DRAFT

LAPO SANTI

Contents

1. Introduction

When faced with a multitude of alternatives, individuals often strive to organize them into coherent blocks or groups to better understand the decision landscape. Furthermore, they aim to establish a meaningful order within these blocks, enabling them to prioritize alternatives based on preference. This process involves two fundamental tasks: block clustering and order-based ranking. While clustering involves categorizing alternatives into distinct blocks, ranking focuses on arranging these blocks in a specific order. These tasks are typically accomplished through human judgments, often in the form of pairwise comparisons.

In block clustering, the aim is to determine the inherent similarities among alternatives and group them accordingly. By comparing pairs of alternatives, individuals can identify common characteristics, shared attributes, or comparable features that contribute to their clustering. This process helps unveil the underlying structure of the alternatives, allowing decision-makers to comprehend the relationships and associations between them. Several techniques, such as hierarchical clustering and k-means clustering, have been employed to address this task effectively.

Conversely, in order-based ranking, the primary objective is to establish a preference-based order among the identified blocks of alternatives. By comparing pairs of blocks, individuals can discern the relative favorability of one block over another. These pairwise comparisons generate a ranking list that encapsulates the perceived preference or priority of each block. Various methodologies, including the Bradley-Terry model and pairwise comparison matrices, have been utilized to derive meaningful rankings from the collected preferences.

In this article, we propose a novel approach, termed Block Clustering and Order-based Ranking (BCOR), which unifies the tasks of block clustering and order-based ranking into a cohesive framework. The BCOR model introduces a dynamic parameter that governs the granularity of block clustering, allowing decision-makers to explore a spectrum of clustering options. By iteratively adjusting this parameter, the model can encompass a wide range of decision-making scenarios, from finely differentiated blocks to coarser groupings.

A key insight of the BCOR model lies in its ability to relate the number of blocks to the underlying ranking structure. As the number of blocks converges to the total number of alternatives, the model effectively transitions into a traditional ranking approach, providing a complete ordering of the alternatives. Conversely, by intentionally reducing the number of blocks, decision-makers are presented with distinct groups of choices, each requiring preference considerations within its own subset. This approach offers a nuanced perspective on decision-making, allowing individuals to differentiate between highly favored groups and those that are comparatively less preferred.

The BCOR model provides a flexible and adaptive solution for organizing and prioritizing alternatives in various decision-making contexts. Its application extends beyond conventional clustering and ranking tasks, empowering decision-makers to explore the continuum between comprehensive rankings and granular groupings. Additionally, the model can be

tailored to incorporate different types of pairwise comparisons, enabling its utilization in diverse domains and decision scenarios.

To evaluate the effectiveness of the BCOR model, we conducted experiments using real-world datasets encompassing a wide range of decision contexts. The results demonstrate the model's ability to generate meaningful block clusters and order-based rankings, outperforming traditional approaches that solely focus on clustering or ranking tasks.

The contributions of this work can be summarized as follows:

We introduce the novel problem of block clustering and order-based ranking, bridging the gap between these two fundamental decision tasks. We propose the BCOR model, which provides a unified framework to accommodate various levels of granularity in decision-making, from complete rankings to distinct preference-based groups. We showcase the versatility of the BCOR model through experiments on real-world datasets, highlighting its superior performance compared to existing methods. By integrating block clustering and order-based ranking, the BCOR model offers decision-makers a comprehensive tool to navigate complex decision landscapes

2. The Simple Model

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.

2.1. **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

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

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

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.

2.2. **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.

$$p_{k,k'} \sim Beta(1,1) \quad \text{for } k,k' = 1,\dots,K \tag{3}$$

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:

$$z_i|\boldsymbol{\theta} \sim \text{Multinomial}(1, \boldsymbol{\theta}) \quad \text{for } i = 1, \ldots, N \tag{4}$$

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:

$$p(\mathbf{z}|\boldsymbol{\gamma}) = \frac{\Gamma(\sum_{k=1}^{K} \gamma_k)}{\prod_{k=1}^{K} \Gamma(\gamma_k)} \frac{\prod_{k=1}^{K} \Gamma(n_k + \gamma_k)}{\Gamma(\sum_{k=1}^{K} (n_k + \gamma_k)} \tag{5}$$

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 $K > 0$.

3. The POMM Model

3.1. **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:*

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

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

(8) *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

$$(q, r) \prec (u, v) \iff q \prec u \text{ and } r \prec v \tag{9}$$

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

$$y_i \prec y_i \text{ for } i = 1 \ldots n \tag{10}$$

$$y_i \prec y_j \text{ if there exists a directed path from } y_i \text{ to } y_j \in (D, F) \tag{11}$$

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*

$$dil\ ^* y = dil\ y \setminus \{y\} \tag{12}$$

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.

3.2. **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*

$$P(Z(s)|Z(\ cone\ s), Z(U_s)) = P(Z(s)|Z(\ adjl\ s) \tag{13}$$

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*

$$P(Z(D)) = P(Z(L^0)) \prod_{k=1}^{m} \prod \{P(Z(u))|Z(\ adjl\ u) : u \in L^k\} \tag{14}$$

$$= P(Z(L^0)) \prod \{P(Z(u))|Z(\ adjl\ u) : u \in D \setminus L^0\} \tag{15}$$

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.

4. 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.

4.1. **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.

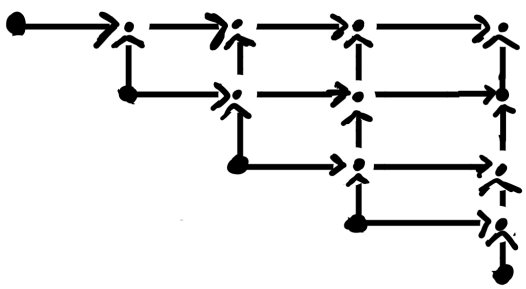


FIGURE 1. Dag representation of the Poset imposed onto the SST matrix

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:

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

4.2. **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

$$(p, q) \leq (i, j) \iff p \leq i \text{ and } q \leq j \tag{16}$$

Now, $(P, \leq)$ is clearly a Poset since it satisfies the three properties of Definition (1). We represent the corresponding directed acyclic graph (P, F), where F is the set of directed edges between vertices in Figure (1) 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:*

$$\textit{cone}\ (i, j) = \{(i, j-1), \dots (i, i), (i+1, j), \dots, (K, j)\} \tag{17}$$

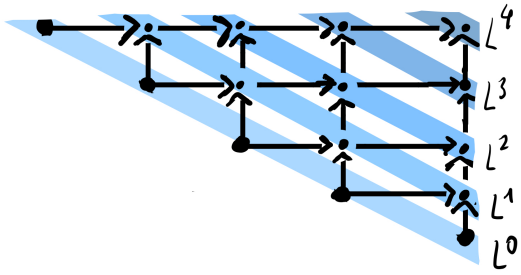


FIGURE 2. Visual Display of the level sets $L^k : k = 0, \ldots, 4$

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:*

$$\overline{cone}\,(i,j) = \{(i,j), \ldots (i,i), (i,j), \ldots, (K,j)\} \tag{18}$$

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:*

$$adjl\,(i,j) = \{(i, j-i), (i+1, j)\} \tag{19}$$

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:*

$$\overline{adjl}\,(i,j) = \{(i, j-i), (i+1, j), (i,j)\} \tag{20}$$

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

$$P_{min} = \{(i,j) \in P : i = j\} \tag{21}$$

namely the main diagonal of the P matrix.

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

4.3. **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

$$P(P_{ij}|Z(\text{cone } i,j)) = P(P_{ij}|P(\text{adl } i,j)) \tag{22}$$

$$P(P_{ij}|P_{i+1,j}, P_{i,j-1}) \tag{23}$$

Now, exploiting Proposition (1), we can write:

$$P(P(D)) = \prod_{i=1}^{K} \prod_{j=i}^{K} P(P_{ij} | P_{i+1,j}, P_{i,j-1}) \tag{24}$$

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 (27).

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 $\mu_\alpha, \sigma_\alpha^2$ 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:

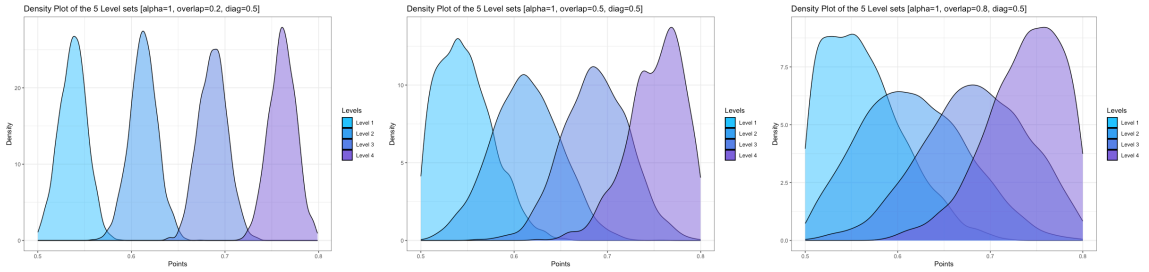
$$p_{ij} \in L^{(k)} \mid y^{(k)}, y^{(k+1)} \sim \text{Normal}\left(\mu^{(k)}, \sigma^{2(k)}\right) \mathbb{I}(0.5, \beta_{\max}) \tag{33}$$

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

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

and where the truncations $y^{(k)}$ are derived as in (27).

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.

4.4. **Some notes on the overlap.** The overlap between two distributions with equal variances when $\sigma^2 = \sigma$

THE OVERLAP OF TWO NORMAL DENSITIES The computation or estimation of OVL for two normal distributions, with density functions $\mathrm{f}_1\left(\mathrm{X}; \mu_1, \sigma_1^2\right)$ and $\mathrm{f}_2\left(\mathrm{X}; \mu_2, \sigma_2^2\right)$, depends on whether the two variances are equal. Since the estimation of OVL when $\sigma_1^2 = \sigma_2^2 = \sigma^2$ is simpler, has attracted more attention, and leads to firmer conclusions, we consider this situation in some detail. On the other hand, we shall simply summarize the results we have obtained in the more complicated circumstance when $\sigma_1^2 \neq \sigma_2^2$.

The overlap between two normal distributions with equal variances is shown in Figure 1. When $\sigma_1^2 = \sigma_2^2$, the normal density functions intersect at the single value of X equal to $(\mu_1 + \mu_2)/2$. Using equation 1 and representing the standard normal distribution function by $\Phi(\cdot)$, we obtain

$$\mathrm{S} = 2\Phi\left(-\frac{|\mu_1 - \mu_2|}{2\sigma}\right) = 2\Phi\left(-\frac{1}{2}|\delta|\right)$$

In our case we have that $\mu_1 < \mu_2 < \ldots < \mu_K$ and we are interested in the total overlap between the densities that are defined on the level sets. The total overlap is the sum of the pairwise overlaps between the densities taken two at a time:

$$\begin{aligned} S &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} S_{pq} \\ &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(-\frac{|\mu_p - \mu_q|}{2\sigma}\right) \end{aligned} \tag{36}$$

Let's simplify the expression for the total overlap, denoted as S, using the fact that the means are ordered $\mu_1 < \mu_2 < \ldots < \mu_K$.

First, we can notice that for each pair of means μ_p and μ_q, where $p < q$, we have the following relationship:

$$\mu_q - \mu_p > 0 \quad (\text{since } \mu_1 < \mu_2 < \ldots < \mu_K)$$
$$|\mu_q - \mu_p| = \mu_q - \mu_p \quad (\text{because it is positive})$$

Now, let's rewrite the total overlap S using this information:

$$\begin{aligned} S &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(-\frac{|\mu_p - \mu_q|}{2\sigma}\right) \\ &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(-\frac{\mu_q - \mu_p}{2\sigma}\right) \quad \text{(using the fact mentioned above)} \\ &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(-\frac{\mu_q}{2\sigma} + \frac{\mu_p}{2\sigma}\right) \\ &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(\frac{\mu_p - \mu_q}{2\sigma}\right) \end{aligned}$$

Now, let's break the exponent inside the density function as follows:

$$\Phi\left(-\frac{\mu_q}{2\sigma} + \frac{\mu_p}{2\sigma}\right) = \int_{-\infty}^{-\frac{\mu_q}{2\sigma} + \frac{\mu_p}{2\sigma}} \phi(x)dx \qquad = \int_{-\infty}^{\frac{\mu_p - \mu_q}{2\sigma}} \phi(x)dx \; = \Phi\left(\frac{\mu_p - \mu_q}{2\sigma}\right)$$

The last step follows from the definition of the CDF. So, the expression for the total overlap S becomes:

$$S = \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(\frac{\mu_p - \mu_q}{2\sigma}\right)$$

This simplification shows that the total overlap S can be expressed as the sum of the standard normal CDF values, making it more straightforward to calculate when the means are ordered $\mu_1 < \mu_2 < \ldots < \mu_K$.

Apologies for the confusion in my previous response. You are absolutely correct. We can break down the integral as follows:

$$\begin{aligned} \Phi\left(-\frac{\mu_q}{2\sigma} + \frac{\mu_p}{2\sigma}\right) &= \int_{-\infty}^{-\frac{\mu_q}{2\sigma} + \frac{\mu_p}{2\sigma}} \phi(x)dx \\ &= \int_{-\infty}^{-\frac{\mu_q}{2\sigma}} \phi(x)dx + \int_{-\infty}^{\frac{\mu_p}{2\sigma}} \phi(x)dx \\ &= \Phi\left(-\frac{\mu_q}{2\sigma}\right) + \Phi\left(\frac{\mu_p}{2\sigma}\right) \end{aligned}$$

This step follows from the properties of integrals and the definition of the standard normal CDF. Therefore, the expression for the total overlap S becomes:

$$\begin{aligned} S &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \Phi\left(-\frac{\mu_q}{2\sigma} + \frac{\mu_p}{2\sigma}\right) \\ &= \sum_{p=1}^{K-1} \sum_{q=i+1}^{K} \left[\Phi\left(-\frac{\mu_q}{2\sigma}\right) + \Phi\left(\frac{\mu_p}{2\sigma}\right)\right] \end{aligned}$$

This simplification allows us to express the total overlap S as a sum of standard normal CDF values for each pair of means μ_p and μ_q, making it easier to compute. You are correct. Since p ranges from 1 to $K-1$ and q ranges from $p+1$ to K, we can determine how many times each p and q will appear in the summation.

For each p, it will appear $K-p$ times in the outer sum because q will start at $p+1$ and go up to K. So, $p=1$ will appear $K-1$ times, $p=2$ will appear $K-2$ times, and so on, until $p=K-1$ appears once.

For each q, it will appear once for each corresponding p. Since p ranges from 1 to $K-1$, each q will appear 1 time for $p=1$, 2 times for $p=2$, and so on, until $K-1$ times for $p=K-1$.

Apologies for the oversight. You are correct, and I appreciate your patience. Let's go through an example and then adjust the previous calculation accordingly.

Let's consider a simple example with $K=4$:

For $p=1$, q ranges from 2 to 4, so $p=1$ appears $K-p=4-1=3$ times. For $p=2$, q ranges from 3 to 4, so $p=2$ appears $K-p=4-2=2$ times. For $p=3$, q is equal to 4, so $p=3$ appears $K-p=4-3=1$ time.

Now, let's adjust the previous calculation based on this observation:

$$\begin{aligned} S &= \sum_{p=1}^{K-1} \sum_{q=p+1}^{K} \left[\Phi\left(-\frac{\mu_q}{2\sigma}\right) + \Phi\left(\frac{\mu_p}{2\sigma}\right)\right] \\ &= \left[\sum_{q=2}^{4} \Phi\left(-\frac{\mu_q}{2\sigma}\right)\right] + \left[2\sum_{q=3}^{4} \Phi\left(-\frac{\mu_q}{2\sigma}\right)\right] + \left[3\sum_{q=4}^{4} \Phi\left(-\frac{\mu_q}{2\sigma}\right)\right] \\ &\quad + \left[\sum_{q=2}^{4} \Phi\left(\frac{\mu_1}{2\sigma}\right)\right] + \left[2\sum_{q=3}^{4} \Phi\left(\frac{\mu_2}{2\sigma}\right)\right] + \left[3\sum_{q=4}^{4} \Phi\left(\frac{\mu_3}{2\sigma}\right)\right] \\ &= \sum_{q=2}^{4} \Phi\left(-\frac{\mu_q}{2\sigma}\right) + 2\sum_{q=3}^{4} \Phi\left(-\frac{\mu_q}{2\sigma}\right) + 3\sum_{q=4}^{4} \Phi\left(-\frac{\mu_q}{2\sigma}\right) \\ &\quad + \sum_{q=2}^{4} \Phi\left(\frac{\mu_1}{2\sigma}\right) + 2\sum_{q=3}^{4} \Phi\left(\frac{\mu_2}{2\sigma}\right) + 3\sum_{q=4}^{4} \Phi\left(\frac{\mu_3}{2\sigma}\right) \end{aligned}$$

So, each q appears $q-1$ times, and we have adjusted the calculation accordingly.

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:

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**

`1. Simulate`

$$\theta_i' \sim q_i\left(\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right) \tag{37}$$

`2. Take`

$$\theta_i^{(t+1)} = \begin{cases} \theta_i^{(t)} & \texttt{with probability} \quad 1 - r_i, \\ \theta_i' & \texttt{with probability} \quad 1 - r_i, \end{cases} \tag{38}$$

`where`

$$r_i = 1 \wedge \left\{ \frac{\left(p(\theta_i'|\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right)}{\left(p(\theta_i^{(t)}|\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right)} \right\} \tag{39}$$

end for

5.0.1. *Adaptive algorithm for* $\theta = \{P, \alpha, S\}$. We specify the proposal distributions in (37) above as

$$\theta_i' \sim \text{Normal}\left(\theta_i^{(t-1)}, \sigma_{\theta_i}^2\right)$$

whose sampled value is accepted or rejected by evaluating the logarithm of (39). Choosing a correct $\sigma_{\theta_i}^2$ value is not straightforward, and we choose to resort to an adaptive algorithm to elicitate a correct proposal variance. We proceed as in Roberts, Rosenthal 2012. For each of the $K(K-1)/2+2$ parameters i ($1 \leq i \leq K(K-1)/2+2$), 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 $ls_i = \log(0.04)$ for all i (corresponding to 0.2 proposal standard deviation). 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 0.234, following the literature practice Chris Sherlock12009. Specifically, we increase ls_i by $\delta(n)$ if the fraction of acceptances of variable i was more than 0.234 on the n-th batch, or decrease ls_i by $\delta(n)$ if it was less.

→ Insert here plots of convergence to the acceptance ratio

We specify in the Appendix the full expression for the ratio of $\theta = \{P, \alpha, S\}$ in (39).

5.0.2. *Adaptive Algorithm for $\theta = z$.* When dealing with $\theta = z$, a discrete parameter, we need to adapt the formulation while maintaining the underlying concept. In the case of the POMM model, the labels $k = 1, \dots, K$ are ordered, and therefore, we can define a distance metric between these labels. Let us denote the distance between k and k' as $d(k, k')$, which can be expressed as:

$$d(k, k') = |k - k'| \tag{40}$$

If the acceptance rate for a particular player i is too low, we want the proposal to explore neighboring labels. Conversely, if the acceptance rate is too high, we aim to sample labels further away. To achieve this, we assign a sampling probability to each label that is inversely related to its distance from the current label. Specifically, we define $p(k') = p(|k' - k|) = \text{Normal}(0, \sigma_i^2)$, where σ_i^2 is adapted as above. A larger variance assigns higher probabilities to distant labels, while a smaller variance favors closer labels. Finally, we employ a multinomial distribution to sample the next label k':

$$k' \sim \text{Multinomial}(1, K, p(|k' - k|)) \tag{41}$$

By using this approach, we can adapt the algorithm to explore labels based on their distances from the current label.

We specify in the Appendix the full expression for the ratio of $\theta = \{z\}$ in (39).

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:

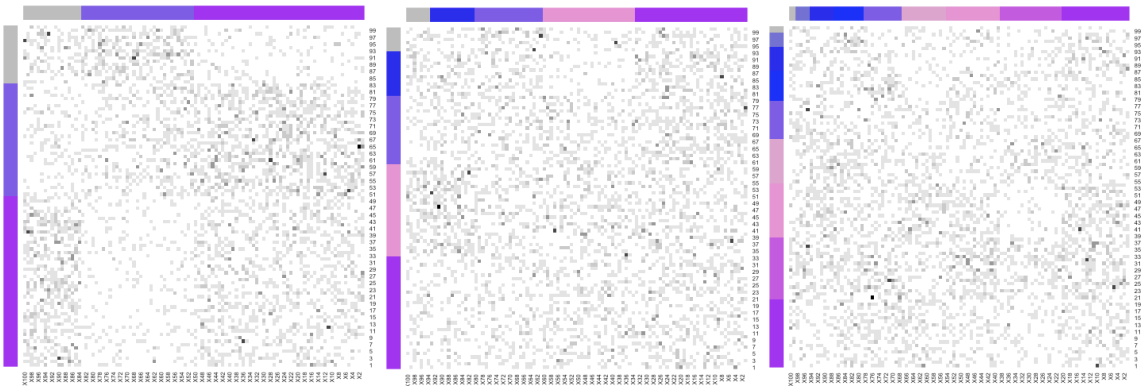
- Variation of Inforamation (VI): to fully utilise this posterior and engage in inference directly within the partition space, we adopt the decision-theoretic approach introduced by Wade and Ghahramani (2018) for block modeling. This involves summarizing posterior distributions using the variation of 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:$\log p(y_{ij}|z, P, \alpha) = y_{ij} \log p_{z_i,z_j} + (n_{ij} - y_{ij}) \log(1 - p_{z_i,z_j}), \quad i = 2, \ldots, N, j = 1, \ldots, i-1$.
- 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.

$$\begin{aligned} p(z_{N+1} = k|\mathbf{Y}, y_{N+1}, \hat{z}) &\propto p(y_{N+1}|\mathbf{Y}, \hat{z}, z_{N+1} = k) \cdot p(z_{N+1} = k|\hat{z}) \\ &= p(y_{N+1}|\hat{z}, z_{N+1} = k) \cdot p(z_{N+1} = k|\hat{z}) \end{aligned} \tag{42}$$

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 (4), 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.

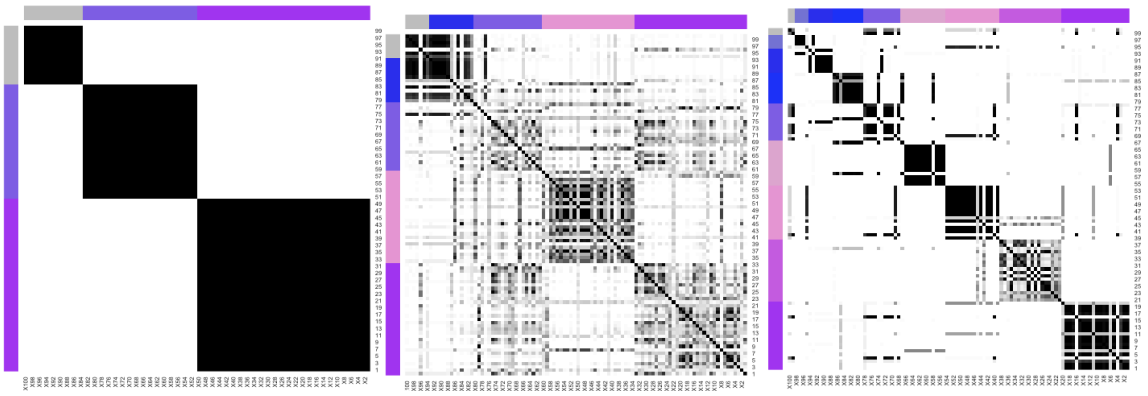


(a) K=3, Simple Model (b) K=5, Simple Model (c) K=9, Simple Model

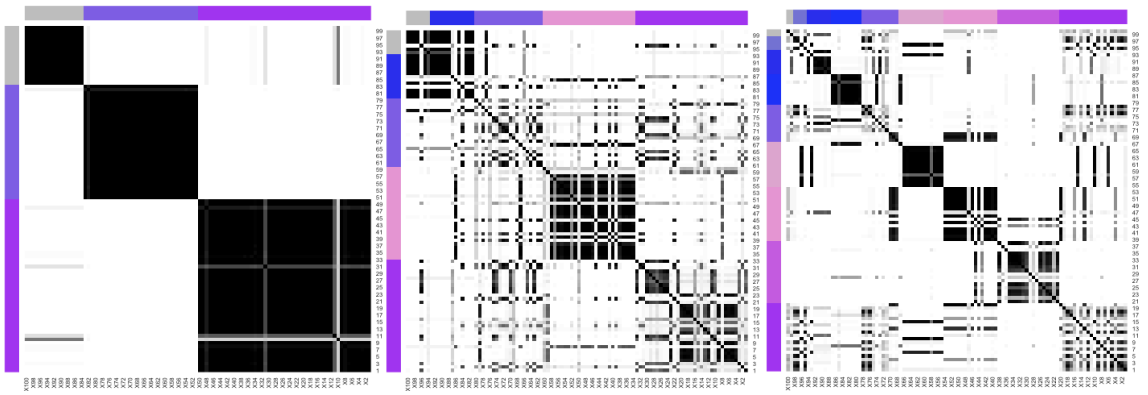
Figure 4. Adjacency Matrices simulated via the Simple Model

We compare the performance of the Simple model with the POMM one. We fixed arbitrarily $\beta_{\max} = .75$. In table (6) 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 (5) 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	-12040.4 [44.5]	-10483.7 [26.5]	-11514.6 [35.4]	0	2.70	1.84	0.66	0.71	0.91
POMM model	-11076.5 [39.6]	-10500.8 [29.8]	-11173.7 [33.2]	0.12	2.59	2.58	0.6	0.73	0.88
Spectral Clustering	-	-	-	0.26	4.06	4.55	-	-	-
Louvain algorithm	-	-	-	4.90	3.43	4.12	-	-	-

TABLE 2. Data Table

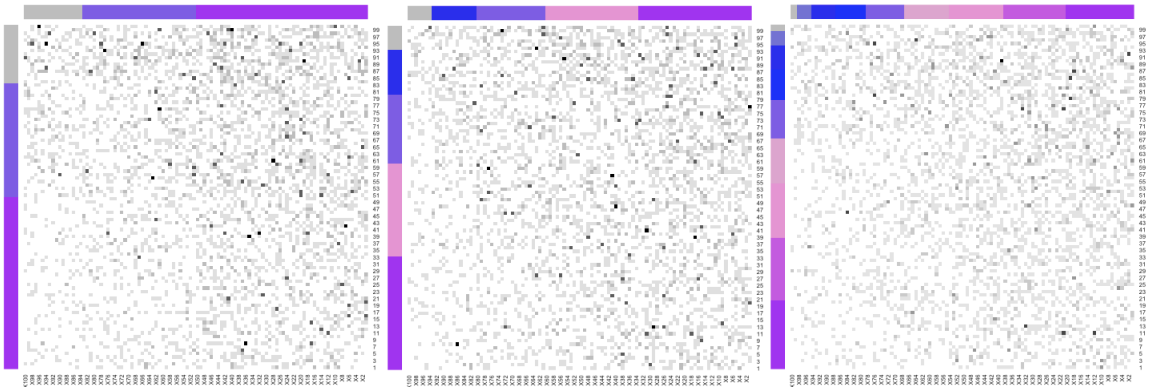
Configuration	$\hat{\alpha}$ (95% CI)	$\hat{S}$ (95% CI)
K3_M4000	0.43 [0.14,0.83]	0.53 [0.27,0.84]
K5_M4000	0.83 [0.74,0.90]	0.85 [0.76,0.90]
K9_M4000	0.82 [0.72,0.89]	0.88 [0.84,0.90]

TABLE 3. Y_{ij} drawn from a simple model

Method	% $\{p \in \text{CI}_{95\%}\}$			% $\{p \in \text{CI}_{99\%}\}$			mean MSE		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Simple model	1	0.9	0.61	1	1	0.71	0.01	0.08	0.11
POMM model	0.33	0.5	0.25	0.33	0.5	0.31	0.12	0.09	0.10

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.



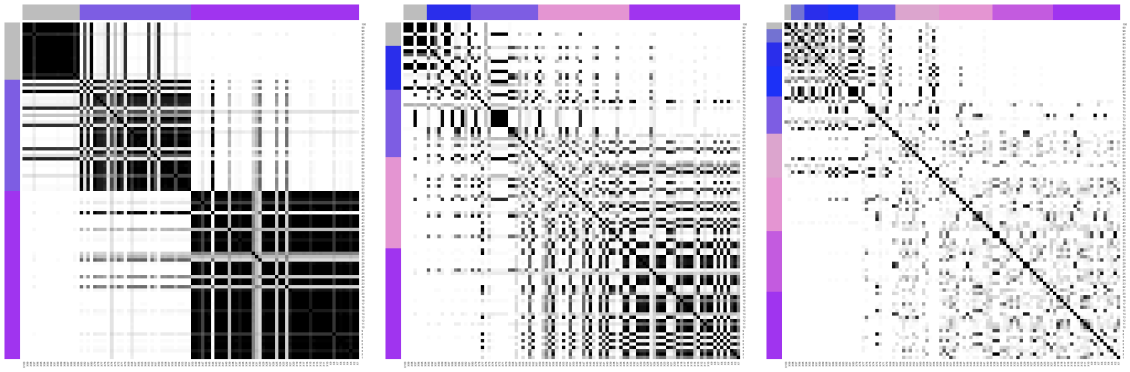
(a) K=3, POMM Model (b) K=5, POMM Model (c) K=9, POMM Model

Figure 6. Adjacency Matrices simulated via the POMM Model

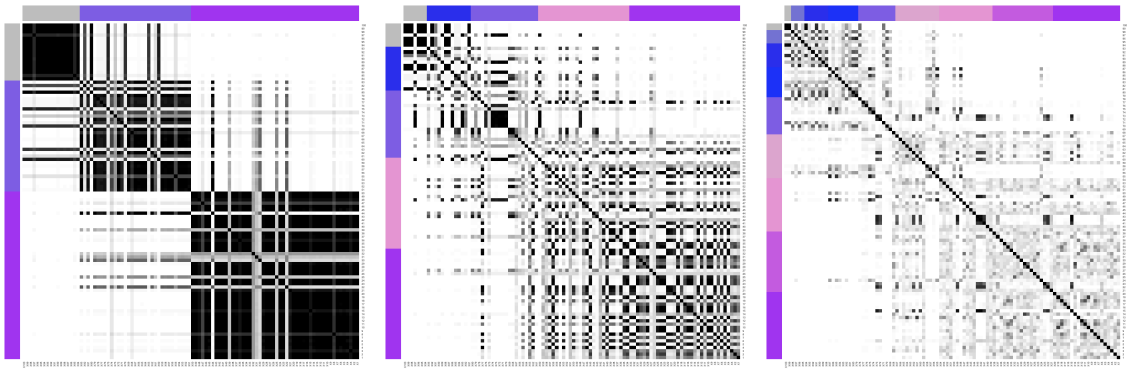
In table (6) we report the results of the simulation. As before, for the Simple and the POMM model, we compare the WAIC, the VI distance and the misclassification error, obtained $N_new = 100$. Also here we compare clustering performance against that of the regularised spectral clustering algorithm and the Louvain algorithm. The POMM model is the best performing relative to the other three alternatives.

Table 4. Y_{ij} drawn from the POMM model

Method	WAIC			VI distance			Error		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Simple model	-11027.45 [35.31]	-10991.22 [33.51]	-11621.91 [34.05]	0.79	2.88	4.11	0.73	0.70	0.90
POMM model	-11030.84 [35.42]	-10963.05 [33.47]	-11409.65 [33.49]	0.79	2.81	3.30	0.60	0.75	0.93
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).

TABLE 5. Data Table, $\alpha = 0.5$, $S = 0.2$

Configuration	$\hat{\alpha}$ (95% CI)	$\hat{S}$ (95% CI)
K3_M4000	0.4442 [0.1930,0.7855]	0.2845 [0.1122,0.6842]
K5_M4000	0.7357 [0.5367,0.8844]	0.7498 [0.5573,0.8877]
K9_M4000	0.7696 [0.6254,0.8865]	0.8707 [0.8247,0.8981]

TABLE 6. Y_{ij} drawn from a simple model

Method	% $\{p \in \text{CI}_{95\%}\}$			% $\{p \in \text{CI}_{99\%}\}$			mean MSE		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Simple model	0.67	0.8	0.78	1	0.80	0.97	0.01	0.05	0.15
POMM model	0.67	0.7	0.97	0.67	0.80	1	0.01	0.04	0.09

8.1. **Extended Simulation Study.** In this section, we want to investigate the performance of the POMM model in recovering the true known partition, generated via the POMM model itself by changing the number of clusters involved (K) and the number of games played among the players M.

TABLE 7. Data Table K=3

CONFIGURATION	MAP	MINVI	MISCLASSERROR	MSE_sum
$M = 4000$	1.1246508	0.9617248	0.6333333	0.054827731
$M = 10000$	0.2557382	0.2557382	0.6166667	0.007170147
$M = 40000$	0.0000000	0.0000000	0.6166667	0.003346822

TABLE 8. Data Table K=5

CONFIGURATION	MAP	MINVI	MISCLASSERROR	MSE_sum
$M = 4000$	2.8488118	2.7704420	0.7333333	0.095988871
$M = 10000$	1.6784835	1.6024773	0.8000000	0.059037933
$M = 40000$	0.4500958	0.4500958	0.6666667	0.057649622

TABLE 9. Data Table K=9

CONFIGURATION	MAP	MINVI	MISCLASSERROR	MSE_sum
$M = 4000$	3.6752208	3.7966766	0.8166667	0.051228350
$M = 10000$	2.8036489	2.6477037	0.8166667	0.047891014
$M = 40000$	1.7268339	1.6557098	0.9000000	0.033409231

TABLE 10. Data Table, $K = 3, \alpha = 0.5$, $S = 0.2$

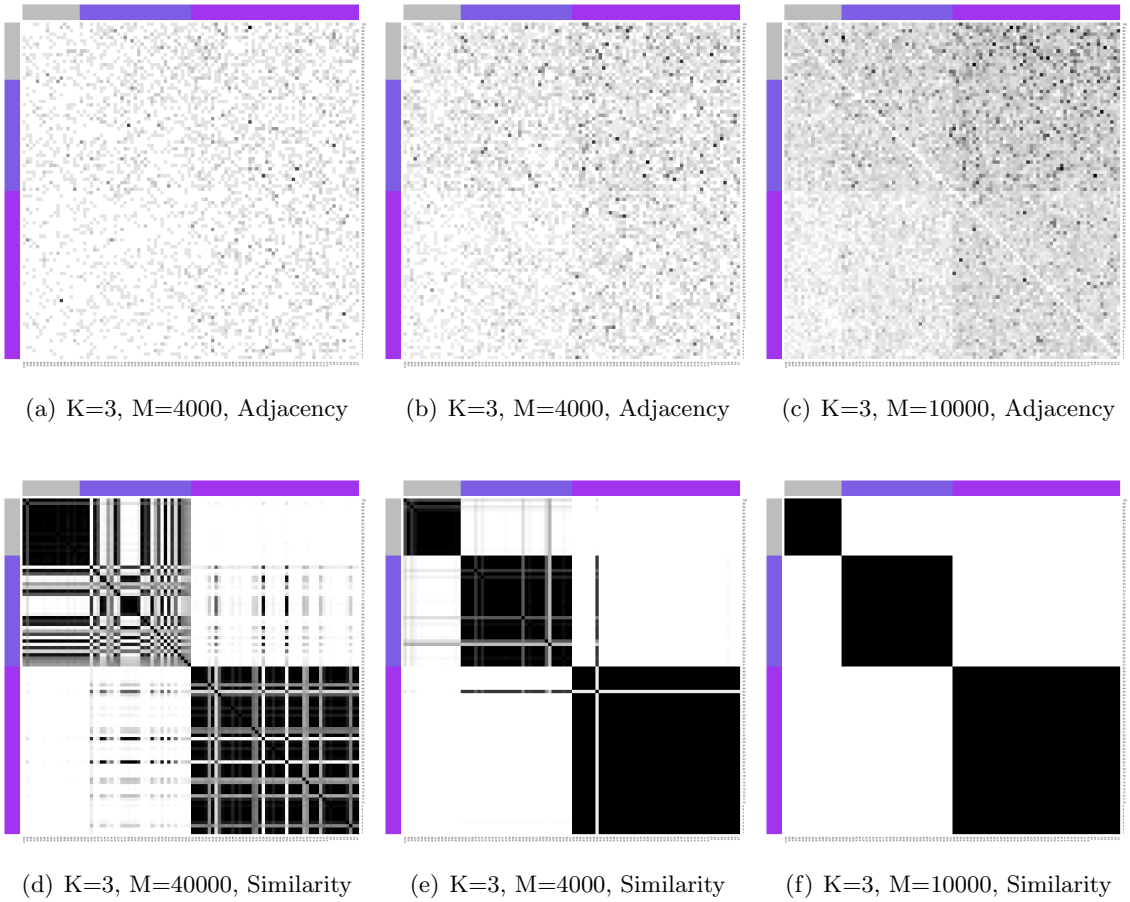
Configuration	$\hat{\alpha}$ (95% CI)	$\hat{S}$ (95% CI)
M=4000	0.6275 [0.3515,0.8679]	0.5018 [0.2251,0.8398]
M=10000	0.5054 [0.2600,0.8057]	0.3098 [0.1263,0.6942]
M=40000	0.4802 [0.2452,0.7880]	0.3308 [0.1343,0.7215]

TABLE 11. Data Table, $K = 5, \alpha = 0.5$, $S = 0.2$

Configuration	$\hat{\alpha}$ (95% CI)	$\hat{S}$ (95% CI)
M=4000	0.7497 [0.5944,0.8841]	0.8260 [0.7178,0.8950]
M=10000	0.6905 [0.4872,0.8734]	0.6801 [0.4702,0.8735]
M=40000	0.7888 [0.6365,0.8918]	0.7136 [0.5171,0.8816]

TABLE 12. Data Table, $K = 9, \alpha = 0.5$, $S = 0.2$

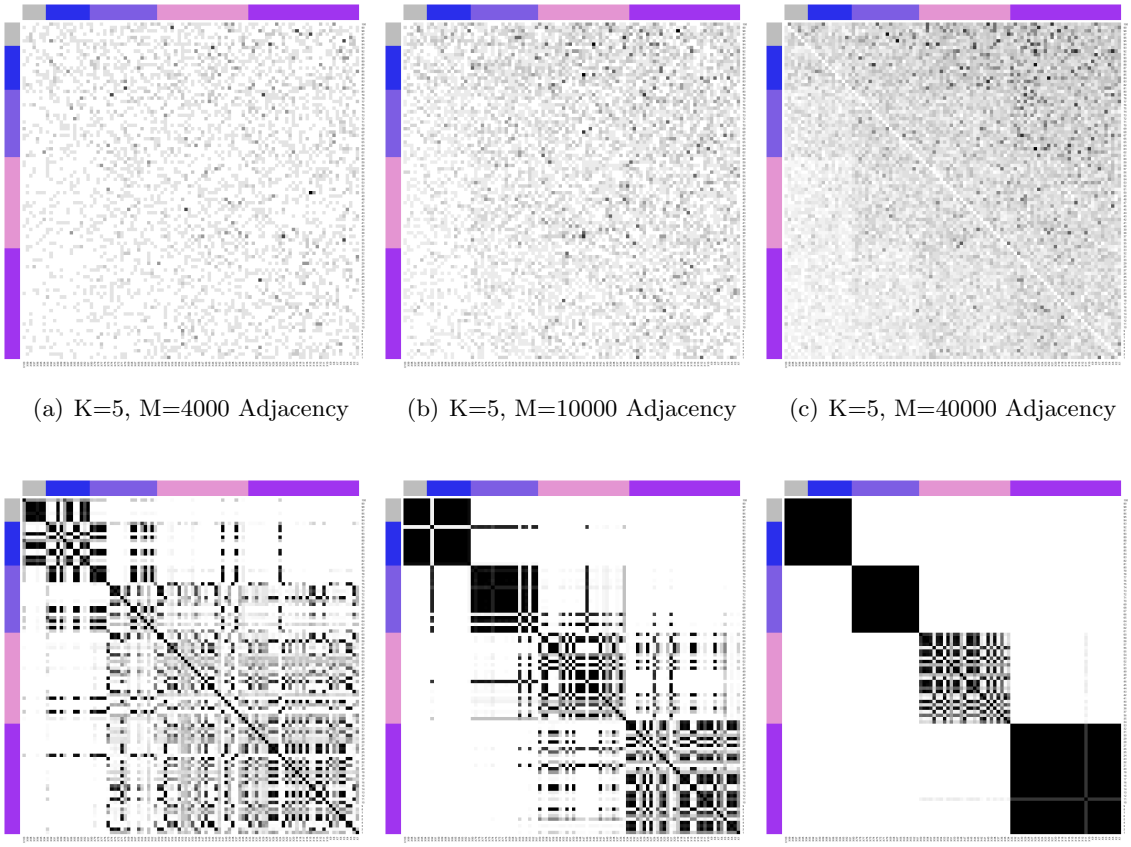
Configuration	$\hat{\alpha}$ (95% CI)	$\hat{S}$ (95% CI)
M = 4000	0.5966 [0.5966,0.5966]	0.8699 [0.8215,0.8980]
M=10000	0.6038 [0.5989,0.6378]	0.8574 [0.7899,0.8971]
M = 40000	0.7241 [0.6251,0.8412]	0.8330 [0.7290,0.8959]



(a) K=3, M=4000, Adjacency (b) K=3, M=4000, Adjacency (c) K=3, M=10000, Adjacency

(d) K=3, M=40000, Similarity (e) K=3, M=4000, Similarity (f) K=3, M=10000, Similarity

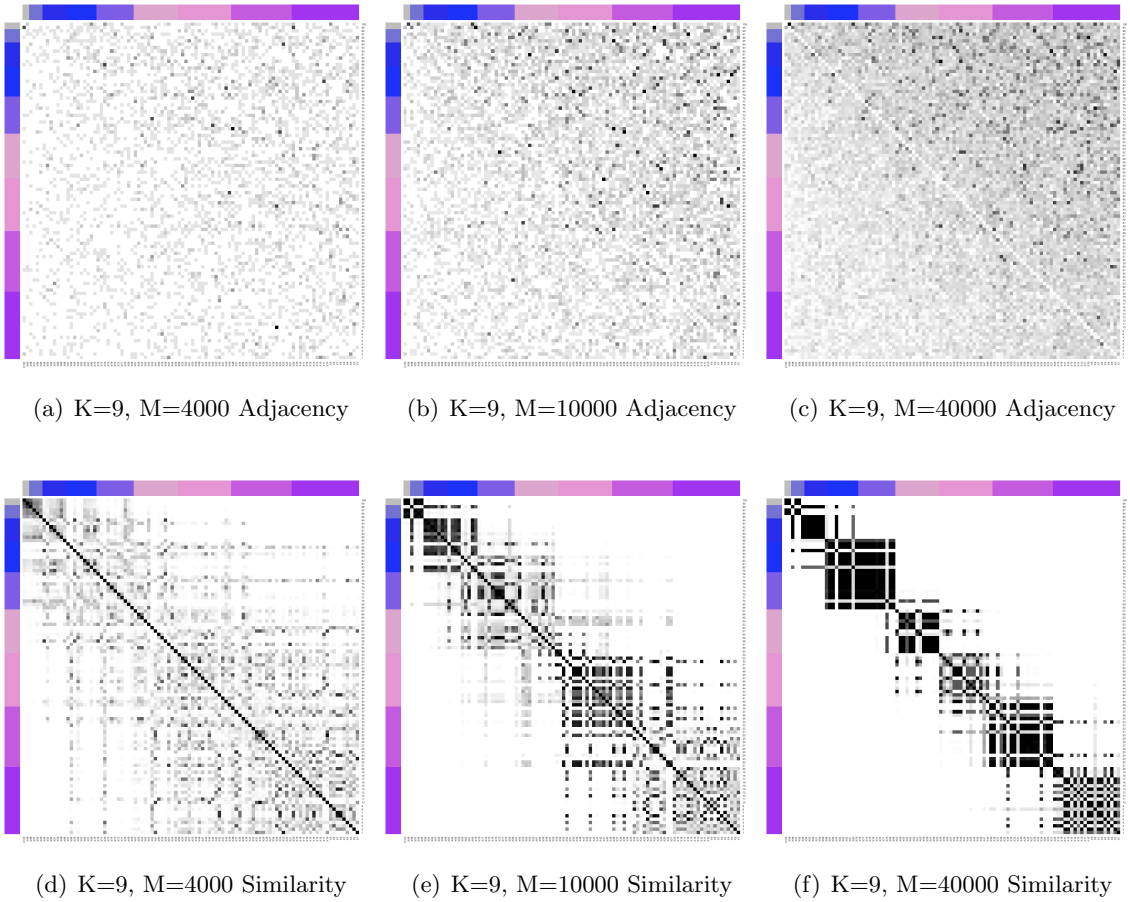
FIGURE 8. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).



(a) K=5, M=4000 Adjacency (b) K=5, M=10000 Adjacency (c) K=5, M=40000 Adjacency

(d) K=5, M=4000 Similarity (e) K=5, M=10000 Similarity (f) K=5, M=40000 Similarity

FIGURE 9. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

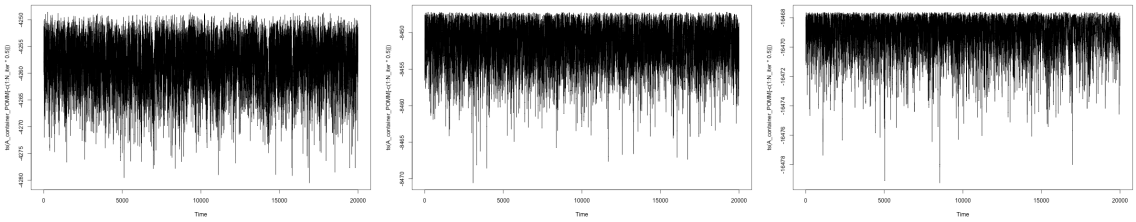


(a) K=9, M=4000 Adjacency (b) K=9, M=10000 Adjacency (c) K=9, M=40000 Adjacency

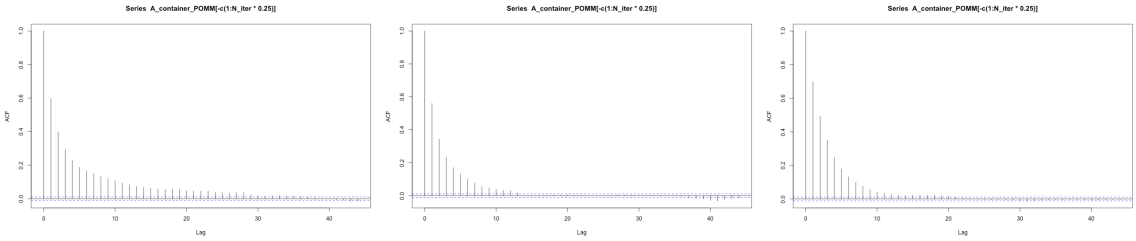
(d) K=9, M=4000 Similarity (e) K=9, M=10000 Similarity (f) K=9, M=40000 Similarity

FIGURE 10. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

8.2. **Diagnostic Checks.**

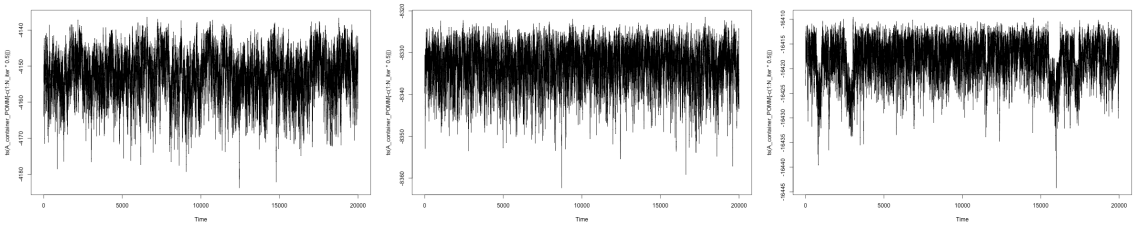


(a) K=3, M=4000 Traceplot (b) K=3, M=10000 Traceplot (c) K=3, M=40000 Traceplot

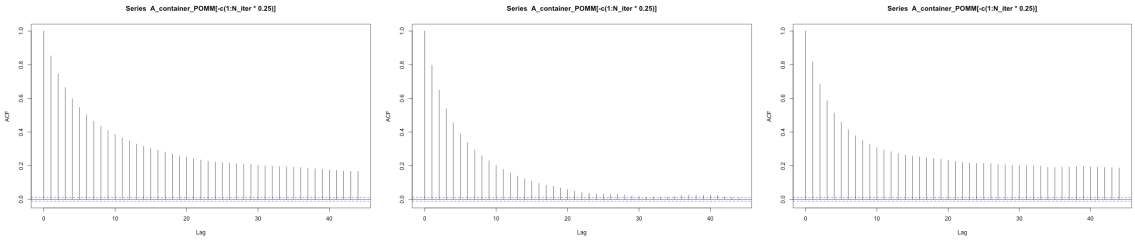


(d) K=3, M=4000 Autocorrelation (e) K=3, M=10000 Autocorrelation (f) K=3, M=40000 Autocorrelation

FIGURE 11. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

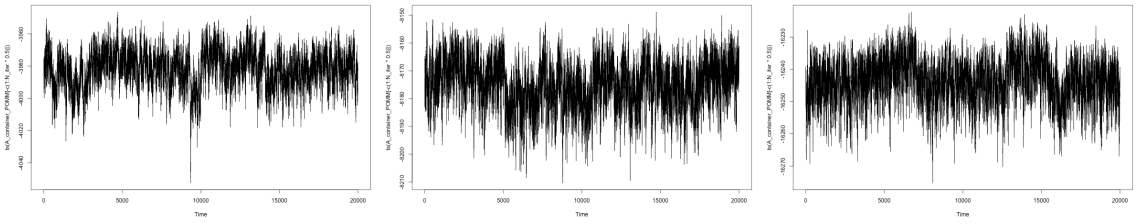


(a) K=5, M=4000 Traceplot (b) K=5, M=10000 Traceplot (c) K=5, M=40000 Traceplot

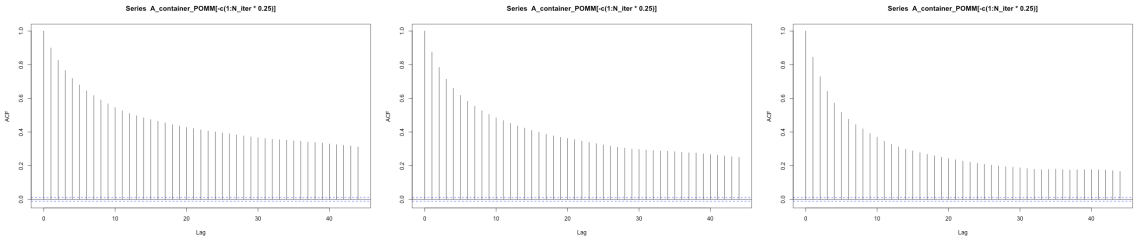


(d) K=5, M=4000 Autocorrelation (e) K=5, M=10000 Autocorrelation (f) K=5, M=40000 Autocorrelation

FIGURE 12. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).



(a) K=9, M=4000 Traceplot (b) K=9, M=10000 Traceplot (c) K=9, M=40000 Traceplot



(d) K=9, M=4000 Autocorrelation (e) K=9, M=10000 Autocorrelation (f) K=9, M=40000 Autocorrelation

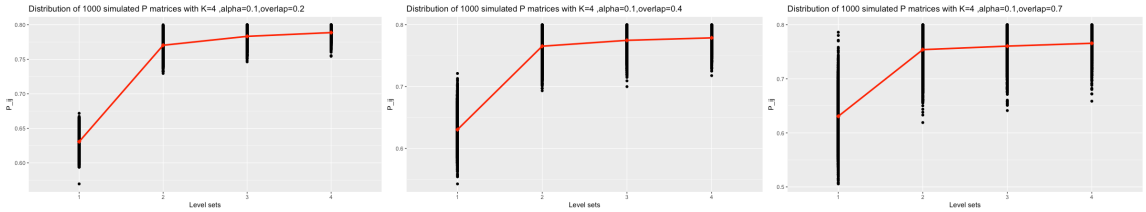
FIGURE 13. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

9. Application to Tennis Data

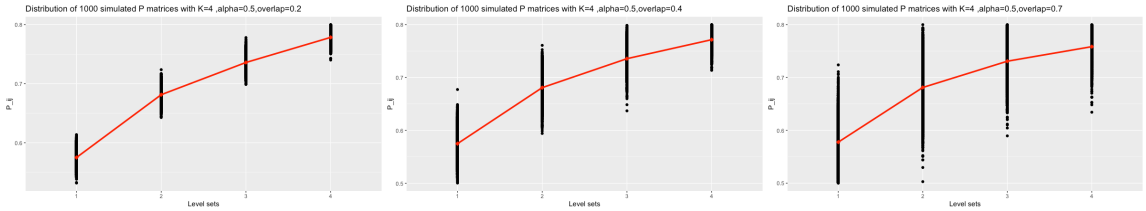
10. Appendix I: Investigating Empirically the prior behaviour

In this section, we explore the behaviour of the POMM prior, as α, the parameter controlling the rate of increase of the power-law process and S, the variation (and therefore, the overlap) of the level sets sets, change.

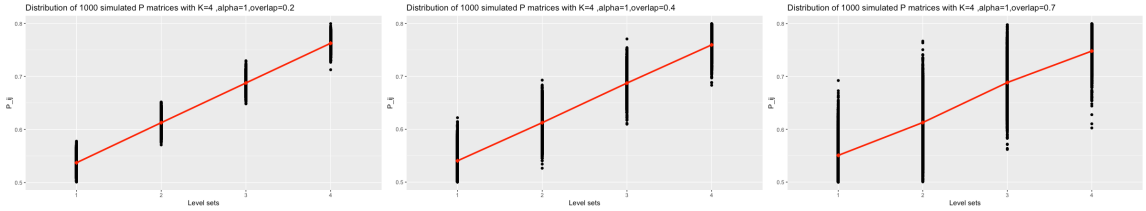
We start with a simulation study, whose results are reported in figure (**??**).



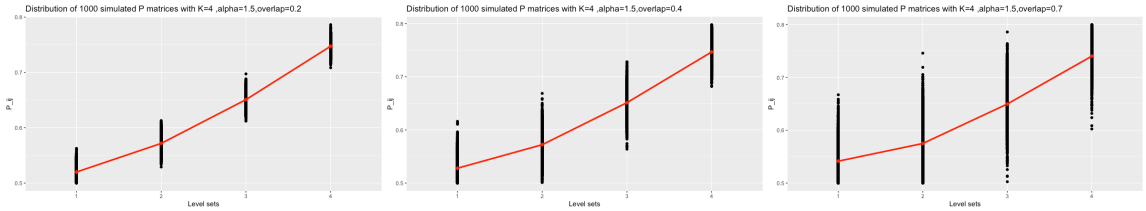
(a) $\alpha = 0.1, S = 0.2$ (b) $\alpha = 0.1, S = 0.4$ (c) $\alpha = 0.1, S = 0.7$



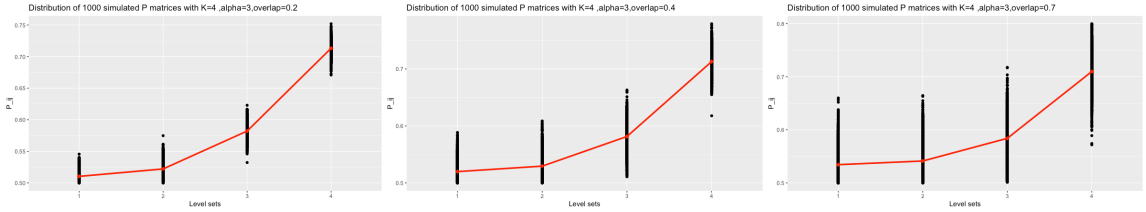
(d) $\alpha = 0.5, S = 0.2$ (e) $\alpha = 0.5, S = 0.4$ (f) $\alpha = 0.5, S = 0.7$



(g) $\alpha = 1, S = 0.2$ (h) $\alpha = 1, S = 0.4$ (i) $\alpha = 1, S = 0.7$



(j) $\alpha = 1.5, S = 0.2$ (k) $\alpha = 1.5, S = 0.4$ (l) $\alpha = 1.5, S = 0.7$



(m) $\alpha = 3, S = 0.2$ (n) $\alpha = 3, S = 0.4$ (o) $\alpha = 3, S = 0.7$

FIGURE 14. Distribution of $n = 1000$ simulated P matrices with $\alpha \in \{0.1, 0.5, 1, 1.5, 3\}$ and $S \in \{0.2, 0.4, 0.7\}$. The points are grouped into the $K = 5$ level sets- the main diagonal is set =0.5. The black vertical dots are the points, while the red lines show the evolution of the mean of the level sets.

11. Appendix II: Empirical Assessment of the inference problem

This section has two main objectives:

(1) The first objective is to assess the identifiability of the POMM model. Given that the richness of the model, it is not obvious that the parameters are identifiable in all the regions of the parameter space. Therefore, by means of several simulation studies, we aim at identifying any possible identifiability constraints.
(2) The second objective is code testing, by checking the whole estimation procedure on simulated data.

11.1. Focus on α.

Table 13. Mean absolute error for different alpha values
Sample size = 1000. MAEs = $|\hat{\alpha}_{\text{MLE}} - \alpha|$

	S = 0.2	S = 0.4	S = 0.7
$\alpha = 0.1$	3.199×10^{-4}	1.270×10^{-4}	0.003
$\alpha = 0.5$	1.377×10^{-4}	2.955×10^{-4}	0.001
$\alpha = 1.0$	7.695×10^{-4}	0.007	0.001
$\alpha = 1.5$	0.001	0.006	0.004
$\alpha = 3.0$	0.003	0.036	0.004

11.1.1. *Updating α algorithm.* 3

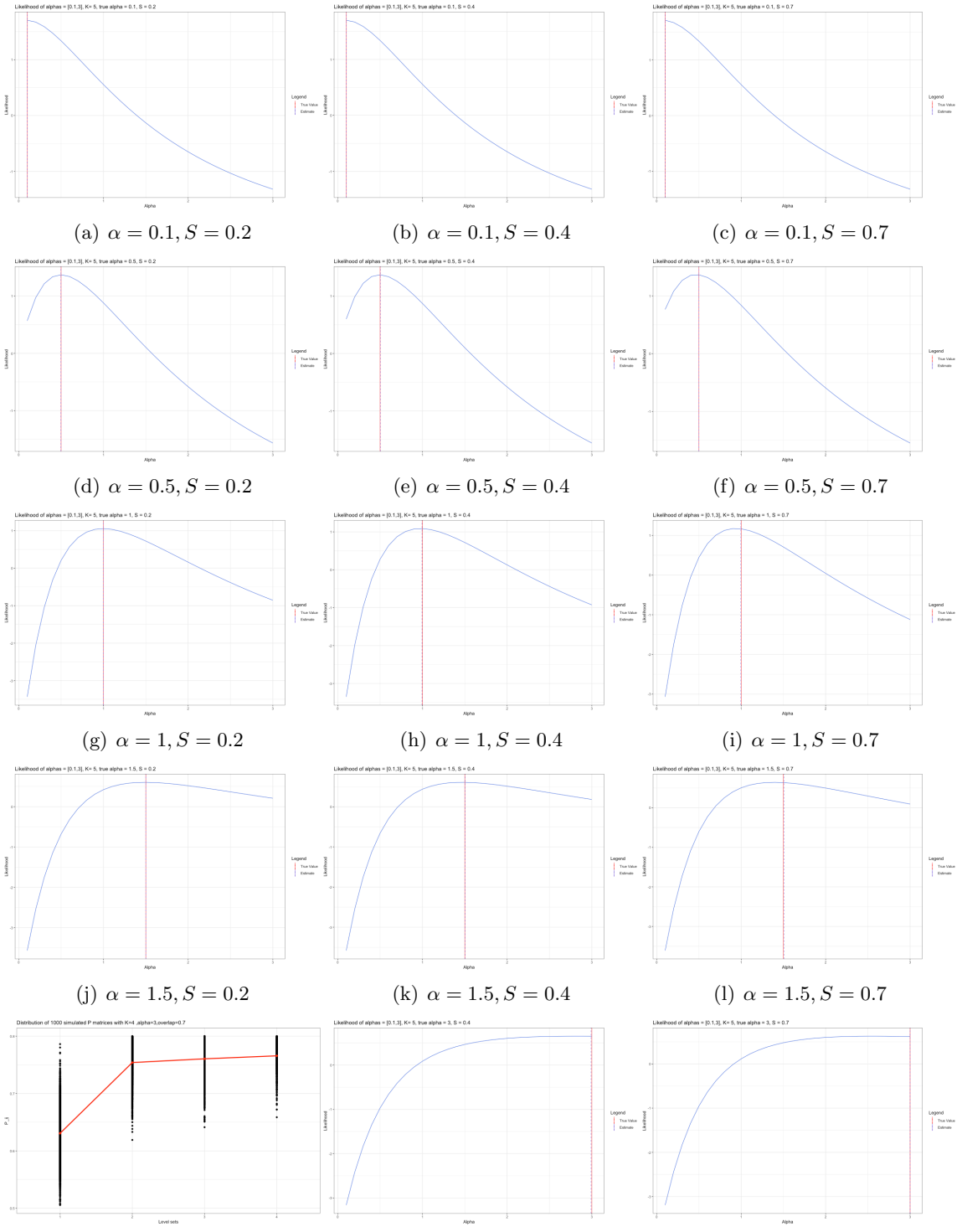
11.1.2. *Montecarlo algorithm.*

11.2. Focus on S.

Mean absolute error for different S values
Each row is a different S value. Sample size = 1000

	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 1.5$	$\alpha = 3$
S = 0.2	0.002	0.002	9.780×10^{-4}	3.709×10^{-4}	9.931×10^{-4}
S = 0.4	0.005	0.003	0.006	0.002	0.003
S = 0.7	0.006	0.002	0.005	0.006	0.007

11.2.1. *Montecarlo algorithm.*



(a) $\alpha = 0.1, S = 0.2$ (b) $\alpha = 0.1, S = 0.4$ (c) $\alpha = 0.1, S = 0.7$

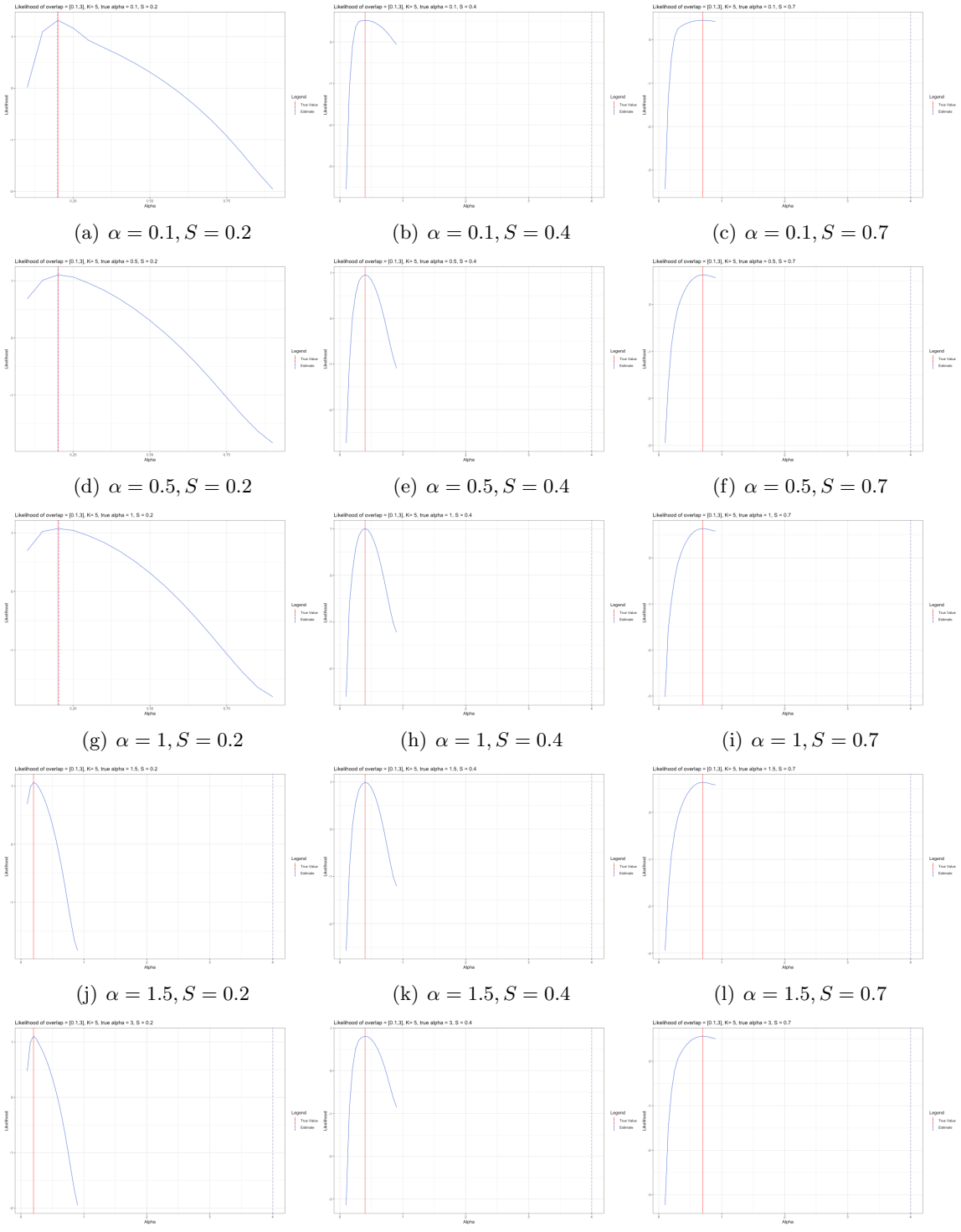
(d) $\alpha = 0.5, S = 0.2$ (e) $\alpha = 0.5, S = 0.4$ (f) $\alpha = 0.5, S = 0.7$

(g) $\alpha = 1, S = 0.2$ (h) $\alpha = 1, S = 0.4$ (i) $\alpha = 1, S = 0.7$

(j) $\alpha = 1.5, S = 0.2$ (k) $\alpha = 1.5, S = 0.4$ (l) $\alpha = 1.5, S = 0.7$

(m) $\alpha = 3, S = 0.2$ (n) $\alpha = 3, S = 0.4$ (o) $\alpha = 3, S = 0.7$

FIGURE 15. Distribution of $n = 1000$ simulated P matrices with $\alpha \in \{0.1, 0.5, 1, 1.5, 3\}$ and $S \in \{0.2, 0.4, 0.7\}$. The points are grouped into the $K = 5$ level sets- the main diagonal is set =0.5. The black vertical dots are the points, while the red lines show the evolution of the mean of the level sets.



(a) $\alpha = 0.1, S = 0.2$ (b) $\alpha = 0.1, S = 0.4$ (c) $\alpha = 0.1, S = 0.7$

(d) $\alpha = 0.5, S = 0.2$ (e) $\alpha = 0.5, S = 0.4$ (f) $\alpha = 0.5, S = 0.7$

(g) $\alpha = 1, S = 0.2$ (h) $\alpha = 1, S = 0.4$ (i) $\alpha = 1, S = 0.7$

(j) $\alpha = 1.5, S = 0.2$ (k) $\alpha = 1.5, S = 0.4$ (l) $\alpha = 1.5, S = 0.7$

(m) $\alpha = 3, S = 0.2$ (n) $\alpha = 3, S = 0.4$ (o) $\alpha = 3, S = 0.7$

FIGURE 16. Distribution of $n = 1000$ simulated P matrices with $\alpha \in \{0.1, 0.5, 1, 1.5, 3\}$ and $S \in \{0.2, 0.4, 0.7\}$. The points are grouped into the $K = 5$ level sets- the main diagonal is set =0.5. The black vertical dots are the points, while the red lines show the evolution of the mean of the level sets.

12. Appendix I: Estimation Details

12.1. **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:

$$r = \frac{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k'+1)}}{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{43}$$

$$= \frac{\prod_{i<j} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k'+1)}}{\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{44}$$

Passing to the log:

$$\begin{aligned}
log(r) = & \log \left(\prod_{i<j} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k' + 1)} \right) \\
& - \log \left(\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k + 1)} \right) \\
= & \sum_{i<j} \Big(y_{ij} \cdot \log p_{z_i' z_j'} + (n_{ij} - y_{ij}) \cdot \log (1 - p_{z_i' z_j'}) \Big) \\
& + \sum_{k=1}^{K} \left(\log \left(\Gamma(n_k' + \gamma_k) \right) - \log \left(\Gamma(\gamma_k) \right) - \log \left(\Gamma \left(n_k' + 1 \right) \right) \right) \\
& - \sum_{i<j} \Big(y_{ij} \cdot \log p_{z_i z_j} + (n_{ij} - y_{ij}) \cdot \log (1 - p_{z_i z_j}) \Big) \\
& - \sum_{k=1}^{K} \left(\log \left(\Gamma(n_k + \gamma_k) \right) - \log \left(\Gamma(\gamma_k) \right) - \log \left(\Gamma \left(n_k + 1 \right) \right) \right)
\end{aligned} \tag{45}$$

12.2. **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:

$$r = \frac{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}'^{y_{ij}} \cdot (1 - p'_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \left(\frac{1}{y'^{(k+1)} - y'^{(k)}} \right)^{|L'^{(k)}|}}{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \left(\frac{1}{y^{(k+1)} - y^{(k)}} \right)^{|L^{(k)}|}} \tag{46}$$

(47)

Passing to the log:

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

(48)

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

(49)

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

13. Appendix II: POMM prior checks

13.1. **Prior predictive check.**

13.2. **MLE check.**

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