Connecting the Popularity Adjusted Block Model to the Generalized Random Dot Product Graph

The popularity adjusted block model (PABM)[1]

Let $G = (V, E)$ be an undirected, unweighted random graph with corresponding affinity matrix $A \in \{0,1\}^{n \times n}$. Then A is a random matrix with corresponding edge probability matrix P such that $A_{ij} \overset{indep}{\sim} Bernoulli(P_{ij})$ for $i > j$ ($A_{ji} = A_{ij}$ and $A_{ii} = 0$). Let there exist K underlying communities in G, and let n_k be the size of the k^{th} community in G such that $\sum_{k=1}^{K} n_k = n$.

If A and P are organized such that $n_k \times n_l$ blocks $A^{(kl)}$ and $P^{(kl)}$ describe the edges and edge probabilities between communities k and l, then $P^{(kl)} = \lambda^{(kl)}(\lambda^{(lk)})^\top$ for a set of fixed vectors $\{\lambda^{(st)}\}_{s,t=1,...,K}$. Each $\lambda^{(st)}$ for $s, t = 1, ..., K$ is a column vector of length n_s (i.e., the community corresponding to the first index provides the vector length).

We will use the notation $A \sim PABM(\{\lambda^{(kl)}\}_K)$ to denote a random affinity matrix A drawn from a PABM with parameters $\lambda^{(kl)}$ consisting of K underlying clusters/communities.

The generalized random dot product graph (GRDPG)[2]

Let $X \in \mathbb{R}^{n \times d}$ be latent positions of the vertices of a graph G. X consists of row vectors $x_i^\top$. Let $A \in \{0,1\}^{n \times n}$ be the corresponding affinity matrix.

Fix p, q such that $p + q = d$ and define $I_{pq} = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}$.

Then $G = (V, E)$ is a generalized random dot product graph with signature (p, q) and patent positions X iff its random affinity matrix can be described as $A_{ij} \overset{indep}{\sim} Bernoulli(P_{ij})$ where $P_{ij} = x_i^\top I_{pq} x_j$.

We will use the notation $A \sim GRDPG_{p,q}(X)$ to denote a random affinity matrix A drawn from latent positions X and signature (p, q).

Connecting the PABM to the GRDPG for $K = 2$

Let $X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & 0 & 0 \\ 0 & 0 & \lambda^{(21)} & \lambda^{(22)} \end{bmatrix}$ and $Y = \begin{bmatrix} \lambda^{(11)} & 0 & \lambda^{(12)} & 0 \\ 0 & \lambda^{(21)} & 0 & \lambda^{(22)} \end{bmatrix}$. Then $P = XY^\top$.

We can note that $Y = X\Pi$ where Π is the permutation matrix $\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Therefore, $P = X\Pi X^\top$.

Taking the spectral decomposition of $\Pi = UDU^\top$, we can see that $P = (XU)D(XU)^\top$. We can then denote $\Sigma = |D|^{1/2}$, the square root of the absolute values of the (diagonal) entries of D and obtain $P = (XU\Sigma)I_{pq}(XU\Sigma)^\top$ where p and q correspond to the number of positive and negative eigenvalues of Π, respectively. Therefore, the PABM with $K = 2$ is a special case of the GRDPG. We can however expand upon this a bit further.

The permutation described by Π has two fixed points and one cycle of order 2. The two fixed points are at positions 1 and 4, so Π has two eigenvalues equal to 1 and corresponding eigenvectors e_1 and e_4.

[1] https://arxiv.org/abs/1910.01931
[2] https://arxiv.org/abs/1709.05506

The cycle of order 2 switching positions 2 and 3 corresponds to eigenvalues 1 and -1 with corresponding eigenvalues $(e_2 + e_3)/\sqrt{2}$ and $(e_2 - e_3)/\sqrt{2}$ respectively. Therefore, $D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = I_{3,1}$ and $U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

Putting it all together, we get $P = (XU)I_{3,1}(XU)^\top$. Therefore, the PABM with $K = 2$ is a GRDPG with $p = 3$, $q = 1$, $d = K^2 = 4$, and latent positions $XU = \begin{bmatrix} \lambda^{(11)} & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(12)}/\sqrt{2} \\ 0 & \lambda^{(22)} & \lambda^{(21)}/\sqrt{2} & -\lambda^{(21)}/\sqrt{2} \end{bmatrix}$.

Generalization to $K > 2$

Let $\Lambda^{(k)} = \begin{bmatrix} \lambda^{(k,1)} & \cdots & \lambda^{(k,K)} \end{bmatrix} \in [0,1]^{n_k \times K}$.
Let X be a block diagonal matrix $X = \text{diag}(\Lambda^{(1)}, ..., \Lambda^{(K)}) \in [0,1]^{n \times K^2}$.

Let $L^{(k)}$ be a block diagonal matrix of column vectors $\lambda^{(lk)}$ for $l = 1, ..., K$. $L^{(k)} = \text{diag}(\lambda^{(1k)}, ..., \lambda^{(Kk)}) \in [0,1]^{n \times K}$.
Let $Y = \begin{bmatrix} L^{(1)} & \cdots & L^{(K)} \end{bmatrix} \in [0,1]^{n \times K^2}$.

Then $P = XY^\top$.

Similar to the $K = 2$ case, we again have $Y = X\Pi$ for a permutation matrix Π, so $P = X\Pi X^\top$.

The permutation described by Π has K fixed points, which correspond to K eigenvalues equal to 1 with corresponding eigenvectors e_k where $k = r(K + 1) + 1$ for $r = 0, ..., K - 1$. It also has $\binom{K}{2} = K(K-1)/2$ cycles of order 2. Each cycle corresponds to a pair of eigenvalues $+1$ and -1 and a pair of eigenvectors $(e_s + e_t)/\sqrt{2}$ and $(e_s - e_t)/\sqrt{2}$[3].

So Π has $K(K+1)/2$ eigenvalues equal to 1 and $K(K-1)/2$ eigenvalues equal to -1. Π has the decomposed form $\Pi = UI_{K(K+1)/2, K(K-1)/2}U^\top$, and we can describe the PABM with K communities as a GRDPG with latent positions XU with signature $\Big(K(K+1)/2, K(K-1)/2\Big)$.

Example: $K = 3$

Using the same notation as before:

$$X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & \lambda^{(13)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda^{(21)} & \lambda^{(22)} & \lambda^{(23)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda^{(31)} & \lambda^{(32)} & \lambda^{(33)} \end{bmatrix}$$

$$Y = \begin{bmatrix} \lambda^{(11)} & 0 & 0 & \lambda^{(12)} & 0 & 0 & \lambda^{(13)} & 0 & 0 \\ 0 & \lambda^{(21)} & 0 & 0 & \lambda^{(22)} & 0 & 0 & \lambda^{(23)} & 0 \\ 0 & 0 & \lambda^{(31)} & 0 & 0 & \lambda^{(32)} & 0 & 0 & \lambda^{(33)} \end{bmatrix}$$

Then $Y = X\Pi$ where $\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

[3]TODO: describe pairs (s, t) in more elegant/succinct closed form

Perhaps a simpler way to look at this is:

- Positions 1, 5, 9 are fixed
- The cycles of order 2 are
  - $(2, 4)$
  - $(3, 7)$
  - $(6, 8)$

Therefore, we can decompose $\Pi = UI_{6,3}U^\top$ where the first three columns of U consist of e_1, e_5, and e_9 corresponding to the fixed positions 1, 5, and 9, the next three columns consist of eigenvectors $(e_k + e_l)/\sqrt{2}$, and the last three columns consist of eigenvectors $(e_k - e_l)/\sqrt{2}$, where pairs (k, l) correspond to the cycles of order 2 described above.

The latent positions are the rows of

$$XU = \begin{bmatrix} \lambda^{(11)} & 0 & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(13)}/\sqrt{2} & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(13)}/\sqrt{2} & 0 \\ 0 & \lambda^{(22)} & 0 & \lambda^{(21)}/\sqrt{2} & 0 & \lambda^{(23)}/\sqrt{2} & -\lambda^{(21)}/\sqrt{2} & 0 & \lambda^{(23)}/\sqrt{2} \\ 0 & 0 & \lambda^{(33)} & 0 & \lambda^{(31)}/\sqrt{2} & \lambda^{(32)}/\sqrt{2} & 0 & -\lambda^{(31)}/\sqrt{2} & -\lambda^{(32)}/\sqrt{2} \end{bmatrix}.$$

Additional notes and observations

- XU is a convenient but not necessarily unique characterization of the latent positions. This comes into play when we try to estimate the latent positions—the estimate is only unique up to matrix multiplication by an arbitrary matrix in $\mathbb{O}(p, q)$.
  - i.e., let $Q \in \mathbb{O}(p, q)$, and let $P = (XU)I_{pq}(XU)^\top$. Then we can also decompose $P = (XUQ)I_{pq}(XUQ)^\top = (XU)(QI_{pq}Q^\top)(XU)^\top = (XU)I_{pq}(XU)^\top = P$
  - $\mathbb{O}(p, q) = \{Q \in \mathbb{R}^{(p+q)\times(p+q)} : QI_{pq}Q^\top = I_{pq}\}$
- If we use the latent positions described by XU, the dot product of two points in different clusters equals 0. Points that are not within the same cluster are orthogonal to each other (again, if we take XU as the true latent positions).
- Even though the latent positions XU follow these conditions, we can multiply by any matrix $Q \in \mathbb{O}(p, q)$ and have these conditions fail.

Recovery/Estimation/Inference

The main goals are:

1. Determining the underlying cluster/community memberships/structure
2. Estimating the latent positions XU
3. Estimating $\lambda^{(kl)}$ for $k, l = 1, ..., K$

Previous results

Theorem 5 from the GRDPG paper[4] says the following:

- Let
  - X be an latent position matrix with latent positions as n rows $x_i^\top$ in d dimensions
  - $A \sim GRDPG_{p,q}(X)$
  - $\hat{x}_i$'s be the adjacency embedding of A
- Then
  - $\max_{i=1,...,n} ||Q\hat{x}_i - x_i|| = O_P(\frac{(\log n)^c}{n^{1/2}})$
  - $Q \in \mathbb{O}(p, q)$ (unidentifiable)
  - $c > 0$

Under some sparsity conditions.

So we can only identify the embedding up to matrix multiplication.

[4] https://arxiv.org/abs/1709.05506

Both the GRDPG paper and the PABM paper[5] starting with the spectral embedding (adjacency or Laplacian) of A and performing a clustering step (K-means, GMM, etc.). Unfortunately, there aren't any theoretical guarantees for the clustering step.

The PABM paper suggests subspace clustering on the embedding, and we can clearly see the justification by looking at XU. The rows of XU that are within the same cluster lie in a K-dimensional subspace. We can further see that the K subspaces defined by the K clusters are all orthogonal to each other. However, in practice, the unidentifiable Q matrix breaks this property when we decompose A.

TODO: Investigate CLT

Proposed Methods

Let $z_i^\top$ denote the i^{th} row of XU.
Let $\gamma(x, y)$ denote the cosine similarity between x and y.
Then $\gamma(z_i, z_j) = 0$ if vertices i and j belong to different clusters, and $\gamma(z_i, z_j) > 0$ almost surely if i and j belong to the same cluster.

This then gives us the result:

- Let $\hat{Z}$ denote the adjacency embedding of $A \sim PABM(\{\lambda^{(kl)}\}_K)$ with rows $\hat{z}_i^\top$.
- Then $\exists Q \in \mathbb{O}(p, q)$ s.t. $\max_{i,j=1,\dots,n} |\gamma(Q\hat{z}_i, Q\hat{z}_j)| = O_P(\frac{(\log n)^c}{n^{1/2}})$ if vertices i and j are in different clusters.[6]
- We can state a similar result using the Laplacian embedding of A.

TODO: CLT results

Thus if we can recover Q, we can assign cluster memberships based on the behavior of the cosine similarity matrix.[7] The problem here is that we cannot get rid of the intractable matrix Q.

Also note that Q is not merely a rotation matrix. Multiplying by Q changes inter-point distances (and angles).

However, perhaps we can use the properties of XU to estimate Q. XU consists of points on K K-dimensional orthogonal hyperplanes. The PABM paper uses the hyperplanes part to justify subspace clustering but does not use the fact that they are orthogonal hyperplanes—the orthogonality is broken by the decomposition of P. If Z is the adjacency embedding of P, then $Z = XUQ^{-1}$ for some unknown $Q \in \mathbb{O}(p, q)$. However, the multiplication by Q^{-1} does not affect the fact that the points lie along (not necessarily orthogonal) hyperplanes, which is a justification for using subspace clustering to identify the community memberships.

For now, we will focus on he case $K = 2$. In this scenario, we have the following: We wish to find $Q \in \mathbb{O}(3, 1)$ such that ZQ "looks like" XU. $XU = \begin{bmatrix} \lambda^{(11)} & 0 & \lambda^{(12)}/\sqrt{2} & \lambda^{(12)}/\sqrt{2} \\ 0 & \lambda^{(22)} & \lambda^{(21)}/\sqrt{2} & -\lambda^{(21)}/\sqrt{2} \end{bmatrix}$, and we can observe the following properties:

1. If two rows of XU are not in the same cluster, they are orthogonal.
2. If vertex i is in the first cluster, its second entry is 0. Likewise, if vertex j is in the second cluster, its first entry is 0.
3. If vertex i is in the first cluster, its third and fourth entries are identical. Similarly, if vertex j is in the second cluster, the sum of its third and fourth entries is equal to 0.

It is straightforward to generalize these properties to $K > 2$.

So we need to find $Q \in \mathbb{O}(3, 1)$ such that $\hat{Z}Q$ meets these criteria as closely as possible. One optimization problem that could arise from this is:

[5] https://arxiv.org/abs/1910.01931

[6] TODO: Check that the convergence rate is true (I'm pretty sure it is since we are taking the dot product of two vectors and then dividing by their norms, so we should get the same convergence rate as in the GRDPG result). Or alternatively, $|(Q\hat{z}_i)^\top (Q\hat{z}_j)| = O_P(\frac{(\log n)^{2c}}{n})$.

[7] TODO: Use CLT to describe deviation from 1 or 0, e.g., if i, j in different clusters then what is the distribution/density of $\gamma(\hat{z}_i, \hat{z}_j)$, which should be centered at 0.

$$\begin{aligned}
&\arg\min_{Q} ||\gamma^{(1)}||^2 + ||\gamma^{(2)}||^2 \\
&\quad \text{s.t. } \Xi = \hat{Z}Q \text{ with rows } \xi_i^\top \\
&\qquad \gamma_i^{(1)} = \min(|\xi_{i1}|, |\xi_{i2}|) \\
&\qquad \gamma_i^{(2)} = \min(|\xi_{i3} + \xi_{i4}|, |\xi_{i3} - \xi_{i4}|) \\
&\qquad Q \in \mathbb{O}(3,1)
\end{aligned}$$

Again, this part is easily generalizable to $K > 2$.

If we embed P instead of A, the optimal value of this objective is 0.

A possible pseudo-algorithm for this is as follows:

1. Start with affinity matrix A and its adjacency embedding $\hat{Z}$
2. Pick some $\tilde{Q} \in \mathbb{O}(3,1)$.
3. Compute $\Xi = \hat{Z}\tilde{Q}$ and vectors $\gamma^{(1)}$, $\gamma^{(2)}$.
4. Compute the objective function.
5. Pick a new $\tilde{Q} \in \mathbb{O}(3,1)$ that lowers the objective function, and repeat steps 3-5.

Note that this doesn't include orthogonality between points in different clusters, but the other two criteria imply orthogonality.

The advantage of this approach over others is that we can directly estimate XU and therefore X (since we know what U is exactly for any given K) instead of relying on subspace clustering. Once we have the embedding $\hat{Z}$ and we find $\hat{Q}$, we can set $\hat{X} = \hat{Z}\hat{Q}U^\top$, and the elements of $\hat{X}$ will correspond to the PABM vectors $\lambda^{(kl)}$ (exactly if the embedding is of P, approximately if the embedding is of A).

The difficulty is then in choosing Q from $\mathbb{O}(3,1)$. It turns out that $\mathbb{O}(3,1)$ is the Lorentz group[8] and matrices of this group are equivalent to Lorentz transformations, which describe the warping of spacetime in special relativity[9]. Any matrix $Q \in \mathbb{O}(3,1)$ can be expressed as a product of six matrices that depend on four parameters[10], i.e., $Q = R_1(\theta)R_2(\phi)R_3(\psi)S_1(\theta)S_2(\tau)S_3(-\tau)$.The six matrices are analogous to three rotation matrices and six squeeze mappings.

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$R_2(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$R_3(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 & 0 \\ -\sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S_1(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cosh\theta & \sinh\theta \\ 0 & 0 & \sinh\theta & \cosh\theta \end{bmatrix}$$

[8] https://en.wikipedia.org/wiki/Lorentz_group

[9] There is actually a multiplication by -1 to account for here.

[10] https://iopscience.iop.org/book/978-1-6817-4254-0/chapter/bk978-1-6817-4254-0ch1

$$S_2(\tau) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cosh\tau & 0 & \sinh\tau \\ 0 & 0 & 1 & 0 \\ 0 & \sinh\tau & 0 & \cosh\tau \end{bmatrix}$$

$$S_3(-\tau) = \begin{bmatrix} \cosh\tau & 0 & 0 & \sinh(-\tau) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sinh(-\tau) & 0 & 0 & \cosh\tau \end{bmatrix}$$

We can write $Q = Q(\theta, \phi, \psi, \tau)$ and tweak the four angles until convergence.

TODO: Show that for arbitrary K, we need K^2 angles and $K(K+1)$ matrices.

TODO: Compute gradients(?)

Alternative approaches

1. If we know the cluster memberships (or if we estimate them using subspace clustering), then we can write a perhaps easier or more robust optimization problem:
   Rearrange $\hat{Z}$ such that the first n_1 rows are in cluster 1 and the next n_2 rows are in cluster 2. Then solve the following.

$$\begin{aligned} \arg\min_{Q} & ||\Xi_{1:n_1,1}||^2 + ||\Xi_{n_1+1:n,2}||^2 + ||\Xi_{1:n_1,3} - \Xi_{1:n1,4}||^2 + ||\xi_{n_1+1:n,3} + \xi_{n_1+1,4}||^2 \\ \text{s.t. } & \Xi = \hat{Z}Q \\ & Q \in \mathbb{O}(3,1) \end{aligned}$$

   Numerical expeirments thus far suggest that this method does not perform any better than estimating XU directly without a clustering step (although more investigation is required to say anything for sure).

2. $Q \in \mathbb{O}(3,1)$ can be written as:
   $Q = \begin{bmatrix} M & -a \\ -b^\top & \gamma \end{bmatrix}$[11]. This might produce a better parameterization of the Q matrices, although so far I haven't had much luck here. We can see that if $QI_{3,1}Q^\top = I_{3,1}$, then we get the following constraints:
   - $MM^\top - aa^\top = I$
   - $\gamma a = Mb$
   - $\gamma^2 = b^\top b + 1$

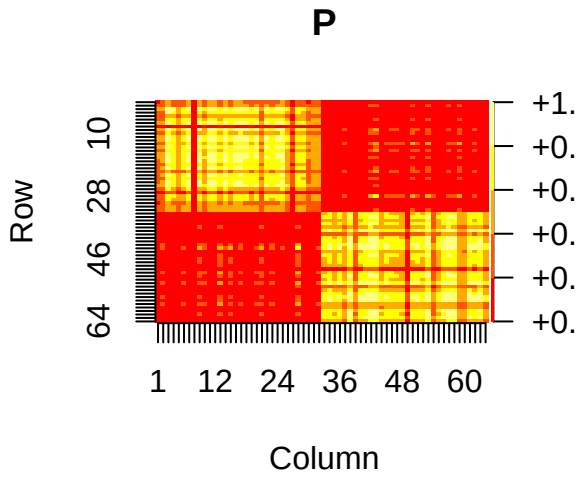
Numerical example

PABM with $K = 2$

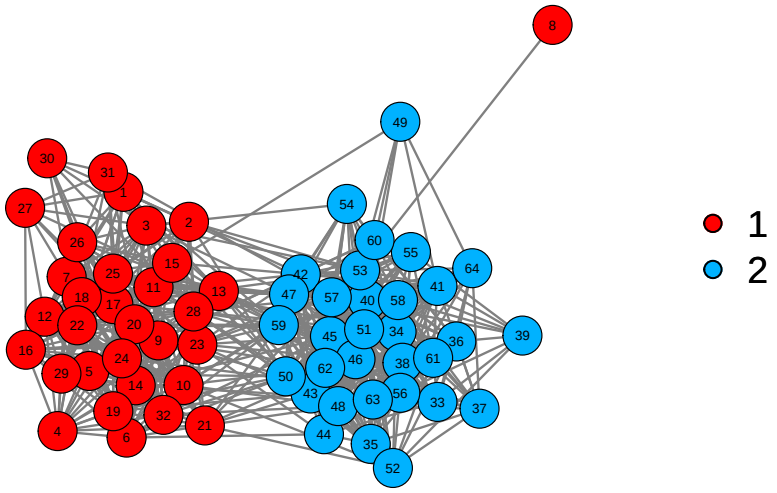
We will draw the elements of $\lambda^{(11)}$ and $\lambda^{(22)}$ independently from a beta distribution with shape parameters a and b such that $a > b$. The elements of $\lambda^{(12)}$ and $\lambda^{(21)}$ will also be independently drawn from a beta distribution, this time with shape parameters c and d such that $c < d$. This ensures that it is more likely for a pair of vertices from the same cluster to be connected than it is for a pair of vertices from different clusters.

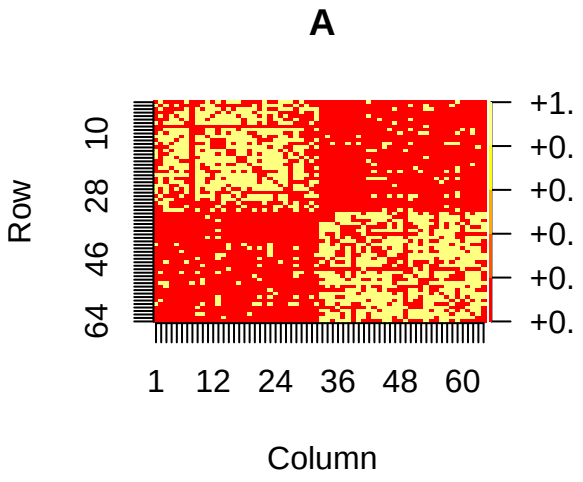
Probability matrix:

[11] https://en.wikipedia.org/wiki/Lorentz_transformation

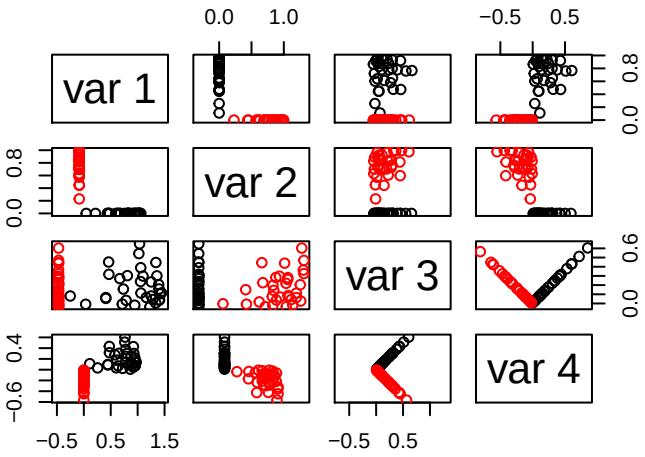


One realization A of P:

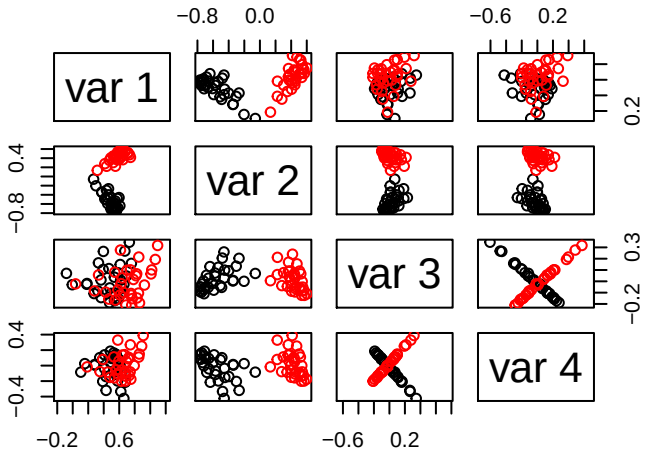




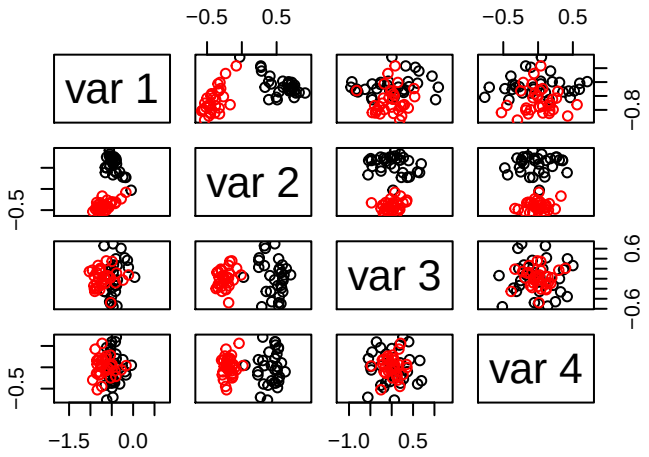
The "true" latent positions XU:



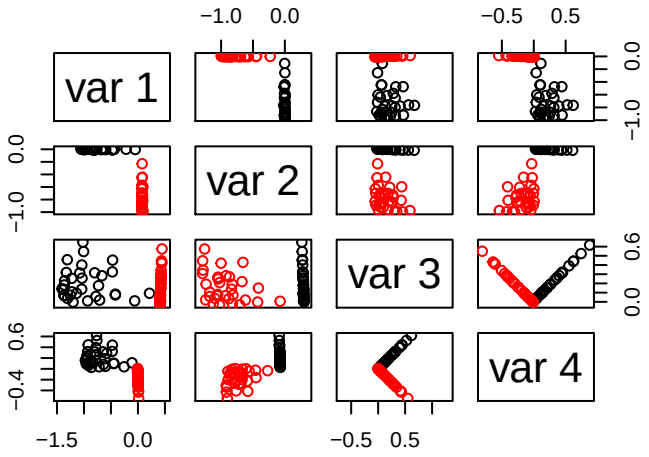
Z, the adjacency embedding of P:



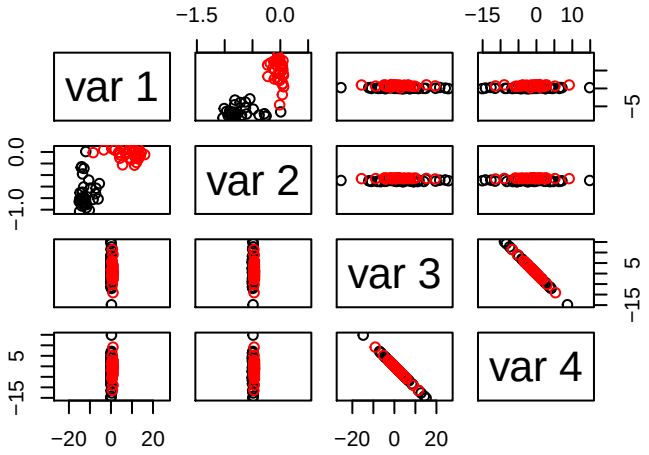
$\hat{Z}$, the adjacency embedding of A:



Recovering XU from Z:



Recovering XU from $\hat{Z}$:



Some brief checking shows that we are able to recover the XU and the $\lambda^{(kl)}$'s from P exactly, which suggests that this method should work asymptotically, but it is not as well behaved when we use A.

Additional TODOs

- We are currently treating the optimization step as a black box (just using Nelder-Mead or simulated annealing at the moment). Since $Q = Q(\theta, \phi, \psi, \tau)$ and is smooth with respect to each of those four parameters, we should be able to use a more disciplined gradient/Hessian-based approach.
- Doing the above should provide us a good way to describe the convergence behavior of estimating Q.
- Is the connection between PABM and special relativity a happy coincidence, or is there something more to explore here?
- Generalize optimization step to $K > 2$.
- The estimate $\hat{Z}\hat{Q}$ doesn't necessarily obey GRDPG rules. How can we force it to?
- What happens when we let $\lambda^{(kl)}$'s be random variables instead of fixed vectors, e.g., $\lambda_i^{(kl)} \sim Beta(a_{kl}, b_{kl})$ (or some Dirichlet equivalent)?