Classification and Regression on Random Dot Product Graphs

Department of YYY, University of XXX

March 22, 2024

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, and multilayer stochastic block model. 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, as well as applications on brain connectivity networks, verify the performance of our 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 [7], 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 [21], 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 [4].

It has been shown [14, 10] that the RDPG (as well as the generalized random dot product graph [14]) can describe a wide range of popular random graph models, such as the Erdös-Rényi model, the stochastic block model (SBM) [11], degree corrected block model (DCBM) [9], and popularity adjusted block model (PABM) [16], 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) [21])**.** 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 $x_1, ..., x_n \overset{\text{iid}}{\sim} F_\theta$. 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. Similarly, if $A \sim \text{RDPG}(F_{\mu,\Sigma})$ and $F_{\mu,\Sigma}$ is the normal distribution with mean vector μ and covariance matrix Σ, then $A \sim \text{RDPG}(F_{W\mu,W\Sigma W^\top})$ is an equivalent RDPG for any orthogonal matrix W.

Definition 2 (Latent structure model (LSM) [4])**.** 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. [4] 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 [3].

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 [17] is a consistent estimator of the latent vectors, up to some unknown orthogonal transformation.

Remark 3 (Sparsity parameter). In many real networks, the degree of each vertex often does not grow proportionally with the size of the network. To account for this, a sparsity factor $\rho_n \in (0, 1]$ is introduced in the edge probabilities, i.e., $P_{ij} \leftarrow \rho_n P_{ij}$, for some sequence $\{\rho_n\}$. Oftentimes the additional constraint of $\lim_{n\to\infty} \rho_n = 0$ is included. For example, a sparse SBM has edge probabilities $P_{ij} = \rho_n \theta_{z_i, z_j}$, for which we use the notation $A \sim \text{SBM}(z, \{\theta_{k\ell}\}_K; \rho_n)$ or $A \sim \text{SBM}(\alpha, \{\theta_{k\ell}\}_K; \rho_n)$, depending on whether we treat the labels as random or fixed. Then the expected degree grows as $O(n\rho_n)$ instead of linearly as $O(n)$. For the sake of unifying the sparse and dense regimes, we also allow for the special case $\rho_n = 1$ and include the sparsity factor throughout, unless otherwise stated. Finally, we also note that while ρ_n limits the rate of growth of the expected degree, our theoretical results still require $n\rho_n$ to diverge to infinity, albeit at a slower rate than $O(n)$. For example, if $\rho_n \propto 1/\sqrt{n}$, then the expected degree for each vertex of the SBM is $O(\sqrt{n})$. In general, for

consistency in estimation and inference, most results on Bernoulli random graphs require $n\rho_n = \omega\big((\log n)^c\big)$ for some $c > 1$. (See Abbe [1], Xie [20], and Rubin-Delanchy et al. [14] for further discussion.)

Definition 3 (Adjacency spectral embedding (ASE) [17])**.** Let $A = V\Lambda V^\top$ be the spectral decomposition of A. Define λ_i as the i^{th} largest eigenvalue of A and v_i by its corresponding eigenvector, and let $V_d = \begin{bmatrix} v_1 & \cdots & v_d \end{bmatrix}$ and $\Lambda_d = \text{diag}(\lambda_1, ..., \lambda_d)$. Then $\hat{X} = V_d|\Lambda|^{1/2}$ is the d-dimensional adjacency spectral embedding of A.

Theorem 1 (Consistency of the ASE [12])**.** *Suppose the sparsity parameter is such that* $n\rho_n = \omega(\log^{4c} n)$ *for some constant* $c > 1$*. Then for some orthogonal transformation* $W \in \mathbb{O}(d)$*,*

$$\max_i \|W\hat{x}_i - x_i\| = O_P\left(\frac{\log^c n}{n^{1/2}}\right),$$

where $x_i^\top$ *and* $\hat{x}_i^\top$ *are the rows of* X *and* $\hat{X}$*, respectively.*

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. [4] 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 [8].

Example 1 (Comparison to the multilayer DCBM [2], multi-RDPG [13], and MREG [19])**.** 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 $\sum_{i:z_i=k} \omega_i^2 = 1$ [9]. 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 [14]. 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. [2] 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 [13], or equivalently, a special case of the multiple random eigen graphs model (MREG), as described by Wang et al. [19]. 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. [4] showed that with some additional information, the ASE of $A^{(\ell)}$ can lead to a consistent estimator for θ_ℓ.

2.2 Main Results

We propose our methodology for classification and regression in three steps:

1. Curve-fitting algorithm to recover the LSM curves from the ASEs.
2. Quasi-maximum likelihood estimation to recover the parameters of the underlying distributions.
3. Classification or regression to determine the relationship between the parameters and the responses.

2.2.1 Estimation of the LSM Curves

For curve-fitting, we will restrict the LSM curves to non-self-intersecting Bezier polynomials [6]. A polynomial curve of degree R in $\mathbb{R}^d$ is uniquely defined by a Bezier polynomial within the space of Bezier polynomials of degree at most R, up to reverse order, although there are infinitely many Bezier polynomials of degree greater than R that can describe the curve [15]. We further restrict the curves to the case $p(0) = 0$, i.e., the curve begins at the origin. This is consistent with well-known models such as the DCBM and PABM when viewed as LSMs or mixtures of LSMs [10, 14], as well as real networks that can adequately be described as LSMs [4]. This constraint also removes the reverse order nonidentifiability. Thus, if R, the degree of the polynomial, can be determined, and the curve begins at the

origin, a collection of $n \geq R$ unique vectors on the curve and not at the origin can determine the unique Bezier polynomial that describes the curve.

A Bezier polynomial of this form, is defined as

$$p(t; b_1, ..., b_R) = \sum_{r=1}^{R} \binom{R}{r} b_r (1-t)^r t^{R-r}, \tag{1}$$

where $b_r \in \mathbb{R}^d$ are Bezier coefficients (control points). In the scenario in which an adjacency matrix A sampled from an LSM with curve p that can be described by a Bezier polynomial, the ASE of A consists of embedding vectors that lie on or near a rotation of the curve. Then if embedding vectors $x_1, ..., x_n \in \mathbb{R}^d$ are such that $x_i = p(t_i) + \epsilon_i$ (for simplicity, redefine p to be the rotated version of the original curve), the squared loss function for the curve fit is of the form

$$L(t_1, ..., t_n; b_1, ..., b_R) = \sum_{i=1}^{n} \big(x_i - p(t_i; b_1, ..., b_R)\big)^2. \tag{2}$$

Letting $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top \in \mathbb{R}^{n \times p}$, $T \in \mathbb{R}^{n \times R}$ such that $T_{ir} = \binom{R}{r}(1-t_i)^{R-r} t_i^r$, $b = \begin{bmatrix} b_1 & \cdots & b_R \end{bmatrix}^\top \in \mathbb{R}^{R \times d}$, the loss function can be rewritten as

$$L(T, b) = \|X - Tb\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm. Then if $t_1, ..., t_n$ are known, the solution to the least squares best fit Bezier coefficients is given by the ordinary least squares estimate,

$$\hat{b} = (T^\top T)^{-1} T^\top X. \tag{4}$$

On the other hand, if the coefficients are known but the timepoints $t_1, ..., t_n$ are not,

this turns into n individual minimization problems for polynomials of degree $2R$. More precisely, $L(t_1, ..., t_n; b_1, ..., b_R) = \sum_{i=1}^{n} L(t_i; b_1, ..., b_R)$ where

$$L(t_i; b_1, ..., b_R) = x_i - \sum_r \binom{R}{r} b_r (1 - t_i)^{R-r} t_i^r. \tag{5}$$

To solve, we find the at most $2R - 1$ roots of its derivative,

$$\dot{L}(t_i; b_1, ..., b_R) = \left(\sum_{r=1}^{R} \binom{R}{r} (-1)^r c_r t_i^r \right) \left(\sum_{r=0}^{R-1} \binom{R-1}{r} (-1)^r c_{r+1} t_i^r \right) = 0, \tag{6}$$

where $c_r = \sum_{s=1}^{r} (-1)^{r-s} \binom{r}{s} b_s$. Since each t_i can be solved separately, this method is highly parallelizable.

For initialization, the ordering of $t_1, ..., t_n$ is determined via a one-dimensional Isomap embedding [18].

Combining equation 4 with the solutions to equation 6 for each $i = 1, ..., n$ provides an alternating coordinate descent algorithm, as outlined in algorithm 1.

Algorithm 1: Procedure for estimating an LSM curve as a Bezier polynomial from an adjacency matrix.

Data: Adjacency matrix A, embedding dimension d, polynomial degree R, Isomap neighborhood parameter λ, stopping criterion ϵ.
Result: Bezier coefficients $\hat{b}$, timepoint values $\hat{t}_1, ..., \hat{t}_n$.

```
Compute X̂ ∈ ℝ^{n×d}, the ASE of A.
Initialize timepoints t̂_1, ..., t̂_n, first via a one-dimensional Isomap embedding of X̂
  using neighborhood parameter λ, then by normalizing the embedding to [0, 1].
repeat
    Fit b̂ by equation 4.
    for i = 1, ..., n do
        Fit t̂_i by equation by finding the roots of equation 6 and choosing the roots
          that minimizes equation 5.
    end
until the change in equation 5 is less than ε.
```

In the case of a mixture of LSMs (as in remark 2), algorithm 1 is modified by fitting Bezier polynomials for each vertex label (if the labels are known) or by estimating the labels via an alternating coordinate descent algorithm outlined in algorithm 2.

Algorithm 2: Procedure for estimating multiple LSM curves as a Bezier polynomial from an adjacency matrix.

Data: Adjacency matrix A, embedding dimension d, polynomial degree R, number of latent structures K, Isomap neighborhood parameter λ, stopping criterion ϵ.

Result: Bezier coefficients $\hat{b}$, timepoint values $\hat{t}_1, ..., \hat{t}_n$, curve memberships $\hat{z}_1, ..., \hat{z}_n$.

Compute $\hat{X} \in \mathbb{R}^{n \times d}$, the ASE of A.
Initialize $\hat{z}_1, ..., \hat{z}_n \in \{1, ..., K\}$, the membership of each vertex to each latent curve.
for $k = 1, ..., K$ **do**
    Subset the rows of $\hat{X}$ by $z_i = k$ to obtain $\hat{x}_{k_1}, ..., \hat{x}_{k_{n_k}}$, where n_k is the number of vertices with label k.
    Initialize timepoints $\hat{t}_{k_1}, ..., \hat{t}_{k_{n_k}}$, first via a one-dimensional Isomap embedding of $\hat{x}_{k_1}, ..., \hat{x}_{k_{n_k}}$ using neighborhood parameter λ, then by normalizing the embedding to $[0, 1]$.
end
repeat
    for $k = 1, ..., K$ **do**
        Fit $\hat{b}^{(k)}$ by equation 4, using the row subsets of T and X such that $z_i = k$.
        for $i = 1, ..., n_k$ **do**
            Fit $\hat{t}_{k_i}$ by equation by finding the roots of equation 6 and choosing the roots that minimizes equation 5.
        end
    end
    for $k = 1, ..., n$ **do**
        Reassign z_i based on the latent curve closest to $\hat{x}_i$.
    end
until *the change in equation 5 is less than* ϵ.

Lemma 1. *Suppose* $A \sim \text{LSM}(p, F_\theta)$ *such that* $p : [0, 1] \to \mathbb{R}^d$ *is a Bezier polynomial of degree* R *with coefficients* b*, and* F_θ *is a probability distribution with probability density function* $f_\theta(t)$ *that is absolutely continuous on support* $[0, 1]$ *and* $\min_t f_\theta(t) > 0 \ \forall t \in [0, 1]$*. Let* $\hat{T}$ *and* $\hat{b}$ *be the minimizers of the loss function in equation 5 from* A*, and let* $\tilde{X} = \hat{T}\hat{b}$*, the estimated vectors along the fitted Bezier curve. Then as* $n \to \infty$*,* $\|X - \tilde{X}W\|_{2,\infty} \xrightarrow{p} 0$

for some $W \in \mathbb{O}(d)$.

Proof. Let X be the true latent positions of the LSM and $\hat{X}$ be the ASE of A drawn from X. Then $X = Tb$, and $\hat{X} = \hat{T}\hat{b} + \hat{E}$, where $\hat{E}$ is the matrix of distances from the estimated positions along the fitted Bezier curves and the embedding vectors. Then

$$\begin{aligned}
\|X - \tilde{X}W\| &= \|Tb - \hat{T}\hat{b}W\|_{2,\infty} \\
&= \|Tb - (\hat{T}\hat{b} + \hat{E} - \hat{E})W\|_{2,\infty} \\
&= \|Tb - (\hat{T}\hat{b} + \hat{E})W + \hat{E}W\|_{2,\infty} \\
&\leq \|X - \hat{X}W\|_{2,\infty} + \|\hat{E}W\|_{2,\infty}.
\end{aligned}$$

The first term $\|X - \hat{X}W\|_{2,\infty}$ converges to 0 in probability [12]. Since the ASE is consistent, the embedding vectors will converge in probability to a Bezier curve, so $\|\hat{E}W\|_{2,\infty} \xrightarrow{p} 0$. □

Lemma 2. *Suppose* $A \sim \text{LSM}(p, F_\theta)$ *such that* $p : [0,1] \to \mathbb{R}^d$ *is a Bezier polynomial of degree* R *with coefficients* b*, and* F_θ *is a probability distribution with probability density function* $f_\theta(t)$ *that is absolutely continuous on support* $[0,1]$ *and* $\min_t f_\theta(t) > 0 \ \forall t \in [0,1]$. *Let* $\hat{T}$ *and* $\hat{b}$ *be the minimizers of the loss function in equation 5 from* A. *Then* $\|T - \hat{T}\|_{2,\infty} \xrightarrow{p} 0$ *as if and only if* $\|b - \hat{b}W\|_F \xrightarrow{p} 0$.

Theorem 2 (Consistency of algorithm 1)**.** *Suppose* $A \sim \text{LSM}(p, F_\theta)$ *such that* $p : [0,1] \to \mathbb{R}^d$ *is a Bezier polynomial of degree* R *with finite coefficients* b*, and* F_θ *is a probability distribution with probability density function* $f_\theta(t)$ *that is absolutely continuous on support* $[0,1]$ *and* $\min_t f_\theta(t) > 0 \ \forall t \in [0,1]$. *Let* $\hat{T}$ *and* $\hat{b}$ *be the minimizers of the loss function in equation 5 from* A. *Then as* $n \to \infty$, $\|T - \hat{T}\|_{2,\infty} \xrightarrow{p} 0$ *where* T *is defined as in equation 3, and* $\|b - \hat{b}W\|_F \xrightarrow{p} 0$ *for some* $W \in \mathbb{O}(d)$.

Proof. □

2.2.2 Recovery of the Underlying Distributions

Algorithm 3: Procedure for estimating the underlying distribution of a Bezier LSM curve from an adjacency matrix.

Data: Adjacency matrix A, embedding dimension d, probability distribution family F, polynomial degree R, Isomap neighborhood parameter λ, stopping criterion ϵ.

Result: Bezier coefficients $\hat{b}$, timepoint values $\hat{t}_1, ..., \hat{t}_n$.

Compute $\hat{X} \in \mathbb{R}^{n \times d}$, the d-dimensional ASE of A.

Initialize timepoints $\hat{t}_1, ..., \hat{t}_n$, first via a one-dimensional Isomap embedding of $\hat{X}$ using neighborhood parameter λ, then by normalizing the embedding to $[0, 1]$.

repeat

  Fit $\hat{b}$ by equation 4.

  for $i = 1, ..., n$ **do**

    Fit $\hat{t}_i$ by equation by finding the roots of equation 6 and choosing the roots that minimizes equation 5.

  end

until *the change in equation 5 is less than* ϵ.

Estimate $\hat{\theta}$ from $\hat{t}_1, ..., \hat{t}_n$ via maximum likelihood estimation.

Theorem 3 (Consistency of algorithm 3)**.** *Suppose* $A \sim \text{LSM}(p, F_\theta)$ *such that* $p : [0, 1] \to \mathbb{R}^d$ *is a Bezier polynomial of degree* R *with coefficients* b*, and* F_θ *is a probability distribution with probability density function* $f_\theta(t)$ *that is absolutely continuous on support* $[0, 1]$ *and* $\min_t f_\theta(t) > 0 \ \forall t \in [0, 1]$*. Let* $\hat{t}_1, ..., \hat{t}_n$ *and* $\hat{b}$ *be the minimizers of the loss function in equation 5 from* A*, and let* $\tilde{theta}$ *be the maximum likelihood estimator for* θ *using* $\hat{t}_1, ..., \hat{t}_n$*. Then* $\tilde{\theta} \xrightarrow{p} \theta$*.*

Proof. By theorem 2, □

2.2.3 Classification and Regression

Algorithm 4: Procedure for fitting a classification or regression model for an MLSM.

Data: Adjacency matrices $A^{(1)}, ..., A^{(L)}$, response variables $y_1, ..., y_L$, embedding dimension d, probability distribution family F, polynomial degree R, Isomap neighborhood parameters $\lambda_1, ..., \lambda_L$, stopping criterion ϵ.

Result: Bezier coefficients $\hat{b}$, timepoint values $\hat{t}_1, ..., \hat{t}_n$.

for $\ell = \{1, ..., L\}$ **do**
  Compute $\hat{X}^{(\ell)} \in \mathbb{R}^{n_\ell \times d}$, the d-dimensional ASE of $A^{(\ell)} \in \mathbb{R}^{n_\ell \times n_\ell}$.
  Initialize timepoints $\hat{t}_1, ..., \hat{t}_{n_\ell}$, first via a one-dimensional Isomap embedding of $\hat{X}^{(\ell)}$ using neighborhood parameter λ_ℓ, then by normalizing the embedding to $[0, 1]$.
  repeat
    Fit $\hat{b}$ by equation 4.
    for $i = 1, ..., n_\ell$ **do**
      Fit $\hat{t}_i$ by equation by finding the roots of equation 6 and choosing the roots that minimizes equation 5.
    end
  until *the change in equation 5 is less than* ϵ.
  Estimate $\hat{\theta}_\ell$ from $\hat{t}_1, ..., \hat{t}_{n_\ell}$ via maximum likelihood estimation.
end
Fit a model for predicting $y_1, ..., y_L$ using the estimated parameters $\theta_1, ..., \theta_L$ (e.g., via linear regression).

Theorem 4.

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 $\alpha_1 = 1$, $\beta_1 = 2$, $\alpha_2 = 2$, and $\beta_2 = 1$.
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.

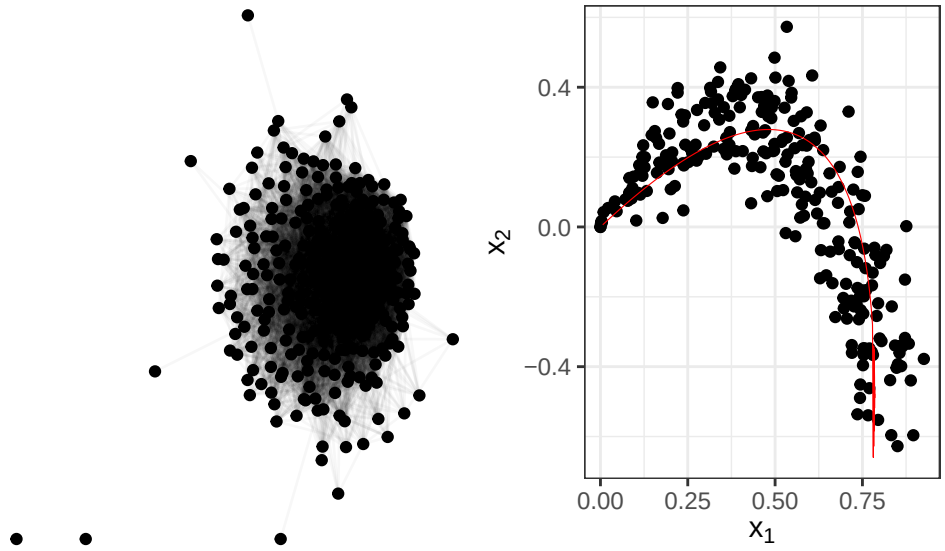


Figure 1: One simulated graph (left) and its ASE (right). The red curve is the fitted quadratic Bezier curve on the ASE.

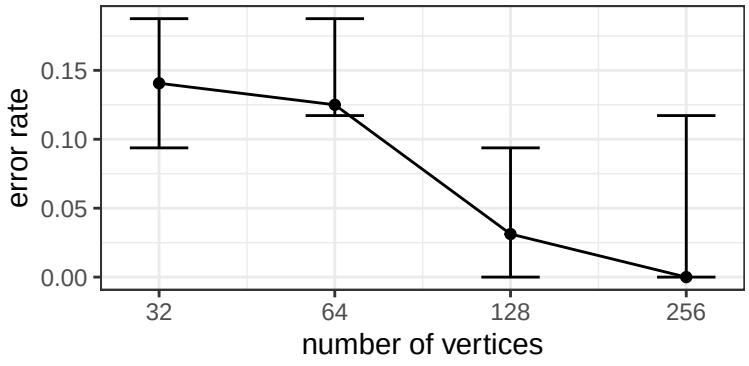


Figure 2: Median classification error rate and its IQR vs. number of vertices in each graph.

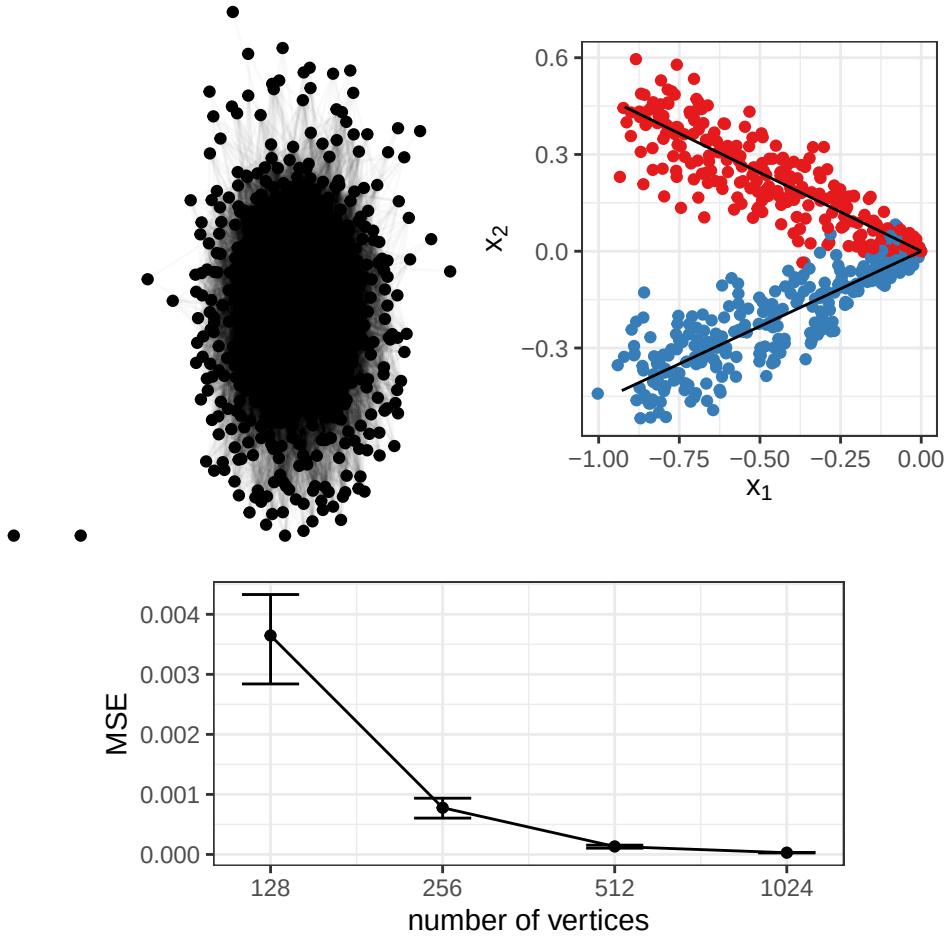
3.2 Regression

In the first regression simulation,

In the next regression simulation, we applied

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 Drosophila Connectome

In the second example, we analyzed the larval *Drosophila* mushroom body connectome [5], which has been studied as a GRDPG by Athreya et al. [4]. This dataset consists of two graphs representing two networks of neurons, one for each hemisphere of the *Drosophila* brain. In these graphs, each vertex is a neuron, and the labels correspond to one of four neuron types (Kenyon Cells, Input Neurons, Output Neurons, and Projection Neurons). The number of neurons in each hemisphere is not equal (209 in the left hemisphere and 213 in the right hemisphere). The resulting graphs are illustrated in figure 3.

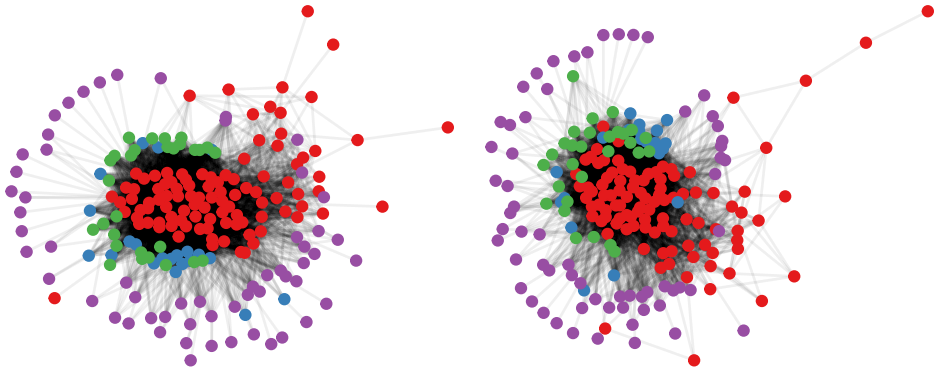


Figure 3: Graphs of the Drosophila connectomes. The left and right are of the left and right hemispheres, respectively. Each vertex represents a neuron, which are labeled by neuron type.

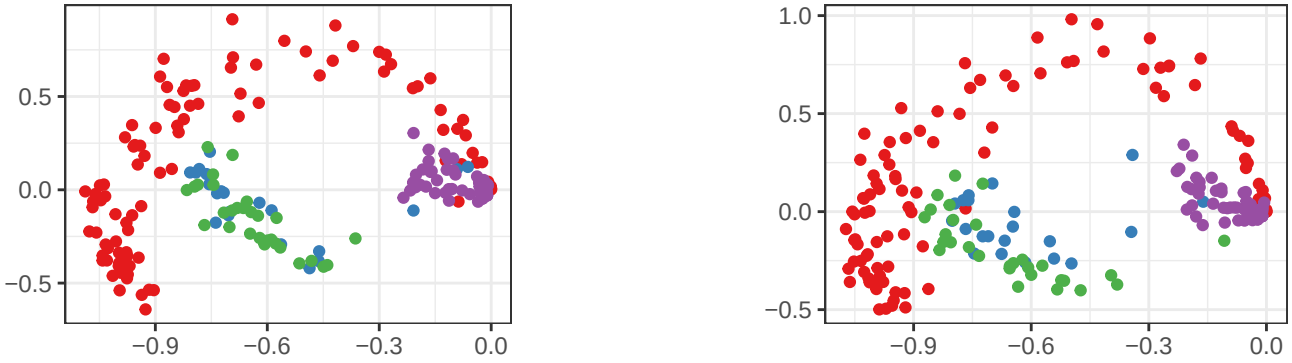


Figure 4: ASEs of the Drosophila connectome graphs. The embedding vectors are labeled by neuron type.

When analyzed as a GRDPG, the ASE of each hemisphere suggests that nodes of each type falls along a curve in the latent space (figure 4). In our analysis, we set the embedding dimension to $d = 4$, with 3 assortative dimensions and 1 disassortative dimension, and the figure only shows the first two assortative dimensions. This matches observations by Athreya et al. [4]. We fit three latent structure Bezier curves, one for Kenyon Cells, one for Input and Output Neurons, and one for Projection Neurons, to the embedding for each hemisphere. Then we fit a Beta distribution to to the timepoints along each curve

and extracted the two Beta parameter estimates (via likelihood maximization) for each curve. If these Beta parameters are informative, we would expect the parameters for each hemisphere to match by neuron type, which is what we observe in these data (fig 5).

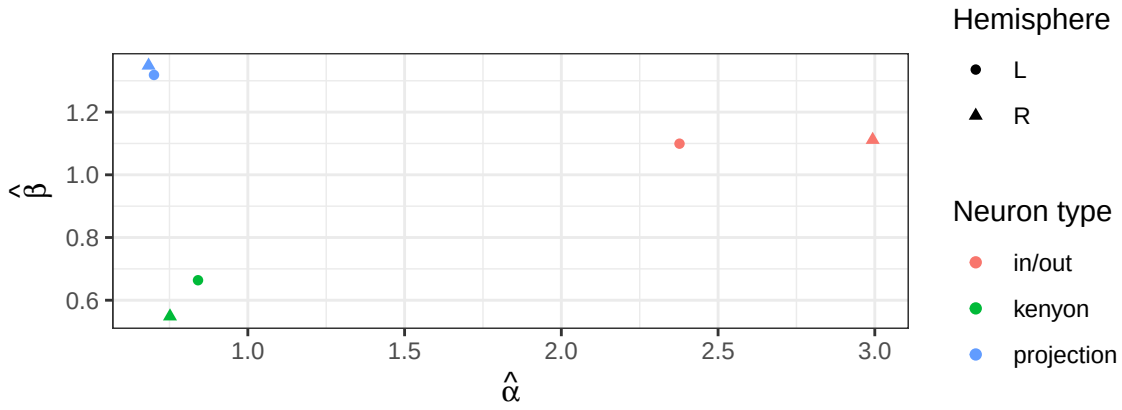


Figure 5: Beta parameter estimates for each curve and hemisphere.

4.2 Human Connectome Project Aging Study

In the second 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 $A_{ij} \overset{\text{ind}}{\sim} \text{Poisson}(\Theta_{ij})$, where $\Theta \in \mathbb{R}_+^{n \times n}$ is a symmetric matrix of Poisson parameters.

In this dataset, there are $L = 516$ graphs (corresponding to individual subjects), each with $n_\ell = n = 84$ vertices (corresponding to brain regions). Analyzing these graphs as RDPGs reveals that the DCBM is a good candidate for these data (figure 6).

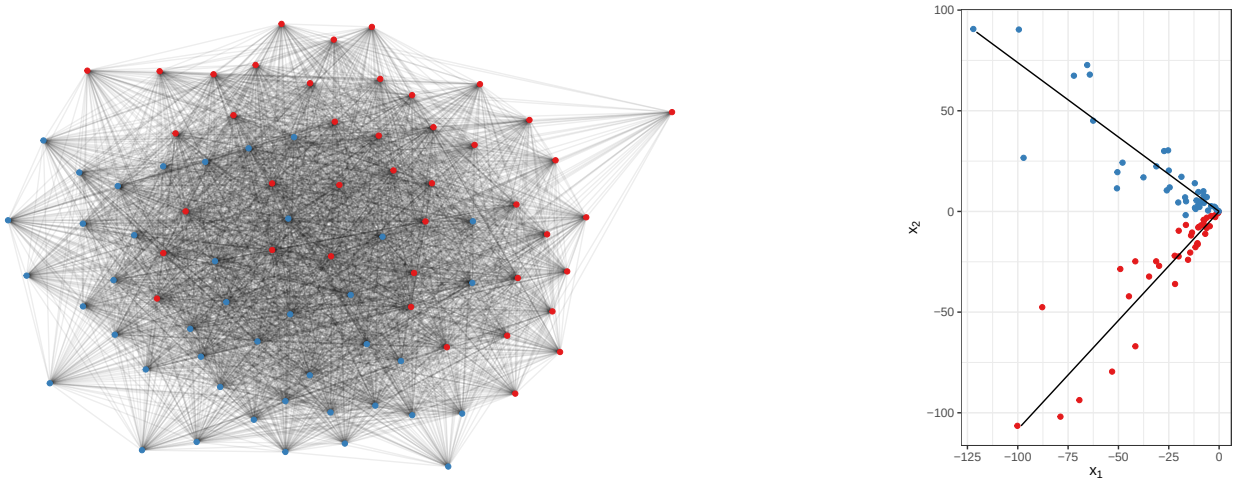


Figure 6: One graph from the HCP dataset (left) and its ASE (right). In the ASE, the lines are fitted via K-curves clustering using degree $= 1$. The outputted clusters correspond exactly to the left (red) and right (blue) hemispheres.

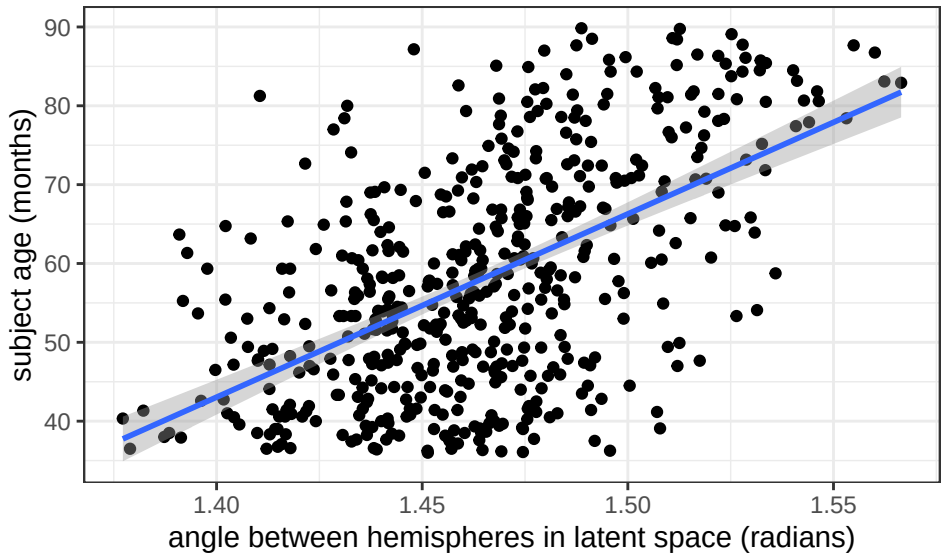


Figure 7: Scatterplot between the subject age (in months) vs. the fitted angle between hemispheres of the brain in the latent space.

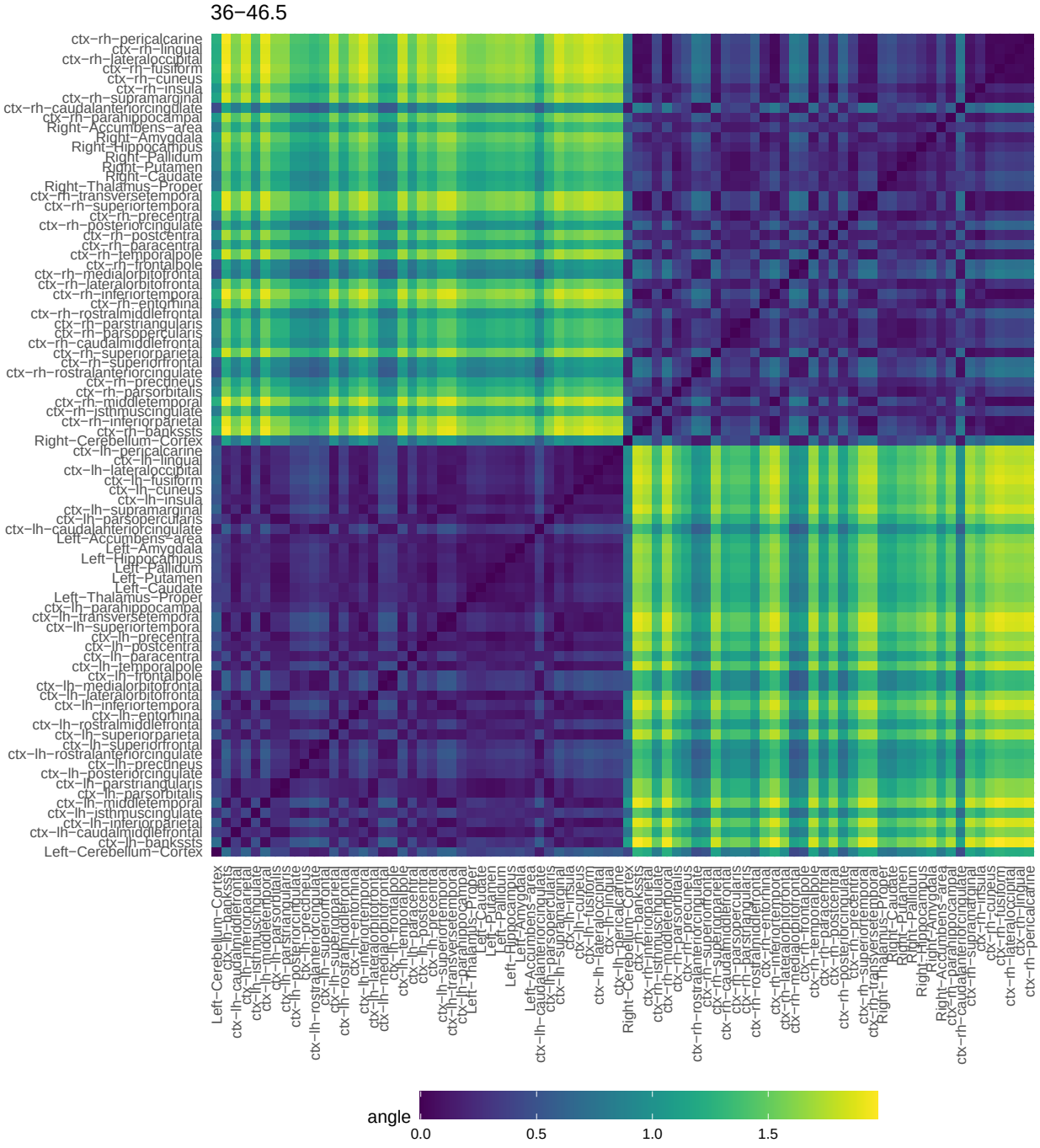
The ASE suggests a latent structure comprised of two Bezier curves of degree 1 (i.e., lines), one for each hemisphere of the brain, that meet at the origin. One possible parameter when analyzing these data as a MLSM is the angle between the two lines. The estimated

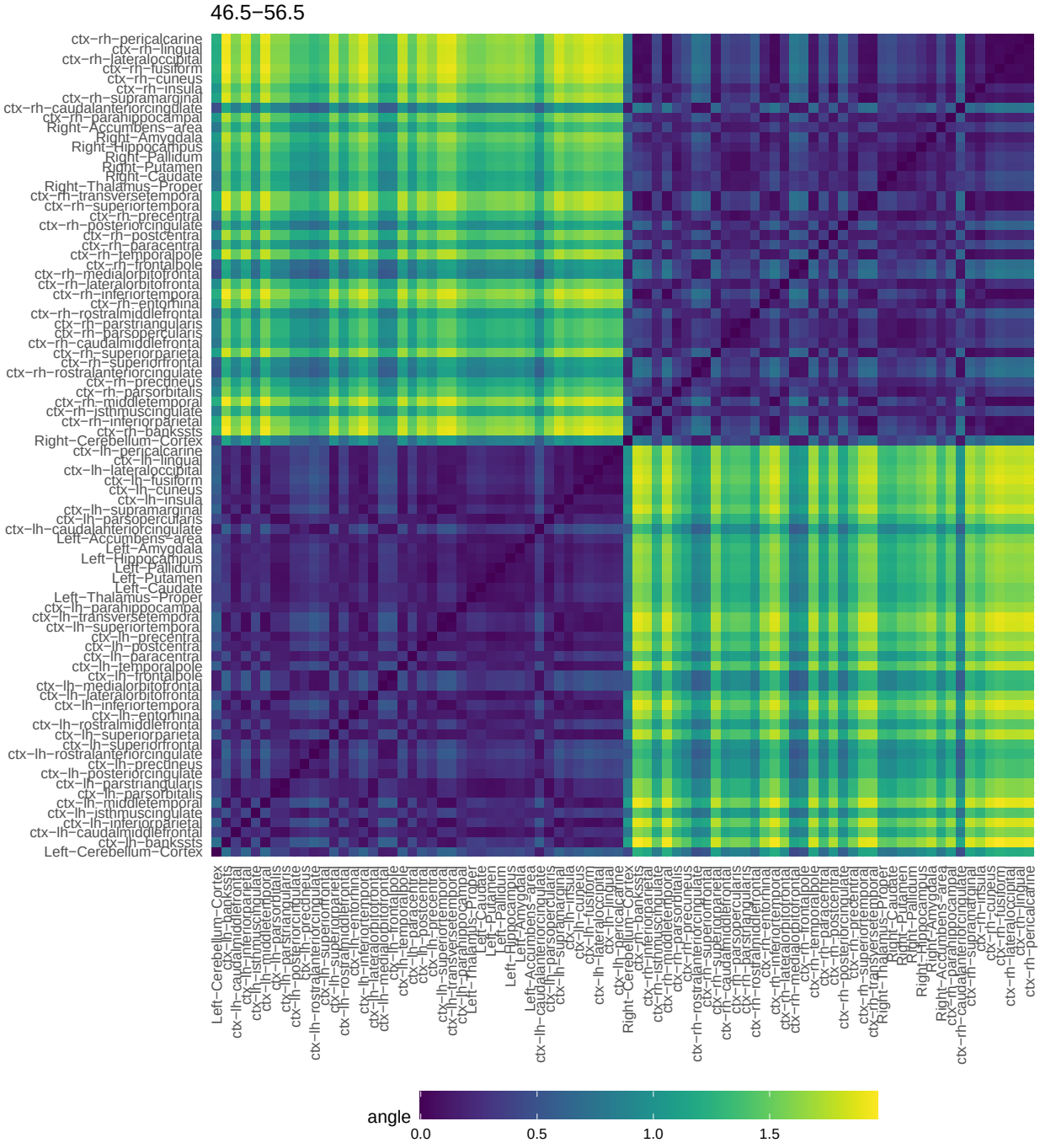
angles were observed to correlate with the subject's age, with wider angles corresponding to older subjects (figure 7). A linear regression setting aside half of the brain connectivity graphs as test data achieves an RMSE of 11.889 months.

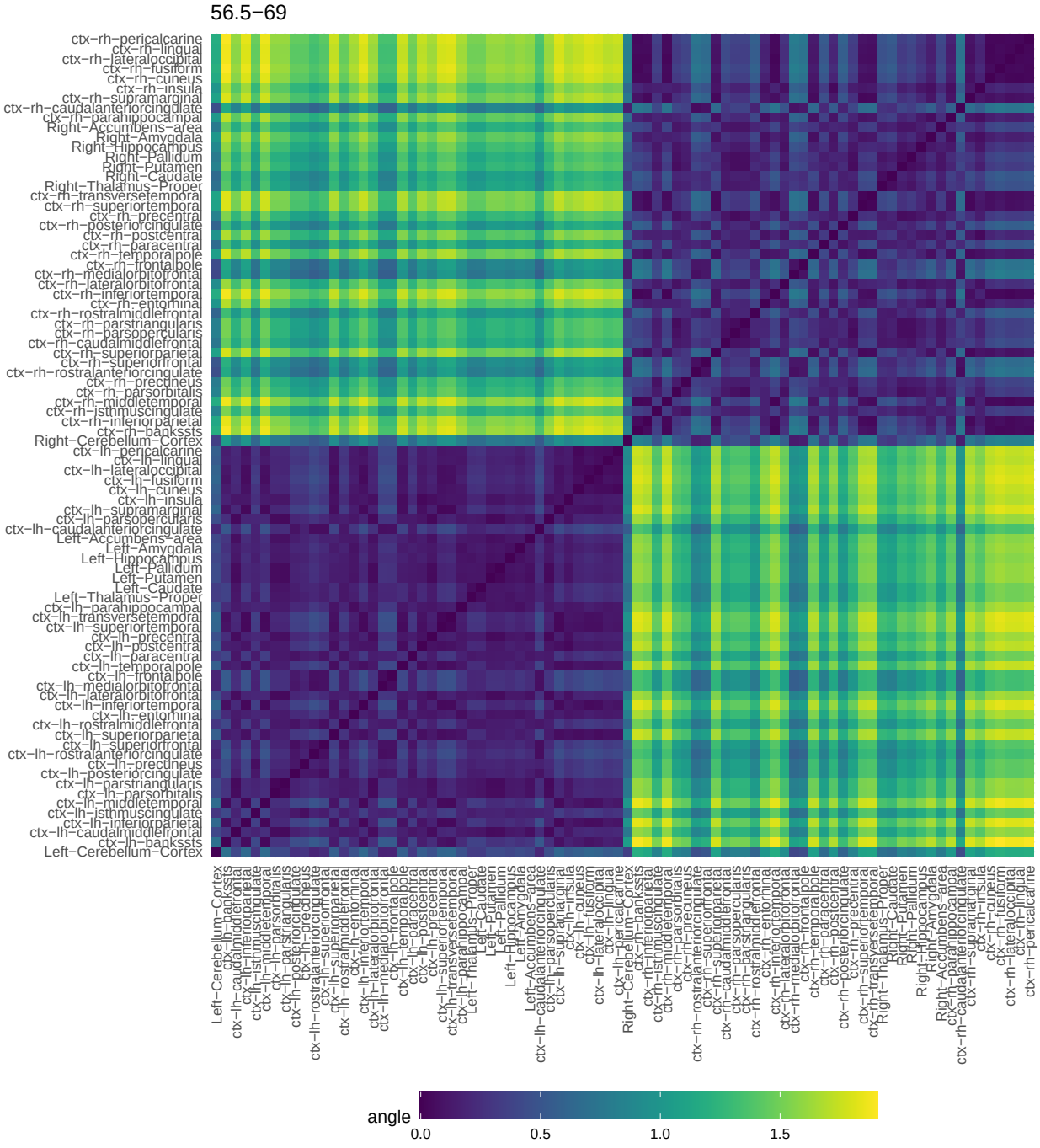
To compare this parameter as a covariate for age against other network statistics, we analyzed these data as a multilayer DCBM, first studied by [2], who proposed the degree corrected multiple adjacency spectral embedding (DC-MASE) algorithm. Since these graphs come with hemisphere labels, we did not apply DC-MASE for community detection but instead used the estimators for the three edge connectivity parameters, B_{LL}, B_{RR}, and B_{LR}. A linear model trained on these parameter estimates achieves a higher RMSE of 12.889 months, despite using three covariates instead of one. In addition, since the angles between the latent structures under the MLSM depend primarily on the shape of the latent structures rather than the exact community memberships, it does not depend on recovery of the original community labels. Ultimately, these two methods are estimators for transformations of the same parameters, and while we cannot determine how close these estimates are on the "true" edge connectivity parameters since they are unknown, we observe that the MLSM estimate is a better linear predictor than the rest.

Other graph statistics were extracted for each brain network, and their correlations to age are reported in table **??**. Note that aside from the angle between the latent structures under the MLSM model, these statistics assume that it is known which vertex belongs to which hemisphere.


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
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

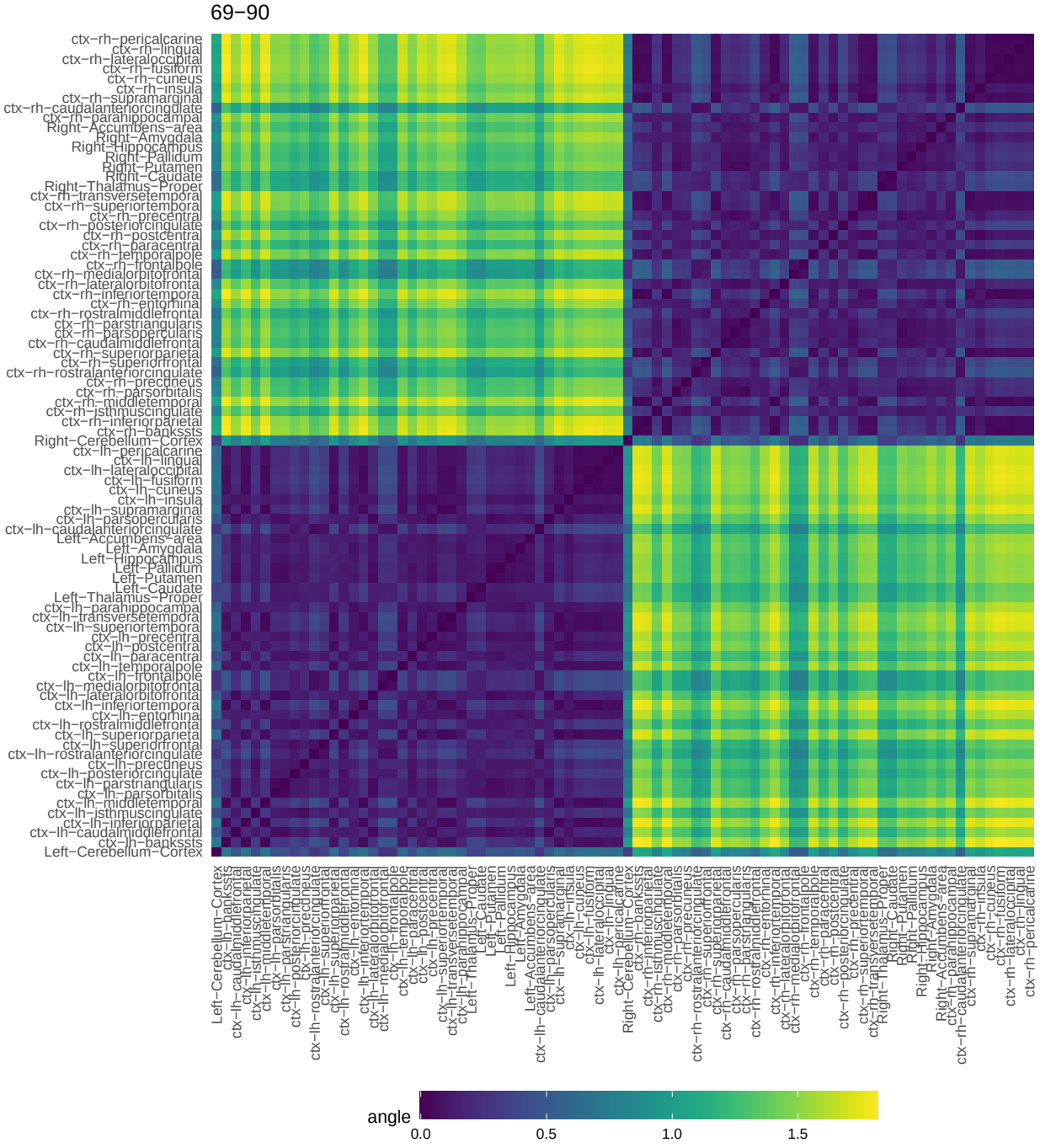

69−90
ctx−rh−pericalcarine
ctx−rh−lingual
ctx−rh−lateraloccipital
ctx−rh−fusiform
ctx−rh−cuneus
ctx−rh−insula
ctx−rh−supramarginal
ctx−rh−caudalanteriorcingulate
ctx−rh−parahippocampal
Right−Accumbens−area
Right−Amygdala
Right−Hippocampus
Right−Pallidum
Right−Putamen
Right−Caudate
Right−Thalamus−Proper
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
angle
0.0
0.5
1.0
1.5

Table 1: Correlation between age and various graph metrics

Metric	Correlation	95% conf. int.
Angle between hemispheres	0.558	(0.496, 0.615)
Edge connectivity within left hemisphere	0.241	(0.158, 0.321)
Edge connectivity within right hemisphere	0.243	(0.16, 0.322)
Edge connectivity between hemispheres	-0.436	(-0.503, -0.363)
Degree within hemisphere	0.436	(0.363, 0.503)
Degree between hemispheres	-0.499	(-0.561, -0.431)
Number of triangles in left hemisphere	-0.025	(-0.111, 0.062)
Number of triangles in right hemisphere	0.002	(-0.084, 0.088)
Assortativity w.r.t. hemisphere	0.436	(0.363, 0.503)
Modularity w.r.t. hemisphere	0.434	(0.362, 0.502)
Joint Embedding	0.556	(0.493, 0.613)

Next, we considered each hemisphere separately and fit a Beta distribution to the embedding of each half-network. The scatterplot of the fitted α and β parameters suggests that there are two clusters of brain networks (figure 8). Greater α values correspond to more embedding vectors farther away from the origin, corresponding to more nodes of higher degree, which in turn correspond to more brain regions with a greater number of connections. Conversely, greater β values correspond to more embedding vectors closer to the origin, which in turn correspond to more brain regions with fewer connections. This may suggest that there are underlying groups, one of which tends to have higher connectivity

in the left hemisphere, and the other having higher connectivity in the right hemisphere.

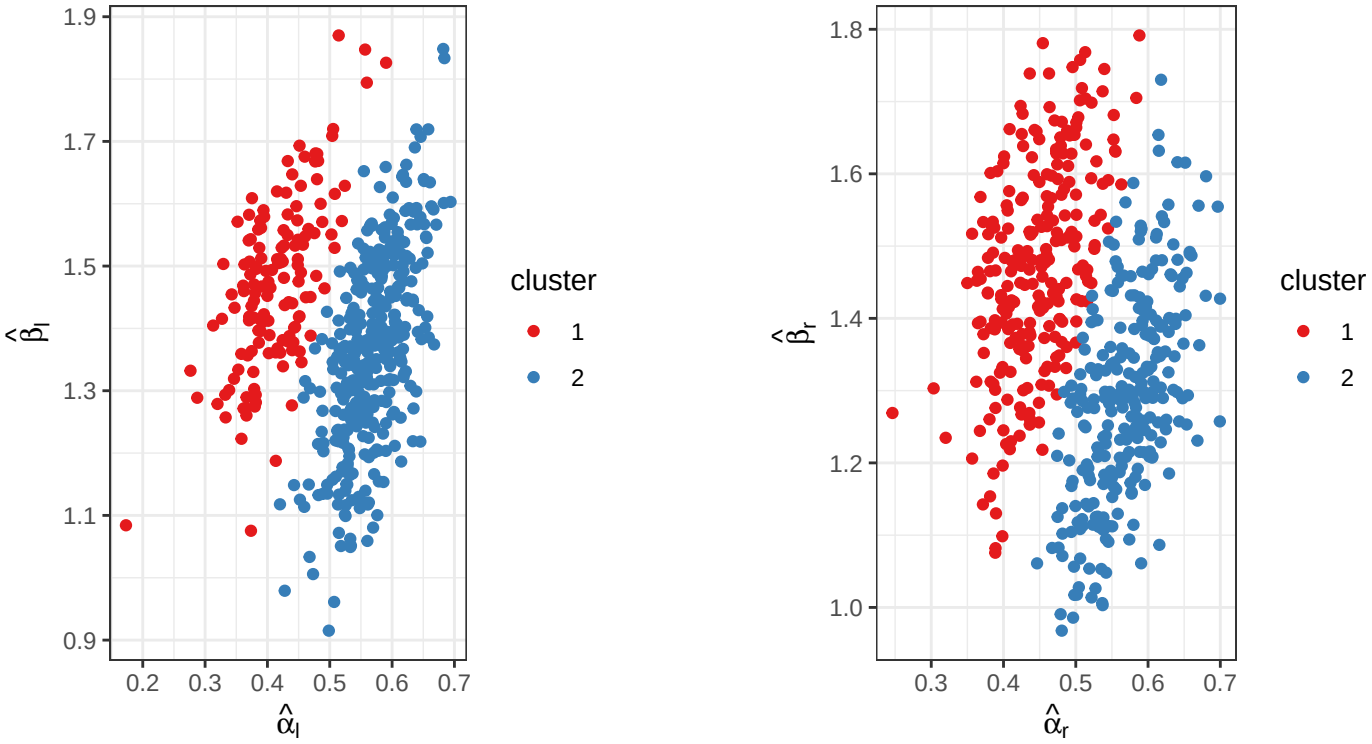


Figure 8: Scatterplot of fitted parameters of the Beta distribution. The left image are the parameters fitted on the embedding of the left hemisphere, and the right image are the parameters fitted on the embedding of the right hemisphere.

5 Discussion

Bibliography

[1] Abbe, E. (2018). Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86.

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

[3] 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.

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

[5] Eichler, K., Li, F., Litwin-Kumar, A., Park, Y., Andrade, I., Schneider-Mizell, C. M., Saumweber, T., Huser, A., Eschbach, C., Gerber, B., Fetter, R. D., Truman, J. W., Priebe, C. E., Abbott, L. F., Thum, A. S., Zlatic, M., and Cardona, A. (2017). The complete connectome of a learning and memory center in an insect brain. *bioRxiv*.

[6] Gallier, J. (1999). *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

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

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

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

[10] 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.

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

[12] 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.

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

[14] 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)*.

[15] Sánchez-Reyes, J. (2022). The uniqueness of the rational bézier polygon is unique. *Computer Aided Geometric Design*, 96:102118.

[16] 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.

[17] 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.

[18] Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

[19] 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.

[20] Xie, F. (2021). Entrywise limit theorems of eigenvectors for signal-plus-noise matrix models with weak signals.

[21] 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.