$$

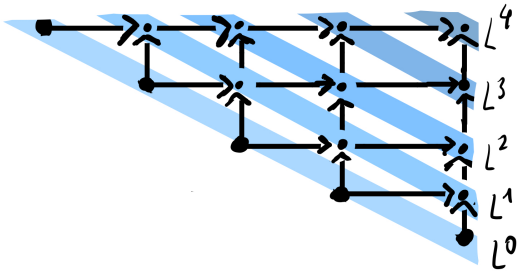


FIGURE 2. Visual Display of the level sets $L^k : k = 0, \ldots, 4$

Definition 11 ($\overline{\text{cone}}\,(i,j)$)**.** *In this case, for any $(i,j) \in P$ the closure of the cone of (i,j) is the set:*

$$\overline{cone}\,(i,j) = \{(i,j), \ldots (i,i), (i,j), \ldots, (K,j)\} \tag{13}$$

Definition 12 (The adjacent lower neighborhood of (i,j))**.** *In this case, for any $(i,j) \in P$ the adjacent lower neighborhood of (i,j) is the set:*

$$adjl\ (i,j) = \{(i, j-i), (i+1, j)\} \tag{14}$$

Definition 13 (closure of adjl (i,j))**.** *In this case, for any $(i,j) \in P$ the closure of adjl (i,j) is the set:*

$$\overline{adjl}\,(i,j) = \{(i, j-i), (i+1, j), (i,j)\} \tag{15}$$

Definition 14 (D^P_{min})**.** *In this case, the minimal element denoted by D_{min} is such that*

$$P_{min} = \{(i,j) \in P : i = j\} \tag{16}$$

namely the main diagonal of the P matrix.

The *level sets* of P, in this case, corresponds to the diagonals above the main one.

3.3. **Partially ordered Markov Models applied to P.** Considering the finite set of random variables $\{P_{1,1}, \ldots, P_{K,K}\}$ indexed by locations where

$$D \equiv \{(1,1), \ldots (K,K)\}$$

Having showed the existence of a directly acyclic graph (P,F) and its corresponding poset $(P, \leq)$, we can write down

$$P(P_{ij}|Z(\text{cone } i,j)) = P(P_{ij}|P(\text{adl } i,j)) \tag{17}$$

$$P(P_{ij}|P_{i+1,j}, P_{i,j-1}) \tag{18}$$

Now, exploiting Proposition (**??**), we can write:

$$P(P(D)) = \prod_{i=1}^{K} \prod_{j=i}^{K} P(P_{ij}|P_{i+1,j}, P_{i,j-1}) \tag{19}$$

4. Full model specification

The following Bayesian model is used to describe and analyze pairwise data, with the specific aim to identify clusters of points with similar connectivity patterns.

The model uses a Poisson distribution to model the number of blocks, a Dirichlet-multinomial distribution to model the distribution of nodes' assignment across blocks, and a binomial distribution to model the distribution of edges within blocks. Additionally, the model includes a POMM process to model the probability of edge formation between nodes within blocks.

The goal of the model is to estimate the number of blocks, the distribution of nodes across blocks, and the probability of edge formation between nodes within blocks, given observed network data. The Bayesian approach allows for uncertainty in these estimates and provides a framework for incorporating prior knowledge and updating beliefs as new data becomes available.

4.1. **Simple Model specification.** This is a model for pairwise count data. We explicitly model the results of the interactions between two individuals i and j. Given N observations, the likelihood is

$$p(y|z,P,K) = \prod_{i=2}^{N-1}\prod_{j=i}^{N} p(y_{ij}|z,P,K) \tag{20}$$

$$= \prod_{i=2}^{N-1}\prod_{j=i}^{N} \binom{n_{ij}}{y_{ij}} p_{z_i,z_j}^{y_{ij}}(1-p_{z_i,z_j})^{n_{ij}-y_{ij}} \tag{21}$$

where n_{ij} denotes the total number of interactions between the two individuals i and j and y_{ij} is the number of successes of the individual i in interacting with j. The probability of success is given by p_{z_i,z_j} which consists of two parameters. The $K \times K$ matrix P and the $N \times 1$ vector z.

The vector z takes values over the discrete and finite set $\{1,\dots,K\}$, and it is an indicator variable such that if $z_i = k$ individual i belongs to block k.

The matrix P contains the probabilities of success for individuals belonging to each possible blocks combination. For this reason P is $K \times K$. Therefore, the parameter p_{z_i,z_j} consists in the probability of success in an interaction between one individual belonging to block z_i and another of block z_j.

4.2. **Prior Specification.** This model has three parameters, and we put a prior on each of them.

Starting with P, we assume that its entries, namely $p_{k,k'}$, are independent and identically $Beta(a,b)$ distributed random variable. By setting $a = b = 1$ they collapse to a uniform distribution.

$$p_{k,k'} \sim Beta(1,1) \quad \text{for } k,k' = 1,\dots,K \tag{22}$$

Second, we assume that the z_is are independent and identically drawn from a multinomial distribution with one trial and probability vector $(\theta_1, \ldots, \theta_K)$. We can write then:

$$z_i|\boldsymbol{\theta} \sim \text{Multinomial}(1, \boldsymbol{\theta}) \quad \text{for } i = 1, \ldots, N \tag{23}$$

To have more flexibility in the blocks sizes, we put an hyper-prior on the $\theta_1, \ldots, \theta_K$, assuming that they are drawn from a Dirichlet distribution with parameter the $K \times 1$ vector $\boldsymbol{\gamma}$.

By marginalizing out θ, following the common practice in the literature, we can express the marginal distribution of z as:

$$p(\mathbf{z}|\boldsymbol{\gamma}) = \frac{\Gamma(\sum_{k=1}^{K} \gamma_k)}{\prod_{k=1}^{K} \Gamma(\gamma_k)} \frac{\prod_{k=1}^{K} \Gamma(n_k + \gamma_k)}{\Gamma(\sum_{k=1}^{K} (n_k + \gamma_k)} \tag{24}$$

where n_k is the number of players assigned to block k.

Finally, we assume that the number of clusters K follow a Poisson distribution Poisson($\lambda = 1$), subject to the condition $K > 0$.

4.3. **POMM Prior.** In order to induce an ordering or ranking among the blocks, we introduce a hierarchical structure. Without loss of generality, we assume that block 1 has the highest probability of success when interacting with any other block, while block K has the lowest probability of success. We require this ranking to be transitive, meaning that if block A has a higher probability of success when interacting with block B, and block B has a higher probability of success when interacting with block C, then block A must still be the preferred choice when interacting with block C. Mathematically, this can be expressed as $p_{k,h} > p_{k',h}$ when $k < k'$ for $h \notin k, k'$.

To achieve this effect, we impose three conditions: probabilities should increase in the columns, decrease in the rows, and be greater than or equal to 0.5 in the upper triangular matrix. To satisfy these conditions, we construct the following scheme.

4.3.1. *Level Sets.* We define the level sets, denoted by $L^{(k)}$, as the diagonals of the upper triangular matrix P. The main diagonal is referred to as level set 0, denoted by $L^{(0)}$. The diagonal above it is denoted by $L^{(1)}$, and so on up to $L^{(K-1)}$. Each level set $L^{(k)}$ is formally defined as:

$$L^{(k)} := p_{ij} \mid j - i = k \quad \text{for } k = 0, \ldots, K-1 \tag{25}$$

It is worth noting that the cardinality of each level set is given by $|L^{(k)}| = K - k$ for $k = 0, \ldots, K-1$.

4.3.2. *Truncation Process.* Requiring that probabilities increase in the rows and decrease in the columns is equivalent to ensuring that the level sets satisfy the condition:

$$\max(L^{(k)}) < \min(L^{(k+1)}) \quad \text{for } k = 0, \ldots, K-1 \tag{26}$$

To enforce this behavior, we employ an increasing truncation process controlled by a parameter α, with an upper bound given by $\beta_{\max}$.

We consider a generic power-law function $y = x^{\alpha}+0.5$, which governs the rate of increase in the truncation process. Setting $f(0) = 0.5$ ensures that $L^{(0)}$ is greater than or equal to 0.5, satisfying the transitivity condition. The function y is monotonically increasing for $x > 0$. To generate the truncations, we partition y effectively by dividing the interval into K equal-sized segments. The segment endpoints are computed as $x_k = \Delta \times k$ for $k = 0, \ldots, K$, where $\Delta = \left((\beta_{\max} - 0.5)^{(1/\alpha)} - 0\right)/K$. Mapping the cumulative sum of the segment endpoints back to y yields K truncation points denoted by $y^{(k)}$, which are defined as:

$$y^{(k)} = \left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times k\right)^{\alpha} + 0.5 \quad \text{for } k = 0, \ldots, K \tag{27}$$

Notice that $f(0) = 0.5$ and $f(K) = \beta_{\max}$ by construction.

These truncation points provide the upper and lower bounds for the entries within the corresponding level sets, thereby ensuring the desired hierarchy and transitivity in the ranking.

Mathematically we have that:

$$y^{(k)} < p_i j \in L^{(k)} < y^{(k+1)} \quad k = 0, \ldots, K-1 \tag{28}$$

4.3.3. *The POMM Prior.* Finally, we put a prior on the matrix P with this new structure in place. We assume that entries $p_{ij} \in L^{(k)} \mid (y^{(k)}+y^{(k+1)}$ are identically and independently distributed according to a Uniform$(y^{(k)}, y^{(k+1)}$. We also put a log-normal hyper-prior on α such that

$$\alpha \sim \text{lognormal}(\mu_\alpha, \sigma_\alpha^2) \tag{29}$$

where $\mu_\alpha, \sigma_\alpha^2$ are specified according to the normal parametrisation of the lognormal and are fixed to 1 and 2 respectively. Altogether, the POMM prior on P is the following:

$$p_{ij} \in L^{(k)} \mid y^{(k)}, y^{(k+1)} \sim (y^{(k)}, y^{(k+1)} \tag{30}$$

$$\alpha \sim \text{Lognormal}\left(\mu_\alpha, \sigma_\alpha^2\right) \tag{31}$$

and where the truncations $y^{(k)}$ are derived as in (**??**).

4.3.4. *The POMM Prior 2.* Finally, we put a prior on the matrix P with this new structure in place. We assume that entries $p_{ij} \in L^{(k)} \mid (y^{(k)}+y^{(k+1)}$ are identically and independently distributed according to a Normal$(\mu^{(k)}, \sigma^{2(k)})$, where $\mu^{(k)} = \frac{y^{(k)}+y^{(k+1)}}{2}$ which corresponds to the midpoint of the level set $L^{(k)}$, and $\sigma^{2(k)} = \left(y^{(k)} + y^{(k+1)}\right) \times S$, where S is a parameter denoted as *overlap*, which intuitively is proportional to the overlap in the distribution support of the level sets. We also put a log-normal hyper-prior on α such that

$$\alpha \sim \text{lognormal}(\mu_\alpha, \sigma_\alpha^2) \tag{32}$$

where $\mu_\alpha, \sigma^2_\alpha$ are specified according to the normal parametrisation of the lognormal and are fixed to 1 and 2 respectively. Altogether, the POMM prior on P is the following:

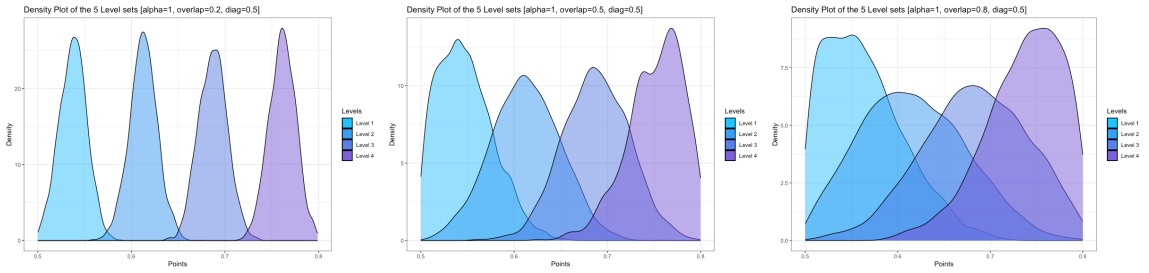
$$p_{ij} \in L^{(k)} \mid y^{(k)}, y^{(k+1)} \sim \text{Normal}\left(\mu^{(k)}, \sigma^{2(k)}\right) \mathbb{I}(0.5, \beta_{\max}) \tag{33}$$

$$\alpha \sim \text{Lognormal}\left(\mu_\alpha, \sigma^2_\alpha\right) \tag{34}$$

$$S \sim \sim \text{Lognormal}\left(\mu_S, \sigma^2_S\right) \tag{35}$$

and where the truncations $y^{(k)}$ are derived as in (**??**).

Insert here the plot for 3 different overlap values



(a) $\alpha = 1, S = 0.2$ (b) $\alpha = 1, S = 0.5$ (c) $\alpha = 1, S = 0.8$

FIGURE 3. Densities for $K = 5$ Level Sets, for different S values and $\alpha = 1$; the main diagonal is set to 0.5, and its collapsed density is not reported.

5. Estimation

For the moment, we want to infer just $\theta = \{z, P, \alpha, S\}$, meaning that we treat K as a known constant. The estimation strategy is a Hybrid MCMC algorithm. Since simulating from the conditional distribution $p(\theta_i|\theta_j, j \neq i)$ is unfeasible or computationally expensive, we substitute the simulation from the full conditional distribution with a simulation from a proposal distribution q_i. Referencing Muller's (1991) work, the Hybrid modification is as follows:

z

Algorithm 1 Metropolis-within-Gibbs MCMC

for $i = 1, \ldots, p$ `given` $\left(\theta_1^{(t+1)}, \ldots, \theta_{i-1}^{(t+1)}, \theta_i^{(t)}, \ldots, \theta_p^{(t)}\right)$ **do**

`Simulate`

$$\theta_i' \sim q_i\left(\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right) \tag{36}$$

Set $z' \leftarrow z$ with the i-th element replaced by `new_label`
Compute new victory probabilities $p_{z_i' z_j'}$ using z'
Compute probability ratio $log(r)$ using $p_{z_i' z_j'}$ and $p_{z_i z_j}$
Set $\alpha_r \leftarrow \min(1, r)$
Sample u from a uniform distribution on $(0, 1)$
if $u < \alpha_r$ **then**
  Update z to z'
  Update $p_{z_i z_j}$ to $p_{z_i' z_j'}$
  Increment $acc.count_z$
end if
Store $z_{current}$ in $z.container$
end for

5.0.1. *Adaptive algorithm.* The proposal distributions mentioned above are Normals $\left(\theta^{(t-1)}, \sigma_\theta^2\right)$. The proposal is accepted or rejected by evaluating the Metropolis ratio (to be precise, the logarithm of the Metropolis ratio). Th To answer these questions, an adaptive algorithm can be used. We proceed as follows. For each of the variables i ($1 \leq i \leq K + 3$), we create an associated variable ls_i giving the logarithm of the standard deviation to be used when proposing a normal increment to variable i. We begin with $ls_i = 0$ for all i (corresponding to unit proposal variance). After the n-th "batch" of 50 iterations, we update each ls_i by adding or subtracting an adaption amount $\delta(n)$. The adapting attempts to make the acceptance rate of proposals for variable i as close as possible to 0.44 (which is optimal for one-dimensional proposals in certain settings; cf. Roberts, Gelman, and Gilks 1997; Roberts and Rosenthal 2001). Specifically, we increase ls_i by $\delta(n)$ if the fraction of acceptances of variable i was more than 0.44 on the n-th batch, or decrease ls_i by $\delta(n)$ if it was less.

Condition (1.1) is satisfied provided $\delta(n) \to 0$; we take $\delta(n) = \min(0.01, n^{-1/2})$. Our approach is to specify a global maximal parameter value $M < \infty$, and restrict each ls_i to the interval $[-M, M]$. For a large class of target densities (which includes all those which are log-concave outside an arbitrary bounded region), this ensures (1.2) holds. In practice, the ls_i stabilize nicely, so the bound on M is not actually needed.

5.1. **Updating z.** To update z we propose a new label for each node, we evaluate the accept/reject move by computing the ratio r as follows:

$$r = \frac{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k'+1)}}{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{37}$$

$$= \frac{\prod_{i<j} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k'+1)}}{\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{38}$$

Passing to the log:

$$\begin{aligned}
log(r) &= \log\left(\prod_{i<j} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k' + 1)} \right) \\
&\quad - \log\left(\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k + 1)} \right) \\
&= \sum_{i<j} \Big(y_{ij} \cdot \log p_{z_i' z_j'} + (n_{ij} - y_{ij}) \cdot \log\left(1 - p_{z_i' z_j'}\right) \Big) \\
&\quad + \sum_{k=1}^{K} \left(\log\left(\Gamma(n_k' + \gamma_k)\right) - \log\left(\Gamma(\gamma_k)\right) - \log\left(\Gamma\left(n_k' + 1\right)\right) \right) \\
&\quad - \sum_{i<j} \left(y_{ij} \cdot \log p_{z_i z_j} + (n_{ij} - y_{ij}) \cdot \log\left(1 - p_{z_i z_j}\right) \right) \\
&\quad - \sum_{k=1}^{K} \left(\log\left(\Gamma(n_k + \gamma_k)\right) - \log\left(\Gamma(\gamma_k)\right) - \log\left(\Gamma\left(n_k + 1\right)\right) \right)
\end{aligned} \tag{39}$$

5.2. **Updating P.** To update P and α we propose a new label for each node, we evaluate the accept/reject move by computing the ratio r as follows:

Algorithm 2 Updating z step

for $i \leftarrow 1$ to N **do**
  Sample `new_label` from $1, ..., K$
  Set $z' \leftarrow z$ with the i-th element replaced by `new_label`
  Compute new victory probabilities $p_{z'_i z'_j}$ using z'
  Compute probability ratio $log(r)$ using $p_{z'_i z'_j}$ and $p_{z_i z_j}$
  Set $\alpha_r \leftarrow \min(1, r)$
  Sample u from a uniform distribution on $(0, 1)$
  if $u < \alpha_r$ **then**
    Update z to z'
    Update $p_{z_i z_j}$ to $p_{z'_i z'_j}$
    Increment $acc.count_z$
  end if
  Store $z_{current}$ in $z.container$
end for

$$r = \frac{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p'^{y_{ij}}_{z_i z_j} \cdot (1 - p'_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \left(\frac{1}{y'^{(k+1)} - y'^{(k)}} \right)^{|L'^{(k)}|}}{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p^{y_{ij}}_{z_i z_j} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \left(\frac{1}{y^{(k+1)} - y^{(k)}} \right)^{|L^{(k)}|}} \tag{40}$$

(41)

Passing to the log:

(42)

$$log(r) = \sum_{i<j} \left(y_{ij} \cdot \log p'_{z_i z_j} + (n_{ij} - y_{ij}) \cdot \log \left(1 - p'_{z_i z_j}\right) \right) - \sum_{k=1}^{K} |L'^{(k)}| \cdot \log \left(y'^{(k+1)} - y'^{(k)} \right)$$

(43)

$$- \sum_{i<j} \left(y_{ij} \cdot \log p_{z_i z_j} + (n_{ij} - y_{ij}) \cdot \log \left(1 - p_{z_i z_j}\right) \right) + \sum_{k=1}^{K} |L^{(k)}| \cdot \log \left(y^{(k+1)} - y^{(k)} \right)$$

Algorithm 3 Updating P step

$j \leftarrow 1$
while $j \leq N_{iter}$ **do**
    Sample α' from a truncated normal distribution
    Generate a new proposal matrix P'
    Compute new victory probabilities $p'_{z_i z_j}$ using P' and $z_{current}$
    Compute probability ratio $log(r)$ using $p'_{z_i z_j}$ and $p_{z_i z_j}$
    Set $\alpha_r \leftarrow \min(1, r)$
    Sample u from a uniform distribution on $(0, 1)$
    if $u < \alpha_r$ **then**
        Update α to α'
        Update P to P'
        Update $p_{z_i z_j}$ to $p'_{z_i z_j}$
        Increment $acc.count_p$
    end if
    Store P in $P.container$
    Store α in $\alpha.container$
    $j \leftarrow j + 1$
end while

6. Point Estimate, Model Selection, and inference

While algorithmic methods produce a single estimated partition, our model offers the entire posterior distribution across different node partitions. We are comparing the results from the simulation study via the following three main measures:

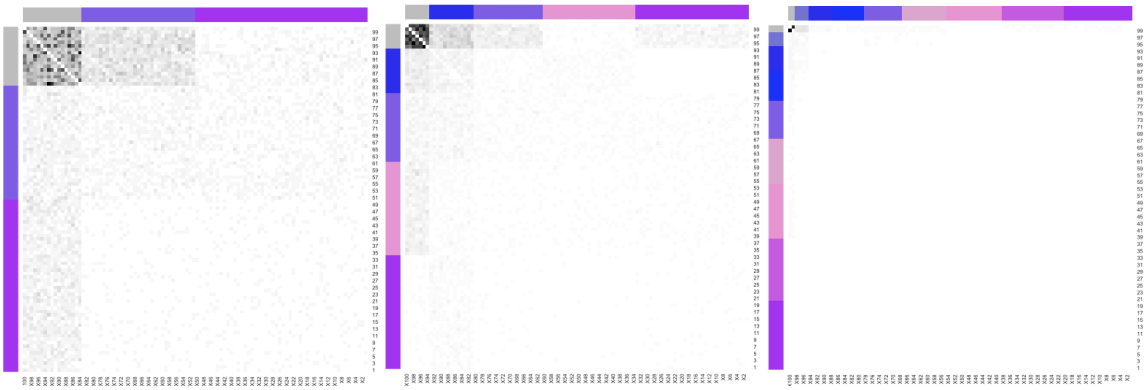
- Variation of Inforamation (VI): to fully utilise this posterior and engage in inference directly within the partition space, we adopt the decision-theoretic approach introduced by Wade and Ghahramani (2018) for block modeling. This involves summarizing posterior distributions using the variation of information (vi) metric, developed by Meilă (2007), which measures the distance between two clusterings by comparing their individual and joint entropies. The vi metric ranges from 0 to log2 V, where V represents the number of nodes. Intuitively, the vi metric quantifies the amount of information contained in two clusterings relative to the shared information between them. As a result, it decreases towards 0 as the overlap between two partitions increases. Refer to Wade and Ghahramani (2018) for a detailed exploration of the key properties of the vi metric. Within this framework, a formal Bayesian point estimate for z is obtained by selecting the partition with the lowest averaged vi distance from the other clusterings
- WAIC: While the WAIC yields practical and theoretical advantages and has direct connections with Bayesian leave-one-out cross-validation, thus providing a measure of edge predictive accuracy, the calculation of the WAIC only requires posterior samples of the log-likelihoods for the edges:$\log p(y_{ij}|z,P,\alpha) = y_{ij}\log p_{z_i,z_j} + (n_{ij} - y_{ij})\log(1-p_{z_i,z_j}), \quad i = 2,\dots,N, j = 1,\dots,i-1.$
- Misclassification error: predicting the group membership z_{N+1} of a new player may also be of interest. We can derive the estimate of the block probabilities for new players based on their early matches with some of the existing players.

$$\begin{aligned} p(z_{N+1} = k|\mathbf{Y}, y_{N+1}, \hat{z}) &\propto p(y_{N+1}|\mathbf{Y}, \hat{z}, z_{N+1} = k) \cdot p(z_{N+1} = k|\hat{z}) \\ &= p(y_{N+1}|\hat{z}, z_{N+1} = k) \cdot p(z_{N+1} = k|\hat{z}) \end{aligned} \tag{44}$$

where $p(z_{N+1} = k|\mathbf{Y}, y_{N+1}, \hat{z})$ is the posterior probability of the new node $N+1$ to belong to the block k, given the previously observed data Y, the new node's data y_{N+1} and the estimated labels $\hat{z}$. On the right hand side of the expression above, $p(y_{N+1}|\mathbf{Y}, \hat{z}, z_{N+1} = k)$ represents the likelihood of observing y_{N+1} given the previously observed data Y and the estimated labels $\hat{z}$, which, due to conditional independence, is the same as conditioning just on $\hat{z}$. Finally, $p(z_{N+1} = k|\hat{z})$ represents the prior probability of label k for the new node $N+1$ given $\hat{z}$, which we can approximate with the relative size of the blocks n_k.

7. Simulation Study from the Simple Model

In order to evaluate how well our model performs in a situation similar to our intended use, and measure its advantages compared to the best existing alternatives, we generated three simulated tournaments with 100 players from the Simple Model. We want to compare how it performs compared to the POMM extension and other state-of-the-art alternatives. Each tournament had a different number of blocks in the underlying structure. We set the total number of games $M := 0.5 * \sum_{i,j}^{N} n_{ij} = \sum_{i,j}^{N} y_{ij} = 4000$, which is the average number of matches played in one year of tennis tournaments. We divided the players into three, five and nine blocks ($K = 3, 5, 9$ respectively). In Figure (**??**), we display the three simulated tournaments, where the difficulty of accurately determining the group membership increases as the number of games increases with the number of blocks.

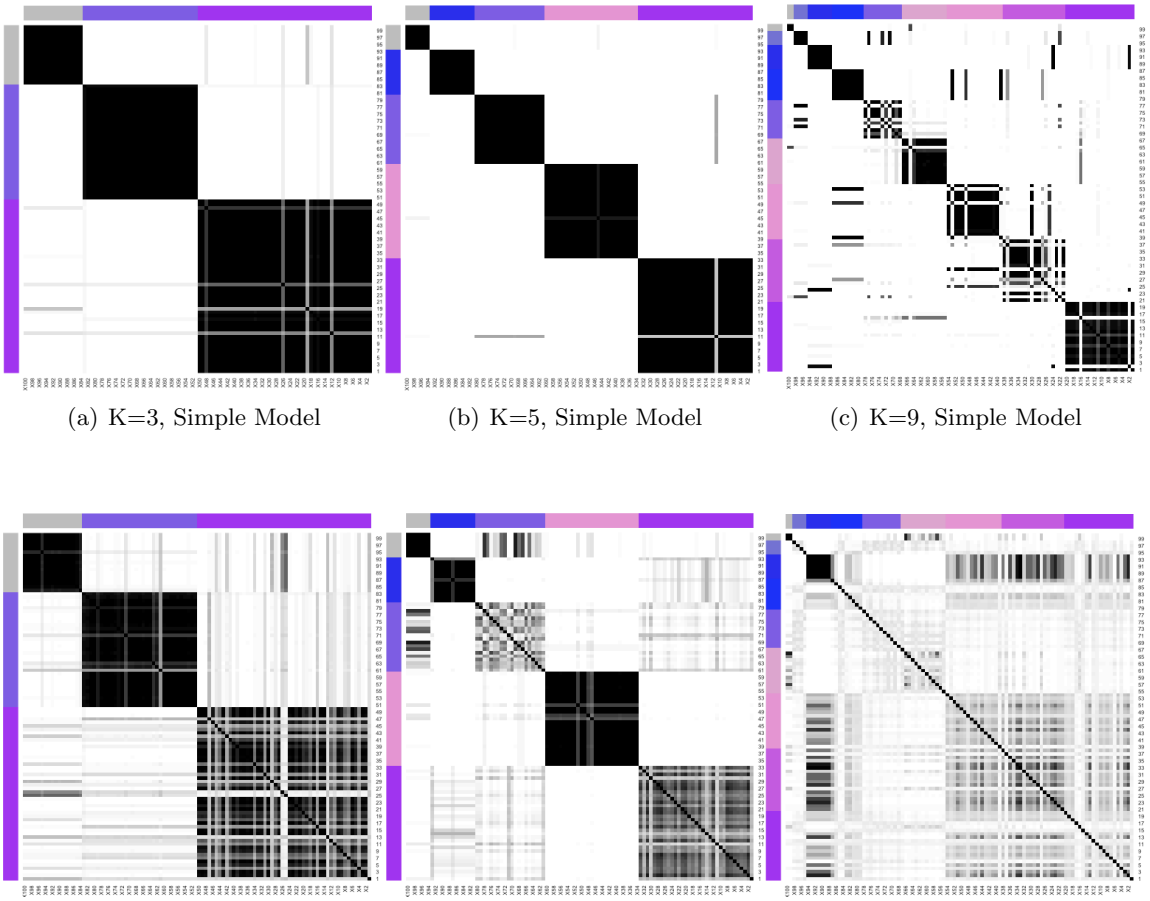


(a) K=3, Simple Model (b) K=5, Simple Model (c) K=9, Simple Model

Figure 4. Adjacency Matrices simulated via the Simple Model

We compare the performance of the Simple model with the POMM one. We fixed arbitrarily $\beta_{\max} = .75$. In table (**??**) we report the results of the simulation. In the three cases, for the Simple and the POMM model, we compare the WAIC, the VI distance and the misclassification error, obtained by considering 100 new incoming players which get to play just with 10 players each. We also compare the labels estimated against the regularised spectral clustering algorithm and the Louvain algorithm. The Simple model is the best performing relative to the other three alternatives.

In figure (**??**) we report the estimated co-clustering matrices resulting from the simulation process.



(a) K=3, Simple Model (b) K=5, Simple Model (c) K=9, Simple Model

(d) K=3, POMM Model (e) K=5, POMM Model (f) K=9, POMM Model

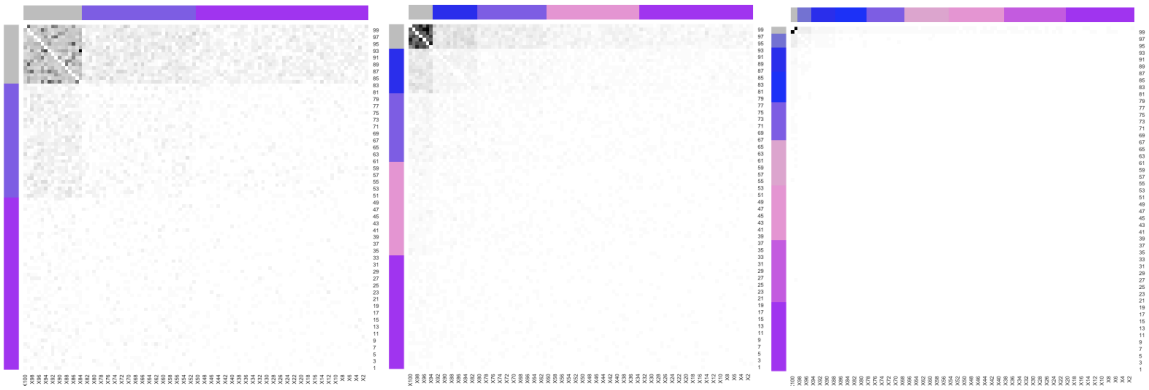
FIGURE 5. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

TABLE 1. Y_{ij} drawn from a simple model

Method	WAIC			VI distance			Error		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Simple model	1805.5	1805.8	1336.5	0	0.12	1.18	0.18	0.61	0.76
POMM model	1590.0	1454.3	954.7	0.51	0.65	2.97	1	0.85	0.92
Spectral Clustering	-	-	-	1.37	0.58	2.82	-	-	-
Louvain algorithm	-	-	-	4.06	2.78	3.79		-	-

8. Simulation Study from the POMM Model

In this section we reverse the exercise performed in previous one. Before we were simulating from the Simple model, now we are doing the same, with similar parameters ($K = 3, 5, 9, M = 4000$ and $\beta_{\max} = .75$). Here are the results.



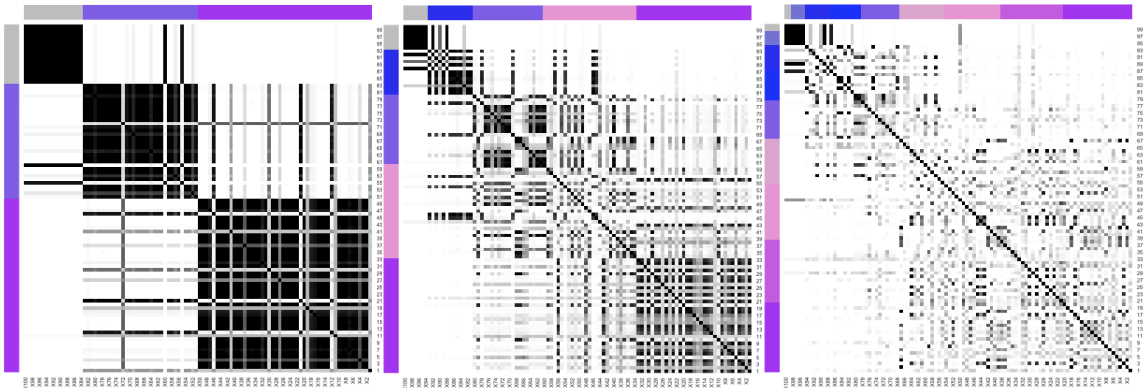
(a) K=3, POMM Model (b) K=5, POMM Model (c) K=9, POMM Model

Figure 6. Adjacency Matrices simulated via the POMM Model

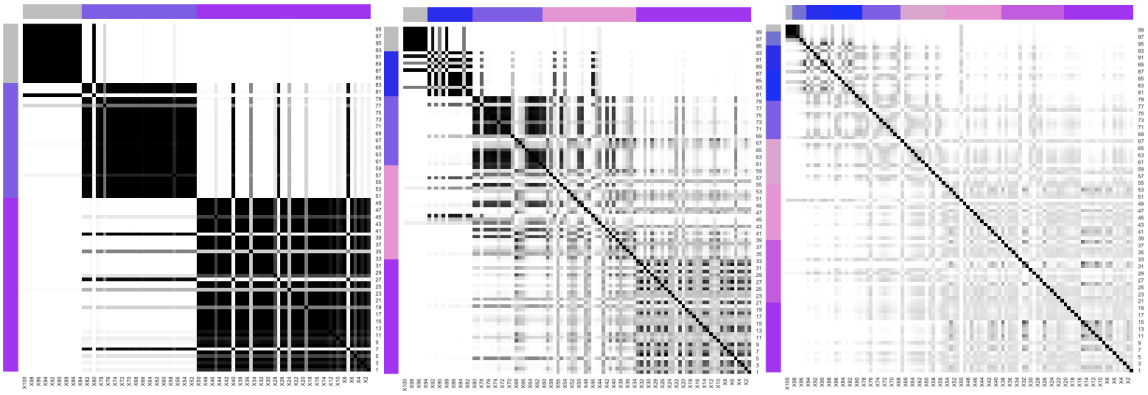
In table (**??**) we report the results of the simulation. As before, for the Simple and the POMM model, we compare the WAIC, the VI distance and the misclassification error, obtained $N_new = 100$. Also here we compare clustering performance against that of the regularised spectral clustering algorithm and the Louvain algorithm. The POMM model is the best performing relative to the other three alternatives.

Table 2. Y_{ij} drawn from the POMM model

Method	WAIC			VI distance			Error		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Simple model	2353.7	2019.1	1692.0	0.83	2.37	3.93	0.90	0.81	0.69
POMM model	2438.5	1992.7	1560.9	0.56	2.00	3.21	0.26	0.39	0.6
Spectral Clustering	-	-	-	0.95	1.85	-	-	-	-
Louvain algorithm	-	-	-	3.94	4.26	-	-	-	-



(a) K=3, Simple Model Estimates (b) K=5, Simple Model Estimates (c) K=9, Simple Model Estimates



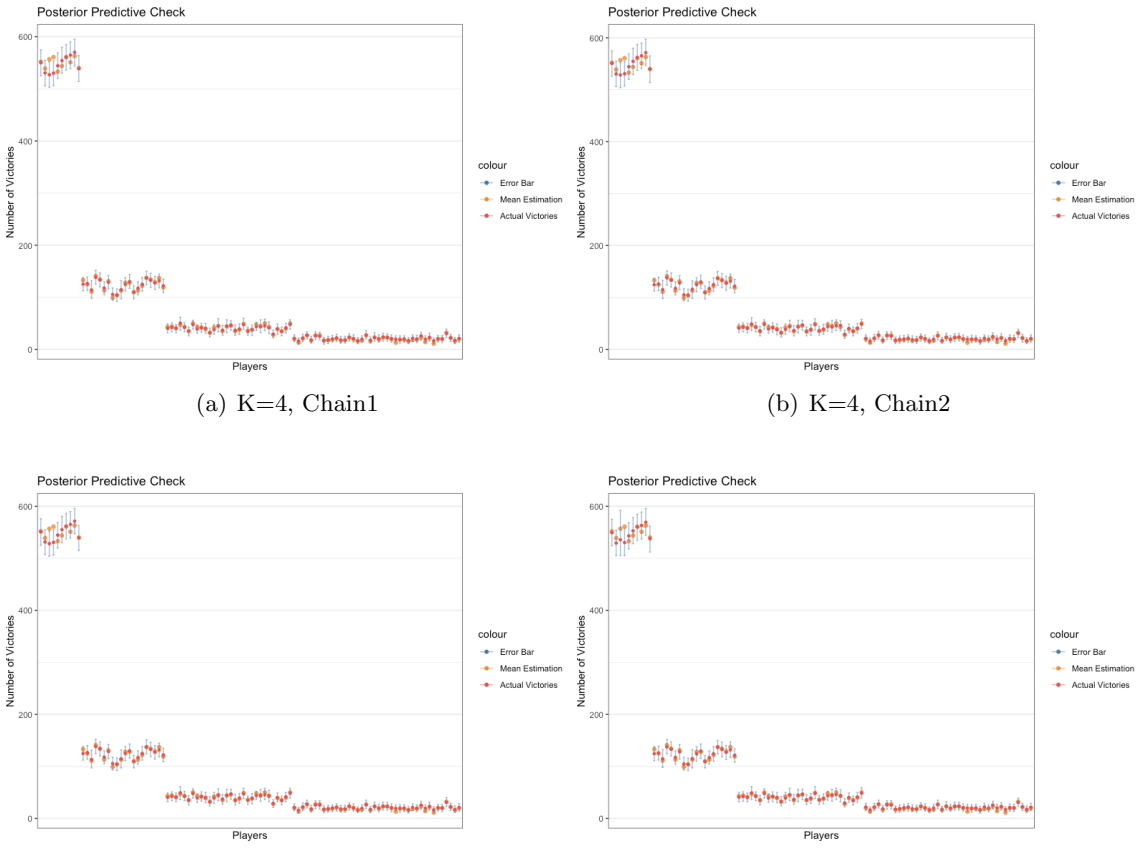
(d) K=3, POMM Model Estimates (e) K=5, POMM Model Estimates (f) K=9, POMM Model Estimates

FIGURE 7. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

8.1. **Diagnostic Checks.** In a Bayesian framework, especially when dealing with MCMC methods, it is important to check whether there has been convergence of the distribution to the stationary one and to check the overall performance of the method.

Here we report several measures to assess the convergence of the estimation procedure.

We simulated according to the same procedure as before a tournament from the POMM prior model, with $N = 100$,$K = 4$, $M = 4000$, $\alpha = 1$, and $\beta_{\max} = 0.8$.In this estimation procedure we run four different MCMC chains for $N_iter = 20000$ according to the POMM prior model($K = 4, \beta_{\max} = 0.8$,; to fasten the computations, the code is parallelised on 4 different cores. The results that we get are therefore regarding the 4 different MCMC chains.



(a) K=4, Chain1

(b) K=4, Chain2

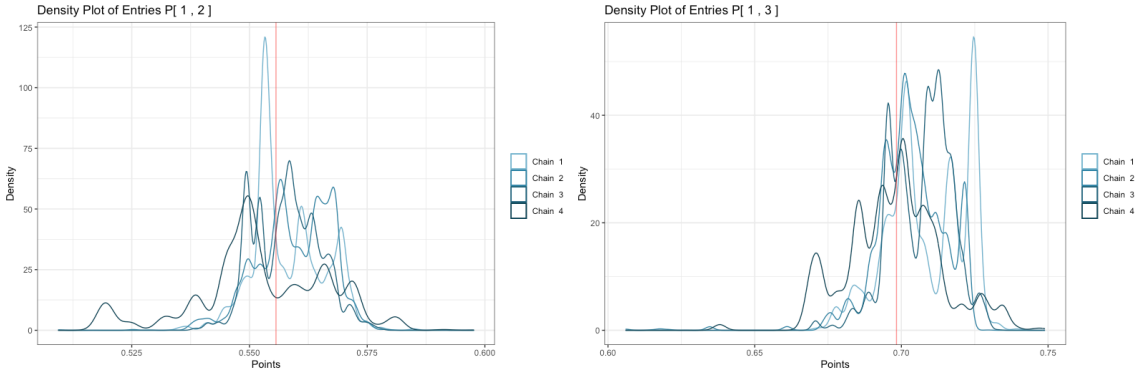
(c) K=4, Chain3

(d) K=4, Chain4

FIGURE 8. Posterior predictive check: simulating from the estimated parameters

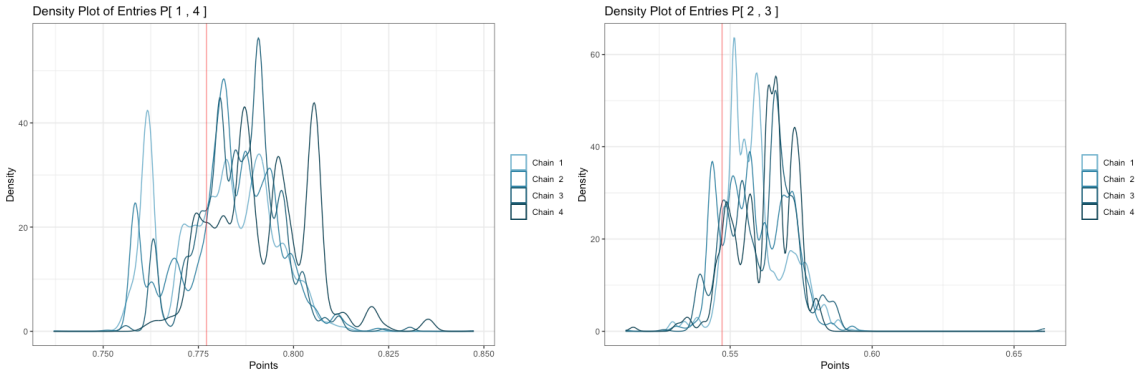
8.1.1. *Posterior Predictive Check.*

8.1.2. *Density Plot.*



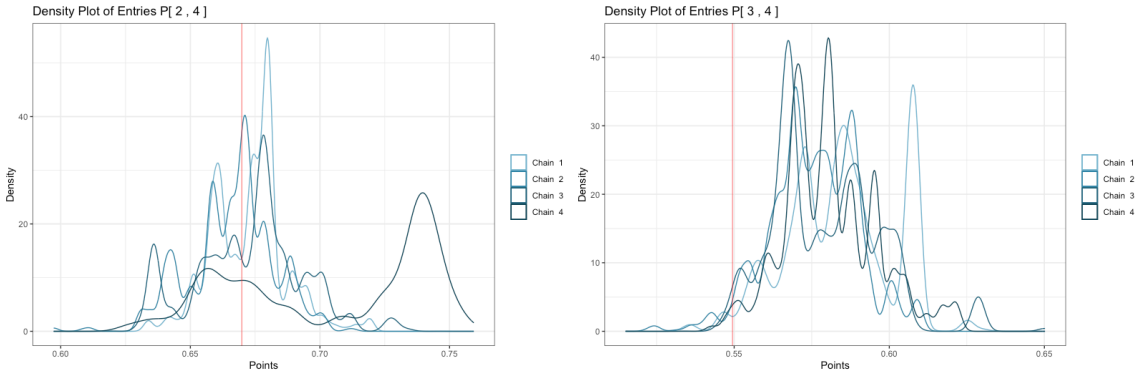
(a) K=4, Chain1-4, $P_{1,2}$

(b) K=4, Chain1-4, $P_{1,3}$



(c) K=4, Chain1-4, $P_{1,4}$

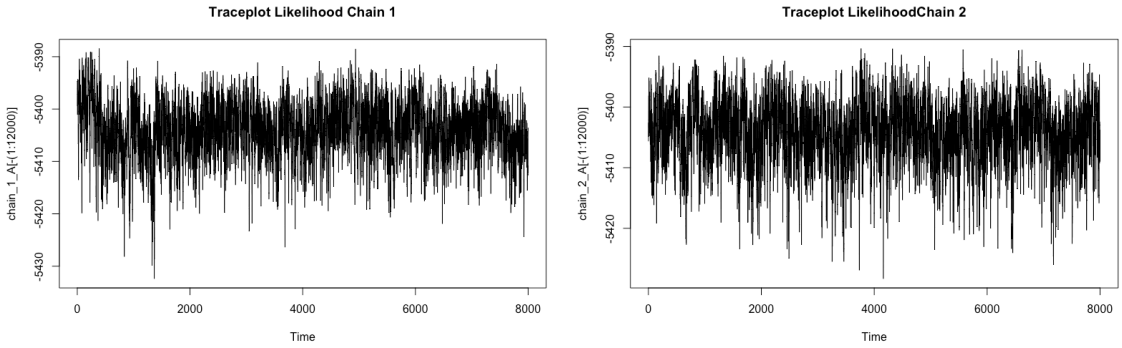
(d) K=4, Chain1-4, $P_{2,3}$



(e) K=4, Chain1-4, $P_{2,4}$

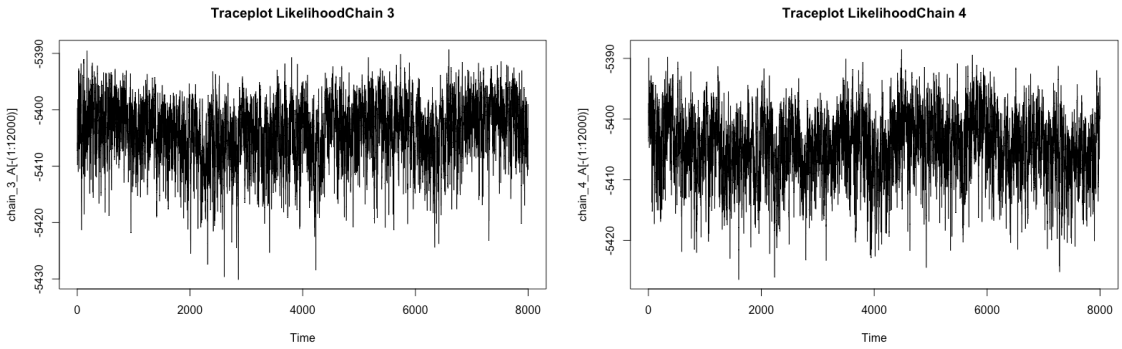
(f) K=4, Chain1-4, $P_{3,4}$

FIGURE 9. Density plot for each P entry: in red the true P value. Different lines correspond to different chains



(a) K=4, Chain1

(b) K=4, Chain2



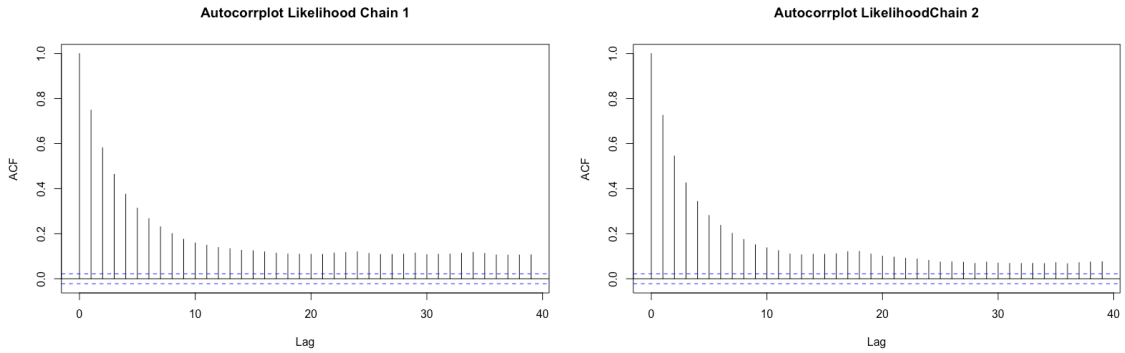
(c) K=4, Chain3

(d) K=4, Chain4

FIGURE 10. Traceplot of the likelihood at each iteration for the 4 chains, namely $\sum_{ij} \log\left(p\left(y_i j|\hat{z}^{(t)}, \hat{P}^{(t)}\right)\right)$

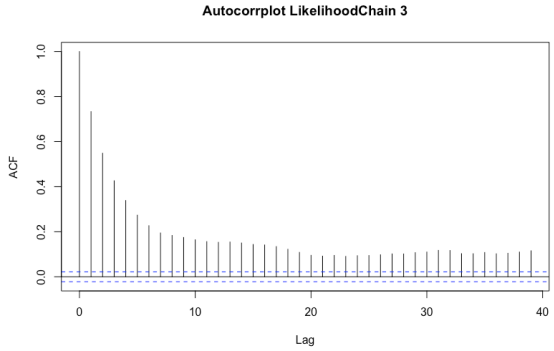
8.1.3. *Traceplot.*

8.1.4. *Autocorrelation Plot.*

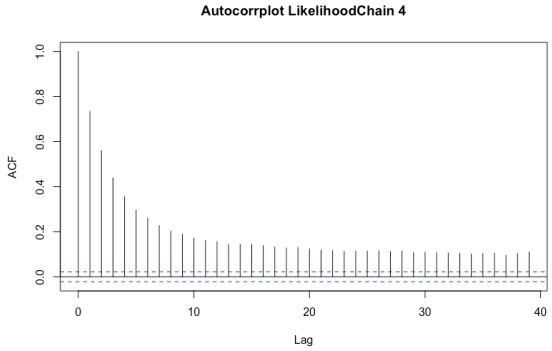


(a) K=4, Chain1

(b) K=4, Chain2



(c) K=4, Chain3



(d) K=4, Chain4

FIGURE 11. Autocorrplot of the likelihood at each iteration for the 4 chains, namely