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. 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

$$Z = \{Z(s) : s \in D\} \tag{1}$$

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

$$D = \{(u, v) : u = 1, \ldots, K; v = 1, \ldots, K\}. \tag{2}$$

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:

$$P^{WST} = \begin{pmatrix} p_{1,1} & p_{1,2} & ... & p_{1,K} \\ p_{2,1} & p_{2,2} & ... & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & ... & p_{K,K} \end{pmatrix} = \begin{pmatrix} 1/2 \leq & p_{1,2} & ... & p_{1,K} \\ p_{2,1} \leq & 1/2 \leq & ... & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & ... & 1/2 \end{pmatrix} \tag{3}$$

Instead, under SST, we would observe:

$$P^{SST} = \begin{pmatrix} p_{1,1} & p_{1,2} & ... & p_{1,K} \\ p_{2,1} & p_{2,2} & ... & p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} & p_{K,2} & ... & p_{K,K} \end{pmatrix} = \begin{pmatrix} 1/2 \leq & p_{1,2} \leq & ... & p_{1,K} \\ \vee| & \vee| & & \vee| \\ p_{2,1} \leq & 1/2 & ... & \leq p_{2,K} \\ \vdots & \vdots & \vdots & \vdots \\ p_{K,1} \leq & p_{K,2} \leq & ... & \leq 1/2 \end{pmatrix} \tag{4}$$

With regard of LST, it is a generalisation of the two axioms and therefore it includes both aforementioned cases (3) and (4), depending how one specifies F and μ. Given the LST definition, we can calculate p_{ij} as follows:

$$p_{ij} = F(\mu(i) - \mu(j)) \tag{5}$$

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

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

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.

3. 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:

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

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 $L^{(1)}, \ldots, L^{(K)}$.
- 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:

$$y^{(k)} = \left(\frac{(\beta_{\max} - 0.5)^{(1/\alpha)}}{K} \times k \right)^{\alpha} + 0.5 \quad \text{for } k = 0, \ldots, K \tag{6}$$

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.

3.1. **Some considerations.** This model is very flexible, since for a given number of clusters K, it can accommodate for different means across the level sets by changing the slope of α. For an $\alpha \in [0, 1]$, we will have the clusters concentrated towards $\beta_{\max}$, meaning that the probabilities of victory in the tournament are clear-cut, symptom of an higher predictability overall. Instead, if $\alpha \in [1, 3]$, this measn that the probabilities are more concentrated towards $1/2$, meaning the chances of victory are not so much clear.

However, the issue is that the number of clusters is not fixed. We would like the model to select by itself the number of clusters and ideally also to place the mean of the distribution wherever it prefers.

One question may be: what is the difference between a model with a very high α, meaning with the probabilities concentrated towards $1/2$ but with three clusters, and a model with just one cluster, and all the players having the same mean?

Maybe the parameter α is redundant and we should find a clever way so that, when we add or remove a cluster, we assign a mean of victory to that given level set.

However, the means of the level sets $\mu_1, ..., \mu_K$ should be ordered as well and we should put a constraint on them.

Another point that I have is the following:

4. 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:

For each cluster c :

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

For each match between clusters i and j :

$$p_{i,j} = \frac{1}{1 + \exp(-\beta \cdot (s_i - s_j))}$$

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

5. additional consideration

The LST and the SST and the WST models can all of them being expressed as part of the same class of models.

6. Modelling ordered blocks

Let $Y_{ij} \in 0, 1, \ldots, n_{ij}$ denote the random outcome of n_{ij} comparisons between items i and j, where $Y_{i,j} = y_{ij}$ if item i is chosen y_{ij} times over item j.

We introduce an indicator variable z_i such that $z_i = k$ if item i belongs to block k, where k ranges over the set of all blocks.

Let $p_{ij}^{(k,k')} = \mathbb{P}\left(Y_{ij} = y_{ij}, |, z_i = k, z_j = k'\right)$ be the probability that item i, belonging to block k, is chosen y_{ij} times out of n_{ij} over item j, belonging to block k'.

The collection of labeled choice probabilities associated with J is then $\mathcal{P}_J = \{p_{ij}^{(k,k')} : (i, j) \in J \times J, , k, k'$ are block indices$\}$.

We define $i \succeq j$ when $p_{ij} \geq p_{ji}$. We say $i \succ j$ when $p_{ij} > p_{ji}$, and $i \sim j$ when $p_{ij} = p_{ji}$. Given our assumption that $p_{ij} + p_{ji} = 1$, it follows that $i \succeq j$ if and only if $p_{ij} \geq \frac{1}{2}$. Additionally, this implies that $p_{ii} = \frac{1}{2}$ for all i in the set J.

This definition ensures that preferences over J are complete but not necessarily transitive. To achieve transitivity, specific constraints on choice probabilities can be imposed. Traditionally, these constraints are referred to as stochastic transitivity axioms or relations (Fishburn, 1973), although, strictly speaking, transitivity axioms simply specify different statistical models (Cavagnaro & Davis-Stober, 2014).

Here, we study three such axioms denoted as $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}_3$. These axioms progressively impose more stringent constraints on triplets of choice probabilities p_{ij}, p_{jk}, and p_{ik}. To facilitate this discussion, let's introduce a few definitions:

A composition rule is a function H that maps $(0, 1) \times (0, 1)$ to $[0, 1]$. A composition rule is said to be strictly monotone if $1 > x > y > 0$, then $H(x, z) \geq H(y, z)$ for all $z \in (0, 1)$, and if $H(x, z) \in (0, 1)$, then $H(x, z) > H(y, z) \iff x > y$. Furthermore, a composition rule is said to be symmetric if $H(x, y) = H(y, x)$ for all $x, y \in (0, 1)$.

Stochastic versions of transitivity include:

7. 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`

$$\theta_i' \sim q_i\left(\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right) \tag{7}$$

`2. Take`

$$\theta_i^{(t+1)} = \begin{cases} \theta_i^{(t)} & \texttt{with probability } 1 - r_i, \\ \theta_i' & \texttt{with probability } 1 - r_i, \end{cases} \tag{8}$$

`where`

$$r_i = 1 \wedge \left\{ \frac{\left(p(\theta_i'|\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right)}{\left(p(\theta_i^{(t)}|\theta_i^{(t)}|\theta_1^{(t+1)}, \ldots, \theta_i^{(t)}, \theta_{i+1}^{(t)}, \ldots, \theta_p^{(t)}\right)} \right\} \tag{9}$$

end for

7.0.1. *Adaptive algorithm for* $\theta = \{P, \alpha, S\}$. We specify the proposal distributions in (7) 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 (9). 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 (9).

7.0.2. *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 $k = 1, \ldots, K$ 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:

$$d(k, k') = |k - k'| \tag{10}$$

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':

$$k' \sim \text{Multinomial}(1, K, p(|k' - k|)) \tag{11}$$

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 (9).

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 three main measures:

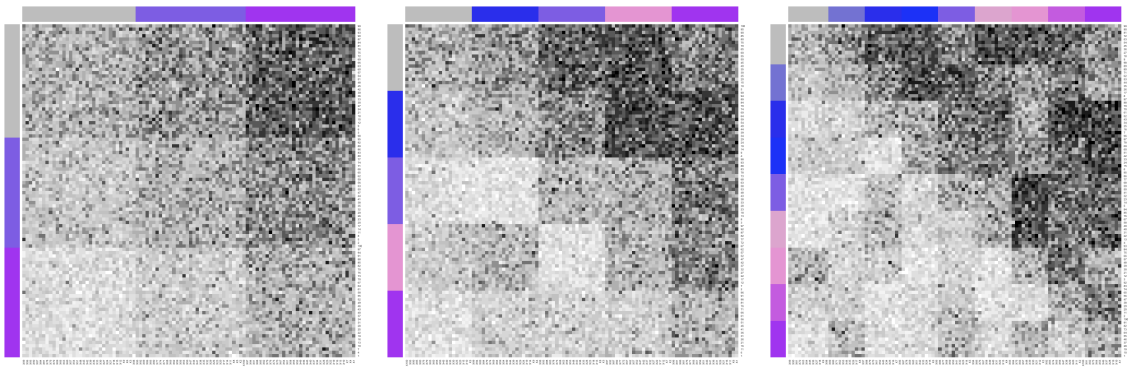
- Variation of Inforamation (VI): to fully utilise this posterior and engage in inference directly within the partition space, we adopt the decision-theoretic approach introduced by Wade and Ghahramani (2018) for block modeling. This involves summarizing posterior distributions using the variation of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

$$\begin{aligned} p(z_{N+1} = k|\mathbf{Y}, y_{N+1}, \hat{z}) &\propto p(y_{N+1}|\mathbf{Y}, \hat{z}, z_{N+1} = k) \cdot p(z_{N+1} = k|\hat{z}) \\ &= p(y_{N+1}|\hat{z}, z_{N+1} = k) \cdot p(z_{N+1} = k|\hat{z}) \end{aligned} \tag{12}$$

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.

9. Simulation Study from the Simple 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 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 (1), 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.

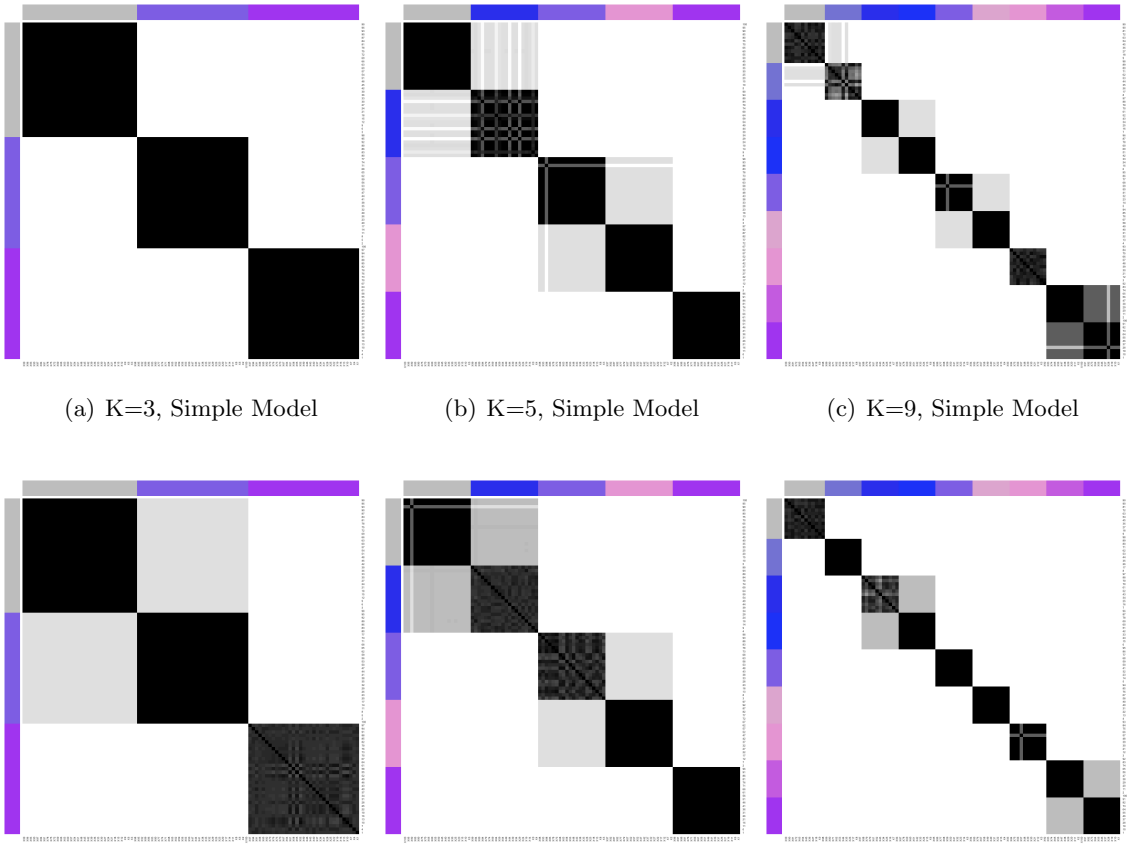


(a) K=3, Simple Model (b) K=5, Simple Model (c) K=9, Simple Model

Figure 1. Adjacency Matrices simulated via the Simple Model

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

In figure (2) 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 2. Co-Clustering Matrices obtained via the Simple Model(above) and the POMM model (below).

P summary table
True Model Simple, $N = 100$

Fitted Model	$\overline{MAE}$			% within-95% CI interval			$\overline{\text{CI interval length}}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.0211	0.0735	0.0709	100%	50%	69.4%	0.1017	0.0955	0.1788
Simple model	0.0036	0.0606	0.0826	100%	60%	55%	0.0108	0.1077	0.1798

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

Method	VI distance$_{\text{MAP}}$			VI distance$_{\text{VI lb}}$			WAIC		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0	0	0.33	0	0.4	0.44	$-6418.29_{17.01}$	$-6196.77_{18.74}$	$6320.08_{18.58}$
Simple model	0	0.5	0.58	0	0	0.22	$-6506.41_{17.60}$	$-6248.25_{18.54}$	$-6285.03_{18.61}$

POMM hyperparameters summary table
True Model Simple, $N = 100$

Fitted Model	$\hat{\theta}$			95% CI interval		
	(a)	(b)	(c)	(a)	(b)	(c)
S	0.21	0.25	0.31	[0.010.79]	[0.040.78]	[0.090.81]
α	0.58	0.41	0.38	$[0.30, 0.90]$	$[0.11, 0.79]$	$[0.10, 0.80]$

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

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	232	126	924	0.015	0.015	0.203	0.0066	0.0083	1.1610	319.891	92.569	50.05
Simple	0	85	26	-0.012	0.004	0.040	0.0058	0.0092	0.0104	1.001	163.099	17.72

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

Fitted Model	ESS			ACF_{30}			% accepted			Gelma	
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(
POMM	1710	1413	1301	0.0513	0.0657	0.0976	16.1023	7.2795	8.221472	30.36944	34.3
Simple	2100	1767	1272	-0.0003	NaN	0.0853	1.0003	9.8198	8.3113	29.97556	33.

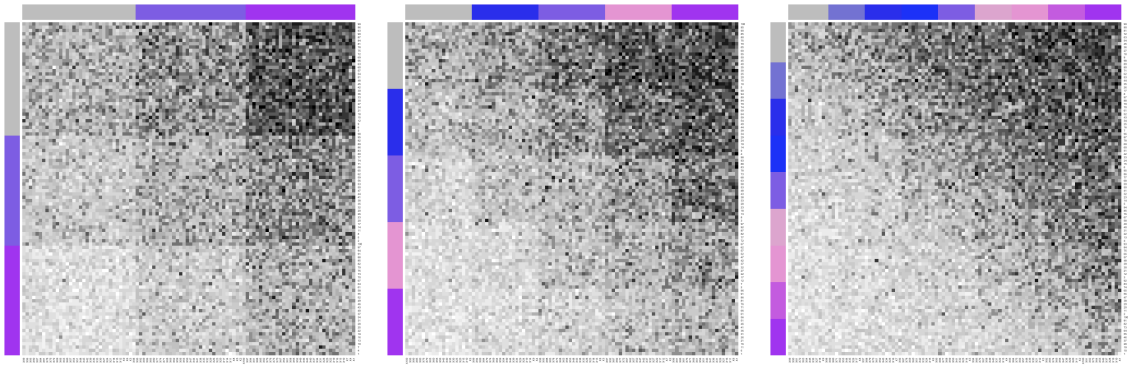
POMM hyperparameters diagnostic table
True Model Simple, $N = 100$

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
S	716	771	0	0.511	0.343	0	1.698	1.446	0	32.50417	32.0075	
α	13	13	13	0.93	0.93	0.93	1.584	1.584	1.584	24.91917	24.91917	24.91917

9.0.1. *Simple Model check.*

10. 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 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

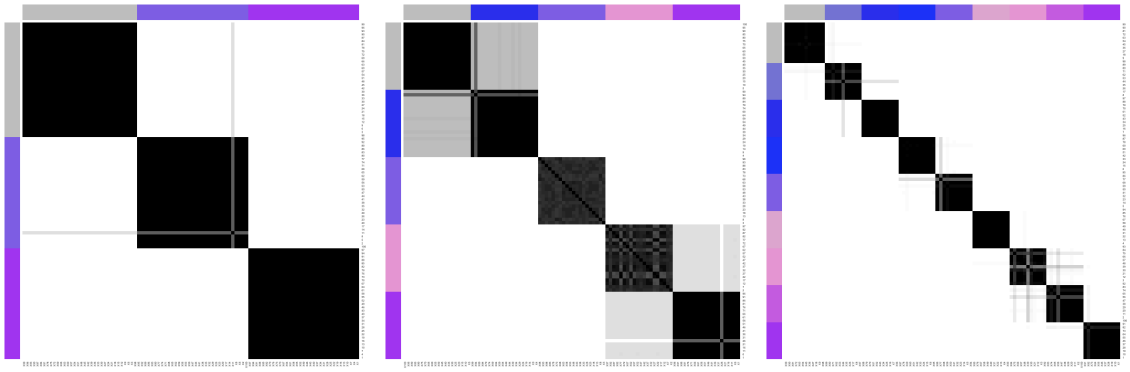
Figure 3. Adjacency Matrices simulated via the POMM Model

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

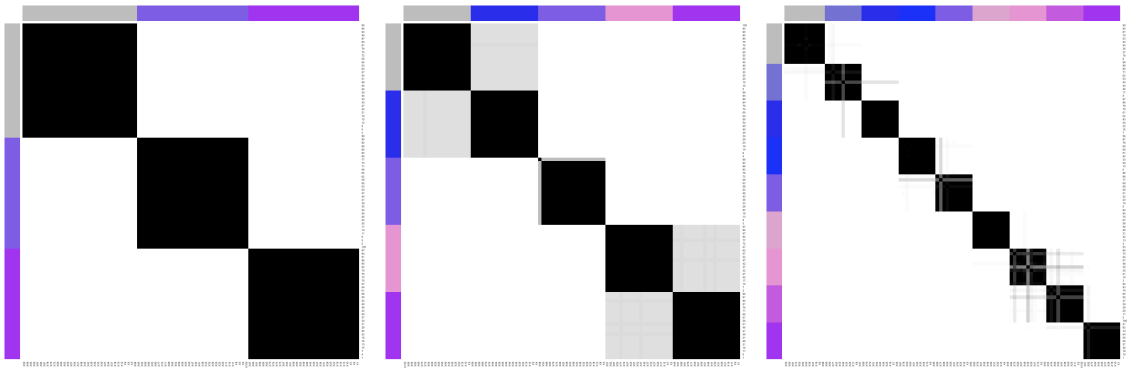
P summary table
True Model POMM, $K = 3$, $N = 100$

Fitted Model	$\overline{MAE}$			% within-95%-CI interval			$\overline{\text{CI interval length}}$		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM model	0.003	0.041	0.010	100%	100%	94.4%	0.010	0.179	0.066
Simple model	0.060	0.027	0.014	100%	100%	100%	0.134	0.13	0.087

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

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	0	0	0	0.0	0.38	-6553.31 17.83	-6551.36 17.68	6652.43 17.78
Simple model	0	0	0	0	0.4	0.0	-6415.70 17.60	-6536.58 17.84	6667.41 17.96

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)
S	0.0178	0.312	0.0221	[0.0039, 0.0425]	[0.0015, 0.8294]	[0.001, 0.0608]	0.01	0.01	0.01
α	0.4214	0.5345	0.4833	[0.23, 0.58]	[0.21, 0.89]	[0.43, 0.56]	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	0	32	336	0.001	0.081	0.473	0.008	0.009	7.280	1.000	101.742	7.268
Simple	0	122	670	0.109	0.030	0.007	0.005	0.008	0.079	470.471	90.074	1.969

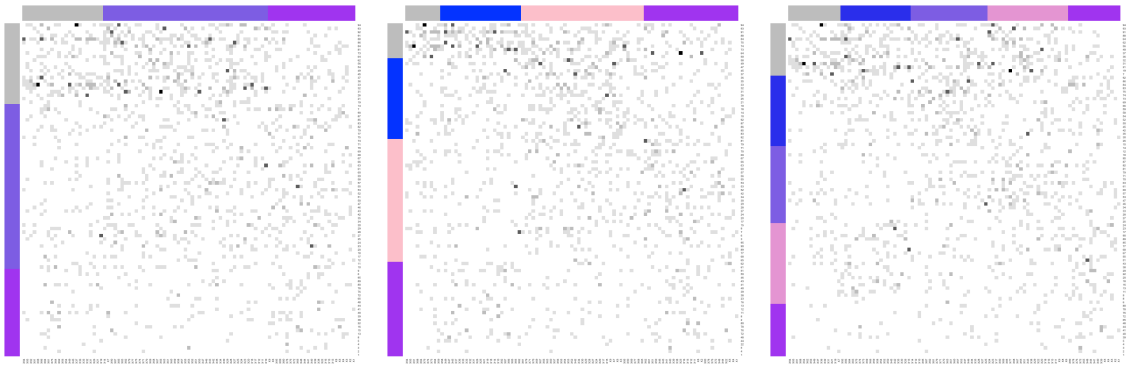
P diagnostic table
True Model POMM, $N = 100$

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubi	
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)
POMM	1948	1489.3	1015.778	0.006	0.0256	0.076	29.253	32.468	33.203	1.001	20.680
Simple	1807	1362.3	1439.056	0.072	0.0552	0.0105	24.939	33.82367	32.44970	54.698	11.999

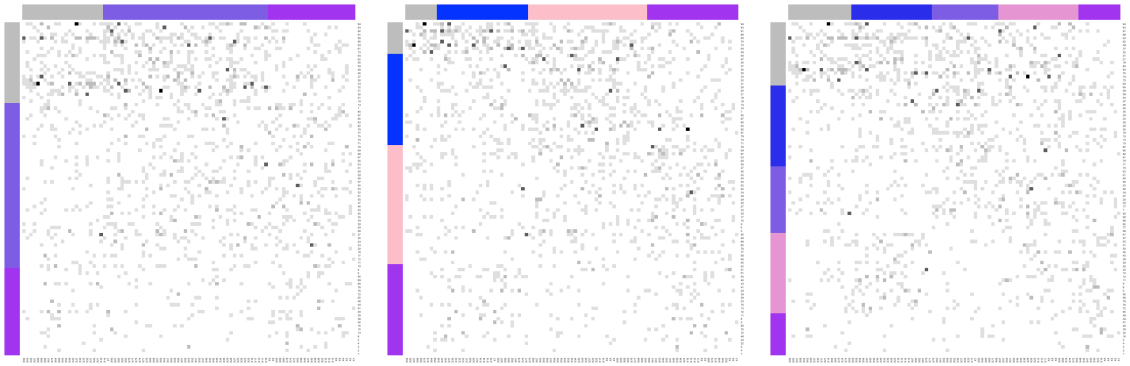
POMM hyperparameters diagnostic table
True Model POMM, $N = 100$

Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
S	319	1219	0	0.188	0.214	0	21.00	27.72	0	1.0	1.9	0
α	40	126	187	0.79	0.76	0.34	23.5	22.4	15.74	1.108	1.238	1.884

11. Application to Tennis Data



(a) K=3, Simple Model (b) K=4, Simple Model (c) K=5, Simple Model

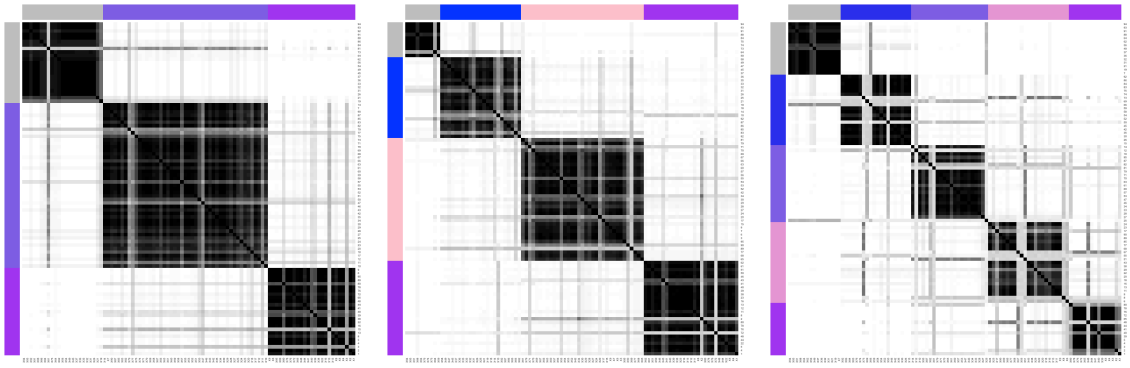


(d) K=3, POMM Model (e) K=4, POMM Model (f) K=5 POMM Model

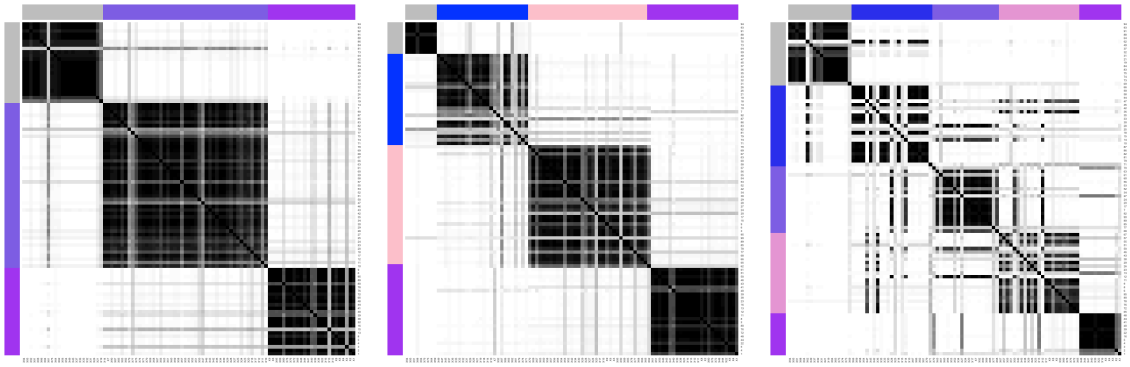
Figure 5. Adjacency Matrices simulated via the POMM Model

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

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



(a) K=3, Simple Model (b) K=4, Simple Model (c) K=5, Simple Model

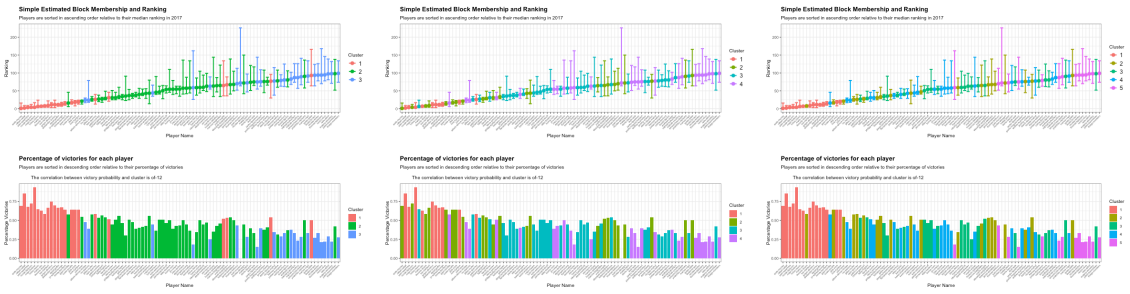


(d) K=3, POMM Model (e) K=4, POMM Model (f) K=5 POMM Model

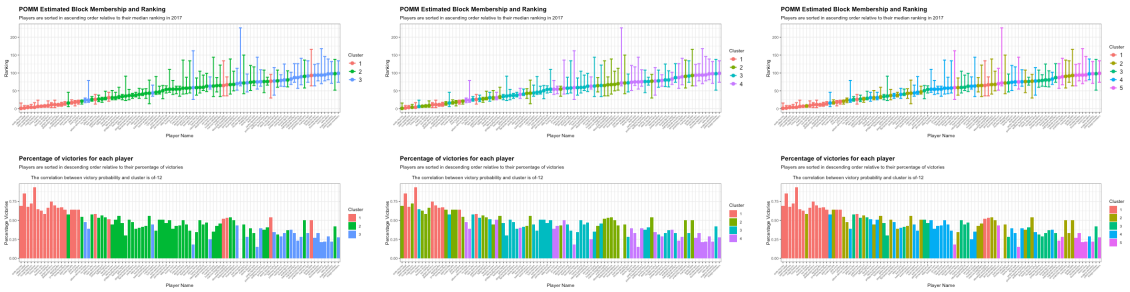
FIGURE 6. Adjacency Matrices simulated via the POMM Model

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

11.1. **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 7. 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	-5536.49	-5637.33
	24.85	24.78	25.51
Simple model	-5411.18	-5535.89	-5637.28
	24.87	24.76	25.48

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

Method	$\hat{\theta}$			95% CI interval		
	(a)	(b)	(c)	(a)	(b)	(c)
S	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, $N = 100$

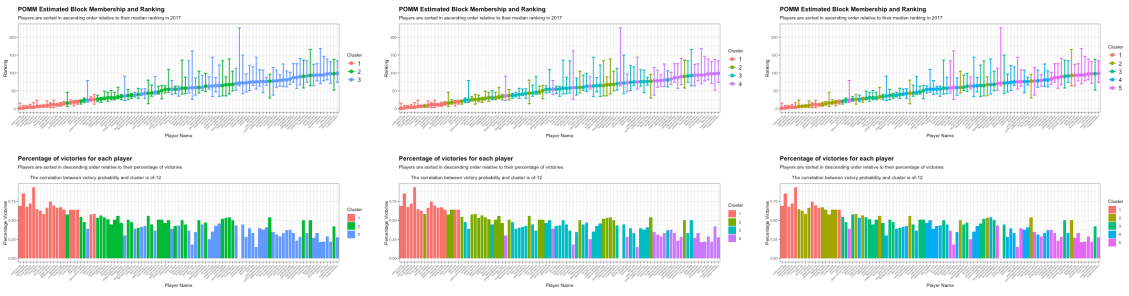
Fitted Model	ESS			ACF_{30}			% accepted			Gelman-Rubin		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
POMM	11458.19	8062.46	11734.00	0.14	0.13	0.06	1.76	1.15	0.94	1.01	1.01	1.01
Simple	12846.46	8137.83	10373.67	0.10	0.18	0.05	1.63	1.17	0.92	1.01	1.07	1.01

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)
S	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

11.2. **Fixing S=0.01.**

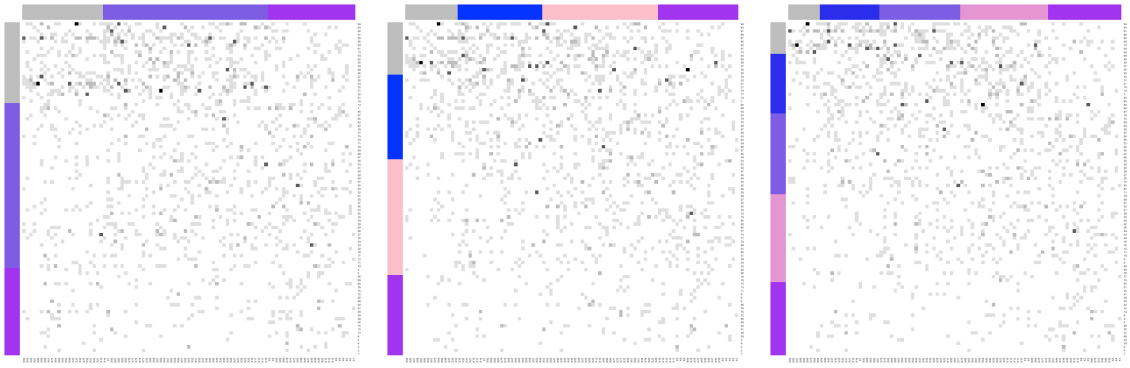
$$\hat{P}^{POMM} = \begin{bmatrix} 0.50 & 0.65 & 0.79 \\ 0.34 & 0.50 & 0.64 \\ 0.21 & 0.35 & 0.50 \end{bmatrix}$$

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

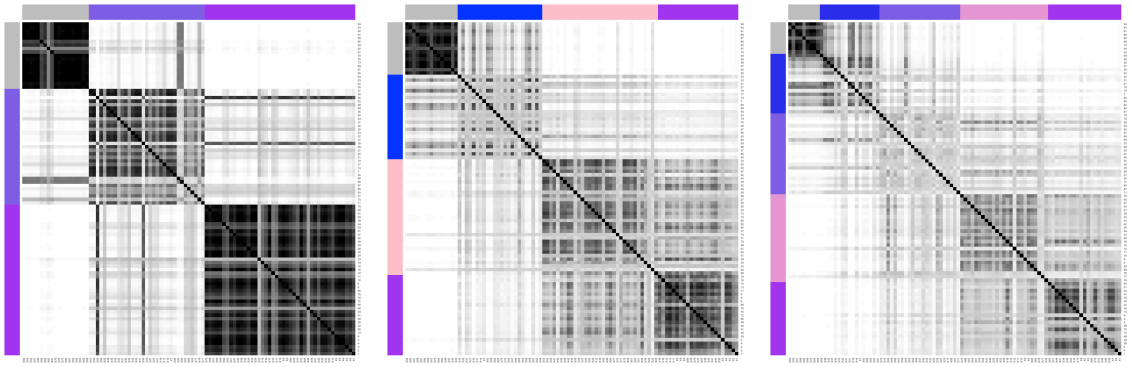
$$\hat{P}^{POMM} = \begin{bmatrix} 0.50 & 0.58 & 0.67 & 0.72 & 0.79 \\ 0.42 & 0.50 & 0.57 & 0.67 & 0.71 \\ 0.33 & 0.43 & 0.50 & 0.57 & 0.67 \\ 0.28 & 0.33 & 0.43 & 0.50 & 0.58 \\ 0.22 & 0.29 & 0.33 & 0.42 & 0.50 \end{bmatrix}$$

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 8. 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)
S	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

11.2.1. *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.

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:

$$r = \frac{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k'+1)}}{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \frac{\Gamma(\gamma_0)\Gamma(n+1)}{\Gamma(n+\gamma_0)} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{13}$$

$$= \frac{\prod_{i<j} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k'+1)}}{\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k+1)}} \tag{14}$$

Passing to the log:

$$\begin{aligned}
log(r) = &\log \left(\prod_{i<j} p_{z_i' z_j'}^{y_{ij}} \cdot (1 - p_{z_i' z_j'})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k' + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k' + 1)} \right) \\
&- \log \left(\prod_{i<j} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \frac{\Gamma(n_k + \gamma_k)}{\Gamma(\gamma_k)\Gamma(n_k + 1)} \right) \\
= &\sum_{i<j} \Big(y_{ij} \cdot \log p_{z_i' z_j'} + (n_{ij} - y_{ij}) \cdot \log (1 - p_{z_i' z_j'}) \Big) \\
&+ \sum_{k=1}^{K} \left(\log \left(\Gamma(n_k' + \gamma_k) \right) - \log \left(\Gamma(\gamma_k) \right) - \log \left(\Gamma \left(n_k' + 1 \right) \right) \right) \\
&- \sum_{i<j} \left(y_{ij} \cdot \log p_{z_i z_j} + (n_{ij} - y_{ij}) \cdot \log (1 - p_{z_i z_j}) \right) \\
&- \sum_{k=1}^{K} \left(\log \left(\Gamma(n_k + \gamma_k) \right) - \log \left(\Gamma(\gamma_k) \right) - \log \left(\Gamma \left(n_k + 1 \right) \right) \right)
\end{aligned} \tag{15}$$

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:

$$r = \frac{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}'^{y_{ij}} \cdot (1 - p'_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \left(\frac{1}{y'^{(k+1)} - y'^{(k)}} \right)^{|L'^{(k)}|}}{\prod_{i<j} \binom{n_{ij}}{y_{ij}} p_{z_i z_j}^{y_{ij}} \cdot (1 - p_{z_i z_j})^{n_{ij} - y_{ij}} \cdot \prod_{k=1}^{K} \left(\frac{1}{y^{(k+1)} - y^{(k)}} \right)^{|L^{(k)}|}} \tag{16}$$

(17)

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

(18)

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

(19)

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

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