DRAFT

LAPO SANTI

Contents

1. Introduction

In the present work, we specify and estimate a model for pairwise or binary data capable of clustering the data within blocks. The novelty lies in the fact that these blocks are ordered, meaning that they present a hierarchical relationship of some sort.

We start by presenting the unordered model, which is essentially a traditional Stochastic Block Models (SBMs). Pairwise data are modelled as binomially distributed, where the probability of success depends both on the membership of the two pair members, stored within the parameter z, and on a blocks-related probabilities matrix P.

The classical SBM models P with a Beta(1,1) prior. In the present work instead we model this P matrix in different ways in order to induce an ordering among the clusters.

We start by reviewing the unordered model, then, by drawing inspiration from the literature on image recognition, we start representing the matrix P as an ordered poset, meaning a discrete set where we impose an ordering structure. This structure is strongly influence by the properties that we would like our ranked cluster to have.

For example, if we work under the Strong Stochastic Transitivity (SST) axiom, we are asking that if the individual i is ranked higher than j and j is ranked higher than q, then i must be ranked higher than the higher between i and j. There are other axioms that we might choose, like the Weak Stochastic Transitivity one (WST), which is less stringent than the SST, or the Linear Stochastic Transitivity one (LST), which encompasses both the two aforementioned ones.

However, assuming the SST axiom and imposing it over the poset defined on P lets emerge an interesting structure, meaning that of the Level Sets $L_{(k)}$. Each Level Set is defined as the set of entries on a specific diagonal of the upper triangular matrix P, starting from $L_{(0)}$ which is the main diagonal, $L_{(1)}$ is the diagonal above and so on, until $L_{(K-1)}$. We impose that $L_{(0)}$ is always equal to $\frac{1}{2}$, since we assume that individuals within the same block have equal probability of being preferred.

We proceed to specify over each Level Set a prior probability distribution, which governs the probability law of the entries within that diagonal. The probability distribution is a truncated normal distribution $TRUNCNORM(\mu^{(k)}|\alpha, \beta_{\max}, \sigma)$.

Each distribution is endowed with a specific and increasing $\mu^{(k)}|\alpha, \beta_{\max}$, which effectively induces a hierarchical model, given that $\alpha, \beta_{\max}$ are parameters common to each level set. While α specifies the rate of increase of the Level Sets' means $\{\mu^{(k)}\ , k = 0, \ldots, K-1\}$, $\beta_{\max}$ specifies the maximum attainable probability within the matrix P.

The last and most important hyperparameter σ is common to each distribution over the level sets. As σ increases, the truncated normals become more and more flat, and therefore we are back to a case where the entries of P are not distributed in a way that is not significantly different from a uniform.

This model should therefore encompass the unordered one, for values of σ that are large enough.

2. The Unordered 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 has entries z_i taking values over the discrete and finite set $\{1,\ldots,K\}$, 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.

2.1. Prior Specification.

Starting with the parameter 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.

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

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. Connection with Image Recognition

By drawing inspiration from the literature in Image Recognition, in particular an article of Noel Cressie and Jennifer Davidson, we represent our matrix P as if it was an image, its probabilities as if they were pixels of different intensities, and its entries' indices as if they were pixels' locations.

Let us denote a generic entry of P as a vector s in $\mathbb{N}^2$. The quantity $Z(s)$ denotes the probability value at the entry index s. We rewrite the whole matrix as

$$Z = \{Z(s) : s \in D\} \tag{6}$$

where D is the set of entries' indices of the matrix P, that is:

$$D = \{(u, v) : u = 1, \dots, K; v = 1, \dots, K\}. \tag{7}$$

Now, let us consider a temporal Markov process {Z(t): t=1,2,...}. The Markov property can be generalised from a one-dimensional time-process to a two-dimensional space with both a conditional and a joint specification.

We draw a connection between the class of models called Partially Ordered Markov models which allows us to efficiently compute the joint probabilities of the prior on P, and the literature on preference learning.

Then we introduce the notion of partial order among the P entries. Let's take once more D. The binary relation $\succeq$ on D is said to be a partial order if For any $x \in D, x \prec x$ (reflexivity). For any $x, y, z \in D, x \prec y$ and $y \prec z$ implies $x \prec z$ (transitivit y). 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. For example, the set of all subsets of a given set, with the relation $\prec$ being set inclusion, is a poset.

Regarding the matrix P, we can check whether and how to use the definition of Poset under the three different axiomatic frameworks that specify different (stochastic) transitivity requirements.

1. Weak Stochastic Transitivity (WST): $\mathbb{P}(x \prec y) \geq \frac{1}{2}$ and $\mathbb{P}(y \prec z) \geq \frac{1}{2}$ imply $\mathbb{P}(x \prec z) \geq \frac{1}{2}$, for all $x, y, z \in \mathcal{A}$.
2. Strong Stochastic Transitivity (SST): $\mathbb{P}(x \prec y) \geq \frac{1}{2}$ and $\mathbb{P}(y \prec z) \geq \frac{1}{2}$ imply $\mathbb{P}(x \prec z) \geq \max\{\mathbb{P}(x \prec y), \mathbb{P}(y \prec z)\}$, for all $x, y, z \in \mathcal{A}$.
3. Linear Stochastic Transitivity (LST): $\mathbb{P}(a \prec b) = F(\mu(a) - \mu(b))$ for all $a, b \in \mathcal{A}$, where $F : \mathbb{R} \to [0, 1]$ is an increasing and symmetric function (referred to as a "comparison function"), and $\mu : \mathcal{A} \to \mathbb{R}$ is a mapping from the set $\mathcal{A}$ of alternatives to the real line (referred to as a "merit function").

Each of these axioms, produces a different P structure. Assuming, without loss of generality, that block 1 is the strongest, and by imposing the main diagonal to be equal to $\frac{1}{2}$ we can visualise a matrix following WST as:

$$P^{WST} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,K} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & \cdots & p_{K,K} \end{pmatrix} = \begin{pmatrix} 1/2 \leq & p_{1,2} & \cdots & p_{1,K} \\ p_{2,1} \leq & 1/2 \leq & \cdots & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & \cdots & 1/2 \end{pmatrix} \tag{8}$$

Instead, under SST, we would observe:

$$P^{SST} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,K} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & \cdots & p_{K,K} \end{pmatrix} = \begin{pmatrix} 1/2 \leq & p_{1,2} \leq & \cdots & p_{1,K} \\ \vee\!| & \vee\!| & & \vee\!| \\ p_{2,1} \leq & 1/2 & \cdots & \leq p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} \leq & p_{K,2} \leq & \cdots & \leq 1/2 \end{pmatrix} \tag{9}$$

With regard of LST, it is a generalisation of the two axioms and therefore it includes both aforementioned cases (8) and (9), depending how one specifies F and μ. Given the LST definition, we can calculate p_{ij} as follows:

$$p_{ij} = F(\mu(i) - \mu(j)) \tag{10}$$

where i and j range from 1 to K and represent the alternatives in the set $\mathcal{A}$.

All the three axiomatic frameworks satisfy (in the LST case we need to check the functional form of F and μ) of the three conditions for being a poset. And therefore, we take advantage of the poset structure.

We can now describe the correspondence referred to above. This connection opens up a large literature on graphical models, outside of statistical image analysis, that we return to in Section 6.2. 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

$$\begin{aligned} &y_i \prec y_i, \text{ for } i = 1, \ldots, n; \\ &y_i \prec y_j, \text{ if there exists a directed path from } y_i \text{ to } y_j \text{ in } (D, F). \end{aligned}$$

Notice that several different directed acyclic graphs can yield the same poset. Conversely, given a finite poset $(D, \prec)$, a corresponding directed acyclic graph can be obtained by defining the set of edges F as follows: $(y_i, y_i) \in F$ if and only if $y_i \prec y_j$ and there does not exist a third element

$$z \neq y_i, y_j \text{ such that } y_i \prec z \prec y_j.$$

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. However, if one starts with a directed acyclic graph, the corresponding poset is unique. From the point of view of image modeling, we arc more interested in the directed-acyclic-graph description because we are able to specify directly the spatial relations between pixel locations.

[to be finished]

4. The WST model

This model implements the Weak Stochastic Transitivity assumption.

To have the WST axiom satisfied, we need the upper triangular entries of the P matrix to be always greater than 0.5

Therefore, to implement this model a Bayesian setting, we must change little with respect to the Unordered model.

We leave everything the same, but we force the upper triangular entries to be greater than 0.5. We can enforce effectively this condition by modifying the proposal distribution, and also the prior distribution.

5. The SST model

We consider a Partially Ordered Markov Model (POMM) for a set of entries $p_{ij} \in L_{(k)}$, where $L_{(k)}$ represents a level set. These entries are assumed to be identically and independently distributed according to a truncated normal distribution:

$$p_{ij}^{(k)} \mid (y^{(k)}, y^{(k+1)}) \sim \text{TruncatedNormal}(\mu^{(k)}, \sigma^{2(k)}; 0.5, \beta_{\max})$$

The full prior is given by:

$$p(P) = \prod_{k=1}^{K} \frac{1}{\sigma} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{p_{ij}-\mu}{\sigma}\right)^2}}{\int_{-\infty}^{\beta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt - \int_{-\infty}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt}$$

Here:

- $\mu^{(k)}$ is the mean, which corresponds to the midpoint of the level set $L_{(k)}$, defined as $\mu^{(k)} = \frac{y^{(k)}+y^{(k+1)}}{2}$.
- σ^2 is the variance parameter constant across the level sets $L_{(1)}, \dots, L_{(k)}$.
- 0.5 and $\beta_{\max}$ are the lower and upper truncation bounds.

We also place a uniform hyper-prior on the parameter α:

$$\alpha \sim \text{Uniform}(0, 3)$$

which in turn, together with $\beta_{\max}$, it provides the truncations of the level sets as follows:

$$y^{(k)} = \left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times k \right)^{\alpha} + 0.5 \quad \text{for } k = 0, \dots, K \tag{11}$$

We also place a uniform hyper-prior on the parameter σ^2

$$\sigma^2 \sim \text{Uniform}(0, 1)$$

which is the variance of the level sets truncated normal distribution.

Supposition: When $\sigma^2 \to \infty$, the joint distribution of the level sets collapses to a uniform and therefore we are back to a uniform distribution.

6. The LST model

The latent skill variables for clusters are denoted by s_c for cluster c. We model these latent skills using a Gaussian Process (GP) prior:

$$s_c \sim \mathcal{GP}(0, k(\cdot, \cdot))$$

Here, $\mathcal{GP}$ is the Gaussian Process distribution, and $k(\cdot, \cdot)$ is the covariance (kernel) function that captures the relationships between clusters.

The victory probability for a player from cluster A against a player from cluster B is modeled using a logistic function f that takes the difference in latent skills into account:

$$p_{A,B} = f(s_A - s_B) = \frac{1}{1 + \exp(-\beta \cdot (s_A - s_B))}$$

Where β is a parameter that controls the slope of the logistic function and affects how fast the probability changes with the difference in skills.

The likelihood of observing y_{ij} victories out of n_{ij} matches between clusters i and j is given by a binomial distribution:

$$y_{ij} \sim \text{Binomial}(n_{ij}, p_{i,j})$$

The complete model, combining the Gaussian Process prior on latent skills and the logistic function for victory probabilities, can be written as:

$$\begin{aligned}
&\text{For each cluster } c: \\
&s_c \sim \mathcal{GP}(0, k(\cdot, \cdot)) \\
&\text{For each match between clusters } i \text{ and } j: \\
&p_{i,j} = \frac{1}{1 + \exp(-\beta \cdot (s_i - s_j))} \\
&y_{ij} \sim \text{Binomial}(n_{ij}, p_{i,j})
\end{aligned}$$

7. Estimation

For the moment, we want to infer just $\theta = \{z, P, \alpha, \sigma^2\}$, meaning that we treat K as a known constant. We report below the posterior distribution that we want to estimate:

$$\begin{aligned} p(z, \alpha, \sigma^2, P \mid y) &= \frac{p(y \mid \alpha, \sigma^2, P) \cdot p(P \mid \alpha, \sigma^2) \cdot p(\alpha) \cdot p(\sigma^2)}{\int p(y, \alpha, \sigma^2, P) dz \; d\alpha \; d\sigma^2 \; dP} \\ &\propto p(y \mid \alpha, \sigma^2, P) \cdot p(P \mid \alpha, \sigma^2) \cdot p(\alpha) \cdot p(\sigma^2) \end{aligned} \tag{12}$$

The estimation strategy for (12) is a Metropolis-within-Gibbs MCMC algorithm. MCMC techniques are designed precisely for scenarios like ours, where direct estimation of the posterior is infeasible or analytically intractable. By generating a sequence of samples through a carefully constructed Markov chain, MCMC allows us to navigate the complex parameter space in a data-driven manner.

Referencing Muller's (1991) work, we present here below an hybrid version of a classical Metropolis-within-Gibbs algorithm that has some self-tuning, or adaptive, features in the variance of the proposal distribution, the parameter τ^2_θ

We start by reviewing the MCMC step for z, and the we move on exploring the other continuous parameters in the remaining of this section.

In algorithm (1) we have that (13) is equal to:

$$\sum_{i=2}^{N-1} \sum_{j=i}^{N} \text{LogBinomial}\left(y_{ij} \mid n_{ij}, p^{(t)}_{z'_i z'_j}\right) - \sum_{i=2}^{N-1} \sum_{j=i}^{N} \text{LogBinomial}\left(y_{ij} \mid n_{ij}, p^{(t)}_{z_i^{(t)} z_j^{(t)}}\right) \tag{14}$$

$$+ \log\left(\frac{\Gamma(\sum_{k=1}^K \gamma_k)}{\prod_{k=1}^K \Gamma(\gamma_k)} \frac{\prod_{k=1}^K \Gamma(n'_k + \gamma_k)}{\Gamma(\sum_{k=1}^K (n'_k + \gamma_k))}\right) - \log\left(\frac{\Gamma(\sum_{k=1}^K \gamma_k)}{\prod_{k=1}^K \Gamma(\gamma_k)} \frac{\prod_{k=1}^K \Gamma(n_k^{(t)} + \gamma_k)}{\Gamma(\sum_{k=1}^K (n_k^{(t)} + \gamma_k))}\right) \tag{15}$$

since the joint prior on the level sets, that is

$$\sum_{k=1}^{K-1} \text{LogTruncatedNormal}\left(L_{(k)} \mid \left[\frac{(\beta_{\max} - 0.5)}{2K^{\alpha'}} \times \left(k^{\alpha'} + (k+1)^{\alpha'}\right) + \frac{1}{2}\right], \sigma^{2(t)}, 0.5, \beta_{\max}\right)$$

, the log prior probability on σ^2, namely $\log p(\sigma)$, and the log prior probability on α are not affected by the new proposed value of z and therefore they subtract out.

This algorithm is meant to propose with higher probability those labels that are adjacent to the current one, and the Normal distribution centred on zero implies that the further the distance. Furthermore, the adaptive variance of the proposal $\tau^2_{z_{\pi(i)}}$ distribution controls the probability of labels that are more far apart from the current one.

Choosing a correct $\tau^2_{z_{\pi(i)}}$ 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 i-th labels 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

Algorithm 1 Metropolis-within-Gibbs update for z

```
Given  $\left(z^{(t),\alpha^{(t)},\sigma^{2(t)},P^{(t)}}\right)$

1.  Randomly sample an order
        $\pi \sim \text{Random permutation of } \{1, 2, \ldots, n\}$
for $i = 1$ to $N$ do
    1.  Compute the difference $\{d_{k,z^{(t)}_{\pi(i)}}\}$ between each label $k = 1, \ldots, K$
    and the current state $z^{(t)}_{\pi(i)}$
        $d_{k,z^{(t)}_{\pi(i)}} = k - z^{(t)}_{\pi(i)}$
    2.  Compute and normalize the probabilities
        $p_k = \frac{p\left(d_{k,z^{(t)}_{\pi(i)}}\right)}{\sum_{k=1}^{K} p\left(d_{k,z^{(t)}_{\pi(i)}}\right)}$  where  $d_{k,z^{(t)}_{\pi(i)}} \sim \text{Normal}\left(0, \tau^2_{z_{\pi(i)}}\right)$
    3.  $z'_{\pi(i)} \leftarrow k'$ sampled from $\{k = 1, ..., K\} \setminus \{z^{(t)}_{\pi(i)}\}$ with probability $p_k$
    4.  Take
        $z^{(t+1)}_{\pi(i)} = \begin{cases} z^{(t)}_{\pi(i)} & \text{with probability } 1 - r', \\ z'_{\pi(i)} & \text{with probability } r', \end{cases}$
    where
        $r' = \log(1) \wedge$
(13)    $\log p\left(z'_{\pi(i)} \mid \alpha^{(t+1)}, \sigma^{2(t+1)}, P^{(t)}, \left\{z^{(t+1)}_{\pi(j)} \mid j < i\right\}, \left\{z^{(t)}_{\pi(j)} \mid j > i\right\}\right)$
        $- \log p\left(z^{(t)}_{\pi(i)} \mid \left\{z^{(t+1)}_{\pi(j)} \mid j < i\right\}, \left\{z^{(t)}_{\pi(j)} \mid j > i\right\} \alpha^{(t)}, \sigma^{2(t)}, P^{(t)}, \right)$
end for
```

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.

Intuitively, if the acceptance rate for a particular label 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.

Now, let us investigate the MCMC move for α and the remaining parameters. The adaptive structure is maintained as for the parameter z, with the required adaptation to a continuous context. The fact that α, σ^2 and P are continuous parameters actually simplifies the notation and the expressions, since there is no need of the intermediate passage in which we compute the difference

Algorithm 2 Metropolis-within-Gibbs update for α

```
Given  $\left(z^{(t+1)}, P^{(t)}, \sigma^{2(t)}\right)$

1.  Sample
(16)        $\alpha' \texttt{from} \, \mathrm{Normal}\left(\alpha^{(t-1)}, (\tau^2_{\alpha})^{(t-1)}\right)$
2.  Take
(17)        $\alpha^{(t+1)} = \begin{cases} \alpha^{(t)} & \texttt{with probability} \;\; 1 - r', \\ \alpha' & \texttt{with probability} \;\; r', \end{cases}$
where
            $r'_i = \log(1) \wedge$
(18)        $\log p\left(\alpha' | z^{(t+1)}, \alpha^{(t)}, P^{(t)}, \sigma^{2(t)}\right) - \log p\left(\alpha^{(t)} | z^{(t+1)}, \alpha^{(t)}, P^{(t)}, \sigma^{2(t)}\right)$
```

In algorithm (2) we have (18) which is given by

$$
\begin{aligned}
&\sum_{k=1}^{K-1} \mathrm{LogTruncatedNormal}\left(L^{(t)}_{(k)} \mid \left[\frac{(\beta_{\max} - 0.5)}{2K^{\alpha'}} \times \left(k^{\alpha'} + (k+1)^{\alpha'}\right) + \frac{1}{2}\right], (\sigma^2)^{(t)}, 0.5, \beta_{\max}\right) \\
&- \sum_{k=1}^{K-1} \mathrm{LogTruncatedNormal}\left(L^{(t)}_{(k)} \mid \left[\frac{(\beta_{\max} - 0.5)}{2K^{\alpha^{(t)}}} \times \left(k^{\alpha^{(t)}} + (k+1)^{\alpha^{(t)}}\right) + \frac{1}{2}\right], (\sigma^2)^{(t)}, 0.5, \beta_{\max}\right) \\
&+ \mathrm{LogUnif}\left(\alpha' \mid 0, 3\right) - \mathrm{LogUnif}\left((\sigma^2)^{(t)} \mid 0, 1\right)
\end{aligned} \tag{19}
$$

since the log-likelihood $\log\left(p(y \mid z, P)\right.$, the log prior probability on z, namely $\log p\left(z \mid \gamma\right)$, and the log prior prior probability on σ^2, namely $\log p\left(\sigma\right)$, are not affected by the new proposed value of α and therefore they subtract out.

In algorithm (3) we have that (22) is given by

$$
\begin{aligned}
&\sum_{k=1}^{K-1} \mathrm{LogTruncatedNormal}\left(L^{(t)}_{(k)} \mid \left[\frac{(\beta_{\max} - 0.5)}{2K^{\alpha^{(t)}}} \times \left(k^{\alpha^{(t)}} + (k+1)^{\alpha^{(t)}}\right) + \frac{1}{2}\right], (\sigma^2)', 0.5, \beta_{\max}\right) \\
&- \sum_{k=1}^{K-1} \mathrm{LogTruncatedNormal}\left(L^{(t)}_{(k)} \mid \left[\frac{(\beta_{\max} - 0.5)}{2K^{\alpha^{(t)}}} \times \left(k^{\alpha^{(t)}} + (k+1)^{\alpha^{(t)}}\right) + \frac{1}{2}\right], (\sigma^2)^{(t)}, 0.5, \beta_{\max}\right) \\
&+ \mathrm{LogUnif}\left(\alpha^{(t)} \mid 0, 3\right) - \mathrm{LogUnif}\left((\sigma^2)^{(t)} \mid 0, 1\right)
\end{aligned} \tag{23}
$$

Algorithm 3 Metropolis-within-Gibbs update for σ^2

```
Given  $\left(z^{(t+1)}, \alpha^{(t+1)}, P^{(t)}\right)$

1. Sample
(20)    $(\sigma^2)' \texttt{from} \,\mathrm{Normal}\left((\sigma^2)^{(t-1)}, (\tau^2_{\sigma^2})^{(t-1)}\right)$
2. Take
(21)    $(\sigma^2)^{(t+1)} = \begin{cases} (\sigma^2)^{(t)} & \texttt{with probability} \;\; 1 - r', \\ (\sigma^2)' & \texttt{with probability} \;\; r', \end{cases}$
where
 $r' = \log(1) \wedge$
(22)  $\log p\left((\sigma^2)' | z^{(t+1)}, \alpha^{(t+1)}, (\sigma^2)^{(t)}, P^{(t)}\right) - \log p\left((\sigma^2)^{(t)} | z^{(t+1)}, \alpha^{(t+1)}, (\sigma^2)^{(t)}, P^{(t)},\right)$
```

since the log-likelihood $\log\left(p(y \mid z, P)\right.$, the log prior probability on z, namely $\log p\left(z \mid \gamma\right)$, and the log prior prior probability on α^2, namely $\log p\left(\alpha\right)$, are not affected by the new proposed value of α and therefore they subtract out.

Algorithm 4 Metropolis-within-Gibbs update for P

```
Given  $\left(z^{(t+1)}, \alpha^{(t+1)}, \sigma^{2(t+1)}, P^{(t)}\right)$

for $i = 1$ to $K - 1$ do
    for $j = i + 1$ to $K$ do
        1. Sample
(24)        $p'_{ij} \texttt{from} \,\mathrm{Normal}\left(p^{(t)}_{ij}, (\tau^2_{p_{ij}})^{(t-1)}\right)$
        2. Take
(25)        $p^{(t+1)}_{ij} = \begin{cases} p^{(t)}_{ij} & \texttt{with probability} \;\; 1 - r', \\ p'_{ij} & \texttt{with probability} \;\; r', \end{cases}$
        where
    $r' = \log(1) \wedge$
      $\log p\left(p'_{i,j} \mid z^{(t+1)}, \alpha^{(t+1)}, \sigma^{2(t+1)}, \left\{p^{(t+1)}_{i^\star,j^\star} \mid i^\star < i, j^\star < j\right\}, \left\{p^{(t)}_{i^\star,j^\star} \mid i^\star > i, j^\star > j\right\}\right)$
(26)  $- \log p\left(p^{(t)}_{i,j} | z^{(t+1)}, \alpha^{(t+1)}, \sigma^{2(t+1)}, \left\{p^{(t+1)}_{i^\star,j^\star} \mid i^\star < i, j^\star < j\right\}, \left\{p^{(t)}_{i^\star,j^\star} \mid i^\star > i, j^\star > j\right\}\right)$
    end for
end for
```

In algorithm (4), (26) is given by:

(27)

$$\sum_{i=2}^{N-1}\sum_{j=i}^{N}\text{LogBinomial}\left(y_{ij}\mid n_{ij},p'_{z_i^{(t)}z_j^{(t)}}\right)-\sum_{i=2}^{N-1}\sum_{j=i}^{N}\text{LogBinomial}\left(y_{ij}\mid n_{ij},p^{(t)}_{z_i^{(t)}z_j^{(t)}}\right)$$

$$\sum_{k=1}^{K-1}\text{LogTruncatedNormal}\left(L'_{(k)}\mid\left[\frac{(\beta_{\max}-0.5)}{2K^{\alpha^{(t)}}}\times\left(k^{\alpha^{(t)}}+(k+1)^{\alpha^{(t)}}\right)+\frac{1}{2}\right],(\sigma^2)^{(t)},0.5,\beta_{\max}\right)$$

$$-\sum_{k=1}^{K-1}\text{LogTruncatedNormal}\left(L^{(t)}_{(k)}\mid\left[\frac{(\beta_{\max}-0.5)}{2K^{\alpha^{(t)}}}\times\left(k^{\alpha^{(t)}}+(k+1)^{\alpha^{(t)}}\right)+\frac{1}{2}\right],(\sigma^2)^{(t)},0.5,\beta_{\max}\right)$$

(28)

As before, the adaptive proposal variance is specific to each upper-triangular entry of P. Therefore, each entry's acceptance rate is monitored and the variance is adjusted accordingly.m

→ Insert here plots of convergence to the acceptance ratio

8. 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 measures

We obtain the point estimate $\hat{z}$ from the MCMC samples in two ways. The first one is via the MAP estimate, meaning that partition which maximises the a posteriori distribution.

$$\begin{aligned}\hat{z}_{\text{MAP}}(y) &= \arg\max_{\theta}\ f(\theta \mid y)\\ &= \arg\max_{\theta} \frac{f(y \mid \theta), g(\theta)}{\int_{\Theta} f(x \mid \vartheta), g(\vartheta)\, d\vartheta}\\ &= \arg\max_{\theta}\ f(x \mid \theta)\, g(\theta) \qquad (29)\end{aligned}$$

The second one is the partition that minimises the lowest averaged variation of information (VI distance) from the other clusterings, denoted in the following as $\hat{z}_{lbVI}$.

The VI fully utilise this posterior and it founded on a decision-theoretic approach introduced by Wade and Ghahramani (2018) for block modelling. 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 $\log 2N$, where N 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. The variation of information is a true distance since it obeys the triangle inequality.

Suppose we have two partitions of a set X and Y of a set A into disjoint subsets, namely $X = \{X_1, X_2, \ldots, X_k\}$ and $Y = \{Y_1, Y_2, \ldots, Y_l\}$.

Let: $n = \sum_i |X_i| = \sum_j |Y_j| = |A|$ $p_i = |X_i|/n$ and $q_j = |Y_j|/n$ $r_{ij} = |X_i \cap Y_j|/n$

Then the variation of information between the two partitions is:

$$\mathrm{VI}(X;Y) = -\sum_{i,j} r_{ij} \left[\log(r_{ij}/p_i) + \log(r_{ij}/q_j)\right]$$

.

To compare different models we use the WAIC loss, which yields practical and theoretical advantages with respect to other losses and has direct connections with Bayesian leave-one-out cross-validation, thus providing a measure of edge predictive accuracy.

Moreover, the calculation of the WAIC only requires posterior samples of the log-likelihoods for the edges:$\log p(y_{ij}|z, P, \alpha) = y_{ij} \log p_{z_i,z_j} + (n_{ij} - y_{ij}) \log(1 - p_{z_i,z_j}), \quad i = 2, \ldots, N, j = 1, \ldots, i-1$. These quantities are already available to the user, since the MCMC chain provides sample both of z and P. The use of the WAIC in a context similar to the present one is documented in Durante and Legramanti (2020).

9. Simulation Study from the Unordered Model N=100

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 Unordered Model.

We want to compare the quality of the Unordered model in recovering the ground truth from some data generated from the Unordered model itself. We compare its recovery performance with the one of the POMM model, which in the present context qualifies as a sort of 'mis-specified model', since the blocks do not present an inherent ordering.

Below, in figure (**??**), you may see the adjacency matrix of the simulated data. The darker is the pixel (i, j), the more are the victories of player i vs player j.

On the side, the different colours testify the different block membership of each single player.

Table 1. Time performance

Fitted Model	Seconds per iteration			Expected time for 30000 iterations		
	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.013 sec	0.015 sec	0.018 sec	6.5 min	7.5 min	9 min
Unordered model	0.013 sec	0.015 sec	0.018 sec	6.5 min	7.5 min	9 min

For the estimation purposes, we run 4 different chains of 30000 iterations each. We report in table (1) the execution time for the MCMC. Within that table, and in all other tables that will follow in the simulation study column (a) refers to the case $K = 3$, column (b) refers to the case $K = 5$, and column (c) to the case $K = 9$.

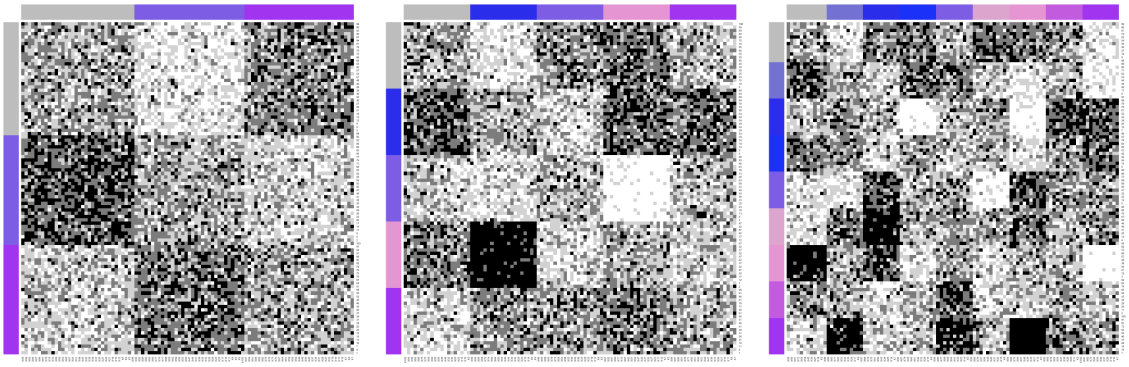
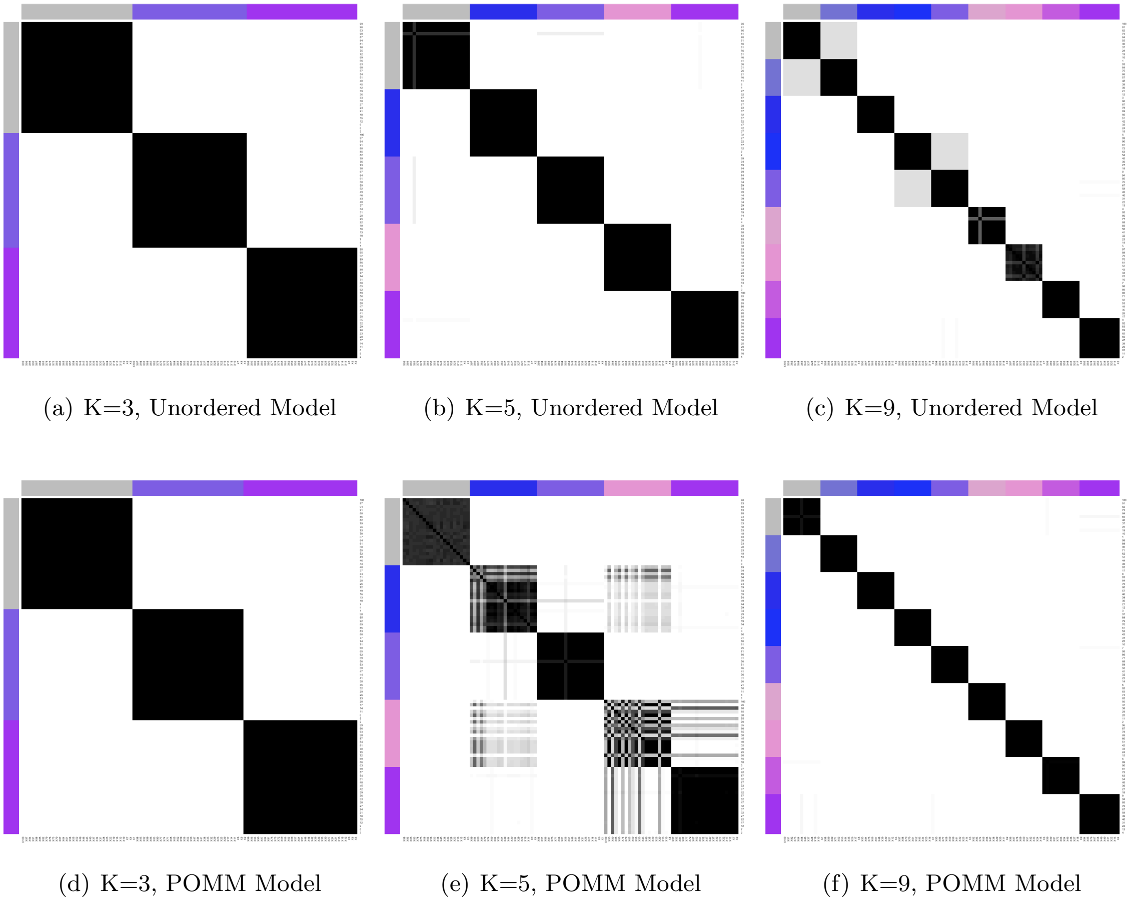

(a) K=3, Unordered Model (b) K=5, Unordered Model (c) K=9, Unordered Model

Figure 1. Adjacency Matrices simulated via the Unordered Model

Each chain is initiated with different starting values and different seeds. The initiation values are saved in order to guarantee the reproducibility of the results.

For the POMM model, we need to choose an appropriate value for $\beta_{\max}$, which controls the maximum attainable value within the matrix P. Here we fix it arbitrarily at 0.85.



(a) K=3, Unordered Model

(b) K=5, Unordered Model

(c) K=9, Unordered Model

(d) K=3, POMM Model

(e) K=5, POMM Model

(f) K=9, POMM Model

FIGURE 2. Co-Clustering Matrices obtained via the Unordered Model(above) and the POMM model (below).

The first results' plot we report is the one in figure (2). Each box contains a co-clustering matrix. Each pixel (i, j) represents the probability that two individuals are placed within the same cluster. The darker the pixel, the higher the probability. Colours on the side signal the true membership of each player.

In the first row, containing figure 2(a),figure 2(b), and figure 2(c), we have the co-clustering matrix for the Simple model estimated on the data generated according to the Simple model itself. This means that the blocks have no inherent ordering, and the model here is correctly specified. We can notice a very good recovery of the true membership.

In the second row instead, the one which contains figure 2(d),figure 2(e), and 2(f), we may observe the co-clustering matrix for the POMM model estimated on the data generated via the Simple one. Therefore, the model is misspecified, but we may notice that the recovery performance is quite competitive with the Simple one.

TABLE 2. P summary table
True Model Unordered, $N = 100$

Fitted Model	MAE			% within-95% CI interval			CI interval length		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.07	0.15	0.15	66.67	40	25.00	0.07	0.12	0.09
Unordered model	0.01	0.13	0.13	100.00	40	41.67	0.02	0.05	0.19

In table (2), we report the summary table of the estimates for the P matrix, both obtained via the Unordered model and the POMM model. For each value of K (again, reported in the sub-columns (a), (b) and (c), respectively), we report the mean absolute error, meaning the mean absolute error $MAE = \frac{1}{(K\cdot(K-1))/2}\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left(\hat{p}_{ij} - p_{ij}^{\star}\right)$ where $p_{ij}^{\star}$ is the true value of that particular entry. Then we also report the percentage of the upper triangular entries of P that are contained by the estimated 95% credible intervals, obtained by computing the 95% higher posterior density region, and the average 95% credible interval length.

Here below, we report the ground truth of the P matrix for each case $K = 3, 5, 9$. Then, next and below we report the estimates, both for the POMM and the Unordered model, within the $\hat{P}_{\text{POMM}}^{K=k}$, $\hat{P}_{\text{Unordered}}^{K=k}$, respectively.

$$P_{true}^{K=3} = \begin{bmatrix} 0.500 & 0.230 & 0.631 \\ 0.770 & 0.500 & 0.327 \\ 0.369 & 0.673 & 0.500 \end{bmatrix} \quad \hat{P}_{\text{POMM}}^{K=3} = \begin{bmatrix} 0.500 & 0.222 & 0.50 \\ 0.778 & 0.500 & 0.41 \\ 0.500 & 0.590 & 0.50 \end{bmatrix}$$

$$\hat{P}_{\text{Unordered}}^{K=3} = \begin{bmatrix} 0.500 & 0.222 & 0.639 \\ 0.778 & 0.500 & 0.321 \\ 0.361 & 0.679 & 0.500 \end{bmatrix}$$

$$P_{true}^{K=5} = \begin{bmatrix} 0.500 & 0.230 & 0.631 & 0.706 & 0.422 \\ 0.770 & 0.500 & 0.327 & 0.752 & 0.714 \\ 0.369 & 0.248 & 0.500 & 0.036 & 0.441 \\ 0.673 & 0.964 & 0.286 & 0.500 & 0.365 \\ 0.294 & 0.578 & 0.559 & 0.635 & 0.500 \end{bmatrix} \quad \hat{P}_{\text{POMM}}^{K=5} = \begin{bmatrix} 0.500 & 0.228 & 0.614 & 0.438 & 0.532 \\ 0.772 & 0.500 & 0.515 & 0.473 & 0.562 \\ 0.386 & 0.485 & 0.500 & 0.403 & 0.421 \\ 0.562 & 0.527 & 0.597 & 0.500 & 0.467 \\ 0.468 & 0.438 & 0.579 & 0.533 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\text{Unordered}}^{K=5} = \begin{bmatrix} 0.500 & 0.226 & 0.622 & 0.522 & 0.560 \\ 0.774 & 0.500 & 0.511 & 0.427 & 0.571 \\ 0.378 & 0.439 & 0.500 & 0.372 & 0.435 \\ 0.478 & 0.591 & 0.628 & 0.500 & 0.350 \\ 0.440 & 0.429 & 0.565 & 0.650 & 0.500 \end{bmatrix}$$

$$P_{true}^{K=9} = \begin{bmatrix} 0.500 & 0.230 & 0.631 & 0.706 & 0.422 & 0.765 & 0.720 & 0.554 & 0.231 \\ 0.770 & 0.500 & 0.327 & 0.752 & 0.714 & 0.363 & 0.197 & 0.512 & 0.118 \\ 0.369 & 0.559 & 0.500 & 0.036 & 0.441 & 0.542 & 0.034 & 0.795 & 0.770 \\ 0.673 & 0.635 & 0.280 & 0.500 & 0.365 & 0.458 & 0.262 & 0.525 & 0.722 \\ 0.294 & 0.235 & 0.803 & 0.446 & 0.500 & 0.082 & 0.764 & 0.567 & 0.553 \\ 0.248 & 0.637 & 0.966 & 0.488 & 0.565 & 0.500 & 0.712 & 0.435 & 0.636 \\ 0.964 & 0.458 & 0.738 & 0.205 & 0.525 & 0.230 & 0.500 & 0.475 & 0.020 \\ 0.578 & 0.542 & 0.236 & 0.475 & 0.769 & 0.278 & 0.364 & 0.500 & 0.382 \\ 0.286 & 0.918 & 0.288 & 0.433 & 0.882 & 0.447 & 0.980 & 0.618 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\text{POMM}}^{K=9} = \begin{bmatrix} 0.500 & 0.293 & 0.559 & 0.488 & 0.518 & 0.499 & 0.404 & 0.485 & 0.453 \\ 0.707 & 0.500 & 0.503 & 0.496 & 0.615 & 0.340 & 0.466 & 0.504 & 0.202 \\ 0.441 & 0.497 & 0.500 & 0.385 & 0.316 & 0.302 & 0.203 & 0.638 & 0.498 \\ 0.512 & 0.504 & 0.615 & 0.500 & 0.509 & 0.501 & 0.523 & 0.549 & 0.632 \\ 0.482 & 0.385 & 0.684 & 0.491 & 0.500 & 0.264 & 0.631 & 0.503 & 0.332 \\ 0.501 & 0.660 & 0.698 & 0.499 & 0.736 & 0.500 & 0.714 & 0.586 & 0.532 \\ 0.596 & 0.534 & 0.797 & 0.477 & 0.369 & 0.286 & 0.500 & 0.535 & 0.201 \\ 0.515 & 0.496 & 0.362 & 0.451 & 0.497 & 0.414 & 0.465 & 0.500 & 0.358 \\ 0.547 & 0.798 & 0.502 & 0.368 & 0.668 & 0.468 & 0.799 & 0.642 & 0.500 \end{bmatrix}$$

$$\hat{P}^{K=9}_{\text{Unordered}} = \begin{bmatrix} 0.500 & 0.351 & 0.620 & 0.484 & 0.576 & 0.756 & 0.371 & 0.449 & 0.456 \\ 0.649 & 0.500 & 0.445 & 0.466 & 0.713 & 0.371 & 0.432 & 0.519 & 0.210 \\ 0.380 & 0.555 & 0.500 & 0.458 & 0.325 & 0.302 & 0.149 & 0.761 & 0.739 \\ 0.516 & 0.534 & 0.542 & 0.500 & 0.481 & 0.474 & 0.582 & 0.536 & 0.652 \\ 0.424 & 0.287 & 0.675 & 0.519 & 0.500 & 0.290 & 0.616 & 0.401 & 0.375 \\ 0.244 & 0.629 & 0.698 & 0.526 & 0.710 & 0.500 & 0.717 & 0.595 & 0.568 \\ 0.629 & 0.568 & 0.851 & 0.418 & 0.384 & 0.283 & 0.500 & 0.559 & 0.058 \\ 0.551 & 0.481 & 0.239 & 0.464 & 0.599 & 0.405 & 0.441 & 0.500 & 0.323 \\ 0.544 & 0.790 & 0.261 & 0.348 & 0.625 & 0.432 & 0.942 & 0.677 & 0.500 \end{bmatrix}$$

TABLE 3. z summary table
True Model Unordered, $N = 100$

Method	VI distance$_{\text{MAP}}$			VI distance$_{\text{VI lb}}$			WAIC		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0	0	0	0	0.56	0	48883.71 145.40	53337.62 152.84	51244.98 153.63
Unordered model	0	0	0	0	0.00	0	46521.89 144.55	52382.24 153.98	49144.47 164.65

In table (3), we report some summary statistics for the parameter z. As above, we have columns (a),(b) and (c) representing the case $K = 3, 5, 9$ respectively.

The first indicator is the VIdistance$_{\text{MAP}}$ computed between the true partition $z^\star$ and the point estimate $\hat{z}^{\text{MAP}}$ obtained with the maximum a posteriori estimate (MAP).

The second one is VIdistance$_{\text{VI lb}}$ computed between the true partition $z^\star$ and the point estimate $\hat{z}^{\text{VI lb}}$ obtained with the partition attaining the VI lower bound.

The third one is the WAIC estimate, along with its standard error below. We see that the Unordered Model is the one preferred in this case.

TABLE 4. POMM hyperparameters summary table
True Model Unordered, $N = 100$

Fitted Model	$\hat{\theta}$			95% CI		
	(a)	(b)	(c)	(a)	(b)	(c)
σ	0.51	0.57	0.64	[0.13 0.89]	[0.24 0.9]	[0.34 0.9]
α	0.48	0.52	0.59	[0.12 0.87]	[0.15 0.88]	[0.19 0.9]

In table (4), we report the results for the hyperparameters of the $POMM$ model. In the $\hat{\theta}$ column we report the estimates both for α and σ, while on the right we have their 95% Credible Interval. Given that the data were generated according the Unordered model, we do not have a ground truth to which these results should be compared. However, we can still try to make sense of these values by inspecting the properties of the induced P^{POMM} matrix resulting from the estimates.

The Unordered model has a prior over $P^{\text{Unordered}} \sim Beta(1,1)$, then we may expect P^{POMM} itself to get as closer as needed to a uniform distribution by selecting the appropriate combination of (α, σ).

9.0.1. *Unordered Model check.* In this subsection, we report some diagnostic checks for the algorithm of the Markov Chains, to assess convergence, quality of mixing, and the overall behaviour of the Metropolis-within-Gibbs algorithm.

Table 5. z diagnostic table
True Model Unordered, $N = 100$

Fitted Model	$\overline{ESS}$			$\overline{ACF_{30}}$			$\overline{\%accepted}$			$\overline{Gelman - Rubin}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	7558.60	301.04	2880.01	0	0.01	0.08	0.02	0.04	0.15	1.61	1.29	1.31
Unordered	6544.85	253.85	140.78	0	0.03	0.11	0.01	0.56	0.15	1.04	1.23	1.26

In table (5) we report the diagnostics for the z parameter. Since the parameter is a label vector with 100 entries in this case, we compute the relevant statistics for each single entries an then we report the average.

- The first statistics is the Effective Sample Size (ESS) averaged over individuals $i = 1, \ldots, N$, which denotes a fairly good sample size. The average is taken as follows:

$$\overline{ESS} = \frac{1}{n} \sum_{i=1}^{N} ESS_i$$

  The same is applied also to the other diagnostic metrics.
- Then, we report the average autocorrelation, $\overline{ACF_{30}}$, computed with a lag of 30 iterations. This values are close to zero, meaning that there is very little correlation within the chain.
- In the third column, we report the average acceptance rate $\overline{\%accepted}$. These are significantly lower than the target acceptance rate that should be hit by the adaptive MCMC, which is 22%. However, the estimates are capable of correctly recovering the true partition. Combining the two facts, we may hypothesise that the simulation study has too "many" data, and proposing for a given individual i, who has already been assigned to the true block, a block different from the true one, leads to a drop in the likelihood which is too large, and as a consequence, we always reject the other labels.
- Finally, we compute the median Gelman-Rubin statistics for each entry, $\overline{Gelman - Rubin}$. Gelman and Rubin (1992) propose a general approach to monitoring convergence of MCMC output in which $m > 1$ parallel chains are run with starting values that are overdispersed relative to the posterior distribution. Convergence is diagnosed when the chains have 'forgotten' their initial values, and the output from all chains is indistinguishable. The Gelman-Rubin diagnostic is applied to a single variable

from the chain. It is based a comparison of within-chain and between-chain variances, and is similar to a classical analysis of variance. Values substantially above 1 indicate lack of convergence. If the chains have not converged,

Table 6. P diagnostic table
True Model Unordered, $N = 100$

Fitted Model	$\overline{ESS}$			$\overline{ACF_{30}}$			$\overline{\%accepted}$			$\overline{Gelman - Rubin}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	345.67	639.8	704.53	0.35	0.05	0.06	7.53	19.06	21.60	1.03	4.23	1.01
Unordered	570.00	828.1	937.67	0.00	0.04	0.03	11.66	18.40	29.06	1.00	1.01	2.23

In table (6) we report the same diagnostics checks for the z parameter. The only difference is that here we do not average the diagnostics indicators over the individuals $i = 1, \dots, N$, but instead over the upper-triangular P indices: $\{i = 1, \dots, K-1, \quad j = i+1, \dots, K\}$.

Just as an example, the average ESS in this case is obtained as

$$\overline{ESS} = \frac{1}{(K \cdot (K-1))/2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ESS_{i,j}$$

Table 7. POMM hyperparameters diagnostic table
True Model Unordered, $N = 100$

Fitted Model	ESS			ACF$_{30}$			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
σ	981	1041	842	0.01	0.01	0.01	39.03	36.59	32	1.08	1	1.02
α	26	17	19	0.82	0.82	0.82	25.36	25.18	24.74	1.33	1.21	1.19

Finally, in table (7) we report again the same diagnostics, but since both α and σ are one-dimensional, we are presenting the diagnostics itself, without any average.

10. Simulation Study from the WST Model N=100

Differing from the Unordered model, the WST model has the constraint that the upper triangular entries of the matrix P are all greater than 0.5.

We want to compare the quality of the POMM model in recovering the ground truth from some data generated from the WST model itself. We compare its recovery performance with the one of the POMM model, which in the present context qualifies as a sort of 'mis-specified model', since the blocks do not present an inherent ordering.

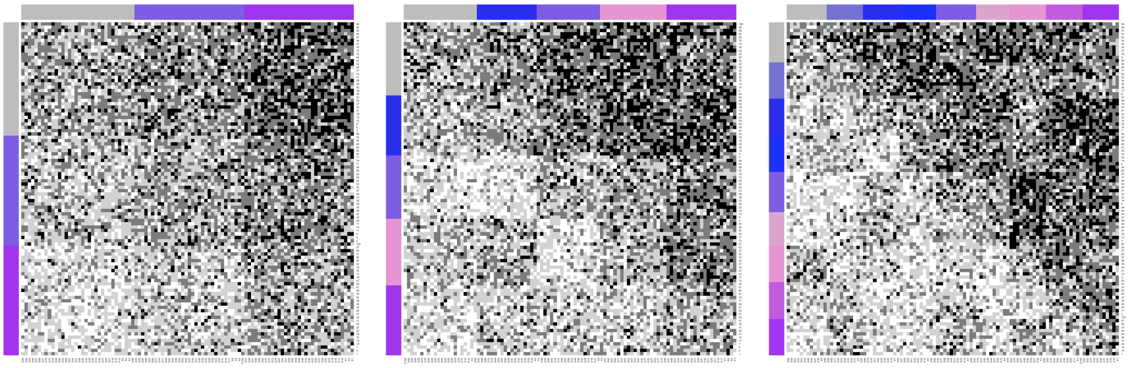
Below, in figure (3), you may see the adjacency matrix of the simulated data. The darker is the pixel (i, j), the more are the victories of player i vs player j.

On the side, the different colours testify the different block membership of each single player.

Table 8. Time performance

Fitted Model	Seconds per iteration			Expected time for 30000 iterations		
	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.013 sec	0.015 sec	0.018 sec	6.5 min	7.5 min	9 min
WST model	0.013 sec	0.015 sec	0.018 sec	6.5 min	7.5 min	9 min

For the estimation purposes, we run 4 different chains of 30000 iterations each. We report in table (8) the execution time for the MCMC. Within that table, and in all other tables that will follow in the simulation study column (a) refers to the case $K = 3$, column (b) refers to the case $K = 5$, and column (c) to the case $K = 9$.


(a) K=3, WST Model

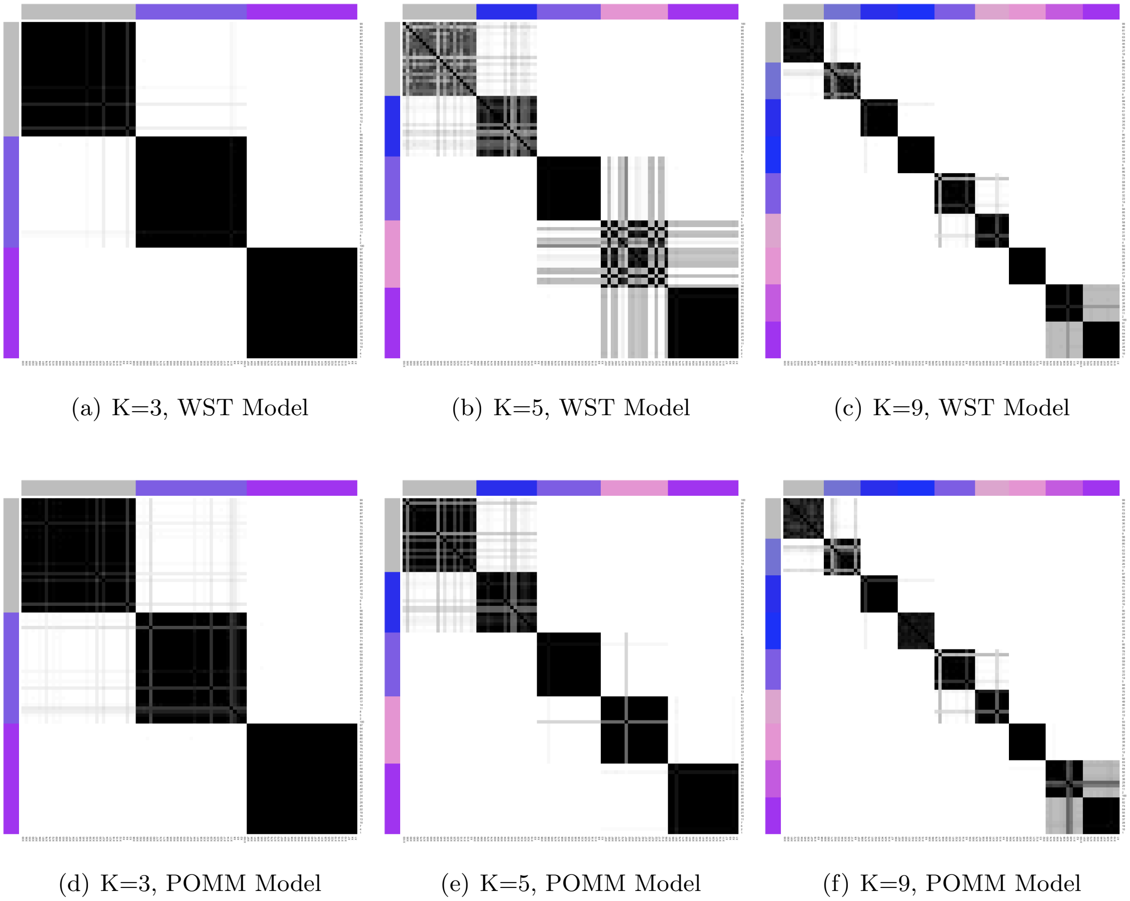
(b) K=5, WST Model

(c) K=9, WST Model

Figure 3. Adjacency Matrices simulated via the WST Model

Each chain is initiated with different starting values and different seeds. The initiation values are saved in order to guarantee the reproducibility of the results.

For the POMM model, we need to choose an appropriate value for $\beta_{\max}$, which controls the maximum attainable value within the matrix P. Here we fix it arbitrarily at 0.85.



(a) K=3, WST Model

(b) K=5, WST Model

(c) K=9, WST Model

(d) K=3, POMM Model

(e) K=5, POMM Model

(f) K=9, POMM Model

FIGURE 4. Co-Clustering Matrices obtained via the WST Model(above) and the POMM model (below).

The first results' plot we report is the one in figure (4). Each box contains a co-clustering matrix. Each pixel (i, j) represents the probability that two individuals are placed within the same cluster. The darker the pixel, the higher the probability. Colours on the side signal the true membership of each player.

In the first row, containing figure 4(a),figure 4(b), and figure 4(c), we have the co-clustering matrix for the WST model estimated on the data generated according to the WST model itself. This means that the blocks have no inherent ordering, and the model here is correctly specified. We can notice a very good recovery of the true membership.

In the second row instead, the one which contains figure 4(d),figure 4(e), and 4(f), we may observe the co-clustering matrix for the POMM model estimated on the data generated via the WST one. Therefore, the model is misspecified, but we may notice that the recovery performance is quite competitive with the WST one.

TABLE 9. P summary table
True Model WST, $N = 100$

Fitted Model	MAE			% within-95% CI interval			CI interval length		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.02	0.06	0.06	100	40	72.22	0.12	0.04	0.16
WST model	0.00	0.06	0.06	100	50	69.44	0.02	0.10	0.15

In table (9), we report the summary table of the estimates for the P matrix, both obtained via the WST model and the POMM model. For each value of K (again, reported in the sub-columns (a), (b) and (c), respectively), we report the mean absolute error, meaning the mean absolute error $MAE = \frac{1}{(K\cdot(K-1))/2}\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left(\hat{p}_{ij} - p^{\star}_{ij}\right)$ where $p^{\star}_{ij}$ is the true value of that particular entry. Then we also report the percentage of the upper triangular entries of P that are contained by the estimated 95% credible intervals, obtained by computing the 95% higher posterior density region, and the average 95% credible interval length.

Here below, we report the ground truth of the P matrix for each case $K = 3, 5, 9$. Then, next and below we report the estimates, both for the POMM and the WST model, within the $\hat{P}^{K=k}_{\text{POMM}}$, $\hat{P}^{K=k}_{\text{WST}}$, respectively.

$$P_{true}^{K=3} = \begin{bmatrix} 0.500 & 0.586 & 0.736 \\ 0.414 & 0.500 & 0.623 \\ 0.264 & 0.377 & 0.500 \end{bmatrix} \quad \hat{P}_{\mathrm{POMM}}^{K=3} = \begin{bmatrix} 0.500 & 0.600 & 0.754 \\ 0.400 & 0.500 & 0.622 \\ 0.246 & 0.378 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\mathrm{WST}}^{K=3} = \begin{bmatrix} 0.500 & 0.592 & 0.739 \\ 0.408 & 0.500 & 0.627 \\ 0.261 & 0.373 & 0.500 \end{bmatrix}$$

$$P_{true}^{K=5} = \begin{bmatrix} 0.500 & 0.586 & 0.736 & 0.765 & 0.658 \\ 0.414 & 0.500 & 0.623 & 0.782 & 0.768 \\ 0.264 & 0.218 & 0.500 & 0.514 & 0.665 \\ 0.377 & 0.486 & 0.232 & 0.500 & 0.637 \\ 0.235 & 0.342 & 0.335 & 0.363 & 0.500 \end{bmatrix} \quad \hat{P}_{\mathrm{POMM}}^{K=5} = \begin{bmatrix} 0.500 & 0.589 & 0.731 & 0.687 & 0.718 \\ 0.411 & 0.500 & 0.703 & 0.640 & 0.700 \\ 0.269 & 0.297 & 0.500 & 0.643 & 0.658 \\ 0.313 & 0.360 & 0.357 & 0.500 & 0.653 \\ 0.282 & 0.300 & 0.342 & 0.347 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\mathrm{WST}}^{K=5} = \begin{bmatrix} 0.500 & 0.566 & 0.660 & 0.700 & 0.710 \\ 0.434 & 0.500 & 0.646 & 0.675 & 0.703 \\ 0.340 & 0.354 & 0.500 & 0.666 & 0.665 \\ 0.300 & 0.325 & 0.334 & 0.500 & 0.652 \\ 0.290 & 0.297 & 0.335 & 0.348 & 0.500 \end{bmatrix}$$

$$P_{true}^{K=9} = \begin{bmatrix} 0.500 & 0.586 & 0.736 & 0.765 & 0.658 & 0.787 & 0.770 & 0.708 & 0.587 \\ 0.414 & 0.500 & 0.623 & 0.782 & 0.768 & 0.636 & 0.574 & 0.692 & 0.544 \\ 0.264 & 0.335 & 0.500 & 0.514 & 0.665 & 0.703 & 0.513 & 0.798 & 0.789 \\ 0.377 & 0.363 & 0.230 & 0.500 & 0.637 & 0.672 & 0.598 & 0.697 & 0.771 \\ 0.235 & 0.213 & 0.426 & 0.292 & 0.500 & 0.531 & 0.786 & 0.713 & 0.707 \\ 0.218 & 0.364 & 0.487 & 0.308 & 0.337 & 0.500 & 0.767 & 0.663 & 0.739 \\ 0.486 & 0.297 & 0.402 & 0.202 & 0.322 & 0.211 & 0.500 & 0.678 & 0.507 \\ 0.342 & 0.328 & 0.214 & 0.303 & 0.413 & 0.229 & 0.261 & 0.500 & 0.643 \\ 0.232 & 0.469 & 0.233 & 0.287 & 0.456 & 0.293 & 0.493 & 0.357 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\mathrm{POMM}}^{K=9} = \begin{bmatrix} 0.500 & 0.572 & 0.666 & 0.707 & 0.723 & 0.712 & 0.705 & 0.626 & 0.695 \\ 0.428 & 0.500 & 0.646 & 0.689 & 0.716 & 0.725 & 0.667 & 0.656 & 0.635 \\ 0.334 & 0.354 & 0.500 & 0.645 & 0.649 & 0.656 & 0.614 & 0.701 & 0.729 \\ 0.293 & 0.311 & 0.355 & 0.500 & 0.651 & 0.666 & 0.666 & 0.687 & 0.751 \\ 0.277 & 0.284 & 0.351 & 0.349 & 0.500 & 0.631 & 0.714 & 0.669 & 0.692 \\ 0.288 & 0.275 & 0.344 & 0.334 & 0.369 & 0.500 & 0.667 & 0.732 & 0.674 \\ 0.295 & 0.333 & 0.386 & 0.334 & 0.286 & 0.333 & 0.500 & 0.687 & 0.668 \\ 0.374 & 0.344 & 0.299 & 0.313 & 0.331 & 0.268 & 0.313 & 0.500 & 0.603 \\ 0.305 & 0.365 & 0.271 & 0.249 & 0.308 & 0.326 & 0.332 & 0.397 & 0.500 \end{bmatrix}$$

$$\hat{P}^{K=9}_{\mathrm{WST}} = \begin{bmatrix} 0.500 & 0.601 & 0.675 & 0.711 & 0.702 & 0.744 & 0.702 & 0.659 & 0.709 \\ 0.399 & 0.500 & 0.623 & 0.671 & 0.734 & 0.702 & 0.653 & 0.647 & 0.581 \\ 0.325 & 0.377 & 0.500 & 0.652 & 0.640 & 0.653 & 0.624 & 0.691 & 0.720 \\ 0.289 & 0.329 & 0.348 & 0.500 & 0.666 & 0.659 & 0.656 & 0.661 & 0.771 \\ 0.298 & 0.266 & 0.360 & 0.334 & 0.500 & 0.631 & 0.709 & 0.671 & 0.686 \\ 0.256 & 0.298 & 0.347 & 0.341 & 0.369 & 0.500 & 0.668 & 0.738 & 0.669 \\ 0.298 & 0.347 & 0.376 & 0.344 & 0.291 & 0.332 & 0.500 & 0.734 & 0.617 \\ 0.341 & 0.353 & 0.309 & 0.339 & 0.329 & 0.262 & 0.266 & 0.500 & 0.646 \\ 0.291 & 0.419 & 0.280 & 0.229 & 0.314 & 0.331 & 0.383 & 0.354 & 0.500 \end{bmatrix}$$

Table 10. z summary table
True Model WST, $N = 100$

Method	VI distance$_{\mathrm{MAP}}$			VI distance$_{\mathrm{VI\ lb}}$			WAIC		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0	0.42	0.1	0	0.42	0.32	-13237.20 30.89	-13352.68 30.77	13657.35 31.35
WST model	0	0.42	0.1	0	0.42	0.32	-13273.59 31.16	-13293.26 30.37	-13673.15 31.34

In table (10), we report some summary statistics for the parameter z. As above, we have columns (a),(b) and (c) representing the case $K = 3, 5, 9$ respectively.

The first indicator is the VIdistance$_{\mathrm{MAP}}$ computed between the true partition $z^\star$ and the point estimate $\hat{z}^{\mathrm{MAP}}$ obtained with the maximum a posteriori estimate (MAP).

The second one is VIdistance$_{\mathrm{VI\ lb}}$ computed between the true partition $z^\star$ and the point estimate $\hat{z}^{\mathrm{VI\ lb}}$ obtained with the partition attaining the VI lower bound.

The third one is the WAIC estimate, along with its standard error below.

Table 11. POMM hyperparameters summary table
True Model WST, $N = 100$

Fitted Model	$\hat{\theta}$			95% CI		
	(a)	(b)	(c)	(a)	(b)	(c)
σ	0.19	0.18	0.15	[0.01 0.79]	[0.04 0.65]	[0.07 0.27]
α	0.60	0.41	0.23	[0.24 0.9]	[0.11 0.75]	[0.1 0.4]

In table (11), we report the results for the hyperparameters of the $POMM$ model. In the $\hat{\theta}$ column we report the estimates both for α and σ, while on the right we have their 95% Credible Interval. Given that the data were generated according the WST model, we do not have a ground truth to which these results should be compared. However, we can still try to make sense of these values by inspecting the properties of the induced P^{POMM} matrix resulting from the estimates.

The WST model has a prior over $P^{\text{WST}} \sim Beta(1,1)$, then we may expect P^{POMM} itself to get as closer as needed to a uniform distribution by selecting the appropriate combination of (α, σ).

Therefore, we simulate $n = 10000$ P^{POMM} matrices via the estimated parameters $\hat{\theta}$ in (11). Then we extract 1000 points from each level set to avoid sample biases, and we compare them with an equally-sized set simulated via the WST model, using the Kolmogorov-Smirnov test, where the null hypothesis is that two sets of points are sampled from the same distribution.

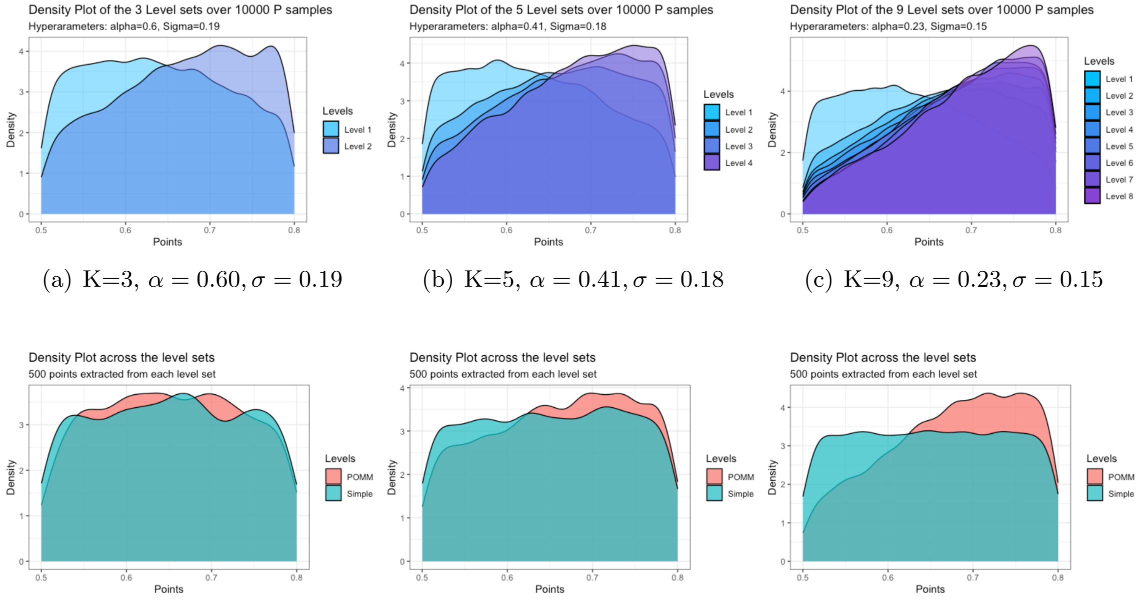
Table 12. Kolmogorov-Smirnov test
Data are generated via the estimated parameters

Method	p-value			% overlap between level sets		
	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.26	0.05	0.00	82%	86%	89%

In table (12) we report first the Kolmogorov-Smirnov test p-values, and we may notice that for $K = 3, 5$ we do not reject with an $\alpha = 5\%$ that the P^{POMM} is compatible with being extracted from the WST model.

Instead with $K = 9$ we cannot say that P^{POMM} has collapsed to a uniform, but at the same time, if we look at the area of overlap between the densities of the level sets, we may notice that there is a significant amount of overlap between them, allowing the POMM model to effectively replicate the Unordered entries of the P^{WST} matrix.

In figure (5) we report the densities of the points generated via the estimated hyperparameters. We may notice the difference for the case $K = 9$ with respect to the other two. In this case, we have very significant distributions for the POMM and the WST one.



(a) K=3, $\alpha = 0.60, \sigma = 0.19$ (b) K=5, $\alpha = 0.41, \sigma = 0.18$ (c) K=9, $\alpha = 0.23, \sigma = 0.15$

(d) K=3, $\alpha = 0.60, \sigma = 0.19$ (e) K=5, $\alpha = 0.41, \sigma = 0.18$ (f) K=9, $\alpha = 0.23, \sigma = 0.15$

FIGURE 5. These are the densities of the entries of 10000 P matrices generated according to the parameters within brackets, that is, the parameters estimated according the POMM model on the data generated via the WST one. In figures (5(a)), (5(b)), (5(c)) we have the densities of the level sets coloured differently. In figures (12(d)), (12(e)), (12(f)) we put together 1000 points extracted from each level sets and we compute the density, so to have an overview of the joint distribution. We also compare the P's entries simulated via the POMM and the WST model

10.0.1. *WST Model check.* In this subsection, we report some diagnostic checks for the algorithm of the Markov Chains, to assess convergence, quality of mixing, and the overall behaviour of the Metropolis-within-Gibbs algorithm.

Table 13. z diagnostic table
True Model WST, $N = 100$

Fitted Model	$\overline{ESS}$			$\overline{ACF_{30}}$			$\overline{\%accepted}$			$\overline{Gelman - Rubin}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	7558.60	7988.02	2880.01	0	0.01	0.08	0.02	0.04	0.15	1.29	1.03	1.31
WST	6544.85	6688.51	2438.17	0	0.03	0.11	0.01	0.56	0.15	1.04	1.29	1.26

In table (13) we report the diagnostics for the z parameter. Since the parameter is a label vector with 100 entries in this case, we compute the relevant statistics for each single entries an then we report the average.

- The first statistics is the Effective Sample Size (ESS) averaged over individuals $i = 1, \dots, N$, which denotes a fairly good sample size. The average is taken as follows:

$$\overline{ESS} = \frac{1}{n} \sum_{i=1}^{N} ESS_i$$

  The same is applied also to the other diagnostic metrics.
- Then, we report the average autocorrelation, $\overline{ACF_{30}}$, computed with a lag of 30 iterations. This values are close to zero, meaning that there is very little correlation within the chain.
- In the third column, we report the average acceptance rate $\overline{\%accepted}$. These are significantly lower than the target acceptance rate that should be hit by the adaptive MCMC, which is 22%. However, the estimates are capable of correctly recovering the true partition. Combining the two facts, we may hypothesise that the simulation study has too "many" data, and proposing for a given individual i, who has already been assigned to the true block, a block different from the true one, leads to a drop in the likelihood which is too large, and as a consequence, we always reject the other labels.
- Finally, we compute the median Gelman-Rubin statistics for each entry, $\overline{Gelman - Rubin}$. Gelman and Rubin (1992) propose a general approach to monitoring convergence of MCMC output in which $m > 1$ parallel chains are run with starting values that are overdispersed relative to the posterior distribution. Convergence is diagnosed when the chains have 'forgotten' their initial values, and the output from all chains is indistinguishable. The Gelman-Rubin diagnostic is applied to a single variable

from the chain. It is based a comparison of within-chain and between-chain variances, and is similar to a classical analysis of variance. Values substantially above 1 indicate lack of convergence. If the chains have not converged,

Table 14. P diagnostic table
True Model WST, $N = 100$

Fitted Model	$\overline{ESS}$			$\overline{ACF_{30}}$			$\overline{\%accepted}$			$\overline{Gelman - Rubin}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	2234.00	2101.1	1454.08	0.02	0.00	0.15	34.52	31.55	29.98	1.06	1.0	1.04
WST	2742.67	1446.9	1565.14	0.00	0.13	0.17	36.85	31.01	30.06	1.00	1.9	1.03

In table (14) we report the same diagnostics checks for the z parameter. The only difference is that here we do not average the diagnostics indicators over the individuals $i = 1, \dots, N$, but instead over the upper-triangular P indices: $\{i = 1, \dots, K-1, \quad j = i+1, \dots, K\}$.

Just as an example, the average ESS in this case is obtained as

$$\overline{ESS} = \frac{1}{(K \cdot (K-1))/2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ESS_{i,j}$$

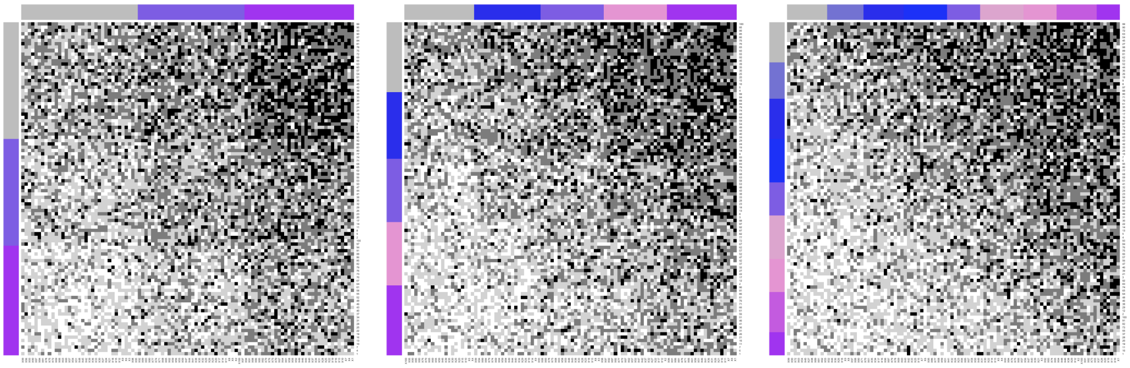
Table 15. POMM hyperparameters diagnostic table
True Model WST, $N = 100$

Fitted Model	ESS			ACF$_{30}$			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
σ	985	155	162	0.5	0.55	0.55	31.47	29.96	29.81	1.96	1.02	1.18
α	13	17	45	0.95	0.94	0.84	24.74	24.93	24.38	1.21	1.62	1.01

Finally, in table (15) we report again the same diagnostics, but since both α and σ are one-dimensional, we are presenting the diagnostics itself, without any average.

11. Simulation Study from the POMM Model N=100

In this section we reverse the exercise performed in previous one. Before we were simulating from the WST model, now we are simulating from the POMM, with $K = 3, 5, 9$. The metrics, indices and summaries are the same as before, so we avoid replicating the same explanations.

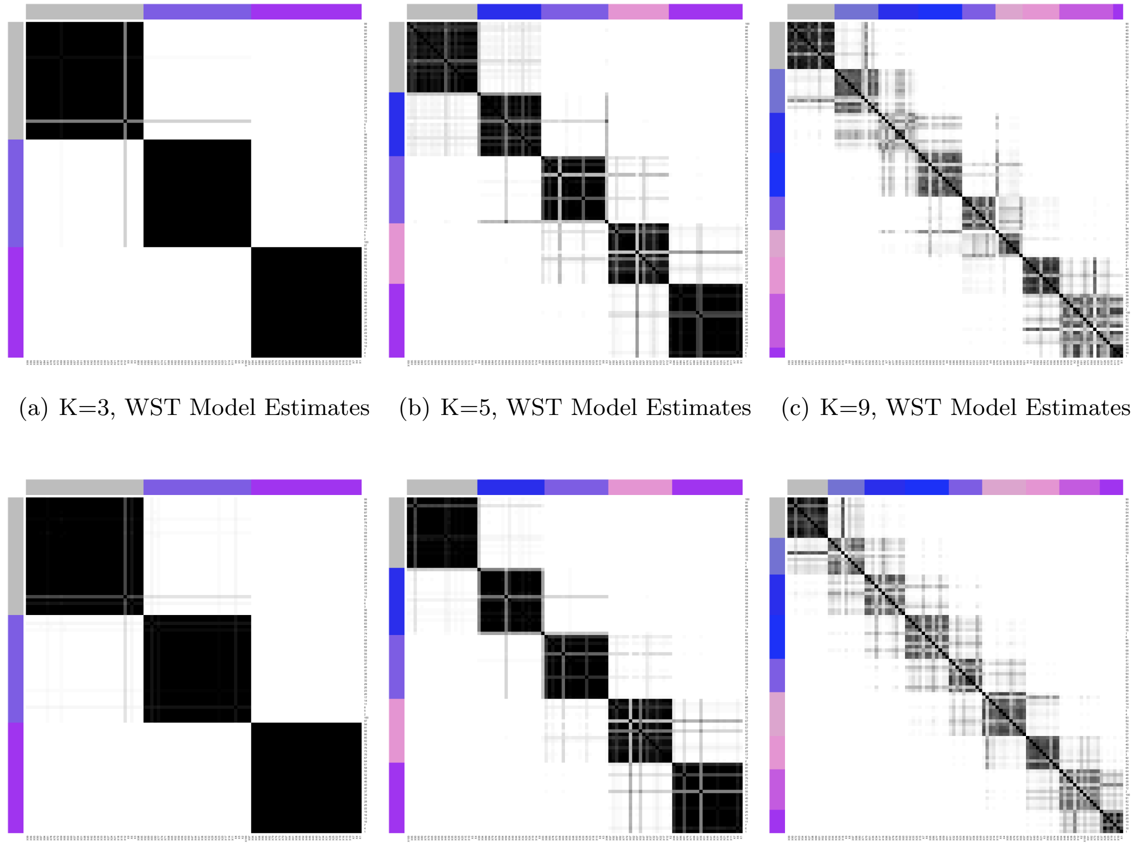


(a) K=3, POMM Model (b) K=5, POMM Model (c) K=9, POMM Model

Figure 6. Adjacency Matrices simulated via the POMM Model

P summary table
True Model POMM, $K = 3$, $N = 100$

Fitted Model	$\overline{MAE}$			% within-95%-CI interval			$\overline{\text{CI interval length}}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.02	0.02	0.01	100%	90%	61.11%	0.14	0.08	0.02
WST model	0.00	0.04	0.02	100%	90%	100.00%	0.02	0.11	0.15



(a) K=3, WST Model Estimates (b) K=5, WST Model Estimates (c) K=9, WST 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 WST Model(above) and the POMM model (below).

$$P_{true}^{K=3} = \begin{bmatrix} 0.500 & 0.600 & 0.754 \\ 0.400 & 0.500 & 0.622 \\ 0.246 & 0.378 & 0.500 \end{bmatrix} \qquad \hat{P}_{\text{POMM}}^{K=3} = \begin{bmatrix} 0.500 & 0.582 & 0.726 \\ 0.418 & 0.500 & 0.649 \\ 0.274 & 0.351 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\text{Simple}}^{K=3} = \begin{bmatrix} 0.500 & 0.606 & 0.758 \\ 0.394 & 0.500 & 0.623 \\ 0.242 & 0.377 & 0.500 \end{bmatrix}$$

$$P_{true}^{K=5} = \begin{bmatrix} 0.500 & 0.569 & 0.679 & 0.752 & 0.781 \\ 0.431 & 0.500 & 0.576 & 0.698 & 0.741 \\ 0.321 & 0.424 & 0.500 & 0.562 & 0.674 \\ 0.248 & 0.302 & 0.438 & 0.500 & 0.571 \\ 0.219 & 0.259 & 0.326 & 0.429 & 0.500 \end{bmatrix} \quad \hat{P}_{\text{POMM}}^{K=5} = \begin{bmatrix} 0.500 & 0.575 & 0.694 & 0.723 & 0.775 \\ 0.425 & 0.500 & 0.604 & 0.655 & 0.736 \\ 0.306 & 0.396 & 0.500 & 0.555 & 0.655 \\ 0.277 & 0.345 & 0.445 & 0.500 & 0.597 \\ 0.225 & 0.264 & 0.345 & 0.403 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\text{Simple}}^{K=5} = \begin{bmatrix} 0.500 & 0.557 & 0.681 & 0.703 & 0.761 \\ 0.443 & 0.500 & 0.659 & 0.641 & 0.744 \\ 0.319 & 0.341 & 0.500 & 0.532 & 0.625 \\ 0.297 & 0.359 & 0.468 & 0.500 & 0.622 \\ 0.239 & 0.256 & 0.375 & 0.378 & 0.500 \end{bmatrix}$$

$$P_{true}^{K=9} = \begin{bmatrix} 0.500 & 0.547 & 0.626 & 0.682 & 0.699 & 0.726 & 0.766 & 0.775 & 0.778 \\ 0.453 & 0.500 & 0.546 & 0.624 & 0.679 & 0.702 & 0.729 & 0.750 & 0.765 \\ 0.374 & 0.454 & 0.500 & 0.571 & 0.633 & 0.647 & 0.705 & 0.720 & 0.738 \\ 0.318 & 0.376 & 0.429 & 0.500 & 0.551 & 0.618 & 0.660 & 0.692 & 0.708 \\ 0.301 & 0.321 & 0.367 & 0.449 & 0.500 & 0.561 & 0.630 & 0.655 & 0.710 \\ 0.274 & 0.298 & 0.353 & 0.382 & 0.439 & 0.500 & 0.557 & 0.625 & 0.676 \\ 0.234 & 0.271 & 0.295 & 0.340 & 0.370 & 0.443 & 0.500 & 0.562 & 0.636 \\ 0.225 & 0.250 & 0.280 & 0.308 & 0.345 & 0.375 & 0.438 & 0.500 & 0.560 \\ 0.222 & 0.235 & 0.262 & 0.292 & 0.290 & 0.324 & 0.364 & 0.440 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\text{POMM}}^{K=9} = \begin{bmatrix} 0.500 & 0.559 & 0.637 & 0.674 & 0.704 & 0.729 & 0.754 & 0.773 & 0.791 \\ 0.441 & 0.500 & 0.558 & 0.637 & 0.675 & 0.704 & 0.730 & 0.751 & 0.772 \\ 0.363 & 0.442 & 0.500 & 0.559 & 0.637 & 0.674 & 0.704 & 0.730 & 0.752 \\ 0.326 & 0.363 & 0.441 & 0.500 & 0.557 & 0.636 & 0.675 & 0.704 & 0.729 \\ 0.296 & 0.325 & 0.363 & 0.443 & 0.500 & 0.557 & 0.638 & 0.676 & 0.705 \\ 0.271 & 0.296 & 0.326 & 0.364 & 0.443 & 0.500 & 0.557 & 0.637 & 0.675 \\ 0.246 & 0.270 & 0.296 & 0.325 & 0.362 & 0.443 & 0.500 & 0.558 & 0.638 \\ 0.227 & 0.249 & 0.270 & 0.296 & 0.324 & 0.363 & 0.442 & 0.500 & 0.558 \\ 0.209 & 0.228 & 0.248 & 0.271 & 0.295 & 0.325 & 0.362 & 0.442 & 0.500 \end{bmatrix}$$

$$\hat{P}_{\text{Simple}}^{K=9} = \begin{bmatrix} 0.500 & 0.547 & 0.626 & 0.682 & 0.699 & 0.726 & 0.766 & 0.775 & 0.778 \\ 0.453 & 0.500 & 0.546 & 0.624 & 0.679 & 0.702 & 0.729 & 0.750 & 0.765 \\ 0.374 & 0.454 & 0.500 & 0.571 & 0.633 & 0.647 & 0.705 & 0.720 & 0.738 \\ 0.318 & 0.376 & 0.429 & 0.500 & 0.551 & 0.618 & 0.660 & 0.692 & 0.708 \\ 0.301 & 0.321 & 0.367 & 0.449 & 0.500 & 0.561 & 0.630 & 0.655 & 0.710 \\ 0.274 & 0.298 & 0.353 & 0.382 & 0.439 & 0.500 & 0.557 & 0.625 & 0.676 \\ 0.234 & 0.271 & 0.295 & 0.340 & 0.370 & 0.443 & 0.500 & 0.562 & 0.636 \\ 0.225 & 0.250 & 0.280 & 0.308 & 0.345 & 0.375 & 0.438 & 0.500 & 0.560 \\ 0.222 & 0.235 & 0.262 & 0.292 & 0.290 & 0.324 & 0.364 & 0.440 & 0.500 \\ & & & & & & & & \end{bmatrix}$$

11.1. **POMM model check.**

z summary table
True Model POMM, $N = 100$

Method	VI distance$_{\text{MAP}}$			VI distance$_{\text{VI lb}}$			WAIC		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.13	0.53	1.9	0.13	0.42	1.73	-13361.79 30.98	-13510.00 31.45	-13645.28 31.03
Simple model	0.13	0.31	2.0	0.13	0.48	1.71	-13409.44 30.74	-13496.19 31.71	-13659.10 30.71

Table 16. POMM Hyperparameters summary table
True Model POMM, $N = 100$

Method	$\hat{\theta}$			95% CI interval			True value		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
σ	0.1487	0.084	0.0196	[0.0033 0.7575]	[4e-04 0.5602]	[0.0012 0.0625]	0.01	0.01	0.01
α	0.4237	0.5265	0.491	[0.1324 0.6215]	[0.439 0.8733]	[0.4258 0.5934]	0.5	0.5	0.5

z diagnostic table
True Model POMM, $N = 100$

Fitted Model	ESS			ACF$_{30}$			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	3418.94	10354.90	4800.38	0.11	0.02	0.09	0.01	0.33	4.50	1.29	1.08	1.00
Simple	1778.56	10191.58	1869.04	0.00	0.01	0.36	0.01	0.50	2.64	1.02	1.17	1.08

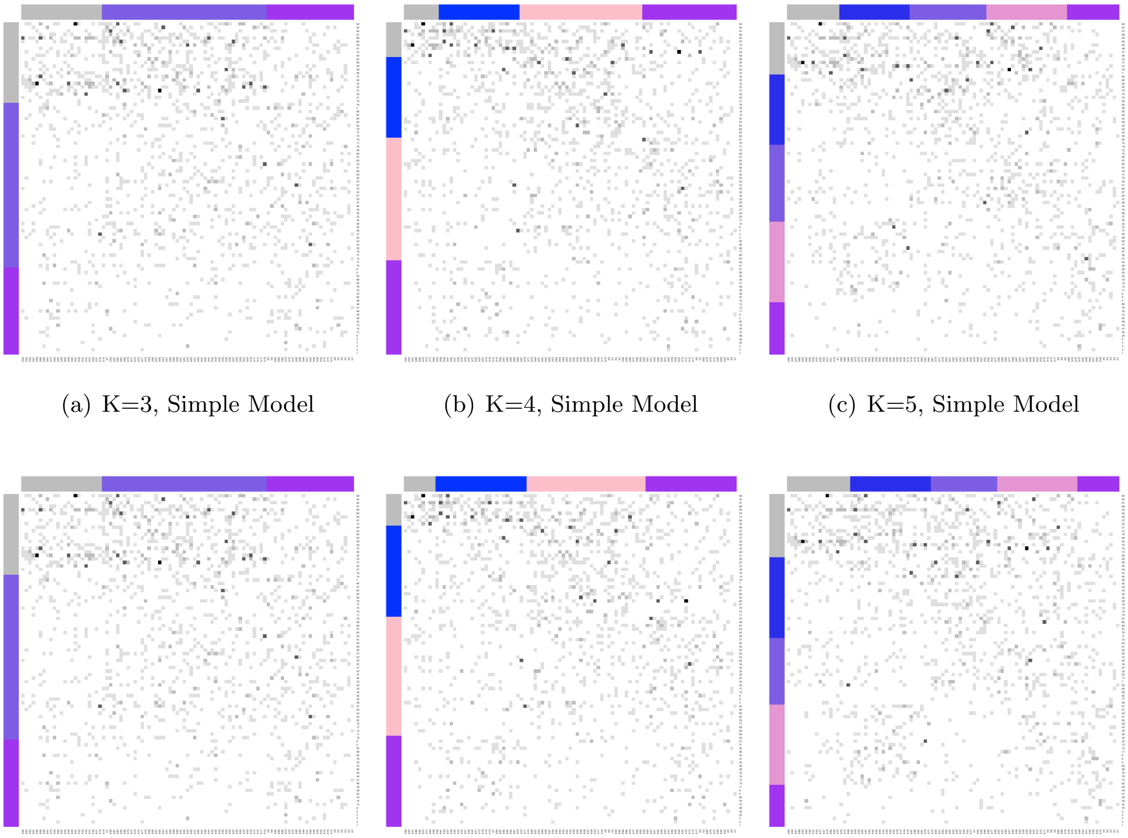
P diagnostic table
True Model POMM, $N = 100$

Fitted Model	ESS			ACF$_{30}$			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	626.67	672.7	308.22	0.29	0.22	0.34	20.89	29.75	29.55	1.36	1.34	1.11
Simple	2798.00	1659.7	121.94	0.00	0.02	0.53	36.70	30.99	29.78	1.00	1.07	1.03

POMM hyperparameters diagnostic table
True Model POMM, $N = 100$

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
σ	981	480	13	0.4	0.58	0.94	15.78	11.38	1.72	3.55	2.15	1.96
α	59	85	29	0.8	0.71	0.88	16.32	15.41	6.33	1.26	1.35	1.25

12. Application to Tennis Data



(a) K=3, Simple Model

(b) K=4, Simple Model

(c) K=5, Simple Model

(d) K=3, POMM Model

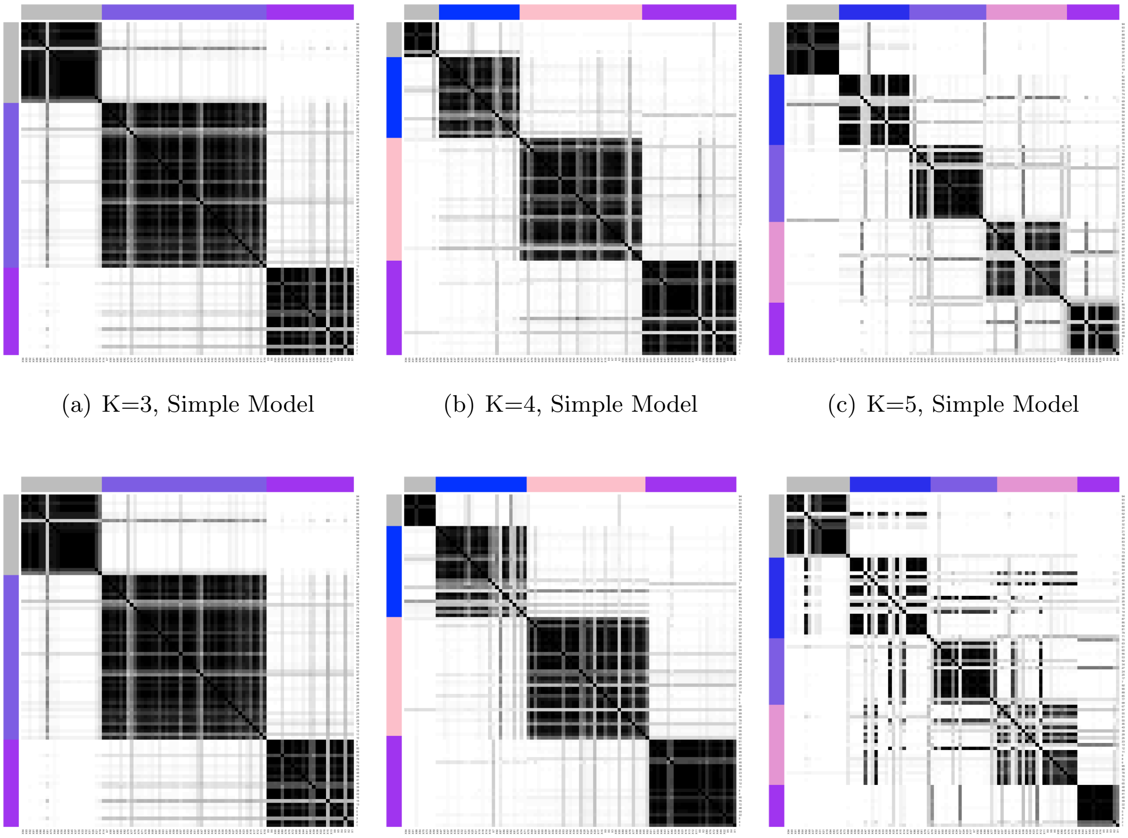
(e) K=4, POMM Model

(f) K=5 POMM Model

Figure 8. Adjacency Matrices simulated via the POMM Model

$$\hat{P}^{POMM} = \begin{bmatrix} 0.500 & 0.779 & 0.648 \\ 0.221 & 0.500 & 0.765 \\ 0.352 & 0.235 & 0.500 \end{bmatrix} \qquad \hat{P}^{Simple} = \begin{bmatrix} 0.500 & 0.779 & 0.648 \\ 0.221 & 0.500 & 0.766 \\ 0.352 & 0.234 & 0.500 \end{bmatrix}$$

$$\hat{P}^{POMM} = \begin{bmatrix} 0.500 & 0.786 & 0.742 & 0.764 \\ 0.214 & 0.500 & 0.775 & 0.532 \\ 0.258 & 0.225 & 0.500 & 0.776 \\ 0.236 & 0.468 & 0.224 & 0.500 \end{bmatrix} \qquad \hat{P}^{Simple} = \begin{bmatrix} 0.500 & 0.787 & 0.742 & 0.763 \\ 0.213 & 0.500 & 0.775 & 0.532 \\ 0.258 & 0.225 & 0.500 & 0.775 \\ 0.237 & 0.468 & 0.225 & 0.500 \end{bmatrix}$$



(a) K=3, Simple Model

(b) K=4, Simple Model

(c) K=5, Simple Model

(d) K=3, POMM Model

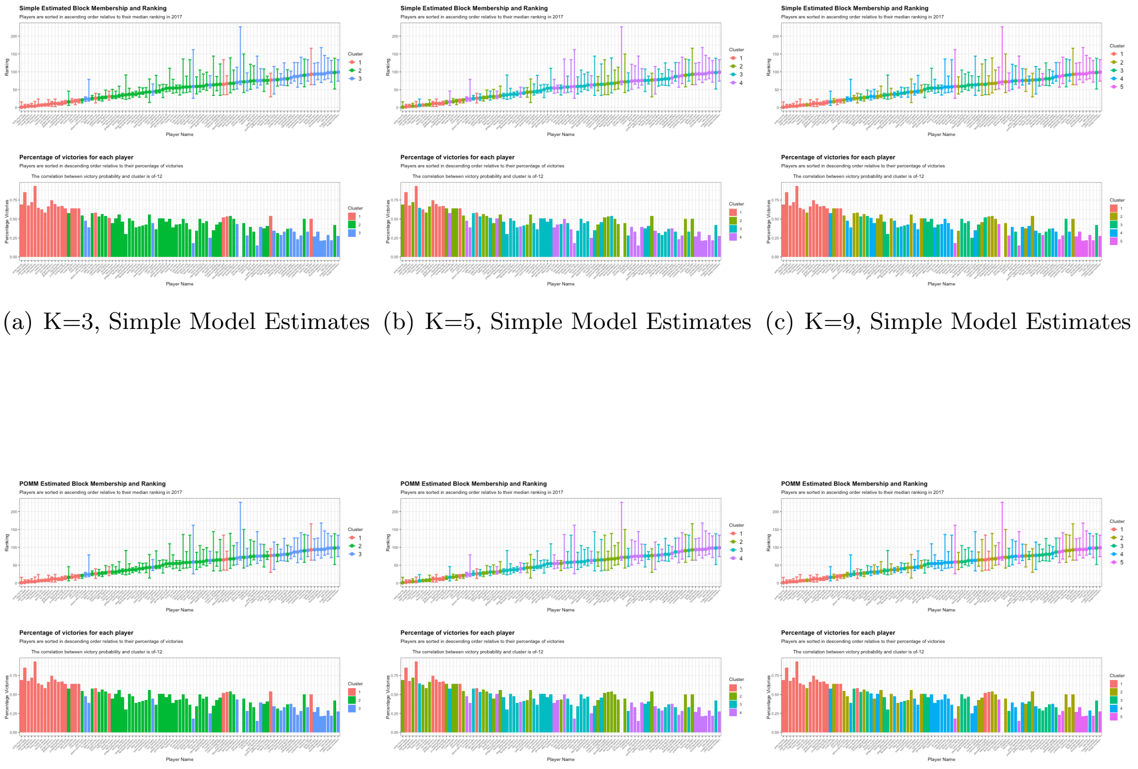
(e) K=4, POMM Model

(f) K=5 POMM Model

FIGURE 9. Adjacency Matrices simulated via the POMM Model

$$\hat{P}^{POMM} = \begin{bmatrix} 0.500 & 0.778 & 0.745 & 0.785 & 0.716 \\ 0.222 & 0.500 & 0.792 & 0.512 & 0.768 \\ 0.255 & 0.208 & 0.500 & 0.747 & 0.652 \\ 0.215 & 0.488 & 0.253 & 0.500 & 0.776 \\ 0.284 & 0.232 & 0.348 & 0.224 & 0.500 \end{bmatrix} \quad \hat{P}^{Simple} = \begin{bmatrix} 0.500 & 0.779 & 0.745 & 0.785 & 0.714 \\ 0.221 & 0.500 & 0.792 & 0.512 & 0.768 \\ 0.255 & 0.208 & 0.500 & 0.748 & 0.652 \\ 0.215 & 0.488 & 0.252 & 0.500 & 0.777 \\ 0.286 & 0.232 & 0.348 & 0.223 & 0.500 \end{bmatrix}$$

12.1. POMM model check.



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

z summary table
True Model POMM, $N = 100$

Method	WAIC		
	(a)	(b)	(c)
POMM model	-5410.20 24.85	-5536.49 24.78	-5637.33 25.51
Simple model	-5411.18 24.87	-5535.89 24.76	-5637.28 25.48

POMM Hyperparameters summary table

True Model POMM, $N = 100$

Method	$\hat{\theta}$			95% CI interval		
	(a)	(b)	(c)	(a)	(b)	(c)
σ	0.54	0.58	0.58	[0.2 0.9]	[0.25 0.9]	[0.001, 0.0608]
α	0.45	0.50	0.42	[0.11 0.84]	[0.15 0.88]	[0.1 0.82]

z diagnostic table

True Model POMM, $N = 100$

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	11458.19	8062.46	11734.00	0.14	0.13	0.06	1.76	1.15	0.94	1.01	1.01	1.01
Simple	12846.46	8137.83	10373.67	0.10	0.18	0.05	1.63	1.17	0.92	1.01	1.07	1.01

P diagnostic table

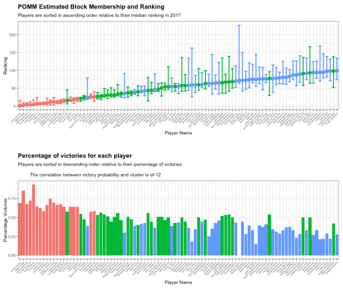
True Model POMM, $N = 100$

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	947.67	1414.33	2051.6	0.11	0.08	0.05	29.96	29.59	32.31	1	1.00	1
Simple	1001.33	1365.83	1928.5	0.10	0.08	0.05	30.17	29.72	32.32	1	1.01	1

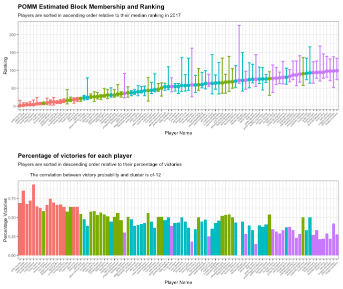
POMM hyperparameters diagnostic table

True Model POMM, $N = 100$

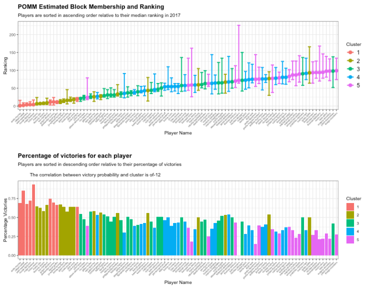
Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
σ	3504	4202	5078	-0.01	0	0.01	37.25	34.56	34.16	1	1	1
α	10	13	15	0.96	0.96	0.97	25.42	25.07	24.88	1.19	1.01	1.1



(a) K=3, POMM Model Estimates



(b) K=5, POMM Model Estimates



(c) K=9, POMM Model Estimates

12.2. **Fixing σ=0.01.**

$$\hat{P}^{POMM} = \begin{bmatrix} 0.50 & 0.65 & 0.79 \\ 0.34 & 0.50 & 0.64 \\ 0.21 & 0.35 & 0.50 \end{bmatrix}$$

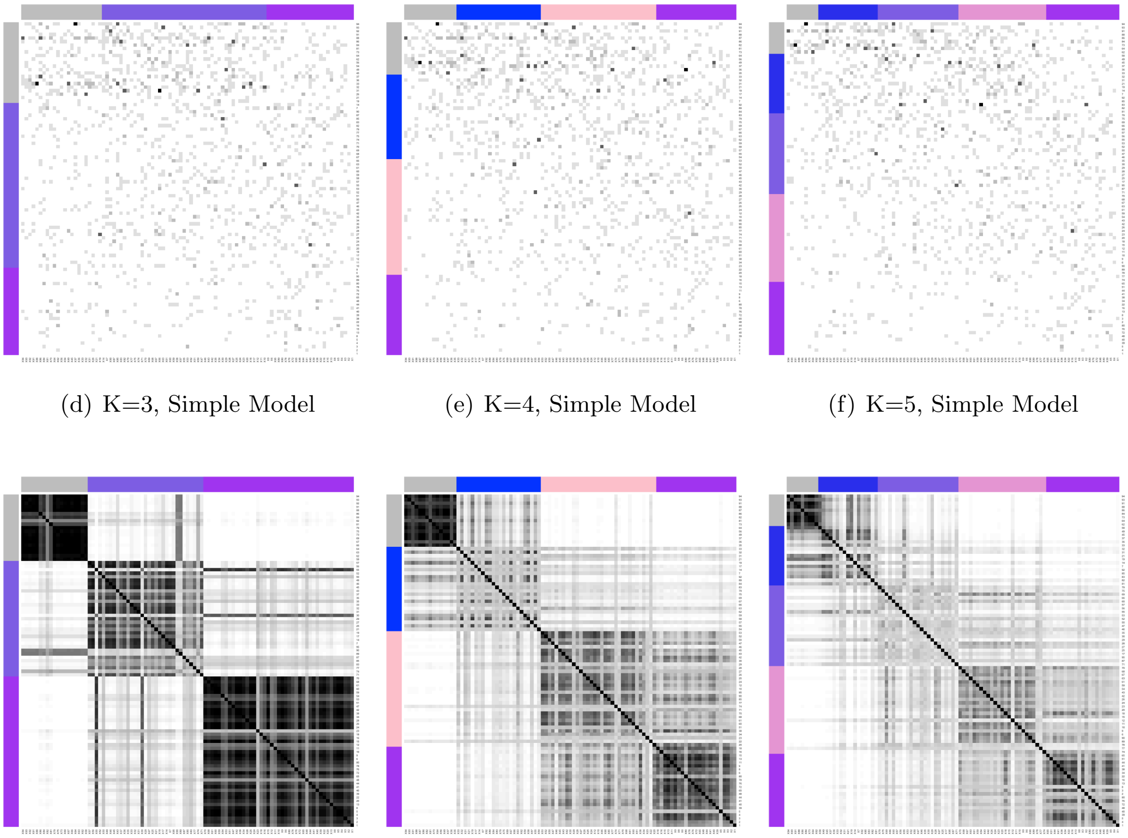
$$\hat{P}^{POMM} = \begin{bmatrix} 0.50 & 0.58 & 0.68 & 0.76 \\ 0.42 & 0.50 & 0.57 & 0.67 \\ 0.32 & 0.43 & 0.50 & 0.57 \\ 0.24 & 0.33 & 0.43 & 0.50 \end{bmatrix}$$

$$\hat{P}^{POMM} = \begin{bmatrix} 0.50 & 0.58 & 0.67 & 0.72 & 0.79 \\ 0.42 & 0.50 & 0.57 & 0.67 & 0.71 \\ 0.33 & 0.43 & 0.50 & 0.57 & 0.67 \\ 0.28 & 0.33 & 0.43 & 0.50 & 0.58 \\ 0.22 & 0.29 & 0.33 & 0.42 & 0.50 \end{bmatrix}$$

z summary table

True Model POMM, $N = 100$

Method	WAIC		
	(a)	(b)	(c)
POMM model	-5220.111 21.17	-5181.395 19.32	-5233.632 20.07



(d) K=3, Simple Model

(e) K=4, Simple Model

(f) K=5, Simple Model

(g) K=3, POMM Model

(h) K=4, POMM Model

(i) K=5 POMM Model

FIGURE 11. Adjacency Matrices simulated via the POMM Model

POMM Hyperparameters summary table

True Model POMM, $N = 100$

Method	$\hat{\theta}$			95% CI interval		
	(a)	(b)	(c)	(a)	(b)	(c)
σ	0.01	0.01	0.01	[0.01 0.01]	[0.01 0.01]	[0.01 0.01]
α	0.12	1.41	0.87	[0.1 0.15]	[0.1 2.74]	[0.1 1.8]

z diagnostic table

True Model POMM, $N = 100$

Fitted Model	ESS			ACF$_{30}$			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	29353.04	20673.88	23681.54	0	0	0.01	8.93	15.05	16.53	1	1.13	1.16

P diagnostic table

True Model POMM, $N = 100$

Fitted Model	ESS			ACF$_{30}$			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	2018.33	1900	2983.4	0.01	0.02	0.01	33.8	27.57	29.75	1	12.35	10.89

12.2.1. *Montecarlo algorithm.* First, it should be noted that within the same block we observe a substantial variability. We may have players that win against other players of the same cluster, against players of weaker clusters, or that simply do not win much. If we have players that substantially win against players of stronger clusters, it means that they are misclassified. So the fundamental variability driver is to be recognised in the blocks of the defeated players. Winning against Federer is not the same as winning against a newcomer. Those two victories should not be accounted in the same fashion.

However, one could argue that if the block exhibits a large amount of variability, this probably means that we should split the block further in two, to possibly account for different patterns of victories.

Another crucial point is data imbalance. Games are not drawn at random, which means that we will observe strongest player playing more with each other, since they are the on remaining within the tournament for longer periods, and weaker players playing less against the strong ones and more among themselves.

The issue is that by assigning one cluster to a player is equivalent to equip them with a single probability to beat all the players within a given cluster.

13. Appendix I: Estimation Details

13.1. **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:

$$r = \frac{\prod_{i<j}\binom{n_{ij}}{y_{ij}} p_{z'_i z'_j}^{y_{ij}} \cdot (1-p_{z'_i z'_j})^{n_{ij}-y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n'_k+\gamma_k)}{\Gamma(\gamma_k)\Gamma(n'_k+1)}}{\prod_{i<j}\binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1-p_{z_i z_j})^{n_{ij}-y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k+\gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{30}$$

$$= \frac{\prod_{i<j} p_{z'_i z'_j}^{y_{ij}} \cdot (1-p_{z'_i z'_j})^{n_{ij}-y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n'_k+\gamma_k)}{\Gamma(\gamma_k)\Gamma(n'_k+1)}}{\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1-p_{z_i z_j})^{n_{ij}-y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k+\gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{31}$$

Passing to the log:

$$\begin{aligned} log(r) = \log &\left(\prod_{i<j} p_{z'_i z'_j}^{y_{ij}} \cdot (1-p_{z'_i z'_j})^{n_{ij}-y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n'_k+\gamma_k)}{\Gamma(\gamma_k)\Gamma(n'_k+1)}\right) \\ &- \log\left(\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1-p_{z_i z_j})^{n_{ij}-y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k+\gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}\right) \\ = &\sum_{i<j}\left(y_{ij} \cdot \log p_{z'_i z'_j} + (n_{ij}-y_{ij}) \cdot \log\left(1-p_{z'_i z'_j}\right)\right) \\ &+ \sum_{k=1}^{K}\left(\log\left(\Gamma(n'_k+\gamma_k)\right) - \log\left(\Gamma(\gamma_k)\right) - \log\left(\Gamma\left(n'_k+1\right)\right)\right) \\ &- \sum_{i<j}\left(y_{ij} \cdot \log p_{z_i z_j} + (n_{ij}-y_{ij}) \cdot \log\left(1-p_{z_i z_j}\right)\right) \\ &- \sum_{k=1}^{K}\left(\log\left(\Gamma(n_k+\gamma_k)\right) - \log\left(\Gamma(\gamma_k)\right) - \log\left(\Gamma\left(n_k+1\right)\right)\right) \end{aligned} \tag{32}$$

13.2. **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:

$$r = \frac{\prod_{i<j}\binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{\prime y_{ij}} \cdot (1-p'_{z_i z_j})^{n_{ij}-y_{ij}} \cdot \prod_{k=1}^{K}\left(\frac{1}{y'^{(k+1)}-y'^{(k)}}\right)^{|L_{\prime(k)}|}}{\prod_{i<j}\binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1-p_{z_i z_j})^{n_{ij}-y_{ij}} \cdot \prod_{k=1}^{K}\left(\frac{1}{y^{(k+1)}-y^{(k)}}\right)^{|L_{(k)}|}} \tag{33}$$

(34)

Passing to the log:

Algorithm 5 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
```

(35)

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

(36)

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

Algorithm 6 Updating P step

```
$j \leftarrow 1$
while $j \leq N_{iter}$ do
    Sample $\alpha'$ 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 $\alpha$ to $\alpha'$
        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 $\alpha$ in $\alpha.container$
    $j \leftarrow j + 1$
end while
```

14. Appendix II: Limiting case for σ

It is interesting to see what happens to the POMM prior when $\sigma \to \infty$. The idea is that,as the variance of the normals increase, the POMM prior should collapse on the Simple prior.

To prove this argument, first we check empirically and visually if the POMM prior distribution really converges to the Simple model distribution. Empirically we can run the Kolmogorov-Smirnov test to assess if there is a statistically significant difference between points $p_{ij} \sim POMM(\beta : \alpha; \sigma = \{0.01, 0.10, 0.50\}$ and the Simple prior $p_{ij} \sim Beta(1, 1)$. Then, we check visually the two distributions. Finally, we try to check analytically a convergence.

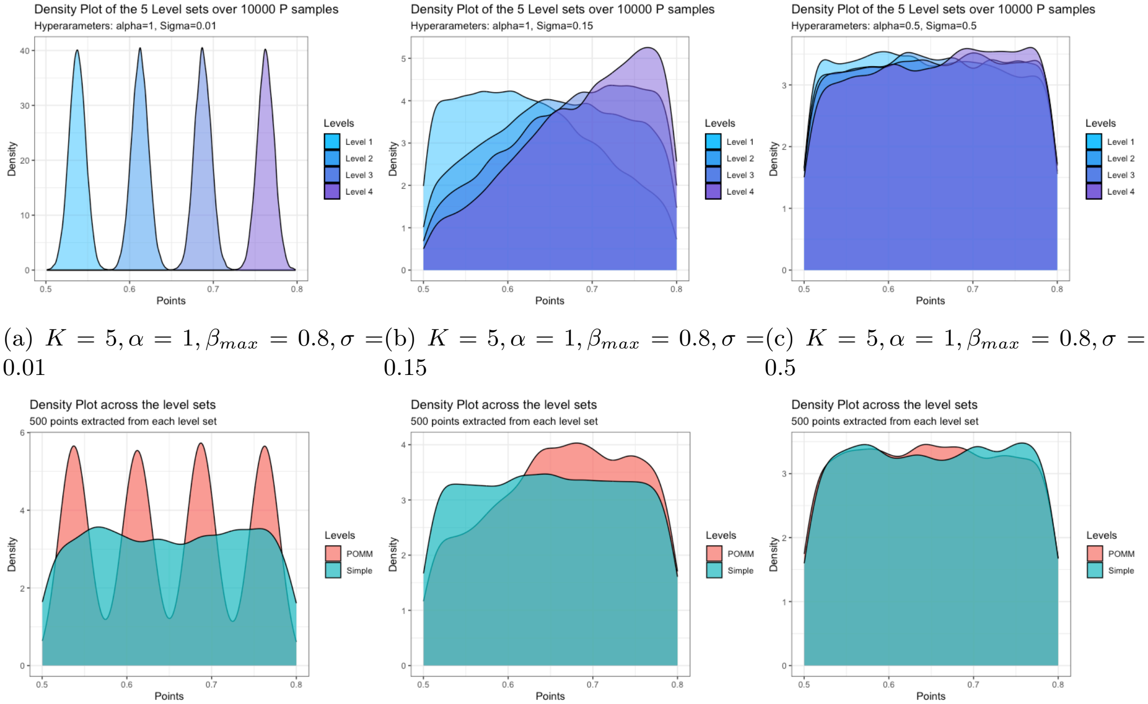
Table 17. Kolmogorov-Smirnov test
Data are generated via $p_{ij} \sim POMM(\alpha = 1, \beta_{max} = .8, \sigma)$

Method	p-value		
	$\sigma = 0.01$	$\sigma = 0.15$	$\sigma = 0.50$
POMM model	4.643e-09	3.998e-13	0.4163

In (17) we see that with values of σ equal to 0.5, we are unable to statistically distinguish points sampled from the Simple model and points sampled from the POMM model

$$\lim_{\sigma \to \infty} \prod_{k=1}^{K} \frac{1}{\sigma} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{p_{ij}-\mu^{(k)}}{\sigma}\right)^2}}{\int_{-\infty}^{\beta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu^{(k)}}{\sigma}\right)^2} dt - \int_{-\infty}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu^{(k)}}{\sigma}\right)^2} dt} = \tag{37}$$

$$\lim_{\sigma \to \infty} \prod_{k=1}^{K} \frac{1}{\sigma} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{p_{ij}-\frac{y^{(k)}+y^{(k+1)}}{2}}{\sigma}\right)^2}}{\int_{-\infty}^{\beta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\frac{y^{(k)}+y^{(k+1)}}{2}}{\sigma}\right)^2} dt - \int_{-\infty}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\frac{y^{(k)}+y^{(k+1)}}{2}}{\sigma}\right)^2} dt} \tag{38}$$



(a) $K = 5, \alpha = 1, \beta_{max} = 0.8, \sigma = 0.01$

(b) $K = 5, \alpha = 1, \beta_{max} = 0.8, \sigma = 0.15$

(c) $K = 5, \alpha = 1, \beta_{max} = 0.8, \sigma = 0.5$

(d) $K = 5, \alpha = 1, \beta_{max} = 0.8, \sigma = 0.01$

(e) $K = 5, \alpha = 1, \beta_{max} = 0.8, \sigma = 0.15$

(f) $K = 5, \alpha = 1, \beta_{max} = 0.8, \sigma = 0.5$

FIGURE 12. As σ increases, the distribution of the level sets of the POMM prior becomes indistinguishable from the distribution of the Simple model prior

If we substitute in the expression for $y^{(k)} = \left(\frac{(\beta_{\max}-0.5)^{(1/\alpha)}}{K} \times k\right)^{\alpha} + 0.5$, we can simplify the following expression:

$$\frac{y^{(k)} + y^{(k+1)}}{2} = \frac{1}{2}\left[\left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times k\right)^{\alpha} + 0.5 + \left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times (k+1)\right)^{\alpha} + 0.5\right] \tag{39}$$

$$= \frac{1}{2}\left[\left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times k\right)^{\alpha} + \left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times (k+1)\right)^{\alpha} + 1\right] \tag{40}$$

$$= \left[\frac{1}{2}\left(\frac{(\beta_{\max} - 0.5)}{K^{\alpha}} \times k^{\alpha}\right) + \left(\frac{(\beta_{\max} - 0.5)}{2K^{\alpha}} \times (k+1)^{\alpha}\right) + \frac{1}{2}\right] \tag{41}$$

$$= \frac{(\beta_{\max} - 0.5)}{2K^{\alpha}} \times (k^{\alpha} + (k+1)^{\alpha}) + \frac{1}{2} \tag{42}$$

Substituting back in the simplified expression:

(43)

$$\lim_{\sigma\to\infty}\prod_{k=1}^{K}\frac{1}{\sigma}\frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{p_{ij}-\left[\frac{(\beta_{\max}-0.5)}{2K^{\alpha}}\times\big(k^{\alpha}+(k+1)^{\alpha}\big)+\frac{1}{2}\right]}{\sigma}\right)^{2}}}{\int_{-\infty}^{\beta}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{t-\left[\frac{(\beta_{\max}-0.5)}{2K^{\alpha}}\times(k^{\alpha}+(k+1)^{\alpha})+\frac{1}{2}\right]}{\sigma}\right)^{2}}dt-\int_{-\infty}^{0.5}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{t-\left[\frac{(\beta_{\max}-0.5)}{2K^{\alpha}}\times(k^{\alpha}+(k+1)^{\alpha})+\frac{1}{2}\right]}{\sigma}\right)^{2}}dt}$$

(44)

$$\lim_{\sigma\to\infty}\prod_{k=1}^{K}\mathrm{TruncatedNormal}(\left[\frac{(\beta_{\max}-0.5)}{2K^{\alpha}}\times(k^{\alpha}+(k+1)^{\alpha})+\frac{1}{2}\right],\sigma^{2};0.5,\beta_{\max})$$