Classification and Regression on Random Dot Product Graphs

Department of YYY, University of XXX

November 10, 2023

Abstract

The random dot product graph (RDPG) has become a powerful modeling tool in uncovering latent structures within graphs. In particular, it has been shown that the RDPG describes a wide range of popular random graph models with rigid latent structures. More recently, joint modeling of mutliple random graphs that share common properties or structures across graphs have been introduced, such as the multilayer RDPG, multiple RPDG, multilayer stochastic block model, etc. In this work, we use these joint random graph models in the context of statistical learning, such as classification and regression, by introducing the multiple latent structure model, in which the graphs share a common latent structure with different parameters that correspond to different response variables. Then we propose various estimation techniques involving manifold learning to estimate these parameters and in turn predict the responses, with theorems guaranteeing convergence of the predictions. Simulations and applications on functional MRI data verify the performance of these methods.

Keywords: latent structure models, random dot product graph, neuroimaging, brain connectivity networks

1 Introduction

Graph and network data are now as ubiquitous as traditional feature data in the fields of sociology (e.g., social networks), neuroimaging (e.g., brain connectivity networks), and deep learning (e.g., graph neural networks). As a result, new statistical and machine learning methods have recently been developed to analyze network data. One approach is to treat the network as a random graph that comes from some probability model. In particular, if we sum up the network in an adjacency matrix $A \in \mathbb{R}^{n \times n}$, then one probability model might be to draw each element A_{ij}, which represents the existence of an edge or the edge weight from vertex i to j, independently from some distribution, perhaps with a unique parameter for the pair (i, j), e.g., $A_{ij} \overset{\text{ind}}{\sim} F_{\theta_{ij}}$. The classical example of this is the Erdös-Rényi model [4], in which every edge is drawn from the same distribution, typically a Bernoulli distribution, using the same parameter, i.e., $A_{ij} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$. The inhomogeneous Bernoulli graph extends this by allowing each edge, represented by A_{ij}, to have its own parameter, P_{ij}, e.g., in the Bernoulli case, $A_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(P_{ij})$. Typically, the parameters are collected into an edge probability matrix (in the case of unweighted graphs) or edge parameter matrix (in the case of weighted graphs), denoted as $P \in \mathbb{R}^{n \times n}$. The network analysis problem in this setting is to estimate P given A.

If we let the parameter matrix P be unconstrained, the inference problem is overparameterized. On the other hand, the classical Erdös-Rényi model is often too restrictive to describe real, observed networks. Much work has been done to develop models that are constrained enough for robust statistical inference while being generalizable to describe a wide range of networks. One such family of random graph models is the random dot product graph (RDPG), first proposed by Young and Scheinerman [15], which is a type of latent space graph in which each vertex of the graph has a corresponding latent vector in a low-dimensional Euclidean space $\mathbb{R}^d$, and the edge parameter between each pair of vertices is determined by the dot product of the corresponding vectors. In this model, the constraint is the low rank of the parameter matrix P, assuming that the latent dimension d is less than the number of vertices n. Further constraints can be imposed on the RDPG in the form of distributional assumptions on the latent vectors or restricting the latent vectors to lie on subspaces or manifolds in the latent space [3].

It has been shown [11, 7] that the RDPG (as well as the generalized random dot product graph [11]) can describe a wide range of popular random graph models, such as the Erdös-Rényi model, the stochastic block model (SBM) [8], degree corrected block model (DCBM) [6], and popularity adjusted block model (PABM) [12], and describing these models as RDPGs (or its generalized version) is useful for parameter estimation. The RDPG captures a wide range of phenomena that can be described as networks, and in fact any network that can be thought of as being sampled from a parameter matrix P can be described as a (generalized) RDPG, especially if P is low rank. The RPDG representation of these networks

2 Methods

2.1 Definitions and Models

We begin by defining the RDPG and the LSM.

Definition 1 (Random dot product graph (RDPG) [15])**.** Let $\mathcal{X}$ be a subset of $\mathbb{R}^d$ for some latent space dimension $d \geq 1$ such that for any $x_1, x_2 \in \mathcal{X}$, $x_1^\top x_2 \in [0, 1]$. Let F_θ be a distribution with support $\mathcal{X}$ and parameters θ, and sample $x_1, ..., x_n \stackrel{\text{iid}}{\sim} F_\theta$. A graph G with adjacency matrix A is a random dot product graph with latent vectors $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$

drawn from distribution F_θ if $A \sim XX^\top$.

We use the notation $A \sim \text{RDPG}(F_\theta)$ to denote a random adjacency matrix A drawn from latent vectors distributed as F_θ.

Remark 1. The latent vectors of an RDPG are not unique. Suppose that $P = XX^\top$ is the edge parameter matrix of an RDPG with latent positions X. Then any orthogonal transformation W on X results in the same edge parameter matrix. More precisely, let $\tilde{X} = XW$. Then it is clear that $\tilde{X}\tilde{X}^\top = XWW^\top X^\top = XX^\top = P$ results in the same edge parameter matrix. Thus, there are infinitely many latent vector configurations that can result in the same P, but for any two latent vector configurations, there exists an orthogonal mapping that connects the two.

Definition 2 (Latent structure model (LSM) [3])**.** Let $\mathcal{C} \subset \mathcal{X} \subset \mathbb{R}^d$ be a smooth, nonintersecting one-dimensional manifold on the domain of a RDPG as defined in definition 1, parameterized by function $p(t) : [0, 1] \to \mathcal{C}$. Then if $t_1, ..., t_n \overset{\text{iid}}{\sim} F_\theta$ for some distribution F_θ with support $[0, 1]$ and parameter θ, each $x_i = p(t_i)$, and $A \sim XX^\top$ for $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, A is the adjacency matrix of a latent structure model on curve $\mathcal{C}$ with parameterization p and underlying distribution F_θ.

We use the notation $A \sim \text{LSM}(\mathcal{C}, F_\theta)$ or $A \sim \text{LSM}(p, F_\theta)$ to denote an adjacency matrix A drawn as an LSM on curve $\mathcal{C}$ or its parameterization p with underlying distribution F_θ.

Remark 2. Although Athreya et al. [3] defined the LSM by a single one-dimensional manifold $\mathcal{C}$ in the latent space, in this paper, we will allow for the existence of multiple one-dimensional manifolds, $\mathcal{C}_1, ..., \mathcal{C}_K$, (i.e., mixture of manifolds distribution). This type of latent space mixture distribution is observed in networks with community structure ([2]). For estimation, if the membership of each latent vector to the manifolds is known, then each manifold can be learned separately using the vectors that belong to that manifold.

If the memberships are not known, then we use an iterative algorithm to both cluster the latent vectors to a known number of manifolds and learn the manifolds using the cluster assignments.

In the case of a mixture of K curves, we use the notation $A \sim \text{LSM}(\{\mathcal{C}_k\}_K, F_\theta, \alpha)$ or $A \sim \text{LSM}(\{p_k\}_K, F_\theta, \alpha)$, where $\alpha = (\alpha_1, ..., \alpha_K)$, $\sum_{k=1}^K \alpha_k = 1$ is the mixture parameter. For simplicity, we only consider the case where the underlying distribution is the same for each curve.

A plausible inference task in the RDPG is to estimate the original latent vectors. The adjacency spectral embedding [13] is a consistent estimator of the latent vectors, up to some unknown orthogonal transformation.

Remark 3 (Sparsity parameter).

Definition 3 (Adjacency spectral embedding (ASE) [13])**.**

Theorem 1 (Consistency of the ASE [9])**.**

Theorem 1 implies that the embedding vectors of the ASE converge to the original latent vectors, up to some unidentifiable orthogonal transformation, and the maximum deviation from the original latent vectors after the orthogonal transformation is bounded by a value that decays to 0. This implies that in the LSM, the ASE can lead to consistent estimation of F_θ, the underlying distribution, and in fact, Athreya et al. [3] showed exactly this.

Definition 4 (Multiple latent structure model (MLSM))**.** Let $\mathcal{C}^{(1)}, ..., \mathcal{C}^{(L)} \subset \mathbb{R}^d$ be a sequence of curves defining the latent positions of a sequence of L LSMs as in definition 2. Each $\mathcal{C}^{(\ell)}$ is parameterized by function $p^{(\ell)}(t) : [0, 1] \to \mathcal{C}^{(\ell)}$. Let $p^{(1)}, ..., p^{(L)}$ be a sequence of functions with domain $[0, 1]$ that parameterize the curves $\mathcal{C}^{(1)}, ..., \mathcal{C}^{(L)}$, let F be a parametric distribution with support $[0, 1]$, and let $\theta_1, ..., \theta_L$ be a sequence of parameters for distribution

F. For each $p^{(\ell)}$, sample $t_1^{(\ell)}, ..., t_{n_\ell}^{(\ell)} \overset{\text{iid}}{\sim} F_{\theta(\ell)}$ for some distribution F parameterized by θ_ℓ, and let $x_i^{(\ell)} = p_\ell(t_i^{(\ell)})$ for each $i = 1, ..., n_\ell$, again as in definition 2. Sample a sequence of L adjacency matrices, each as $A^{(\ell)} \overset{\text{ind}}{\sim} \text{LSM}(p_\ell, F_{\theta_\ell})$. Then the sequence $\{A^{(\ell)}\}_L$ is are the adjacency matrices of a multiple latent structure model with curves $\{C^{(\ell)}\}_L$ parameterized by $\{p^{(\ell)}\}$ and underlying distribution F with parameters $\{\theta_\ell\}_L$.

We use the notation $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$ to denote a sequence of adjacency matrices drawn from an MLSM with parameterizations $\{p^{(\ell)}\}$ and parameters $\{\theta_\ell\}$ on underlying distrubtion F.

Remark 4. Again as in the case of a single LSM, we also allow for each $A^{(\ell)}$ to be sampled from a latent structure composed of K curves with mixture parameter $\alpha^{(\ell)}$. In this case, we use the notation $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{\mathcal{C}_k^{(\ell)}\}_{K,L}, F, \{\theta_\ell\}_L, \{\alpha^{(\ell)}\}_L)$ or $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p_k^{(\ell)}\}_{K,L}, F, \{\theta_\ell\}_L, \{\alpha^{(\ell)}\}_L)$.

2.1.1 Connections to Related Models

In the MLSM defined in definition 4, the only commonality that we assume from graph to graph is that they are all LSMs with the same underlying distribution family. If we impose further restrictions, namely that the latent structures are linear, we obtain the multilayer random dot product graph [5].

Example 1 (Comparison to the multilayer DCBM [1], multi-RDPG [10], and MREG [14])**.** In the K-community DCBM, the probability of an edge between a pair of vertices is given by

$$A_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(\omega_i \omega_j B_{z_i, z_j}),$$

where $z_i \in \{1, ..., K\}$ is the community label for vertex i, $B_{k,\ell}$ is the block connectivity between communities k and ℓ, and ω_i is the degree correction parameter for vertex i. In order to preserve identifiability and uniqueness, a common constraint on these parameters is to set $\sum_{i:z_i=k} \omega_i^2 = 1$ [6]. As in

The edge parameter matrix of a K-community DCBM with n vertices can be decomposed as follows:

$$P = \Omega B \Omega^\top,$$

where B is a $K \times K$ matrix of block connectivities and Ω is an $n \times K$ matrix such that $\Omega_{ik} = \omega_i$ if vertex i is in community k and 0 otherwise. Then it is clear that if B is positive semidefinite matrix of rank K, P can also be viewed as a K-dimensional RDPG, since the rows of Ω are normalized and can be seen as eigenvectors, and B is full rank and can be decomposed into a diagonal matrix and a rotation matrix, i.e., $P = (\Omega V)\Lambda(\Omega V)^\top$. Furthermore, this matrix decomposition implies that the latent vectors lie on one of K line segments that intersect at the origin [11]. Thus, the DCBM is a special case of the LSM in which there are K latent "linear curves" (i.e., lines) in $\mathbb{R}^K$.

The multilayer DCBM as described by Agterberg et al. [1] extends this to a sequence of DCBMs with the same community structure, allowing the block connectivities and degree correction parameters to change for each layer but keeping the same community structure throughout, i.e., each element of $P^{(\ell)} = \Omega^{(\ell)} B^{(\ell)} (\Omega^{(\ell)})^\top$ can vary with ℓ but $\Omega^{(\ell)} = 0 \iff \Omega^{(\ell')} = 0$ and similarly, $\Omega^{(\ell)} > 0 \iff \Omega^{(\ell')} > 0$, for every pair (ℓ, ℓ'). On the other hand, if the further restriction of $\Omega^{(\ell)} = \Omega^{(\ell')}$ for all (ℓ, ℓ'), i.e., $P^{(\ell)} = \Omega B^{(\ell)} \Omega^\top$, we obtain a special case of the multiple RDPG (multi-RDPG) with the identity link function, as described by Nielsen and Witten [10], or equivalently, a special case of the multiple random eigen graphs model (MREG), as described by Wang et al. [14]. However, if the community labels are not identical from graph to graph, the sequence of DCBMs cannot be described as a multilayer DCBM, multi-RDPG, or MREG, but it can still be described as an MLSM.

Example 2 (Nonlinear MLSM and comparison to MREG and multi-RDPG)**.**

2.1.2 Classification and Regression on the MLSM

To use the MLSM for classification or regression problems, we assign response variables $y_1, ..., y_L$ to each graph. Then if each response y_ℓ depends on $p^{(\ell)}(t)$ (the parameterization of the ℓ^{th} curve) or θ_ℓ (the parameter of the ℓ^{th} distribution), or some combination of the two, there is a plausible setup for a predictive modeling task for predicting y_ℓ after observing $A^{(\ell)}$. Four such scenarios are given:

Definition 5 (MLSM for classification 1)**.** Let $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$, $y_1, ..., y_L \in \{1, ..., M\}$ be labels for each of the L graphs of the MLSM, and each θ_ℓ take on one of M values $\{\phi_1, ..., \phi_M\}$ corresponding to its response, y_ℓ, i.e., $\theta_\ell = \phi_{y_\ell}$.

We observe the adjacency matrices $A^{(1)}, ..., A^{(L)}$ and the first r labels $y_1, ..., y_r$, and the task is to predict the unknown labels $y_{r+1}, ..., y_L$.

This can be described by the following generative model:

1. Draw labels $y_1, ..., y_L \overset{\text{iid}}{\sim} \text{Categorical}(\{1, ..., M\}, \{\pi_1, ..., \pi_M\})$.
2. Set each $\theta_\ell = \phi_{y_\ell}$.
3. Draw adjacency matrices as $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$.

Definition 6 (MLSM for classification 2)**.** Let $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{\mathcal{C}^{(\ell)}\}, F, \{\theta_\ell\}_L)$, $y_1, ..., y_L \in \{1, ..., M\}$ be labels for each of the L graphs of the MLSM, and each $\mathcal{C}^{(\ell)}$ take on one of M functional forms $\{p^{(1)}(t), ..., p^{(M)}(t)\}$ corresponding to its response variable, y_ℓ, i.e., $\mathcal{C}^{(\ell)}$ is parameterized by $p^{(y_\ell)}(t)$.

We observe the adjacency matrices $A^{(1)}, ..., A^{(L)}$ and the first r labels $y_1, ..., y_r$, and the task is to predict the unknown labels $y_{r+1}, ..., y_L$.

This can be described by the following generative model:

1. Draw labels $y_1, ..., y_L \overset{\text{iid}}{\sim} \text{Categorical}(\{1, ..., M\}, \{\pi_1, ..., \pi_M\})$.
2. Set each $p^{(\ell)}(t) = p^{(y_\ell)}(t)$.
3. Draw adjacency matrices as $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$.

As in remark 4, we also allow for each adjacency matrix $A^{(\ell)}$ to be drawn from a latent structure consisting of multiple curves $\{p_k^{(\ell)}\}_K$.

Definition 7 (MLSM for regression 1)**.** Let $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$, and suppose that for each $\ell = 1, ..., L$, the ℓ^{th} graph is coupled with a response variable, y_ℓ, as $y_\ell \overset{\text{ind}}{\sim} \mathcal{N}(\theta_\ell^\top \beta, \sigma^2)$.

We observe the adjacency matrices $A^{(1)}, ..., A^{(L)}$ and the first r response variables $y_1, ..., y_r$. In this setting, there are two plausible inference tasks. The first is to estimate the coefficient vector β. The second is to predict the unobserved response variables $y_{r+1}, ..., y_L$.

Definition 8 (MLSM for regression 2)**.** Let $A^{(1)}, ..., A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$ and each $p^{(\ell)}(t) = p(t; \gamma_\ell)$, where γ_ℓ is the vector of parameters for function p. Suppose that for each $\ell = 1, ..., L$, the ℓ^{th} graph is coupled with a response variable, y_ℓ, as $y_\ell \overset{\text{ind}}{\sim} \mathcal{N}(\gamma_\ell^\top \beta, \sigma^2)$.

We observe the adjacency matrices $A^{(1)}, ..., A^{(L)}$ and the first r response variables $y_1, ..., y_r$. In this setting, there are two plausible inference tasks. The first is to estimate the coefficient vector β. The second is to predict the unobserved response variables $y_{r+1}, ..., y_L$.

As in remark 4, we also allow for each adjacency matrix $A^{(\ell)}$ to be drawn from a latent structure consisting of multiple curves $\{p_k^{(\ell)}\}_K$.

In all of these settings, the ASE of $A^{(\ell)}$ provides some insight into both $p^{(\ell)}$ and θ_ℓ. Athreya et al. [3] showed that with some additional information, the ASE of $A^{(\ell)}$ can lead to a consistent estimator for θ_ℓ.

2.2 Main Results

3 Simulation Study

3.1 Classification

In this simulation experiment, the latent vectors were sampled along a Bezier curve defined by $g(t) = \begin{bmatrix} t^2 & 2t(1-t) \end{bmatrix}^\top$. The timepoints t_i were sampled as iid Beta random variables with two sets of parameters, (α_1, β_1) and (α_2, β_2). The setup is as follows:

1. Draw response variables $y_1, ..., y_n \overset{\text{iid}}{\sim} \text{Multinomial}(1/2, 1/2)$.
2. For each $i = 1, ..., n$, draw $t_i \mid y_i \overset{\text{ind}}{\sim} \text{Beta}(\alpha_{y_i}, \beta_{y_i})$, such that
    - $\alpha_1 = 1$
    - $\beta_1 = 2$
    - $\alpha_2 = 2$
    - $\beta_2 = 1$
3. Construct each latent vector as $x_i = g(t_i)$ and compile them in data matrix $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$.
4. Sample graph and its adjacency matrix as $A \sim \text{RDPG}(X)$.

For each graph, we constructed the ASE, which was used to estimate the parameters $(\hat{\alpha}, \hat{\beta})$ for the graph, using the maximum likelihood method. The estimated parameters were then used as predictors $y_1, ..., y_n$, setting aside half for training and half for testing. We investigated graphs of size $|V| = 32, 64, 128, 256$. The number of graphs for each experiment was set to $L = 64$. For each (number of graphs, size of graph) pair, we performed 32 replicates. Figure 1 shows the ASE of one graph.

Figure 2 shows the boxplots of the classification error rates.

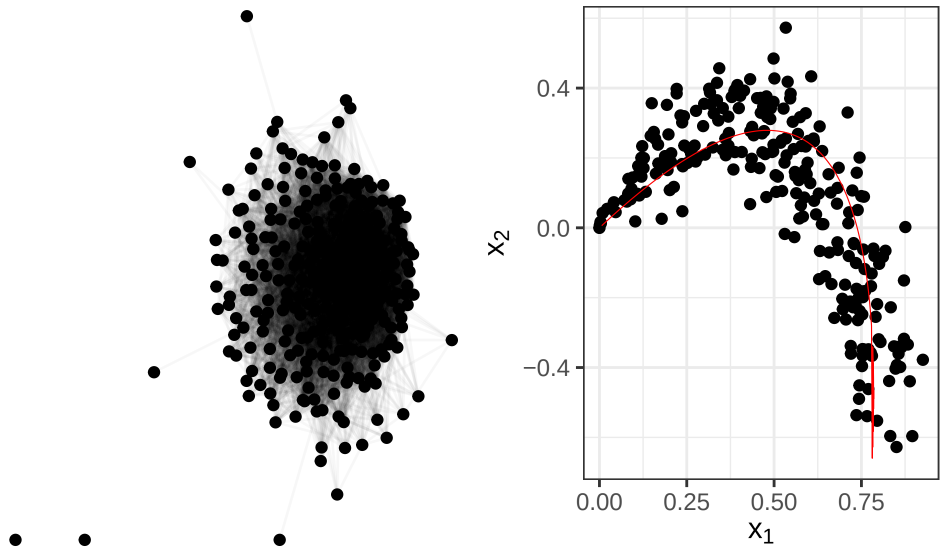


Figure 1: One simulated graph (left) and its ASE (right). The red curve is the fitted quadratic Bezier curve on the ASE.

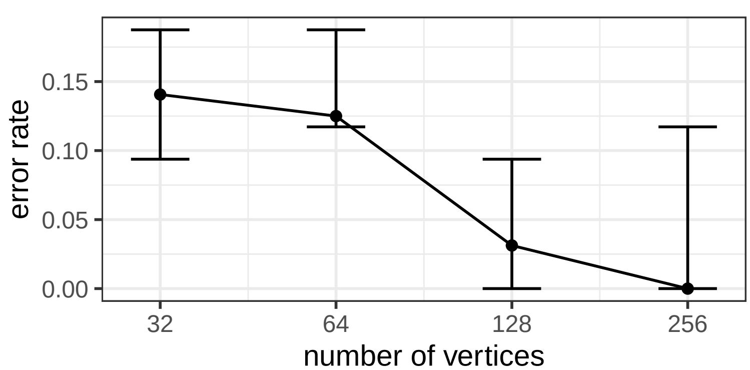


Figure 2: Median classification error rate and its IQR vs. number of vertices in each graph.

3.2 Regression

1. Draw angles $\theta_1, ..., \theta_N \overset{\text{iid}}{\sim} \text{Uniform}(\pi/6, \pi/3)$.

2. For each $k = 1, ..., N$,

   i. Draw $t_1, ..., t_n \overset{\text{iid}}{\sim} \text{Uniform}(0, 1)$;

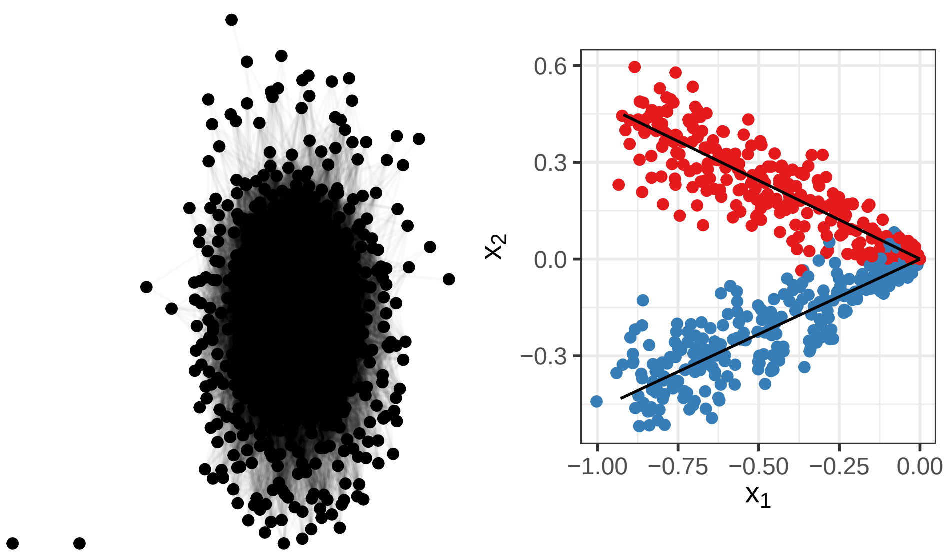
   ii. Draw $z_1, ..., z_n \overset{\text{iid}}{\sim} \text{Multinomial}(1/2, 1/2)$;

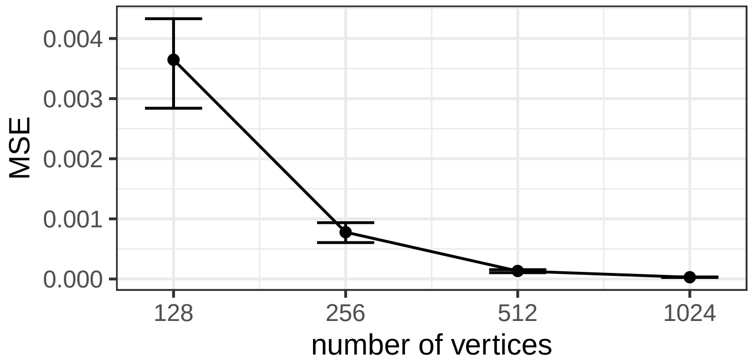
   iii. For each $i = 1, ..., n$, set $x_i = \begin{cases} \begin{bmatrix} t_i & 0 \end{bmatrix}^\top & z_i = 1 \\ \begin{bmatrix} t_i \cos\theta_\ell & t_i \sin\theta_\ell \end{bmatrix}^\top & z_i = 2 \end{cases}$;

   iv. Collect $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$ and draw $A \sim \text{RDPG}(X)$;

   v. Set the response $y_\ell = \beta_0 + \beta_1 \theta_\ell$

In this simulation, we set $\beta_0 = \beta_1 = 1$. The number of graphs was set to $L = 64$, and the number of vertices per graph was set to $n = 128, 256, 512, 1024$. For each n, we simulated 32 replicates.





4 Applications

4.1 Human Connectome Project Aging Study

In the first example, we analyzed fiber count data between brain regions from the Human Connectome Project (HCP). When analyzing these data as graphs, we denote the regions as vertices and the fiber counts between pairs of regions as weighted edges. A plausible statistical model for these data is to assume that the edge weights between pairs of vertices is Poisson distributed, i.e., the adjacency matrix is sampled as $A_{ij} \stackrel{\text{ind}}{\sim} \text{Poisson}(\Theta_{ij})$, where $\Theta \in \mathbb{R}_+^{n \times n}$ is a symmetric matrix of Poisson parameters.

In this dataset, there are $N = 516$ graphs (corresponding to individual subjects), each with $n = 84$ vertices (corresponding to brain regions). Analyzing these graphs as RDPGs reveals that the DCBM is a good candidate for these data (figure 3).

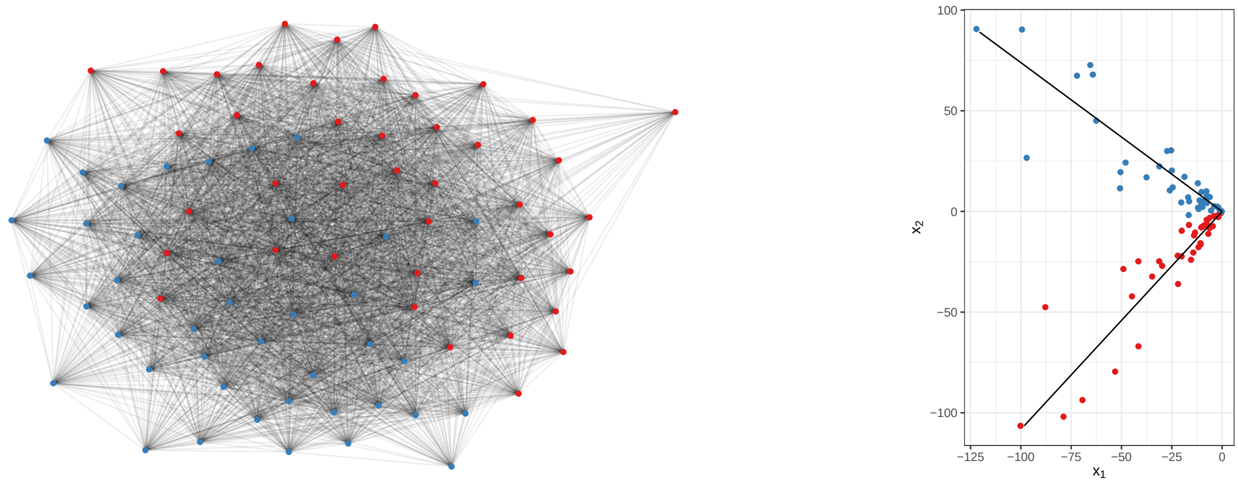


Figure 3: One graph from the HCP dataset (left) and its ASE (right). In the ASE, the lines are fitted via K-curves clustering using degree $= 1$. The outputted clusters correspond exactly to the left (red) and right (blue) hemispheres.

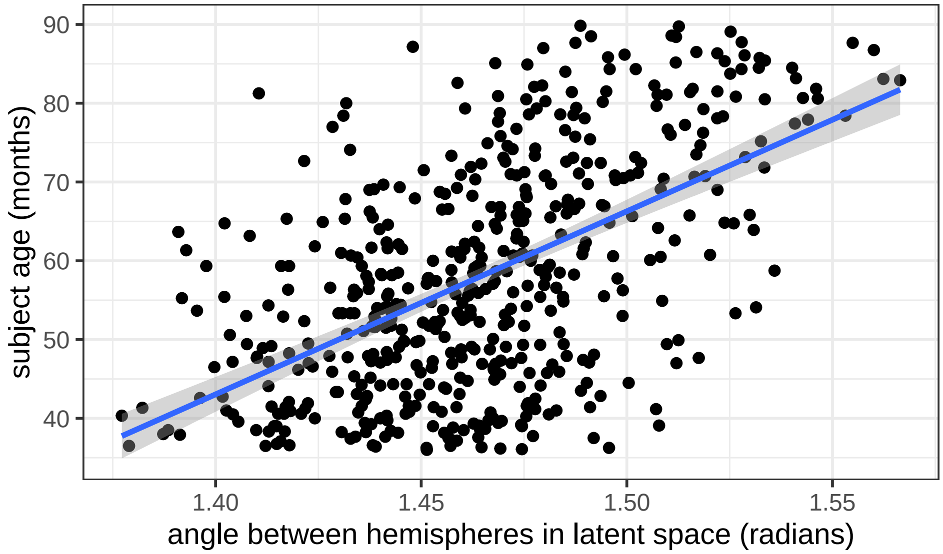


Figure 4: Scatterplot between the subject age (in months) vs. the fitted angle between hemispheres of the brain in the latent space.

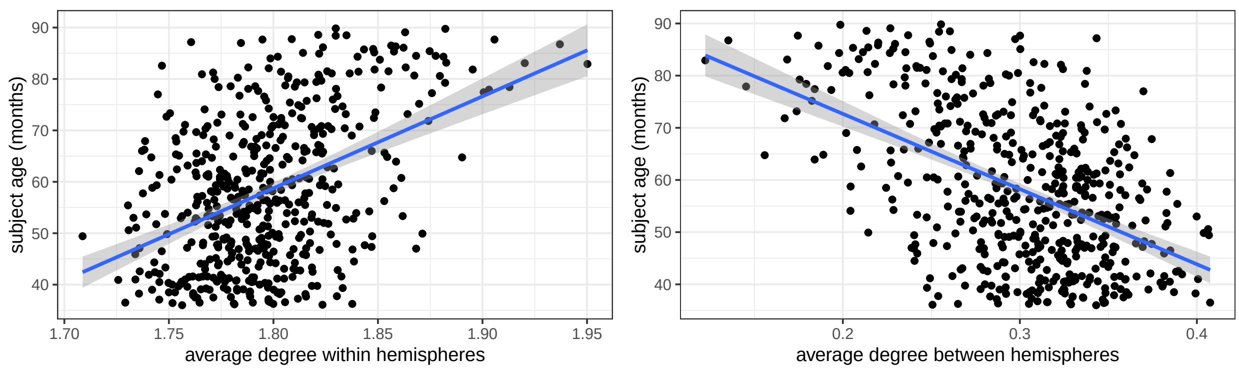
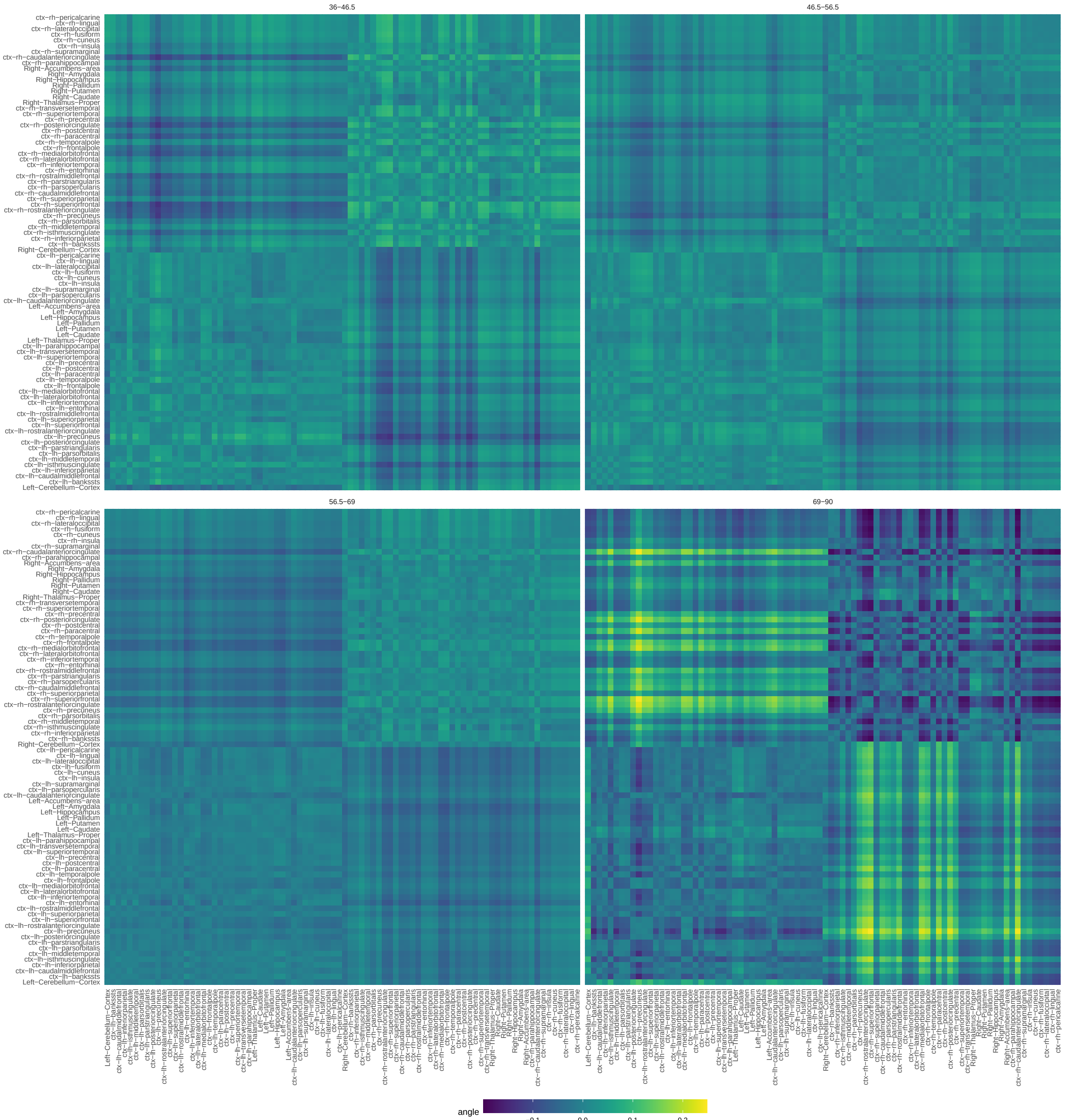


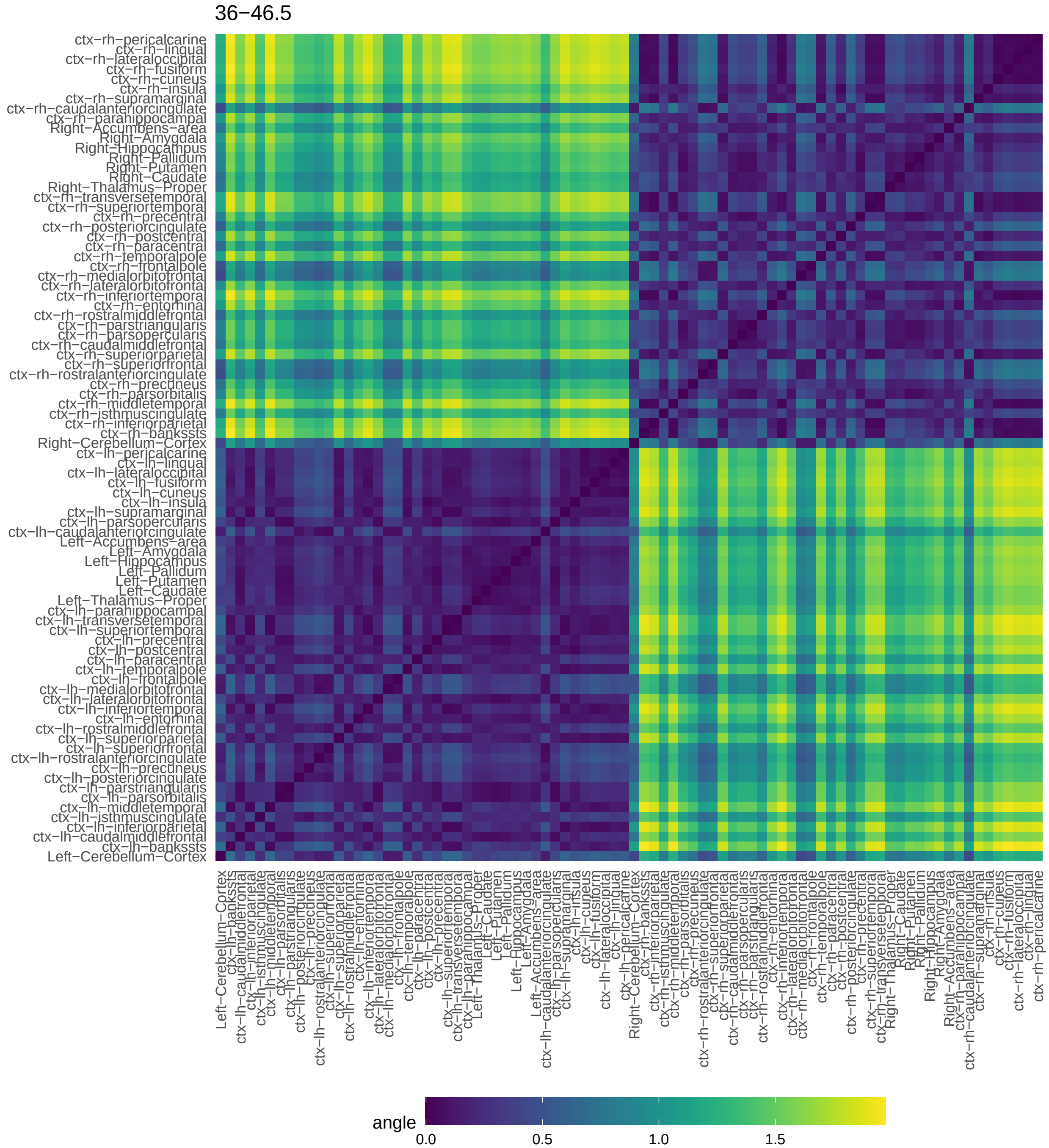
Figure 5: Scatterplot between the subject age (in months) vs. the average degree within (left) and between (right) hemispheres.

The ASE suggests a latent structure comprised of two line segments, one for each hemisphere of the brain, that meet at the origin. One possible parameter is the angle between the two lines. The estimated angles were observed to correlate with the subject's age, with wider angles corresponding to older subjects (figure 4). A linear regression setting aside half of the brain connectivity graphs as test data achieves an RMSE of 11.89 months.

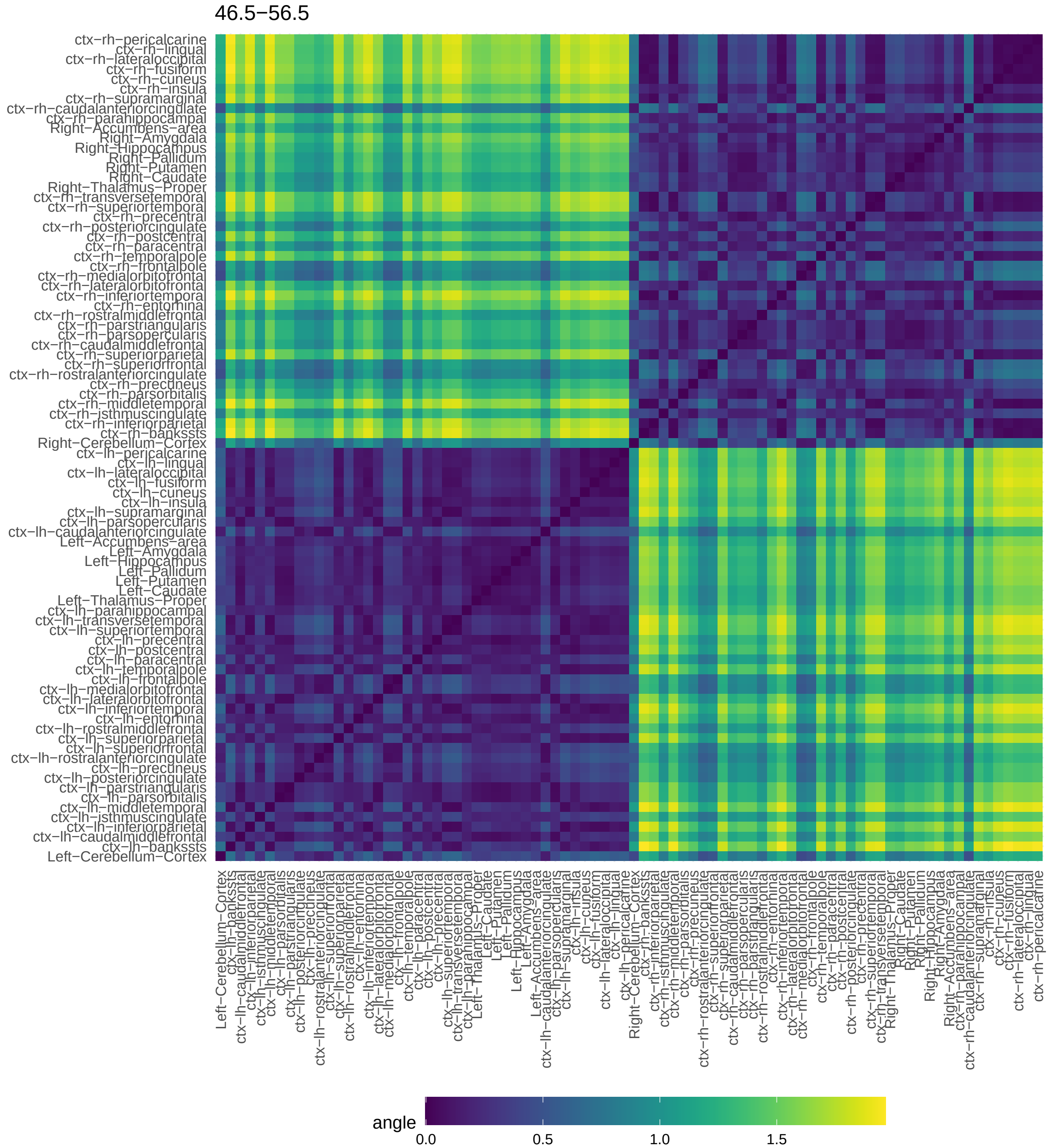
Other plausible statistics from the graphs are the average degrees within and between hemispheres. When analyzing these as stochastic block models (SBM) or degree corrected block models (DCBM), these statistics would roughly correspond to the expected edge weights within and between communities. Both of these statistics are correlated with the target value (figure 5), but a linear model using them (i.e., $age \sim degree_{within} + degree_{between}$) achieves a higher RMSE (12.3 months) while adding another parameter to the model. Similarly, the assortativity of each graph with respect to hemisphere results in a higher RMSE of 156.5 when regressed on age.

We also note that this structure is consistent with the DCBM. The multilayer DCBM was previously studied by Agterberg et al. [1]

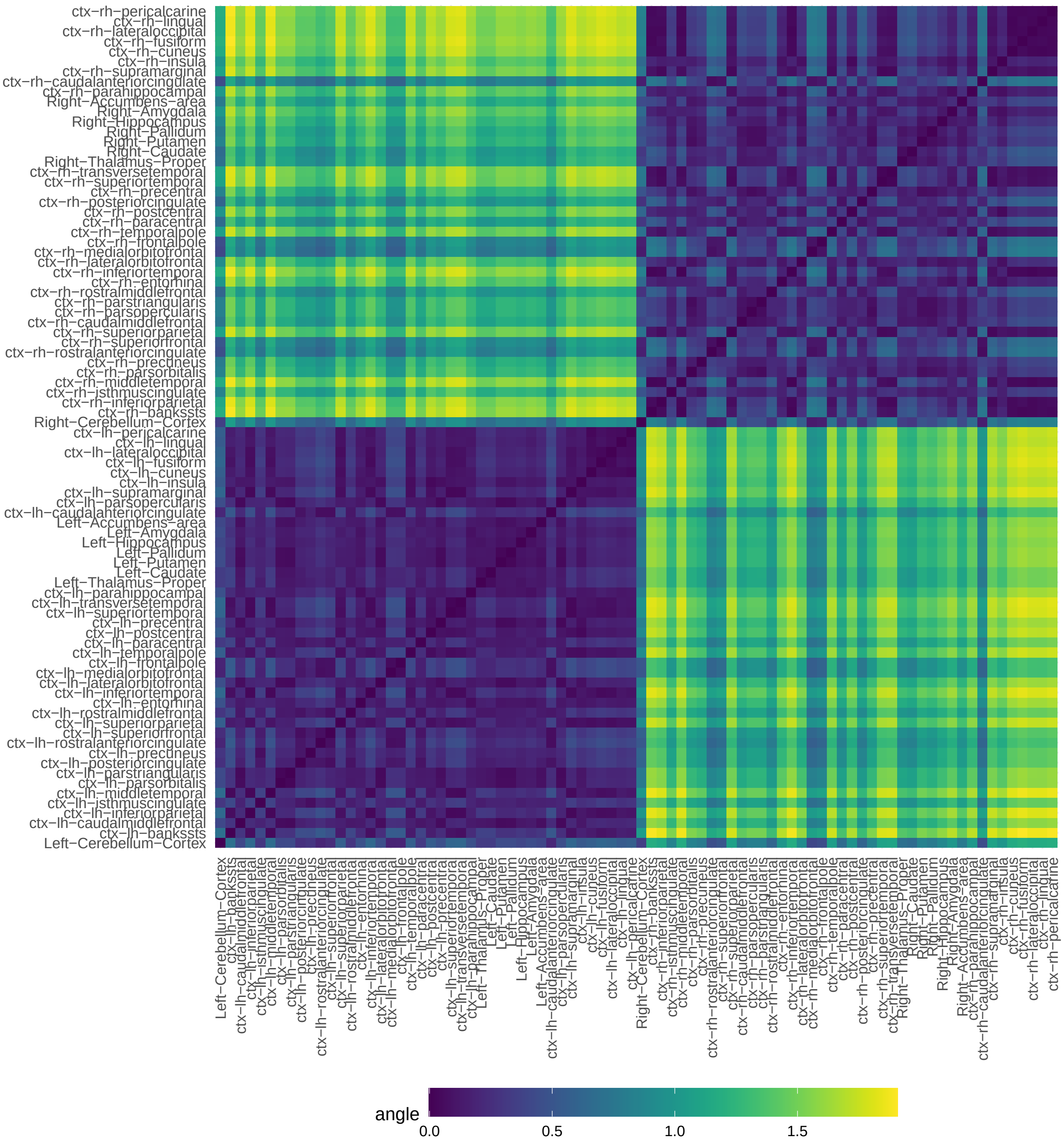

36–46.5
46.5–56.5
56.5–69
69–90
angle
−0.1
0.0
0.1
0.2



36−46.5
ctx−rh−pericalcarine
ctx−rh−lingual
ctx−rh−lateraloccipital
ctx−rh−fusiform
ctx−rh−cuneus
ctx−rh−insula
ctx−rh−supramarginal
ctx−rh−caudalanteriorcingulate
ctx−rh−parahippocampal
Right−Accumbens−area
Right−Amygdala
Right−Hippocampus
Right−Pallidum
Right−Putamen
Right−Caudate
Right−Thalamus−Proper
ctx−rh−transversetemporal
ctx−rh−superiortemporal
ctx−rh−precentral
ctx−rh−posteriorcingulate
ctx−rh−postcentral
ctx−rh−paracentral
ctx−rh−temporalpole
ctx−rh−frontalpole
ctx−rh−medialorbitofrontal
ctx−rh−lateralorbitofrontal
ctx−rh−inferiortemporal
ctx−rh−entorhinal
ctx−rh−rostralmiddlefrontal
ctx−rh−parstriangularis
ctx−rh−parsopercularis
ctx−rh−caudalmiddlefrontal
ctx−rh−superiorparietal
ctx−rh−superiorfrontal
ctx−rh−rostralanteriorcingulate
ctx−rh−precuneus
ctx−rh−parsorbitalis
ctx−rh−middletemporal
ctx−rh−isthmuscingulate
ctx−rh−inferiorparietal
ctx−rh−bankssts
Right−Cerebellum−Cortex
ctx−lh−pericalcarine
ctx−lh−lingual
ctx−lh−lateraloccipital
ctx−lh−fusiform
ctx−lh−cuneus
ctx−lh−insula
ctx−lh−supramarginal
ctx−lh−parsopercularis
ctx−lh−caudalanteriorcingulate
Left−Accumbens−area
Left−Amygdala
Left−Hippocampus
Left−Pallidum
Left−Putamen
Left−Caudate
Left−Thalamus−Proper
ctx−lh−parahippocampal
ctx−lh−transversetemporal
ctx−lh−superiortemporal
ctx−lh−precentral
ctx−lh−postcentral
ctx−lh−paracentral
ctx−lh−temporalpole
ctx−lh−frontalpole
ctx−lh−medialorbitofrontal
ctx−lh−lateralorbitofrontal
ctx−lh−inferiortemporal
ctx−lh−entorhinal
ctx−lh−rostralmiddlefrontal
ctx−lh−superiorparietal
ctx−lh−superiorfrontal
ctx−lh−rostralanteriorcingulate
ctx−lh−precuneus
ctx−lh−posteriorcingulate
ctx−lh−parstriangularis
ctx−lh−parsorbitalis
ctx−lh−middletemporal
ctx−lh−isthmuscingulate
ctx−lh−inferiorparietal
ctx−lh−caudalmiddlefrontal
ctx−lh−bankssts
Left−Cerebellum−Cortex
angle
0.0
0.5
1.0
1.5


46.5−56.5
ctx−rh−pericalcarine
ctx−rh−lingual
ctx−rh−lateraloccipital
ctx−rh−fusiform
ctx−rh−cuneus
ctx−rh−insula
ctx−rh−supramarginal
ctx−rh−caudalanteriorcingulate
ctx−rh−parahippocampal
Right−Accumbens−area
Right−Amygdala
Right−Hippocampus
Right−Pallidum
Right−Putamen
Right−Caudate
Right−Thalamus−Proper
ctx−rh−transversetemporal
ctx−rh−superiortemporal
ctx−rh−precentral
ctx−rh−posteriorcingulate
ctx−rh−postcentral
ctx−rh−paracentral
ctx−rh−temporalpole
ctx−rh−frontalpole
ctx−rh−medialorbitofrontal
ctx−rh−lateralorbitofrontal
ctx−rh−inferiortemporal
ctx−rh−entorhinal
ctx−rh−rostralmiddlefrontal
ctx−rh−parstriangularis
ctx−rh−parsopercularis
ctx−rh−caudalmiddlefrontal
ctx−rh−superiorparietal
ctx−rh−superiorfrontal
ctx−rh−rostralanteriorcingulate
ctx−rh−precuneus
ctx−rh−parsorbitalis
ctx−rh−middletemporal
ctx−rh−isthmuscingulate
ctx−rh−inferiorparietal
ctx−rh−bankssts
Right−Cerebellum−Cortex
ctx−lh−pericalcarine
ctx−lh−lingual
ctx−lh−lateraloccipital
ctx−lh−fusiform
ctx−lh−cuneus
ctx−lh−insula
ctx−lh−supramarginal
ctx−lh−parsopercularis
ctx−lh−caudalanteriorcingulate
Left−Accumbens−area
Left−Amygdala
Left−Hippocampus
Left−Pallidum
Left−Putamen
Left−Caudate
Left−Thalamus−Proper
ctx−lh−parahippocampal
ctx−lh−transversetemporal
ctx−lh−superiortemporal
ctx−lh−precentral
ctx−lh−postcentral
ctx−lh−paracentral
ctx−lh−temporalpole
ctx−lh−frontalpole
ctx−lh−medialorbitofrontal
ctx−lh−lateralorbitofrontal
ctx−lh−inferiortemporal
ctx−lh−entorhinal
ctx−lh−rostralmiddlefrontal
ctx−lh−superiorparietal
ctx−lh−superiorfrontal
ctx−lh−rostralanteriorcingulate
ctx−lh−precuneus
ctx−lh−posteriorcingulate
ctx−lh−parstriangularis
ctx−lh−parsorbitalis
ctx−lh−middletemporal
ctx−lh−isthmuscingulate
ctx−lh−inferiorparietal
ctx−lh−caudalmiddlefrontal
ctx−lh−bankssts
Left−Cerebellum−Cortex
angle
0.0
0.5
1.0
1.5

56.5−69



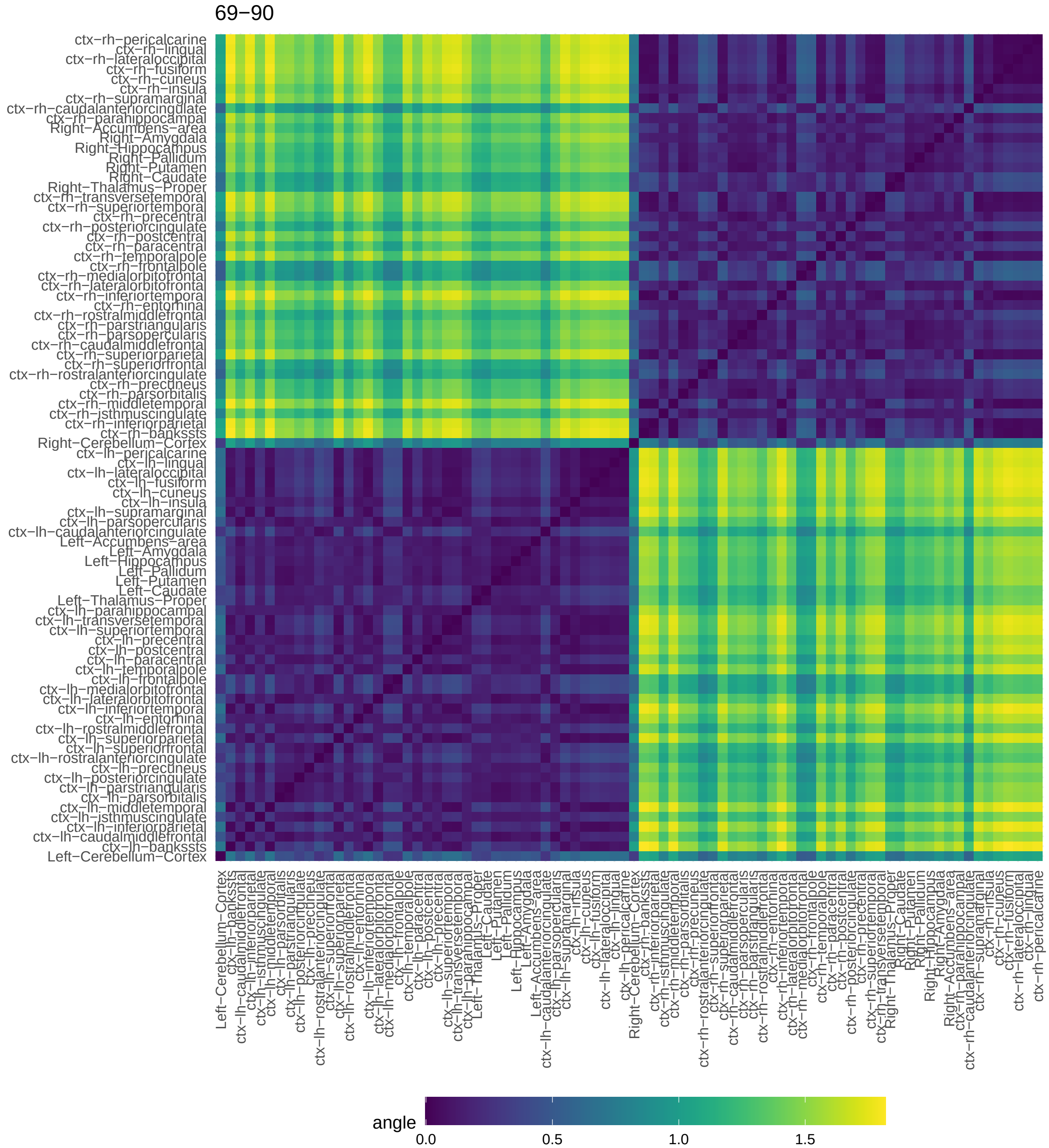

69−90
ctx−rh−pericalcarine
ctx−rh−lingual
ctx−rh−lateraloccipital
ctx−rh−fusiform
ctx−rh−cuneus
ctx−rh−insula
ctx−rh−supramarginal
ctx−rh−caudalanteriorcingulate
ctx−rh−parahippocampal
Right−Accumbens−area
Right−Amygdala
Right−Hippocampus
Right−Pallidum
Right−Putamen
Right−Caudate
Right−Thalamus−Proper
ctx−rh−transversetemporal
ctx−rh−superiortemporal
ctx−rh−precentral
ctx−rh−posteriorcingulate
ctx−rh−postcentral
ctx−rh−paracentral
ctx−rh−temporalpole
ctx−rh−frontalpole
ctx−rh−medialorbitofrontal
ctx−rh−lateralorbitofrontal
ctx−rh−inferiortemporal
ctx−rh−entorhinal
ctx−rh−rostralmiddlefrontal
ctx−rh−parstriangularis
ctx−rh−parsopercularis
ctx−rh−caudalmiddlefrontal
ctx−rh−superiorparietal
ctx−rh−superiorfrontal
ctx−rh−rostralanteriorcingulate
ctx−rh−precuneus
ctx−rh−parsorbitalis
ctx−rh−middletemporal
ctx−rh−isthmuscingulate
ctx−rh−inferiorparietal
ctx−rh−bankssts
Right−Cerebellum−Cortex
ctx−lh−pericalcarine
ctx−lh−lingual
ctx−lh−lateraloccipital
ctx−lh−fusiform
ctx−lh−cuneus
ctx−lh−insula
ctx−lh−supramarginal
ctx−lh−parsopercularis
ctx−lh−caudalanteriorcingulate
Left−Accumbens−area
Left−Amygdala
Left−Hippocampus
Left−Pallidum
Left−Putamen
Left−Caudate
Left−Thalamus−Proper
ctx−lh−parahippocampal
ctx−lh−transversetemporal
ctx−lh−superiortemporal
ctx−lh−precentral
ctx−lh−postcentral
ctx−lh−paracentral
ctx−lh−temporalpole
ctx−lh−frontalpole
ctx−lh−medialorbitofrontal
ctx−lh−lateralorbitofrontal
ctx−lh−inferiortemporal
ctx−lh−entorhinal
ctx−lh−rostralmiddlefrontal
ctx−lh−superiorparietal
ctx−lh−superiorfrontal
ctx−lh−rostralanteriorcingulate
ctx−lh−precuneus
ctx−lh−posteriorcingulate
ctx−lh−parstriangularis
ctx−lh−parsorbitalis
ctx−lh−middletemporal
ctx−lh−isthmuscingulate
ctx−lh−inferiorparietal
ctx−lh−caudalmiddlefrontal
ctx−lh−bankssts
Left−Cerebellum−Cortex
Left−Cerebellum−Cortex
ctx−lh−bankssts
ctx−lh−caudalmiddlefrontal
ctx−lh−inferiorparietal
ctx−lh−isthmuscingulate
ctx−lh−middletemporal
ctx−lh−parsorbitalis
ctx−lh−parstriangularis
ctx−lh−posteriorcingulate
ctx−lh−precuneus
ctx−lh−rostralanteriorcingulate
ctx−lh−superiorfrontal
ctx−lh−superiorparietal
ctx−lh−rostralmiddlefrontal
ctx−lh−entorhinal
ctx−lh−inferiortemporal
ctx−lh−lateralorbitofrontal
ctx−lh−medialorbitofrontal
ctx−lh−frontalpole
ctx−lh−temporalpole
ctx−lh−paracentral
ctx−lh−postcentral
ctx−lh−precentral
ctx−lh−superiortemporal
ctx−lh−transversetemporal
ctx−lh−parahippocampal
Left−Thalamus−Proper
Left−Caudate
Left−Putamen
Left−Pallidum
Left−Hippocampus
Left−Amygdala
Left−Accumbens−area
ctx−lh−caudalanteriorcingulate
ctx−lh−parsopercularis
ctx−lh−supramarginal
ctx−lh−insula
ctx−lh−cuneus
ctx−lh−fusiform
ctx−lh−lateraloccipital
ctx−lh−lingual
ctx−lh−pericalcarine
Right−Cerebellum−Cortex
ctx−rh−bankssts
ctx−rh−inferiorparietal
ctx−rh−isthmuscingulate
ctx−rh−middletemporal
ctx−rh−parsorbitalis
ctx−rh−precuneus
ctx−rh−rostralanteriorcingulate
ctx−rh−superiorfrontal
ctx−rh−superiorparietal
ctx−rh−caudalmiddlefrontal
ctx−rh−parsopercularis
ctx−rh−parstriangularis
ctx−rh−rostralmiddlefrontal
ctx−rh−entorhinal
ctx−rh−inferiortemporal
ctx−rh−lateralorbitofrontal
ctx−rh−medialorbitofrontal
ctx−rh−frontalpole
ctx−rh−temporalpole
ctx−rh−paracentral
ctx−rh−postcentral
ctx−rh−posteriorcingulate
ctx−rh−precentral
ctx−rh−superiortemporal
ctx−rh−transversetemporal
Right−Thalamus−Proper
Right−Caudate
Right−Putamen
Right−Pallidum
Right−Hippocampus
Right−Amygdala
Right−Accumbens−area
ctx−rh−parahippocampal
ctx−rh−caudalanteriorcingulate
ctx−rh−supramarginal
ctx−rh−insula
ctx−rh−cuneus
ctx−rh−fusiform
ctx−rh−lateraloccipital
ctx−rh−lingual
ctx−rh−pericalcarine
angle
0.0
0.5
1.0
1.5

Table 1: Correlation between age and various graph metrics

Metric	Correlation
Angle between hemispheres	0.558
Degree within hemisphere	0.436
Degree between hemispheres	-0.499
Number of triangles in left hemisphere	-0.025
Number of triangles in right hemisphere	0.002
Assortativity w.r.t. hemisphere	0.436
Transitivity	-0.261
Modularity w.r.t. hemisphere	0.434

5 Discussion

Bibliography

[1] Agterberg, J., Lubberts, Z., and Arroyo, J. (2022). Joint spectral clustering in multilayer degree-corrected stochastic blockmodels.

[2] Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., and Qin, Y. (2017). Statistical inference on random dot product graphs: A survey. *J. Mach. Learn. Res.*, 18(1):8393–8484.

[3] Athreya, A., Tang, M., Park, Y., and Priebe, C. E. (2020). On estimation and inference in latent structure random graphs.

[4] Gilbert, E. N. (1959). Random Graphs. *The Annals of Mathematical Statistics*, 30:1141 – 1144.

[5] Jones, A. and Rubin-Delanchy, P. (2021). The multilayer random dot product graph.

[6] Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1).

[7] Koo, J., Tang, M., and Trosset, M. W. (2022). Popularity adjusted block models are generalized random dot product graphs. *Journal of Computational and Graphical Statistics*, 32(1):131–144.

[8] Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1):49–80.

[9] Lyzinski, V., Sussman, D. L., Tang, M., Athreya, A., and Priebe, C. E. (2014). Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electron. J. Statist.*, 8(2):2905–2922.

[10] Nielsen, A. M. and Witten, D. (2018). The multiple random dot product graph model.

[11] Rubin-Delanchy, P., Cape, J., Tang, M., and Priebe, C. E. (2022). A statistical interpretation of spectral embedding: The generalised random dot product graph. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

[12] Sengupta, S. and Chen, Y. (2018). A block model for node popularity in networks with community structure. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 80(2):365–386.

[13] Sussman, D. L., Tang, M., Fishkind, D. E., and Priebe, C. E. (2012). A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107:1119–1128.

[14] Wang, S., Arroyo, J., Vogelstein, J. T., and Priebe, C. E. (2021). Joint embedding of graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1324–1336.

[15] Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. In Bonato, A. and Chung, F. R. K., editors, *Algorithms and Models for the Web-Graph*, pages 138–149, Berlin, Heidelberg. Springer Berlin Heidelberg.