

Classification and Regression on Random Dot Product Graphs

Department of YYY, University of XXX

November 15, 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 multiple random graphs that share common properties or structures across graphs have been introduced, such as the multilayer RDPG, multiple RPDG, and common subspace independent edge model. In this work, we use these joint random graph models in the context of graph 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. One such approach in particular is the independent edge approach. If we sum up the network in an adjacency matrix $A \in \mathbb{R}^{n \times n}$, then this model draws 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., $A_{ij} \stackrel{\text{ind}}{\sim} F_{\theta_{ij}}$. The classical example of this is the Erdős-Rényi model [8], in which every edge is drawn from the same distribution, typically a Bernoulli distribution, using the same parameter, i.e., $A_{ij} \stackrel{\text{iid}}{\sim} \text{Bernoulli}(p)$. 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, $A_{ij} \stackrel{\text{ind}}{\sim} \text{Bernoulli}(P_{ij})$. 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 the parameter matrix P is unconstrained, the inference problem is overparameterized. On the other hand, the classical Erdős-Rényi model is often too restrictive to describe real, observed networks. Various approaches have been taken to restrict P , such as by setting the matrix rank of P to a small value. One such family of random graph models is the random dot product graph (RDPG), first proposed by Young and Scheinerman [23], 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 [5].

It has been shown [11, 16] that the RDPG (as well as the generalized random dot product graph, or GRDPG, [16]) can describe a wide range of popular random graph models, such as the Erdős-Rényi model, the stochastic block model (SBM) [12], degree corrected block model (DCBM) [10], and popularity adjusted block model (PABM) [18]. In these examples, the (G)RPDG representation of these networks allows for inference and analysis in more familiar Euclidean space rather than on graphs, and the geometry of the configuration of latent vectors correspond to the parameters of these graph models. More specifically, when these network models are described as (G)RDPGs, each latent vector lies on a subspace. Athreya et al. [5] and Passino and Heard [15] extended this to the latent structure model (LSM) and latent structure block model (LSBM) to include latent configurations that consist of nonlinear manifolds.

In this paper, we apply these highly structured random dot product graph models to supervised learning problems in which there are multiple graphs, G_1, \dots, G_L , each with a response $y^{(1)}, \dots, y^{(L)}$. Each graph, represented by adjacency matrix $A^{(\ell)}$, is presumed to be drawn from a RDPG with parameter set θ_l , and in turn, the response y_l is drawn from a univariate distribution with parameter θ_l . Since the latent vectors of each $A^{(\ell)}$ are presumed to be distributed on a structured subset of the latent space, e.g., a manifold, we call this the multiple latent structure model (MLSM). Much previous work has been done

on multilayer networks, such as the multilayer degree corrected block model of Agterberg et al. [2], multiple random eigen graphs of Wang et al. [21], and the common subspace independent edge graphs of Arroyo et al. [3]. In each of these models, the main inference problem is to identify parameters that are shared across multiple graphs. Our contribution is in applying similar methods to identify parameters specific to each graph, which are in turn used as features for supervised learning. This further allows us to model graphs with differing vertex sets.

The remainder of this paper is organized as follows: Section 2 builds on the latent structure model to introduce the multiple latent structure model and proposes manifold learning algorithms for estimation. Sections 3 and 4 provide simulation studies and real data analyses. Section 5 concludes. Proofs of theorems are provided in the Appendix.

2 Setup and Methodology

2.1 Notation and Scope

Let $G^{(1)}, \dots, G^{(L)}$ be a collection of undirected graphs where each $G^{(\ell)} = (V^{(\ell)}, E^{(\ell)})$. In this setup, each graph can have its own vertex and edge sets. These graphs are represented by adjacency matrices $A^{(1)}, \dots, A^{(L)}$, where each $A^{(\ell)} \in \mathbb{R}^{n_l \times n_l}$ is a symmetric and hollow matrix such that $A_{ij}^{(\ell)}$ represents the edge weight between vertices i and j , and n_l represents the size of the vertex set of the l^{th} graph. Let $P^{(1)}, \dots, P^{(L)}$ be the corresponding collection of edge parameter matrices such that each $A_{ij}^{(\ell)} \stackrel{\text{ind}}{\sim} H(P_{ij}^{(\ell)})$ for each $1 \leq i < j \leq n_l$ and H is a specified probability distribution such that $E[A_{ij}^{(\ell)}] = P_{ij}^{(\ell)}$. For ease of notation, we use $A^{(\ell)} \sim H(P^{(\ell)})$ for an adjacency matrix $A^{(\ell)}$ drawn from edge parameter matrix $P^{(\ell)}$ in this way. Denote $X^{(1)}, \dots, X^{(L)}$ as the sets of latent vectors corresponding to each

graph under the GRDPG framework. Each $X^{(\ell)} \in \mathbb{R}^{n_\ell \times d}$, i.e., the graphs share the same latent dimension, and $x_i^{(\ell)} \in \mathbb{R}^d$ is the latent vector corresponding to the i^{th} vertex in the ℓ^{th} graph. $\hat{X}^{(\ell)}$ is the adjacency spectral embedding of $A^{(\ell)}$ and is an estimator for $X^{(\ell)}$.

2.2 Definitions and Models

We begin by defining the random dot product graph and the latent structure model, then building the Multiple Latent Structure Model from there.

Definition 1 (Random dot product graph (RDPG) [23]). 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 $x_1, \dots, x_n \stackrel{\text{iid}}{\sim} F_\theta$. A graph G with adjacency matrix A is a random dot product graph with latent vectors $X = [x_1 \ \cdots \ x_n]^\top$ drawn from distribution F_θ if $A \sim H(XX^\top)$.

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 .

Remark 2. The RDPG requires that $P = XX^\top$ be a positive semidefinite matrix. To include matrices that are not positive semidefinite, the generalized random dot product graph (GRDPG) was introduced by Rubin-Delanchy et al. [16], in which $P = XI_{p,q}X^\top$. Here, $I_{p,q}$ is a diagonal matrix of p 1's followed by $q - 1$'s. While the focus of this work is on the RDPG, the theoretical results also apply to the GRDPG.

Definition 2 (Latent structure model (LSM) [5]). Let $\mathcal{C} \subset \mathcal{X} \subset \mathbb{R}^d$ be a smooth, non-intersecting one-dimensional manifold on the domain of a RDPG as defined in definition 1, parameterized by function $p(t) : [0, 1] \rightarrow \mathcal{C}$. Then if $t_1, \dots, t_n \stackrel{\text{iid}}{\sim} F_\theta$ for some distribution F_θ with support $[0, 1]$ and parameter θ , each $x_i = p(t_i)$, and $A \sim H(XX^\top)$ for $X = [x_1 \ \cdots \ x_n]^\top$, 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 3. Although Athreya et al. [5] 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, $\mathcal{C}_1, \dots, \mathcal{C}_K$, (i.e., mixture of manifolds distribution). This type of latent space mixture distribution is observed in networks with community structure [4]. 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 $\alpha = (\alpha_1, \dots, \alpha_K)$, $\sum_{k=1}^K \alpha_k = 1$ 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 [19] is a consistent estimator of the latent vectors, up to some unknown orthogonal transformation.

Remark 4 (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 $\lim_{n \rightarrow \infty} \rho_n = 0$ 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 $n\rho_n = \omega((\log n)^c)$ for some $c > 1$. (See Abbe [1], Xie [22], and Rubin-Delanchy et al. [16] for further discussion.) For ease of exposition, we set $\rho_n \equiv 1$ for the remainder of this paper.

Definition 3 (Adjacency spectral embedding (ASE) [19]). 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 $V_d = [v_1 \cdots v_d]$ and $\Lambda_d = \text{diag}(\lambda_1, \dots, \lambda_d)$. Then $\hat{X} = V_d|\Lambda|^{1/2}$ is the

d -dimensional adjacency spectral embedding of A .

Theorem 1 (Consistency of the ASE [13]). *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. [5] showed exactly this.

Definition 4 (Multiple latent structure model (MLSM)). Let $\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(L)} \subset \mathbb{R}^d$ 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 $p^{(\ell)}(t) : [0, 1] \rightarrow \mathcal{C}^{(\ell)}$. Let $p^{(1)}, \dots, p^{(L)}$ be a sequence of functions with domain $[0, 1]$ that parameterize the curves $\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(L)}$, let F be a parametric distribution with support $[0, 1]$, and let $\theta_1, \dots, \theta_L$ be a sequence of parameters for distribution F . For each $p^{(\ell)}$, sample $t_1^{(\ell)}, \dots, t_{n_\ell}^{(\ell)} \stackrel{\text{iid}}{\sim} F_{\theta^{(\ell)}}$ for some distribution F parameterized by θ_ℓ , and let $x_i^{(\ell)} = p_\ell(t_i^{(\ell)})$ for each $i = 1, \dots, n_\ell$, again as in definition 2. Sample a sequence of L adjacency matrices, each as $A^{(\ell)} \stackrel{\text{ind}}{\sim} \text{LSM}(p_\ell, F_{\theta_\ell})$. Then the sequence $\{A^{(\ell)}\}_L$ are the adjacency matrices of a multiple latent structure model with curves $\{\mathcal{C}^{(\ell)}\}_L$ parameterized by $\{p^{(\ell)}\}$ and underlying distribution F with parameters $\{\theta_\ell\}_L$.

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

Remark 5. 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 $A^{(1)}, \dots, A^{(L)} \sim \text{MLSM}(\{\mathcal{C}_k^{(\ell)}\}_{K,L}, F, \{\theta_\ell\}_L, \{\alpha^{(\ell)}\}_L)$ or $A^{(1)}, \dots, A^{(L)} \sim \text{MLSM}(\{p_k^{(\ell)}\}_{K,L}, F, \{\theta_\ell\}_L, \{\alpha^{(\ell)}\}_L)$.

2.2.1 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. Unlike multilayer models, such as the multilayer random dot product graph [9] or CCommon Subspace Independent Edge (COSIE) graph [3], there is no assumption that the graphs are connected in some way, such as via a shared vertex set. In the MLSM, each graph is drawn from the same process with different parameters, whereas in multilayer models the graphs share the same vertex set but the edges are drawn from different distributions. The MLSM can describe multilayer graphs if further restrictions are imposed on it.

Example 1 (Comparison to the multilayer DCBM [2], multi-RDPG [14], and MREG [21]).

In the K -community DCBM, the probability of an edge between a pair of vertices is given by

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

where $z_i \in \{1, \dots, K\}$ 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 $\sum_{i:z_i=k} \omega_i^2 = 1$ [10]. As in

The edge parameter matrix of a K -community DCBM with n vertices can be decom-

posed 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 [16]. 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 RDG (multi-RDG) with the identity link function, as described by Nielsen and Witten [14], or equivalently, a special case of the multiple random eigen graphs model (MREG), as described by Wang et al. [21]. However, if the community labels are not identical from graph to graph, the sequence of DCBMs cannot be described as a multilayer DCBM, multi-RDG, or MREG, but it can still be described as an MLSM.

2.2.2 Classification and Regression on the MLSM

To use the MLSM for classification or regression problems, we assign response variables y_1, \dots, 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 $A^{(1)}, \dots, A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$, $y_1, \dots, y_L \in \{1, \dots, M\}$ be labels for each of the L graphs of the MLSM, and each θ_ℓ take on one of M values $\{\phi_1, \dots, \phi_M\}$ corresponding to its response, y_ℓ , i.e., $\theta_\ell = \phi_{y_\ell}$.

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

This can be described by the following generative model:

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

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

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

This can be described by the following generative model:

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

As in remark 5, 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 $A^{(1)}, \dots, A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta_\ell\}_L)$, and suppose that for each $\ell = 1, \dots, L$, the ℓ^{th} graph is coupled with a response variable, y_ℓ , as $y_\ell \stackrel{\text{ind}}{\sim} \mathcal{N}(\theta_\ell^\top \beta, \sigma^2)$.

We observe the adjacency matrices $A^{(1)}, \dots, A^{(L)}$ and the first r response variables y_1, \dots, 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}, \dots, y_L .

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

We observe the adjacency matrices $A^{(1)}, \dots, A^{(L)}$ and the first r response variables y_1, \dots, 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}, \dots, y_L .

As in remark 5, 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. [5] showed that with some additional information, the ASE of $A^{(\ell)}$ can lead

to a consistent estimator for θ_ℓ .

2.3 Main Results

We propose our methodology for classification and regression in three steps: First, we propose a curve-fitting algorithm to recover the LSM curves from the ASEs. Second, we estimate the parameters of the underlying distributions from the curve fit via maximum likelihood estimation. Third, we apply classification or regression methods to determine the relationship between the estimated parameters and the responses. Finally, we provide theorems that show the consistency of the curve fits and maximum likelihood estimators.

2.3.1 Estimation of the LSM Curves

For curve-fitting, we will restrict the LSM curves to non-self-intersecting Bezier polynomials [7]. A polynomial curve of degree R defined by $p : [0, 1] \rightarrow \mathbb{R}^d$ 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 [17]. 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 [11, 16], as well as real networks that can adequately be described as LSMs [5]. 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

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

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 $x_1, \dots, x_n \in \mathbb{R}^d$ 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

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

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

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

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

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

On the other hand, if the coefficients are known but the timepoints t_1, \dots, t_n are not, this turns into n individual minimization problems for polynomials of degree $2R$. More

precisely, $L(t_1, \dots, t_n; b_1, \dots, b_R) = \sum_{i=1}^n L(t_i; b_1, \dots, b_R)$ where

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

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

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

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.

Combining equation 4 with the solutions to equation 6 for each $i = 1, \dots, n$ 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 , stopping criterion ϵ .

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

- 1 Compute $\hat{X} \in \mathbb{R}^{n \times d}$, the ASE of A .
 - 2 Initialize timepoints $\hat{t}_1, \dots, \hat{t}_n$, (e.g., via a one-dimensional Isomap embedding [20]).
 - 3 **repeat**
 - 4 Fit \hat{b} by equation 4.
 - 5 **for** $i = 1, \dots, n$ **do**
 - 6 Fit \hat{t}_i by equation by finding the roots of equation 6 and choosing the roots that minimizes equation 5.
 - 7 **end**
 - 8 **until** the change in equation 5 is less than ϵ .
-

In the case of a mixture of LSMs (as in remark 3), 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 , stopping criterion ϵ .

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

- 1 Compute $\hat{X} \in \mathbb{R}^{n \times d}$, the ASE of A .
- 2 Initialize $\hat{z}_1, \dots, \hat{z}_n \in \{1, \dots, K\}$, the membership of each vertex to each latent curve.
- 3 **for** $k = 1, \dots, K$ **do**
- 4 Subset the rows of \hat{X} by $z_i = k$ to obtain $\hat{x}_{k_1}, \dots, \hat{x}_{k_{n_k}}$, where n_k is the number of vertices with label k .
- 5 Initialize timepoints $\hat{t}_{k_1}, \dots, \hat{t}_{k_{n_k}}$, (e.g., via a one-dimensional Isomap embedding [20]).
- 6 **end**
- 7 **repeat**
- 8 **for** $k = 1, \dots, K$ **do**
- 9 Fit $\hat{b}^{(k)}$ by equation 4, using the row subsets of T and X such that $z_i = k$.
- 10 **for** $i = 1, \dots, n_k$ **do**
- 11 Fit \hat{t}_{k_i} by equation by finding the roots of equation 6 and choosing the roots that minimizes equation 5.
- 12 **end**
- 13 **end**
- 14 **for** $k = 1, \dots, n$ **do**
- 15 Reassign z_i based on the latent curve closest to \hat{x}_i .
- 16 **end**
- 17 **until** the change in equation 5 is less than ϵ .

Theorem 2 (Consistency of algorithm 1). *Suppose $A \sim \text{LSM}(p, F_\theta)$ such that $p : [0, 1] \rightarrow \mathbb{R}^d$ 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 $n \rightarrow \infty$, $\|T - \hat{T}\|_{2,\infty} \xrightarrow{p} 0$ where T is defined as in equation 3, and $\|b - \hat{b}W\|_F \xrightarrow{p} 0$ for some $W \in \mathbb{O}(d)$.*

2.3.2 Quasi-Maximum Likelihood Estimation for θ

The estimation of T and the restrictions imposed on the family of Bezier curves allows for the estimation of θ via maximum likelihood estimation (algorithrm 3). Since \hat{T} is a consistent estimator for T by theorem 2, if $t_1, \dots, t_n \sim F_\theta$, a maximum likelihood estimator using $\hat{t}_1, \dots, \hat{t}_n$ is a consistent estimator for θ . Algorithm 3 details the procedure.

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 .

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

- 1 Compute $\hat{X} \in \mathbb{R}^{n \times d}$, the d -dimensional ASE of A .
 - 2 Initialize timepoints $\hat{t}_1, \dots, \hat{t}_n$.
 - 3 **repeat**
 - 4 Fit \hat{b} by equation 4.
 - 5 **for** $i = 1, \dots, n$ **do**
 - 6 Fit \hat{t}_i by equation by finding the roots of equation 6 and choosing the roots that minimizes equation 5.
 - 7 **end**
 - 8 **until** the change in equation 5 is less than ϵ .
 - 9 Estimate $\hat{\theta}$ from $\hat{t}_1, \dots, \hat{t}_n$ via maximum likelihood estimation.
-

Theorem 3 (Consistency of algorithm 3). *Suppose $A \sim \text{LSM}(p, F_\theta)$ such that $p : [0, 1] \rightarrow \mathbb{R}^d$ 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 $\hat{t}_1, \dots, \hat{t}_n$ 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 $\hat{t}_1, \dots, \hat{t}_n$. Then $\tilde{\theta} \xrightarrow{p} \theta$.*

2.3.3 Classification and Regression

Applying algorithm 3 to supervised learning problems is straightforward. First, we extract each $\hat{\theta}^{(\ell)}$ from each graph, then apply a supervised learning algorithm (e.g., ordinary least squares) for predicting each $y^{(\ell)}$ from each $\hat{\theta}^{(\ell)}$. Since 3 produces consistent estimators, for

large graphs the supervised learning algorithm in this regime will perform just as well as if we already knew each $\theta^{(\ell)}$. Therefore, $\tilde{\beta}$, the estimator based on $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(L)}$, is consistent for β (theorem 4).

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

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

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

```

1 for  $\ell = \{1, \dots, L\}$  do
2   Compute  $\hat{X}^{(\ell)} \in \mathbb{R}^{n_\ell \times d}$ , the  $d$ -dimensional ASE of  $A^{(\ell)} \in \mathbb{R}^{n_\ell \times n_\ell}$ .
3   Initialize timepoints  $\hat{t}_1, \dots, \hat{t}_{n_\ell}$ ; repeat
4     Fit  $\hat{b}$  by equation 4.
5     for  $i = 1, \dots, n_\ell$  do
6       Fit  $\hat{t}_i$  by equation by finding the roots of equation 6 and choosing the
7         roots that minimizes equation 5.
8     end
9     until the change in equation 5 is less than  $\epsilon$ .
10    Estimate  $\hat{\theta}^{(\ell)}$  from  $\hat{t}_1, \dots, \hat{t}_{n_\ell}$  via maximum likelihood estimation.
11  end
12  Fit a model for predicting  $y^{(\ell)}, \dots, y^{(\ell)}$  using the estimated parameters  $\theta^{(\ell)}, \dots, \theta^{(\ell)}$ 
13  (e.g., via linear regression).

```

Theorem 4. Suppose $A^{(1)}, \dots, A^{(L)} \sim \text{MLSM}(\{p^{(\ell)}\}, F, \{\theta^{(\ell)}\})$ where $p : [0, 1] \rightarrow \mathbb{R}^d$ is a Bezier polynomial of degree $R < d$ and $p(0) = 0$. Let each $y^{(\ell)} \stackrel{\text{ind}}{\sim} \mathcal{N}((\theta^{(\ell)})^\top \beta, \sigma^2)$, and let $\tilde{\beta}$ be the ordinary least squares estimate of β using $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(L)}$, the outputs of 3 for each graph. Then as each $n_\ell \rightarrow \infty$, $\tilde{\beta} \xrightarrow{P} \beta$.

3 Simulation Study

In this section, we evaluate algorithm 4 and theorem 4 empirically via simulation and compare to existing methods. Two settings are examined. In the first, the data generating model reduces to a simpler model and can be adequately described by the COSIE model or multilayer DCBM. The second setting involves graphs of variable sizes drawn from a

nonlinear latent structure that can be described as a polynomial curve and thus does not reduce to a simpler model or be analyzed by their corresponding methods.

In these simulations, the responses $y^{(\ell)}$ are generated as linear responses of $\theta^{(\ell)}$ (in scenario 1) or $a^{(\ell)}$ and $b^{(\ell)}$ (in scenario 2). More specifically, in scenario 1, each $y_\ell \stackrel{\text{ind}}{\sim} \mathcal{N}(\beta_0 + \beta_1 \theta^{(\ell)}, \sigma^2)$, where $\theta^{(\ell)}$ is the angle between two line segments in the latent space, and in scenario 2, each $y_\ell \stackrel{\text{ind}}{\sim} \mathcal{N}(\beta_0 + \beta_1 a^{(\ell)} + \beta_2 b^{(\ell)}, \sigma^2)$, where $a^{(\ell)}$ and $b^{(\ell)}$ describe the Beta parameters of the distribution of latent vectors along a curve. We set $\sigma^2 = 0.25$ for both scenarios.

The simulation setup for the first scenario is as follows: For each $N = 128, 256, 512, 1024$, sample $L = 128$ graphs as:

- i. Draw n_1, \dots, n_L as $n_\ell \stackrel{\text{iid}}{\sim} \text{Poisson}(N)$.
- ii. Draw $\theta^{(1)}, \dots, \theta^{(L)} \stackrel{\text{iid}}{\sim} \text{Uniform}(\pi/6, \pi/3)$.
- iii. For each $\ell = 1, \dots, L$, draw latent vectors either from the line segment described by $p_1(t) = \begin{bmatrix} t & 0 \end{bmatrix}^\top$ or $p_2(t) = \begin{bmatrix} \cos(\theta^{(\ell)}) & \sin(\theta^{(\ell)}) \end{bmatrix}^\top$ with probability 1/2 for each.
- iv. Use the L sets of latent vectors to draw L RDPGs with adjacency matrices $A^{(1)}, \dots, A^{(L)}$.
- v. Draw each $y_\ell \stackrel{\text{ind}}{\sim} \mathcal{N}(1 + 2\theta^{(\ell)}, \sigma^2)$.
- vi. Apply algorithm 4 to construct a linear model connecting y_1, \dots, y_ℓ and $A^{(1)}, \dots, A^{(L)}$, using the first half as training data and reporting MSE on the last half not used for training.

For the second the simulation setup is similar. For each $N = 128, 256, 512, 1024$, sample $L = 128$ graphs as:

- i. Draw n_1, \dots, n_L as $n_\ell \stackrel{\text{iid}}{\sim} \text{Poisson}(N)$.
- ii. Draw each $(a^{(\ell)}, b^{(\ell)}) \stackrel{\text{iid}}{\sim} \text{Uniform}([0.5, 2] \times [0.5, 2])$.

- iii. For each $\ell = 1, \dots, L$, draw latent vectors by first taking $t_1, \dots, t_{n_\ell} \sim \text{Beta}(a^{(\ell)}, b^{(\ell)})$ then taking each latent vector as $x_i = \begin{bmatrix} t^2 & 2t(1-t) \end{bmatrix}^\top$.
- iv. Use the L sets of latent vectors to draw L RDPGs with adjacency matrices $A^{(1)}, \dots, A^{(L)}$.
- v. Draw each $y_\ell \stackrel{\text{ind}}{\sim} \mathcal{N}(1 + 2\alpha^{(\ell)} + b^{(\ell)}, \sigma^2)$.
- vi. Apply algorithm 4 to construct a linear model connecting y_1, \dots, y_ℓ and $A^{(1)}, \dots, A^{(L)}$, using the first half as training data and reporting MSE on the last half not used for training.

In figure 1, we compare algorithm 4 to fitting a regression model using graph transitivity and average degree of each vertex. Since this scenario is similar to a multilayer DCBM, we expect performance to be very similar, which is what we observe in the results. However, we cannot model this simulation as a multilayer DCBM since although each graph can be described as a DCBM, the graphs do not share the same vertex set. In figure 3, we see that linear regression using graph summary statistics performs better for smaller graphs, but as we draw more vertices, the ASE improves and so algorithm ?? performs better, whereas graph summary statistics cannot improve with the number of vertices. Figures 2 and 4 show that the quasi-MLE estimators computed via algorithm 3 are consistent.

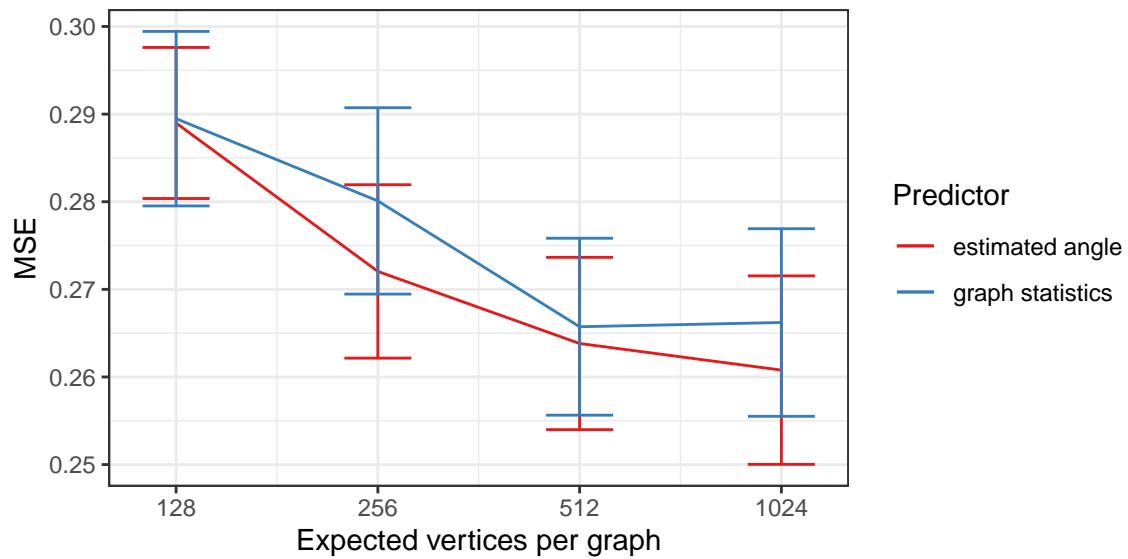


Figure 1: The average MSE over 50 simulations in scenario 1. The errorbars are the standard error.

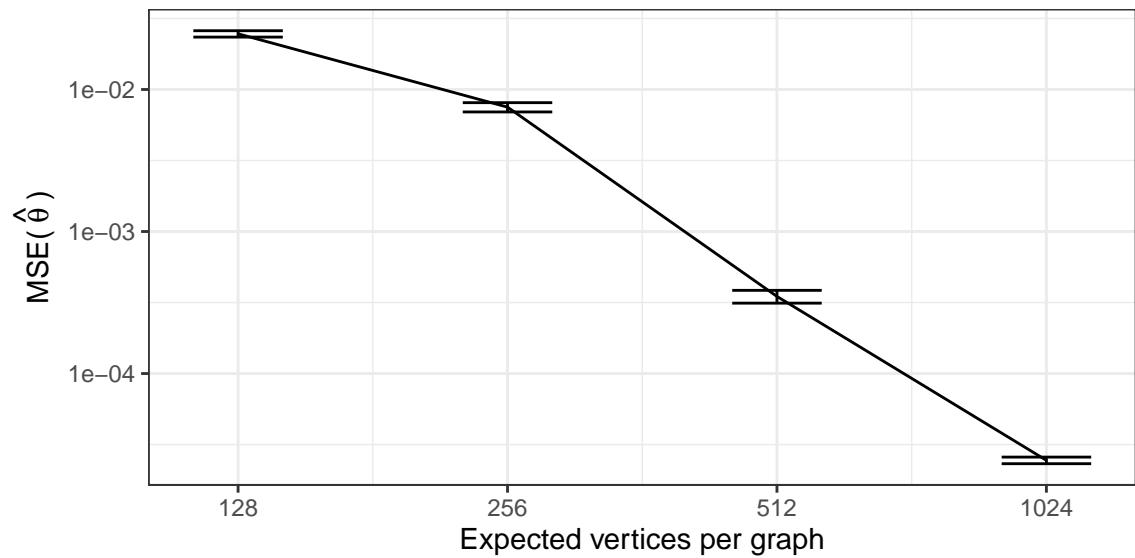


Figure 2: The average MSE of the estimated angles in scenario 1 over 50 simulations. The errorbars are the standard error.

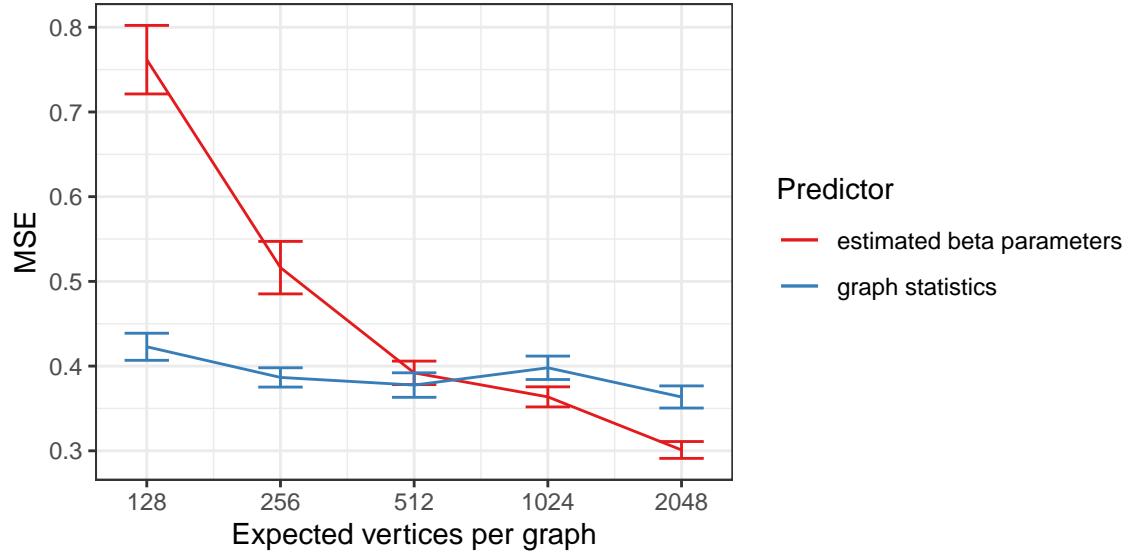


Figure 3: The average MSE over 50 simulations in scenario 2. The errorbars are the standard error.

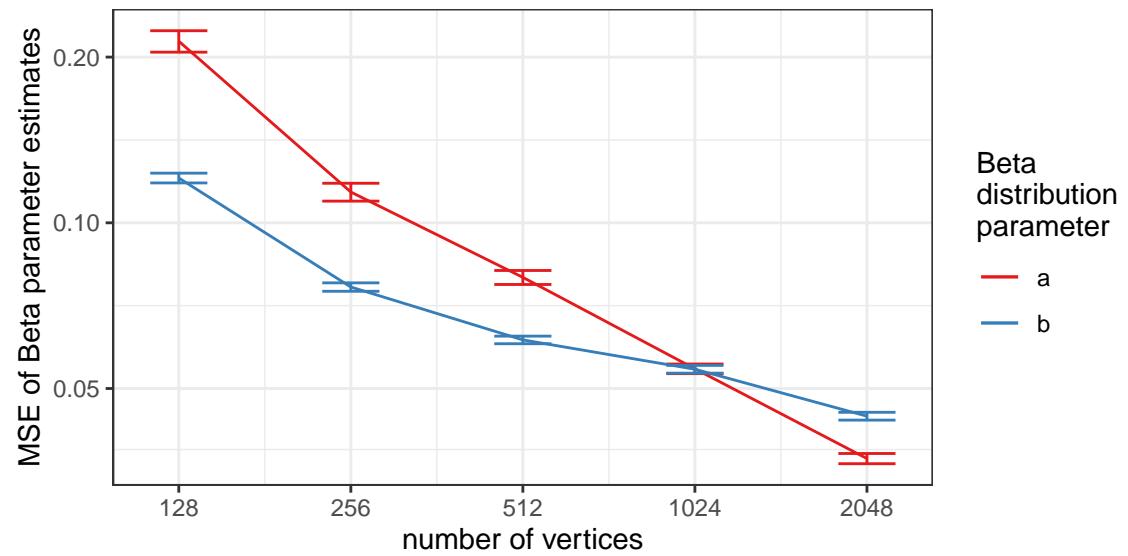


Figure 4: The average MSE of the estimated beta parameters, (a, b) , in scenario 2 over 50 simulations. The errorbars are the standard error.

4 Applications

4.1 Drosophila Connectome

In the second example, we analyzed the larval *Drosophila* mushroom body connectome [6], which has been studied as a GRDPG by Athreya et al. [5]. 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 5.

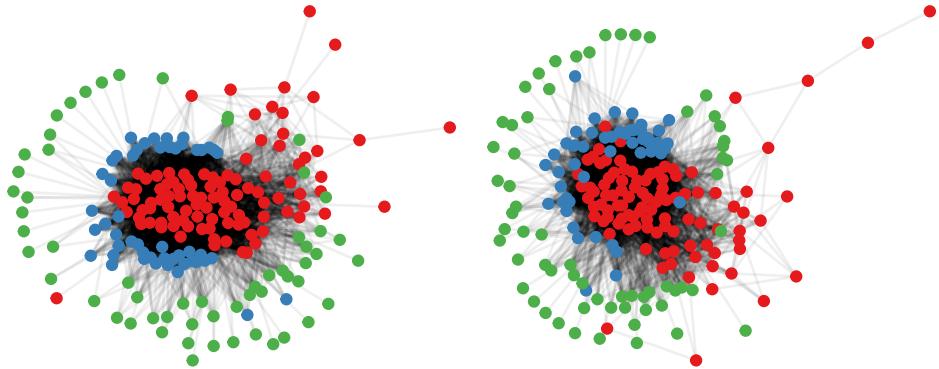


Figure 5: Graphs of the Drosophila connectomes. The left and right are of the left and right hemispheres, respectively. Each vertex represents a neuron, which are labeled by neuron type. The red vertices are Kenyon Cells, the blue vertices are Input and Output Neurons, and the green vertices are Projection Neurons.

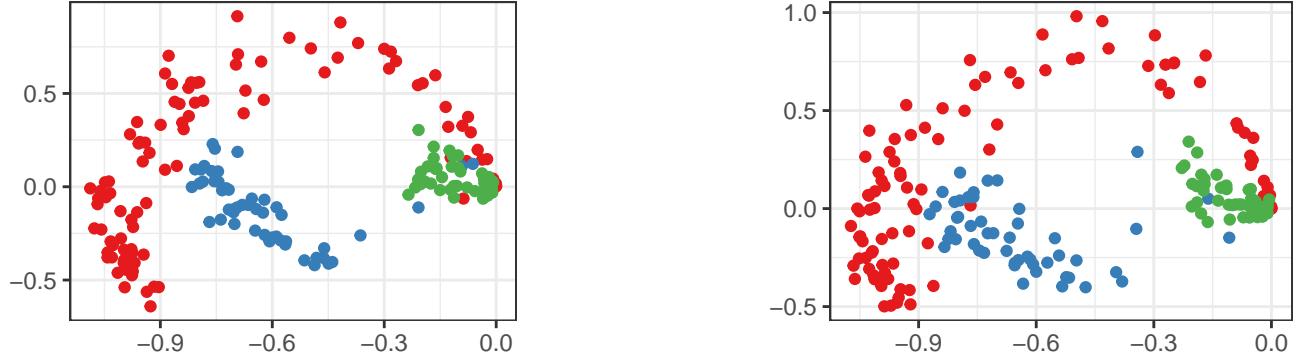


Figure 6: ASEs of the Drosophila connectome graphs. The embedding vectors are labeled by neuron type, with red being Kenyon Cells, blue being Input and Output Neurons, and green being Projection Neurons.

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 6). 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. [5]. 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 7).

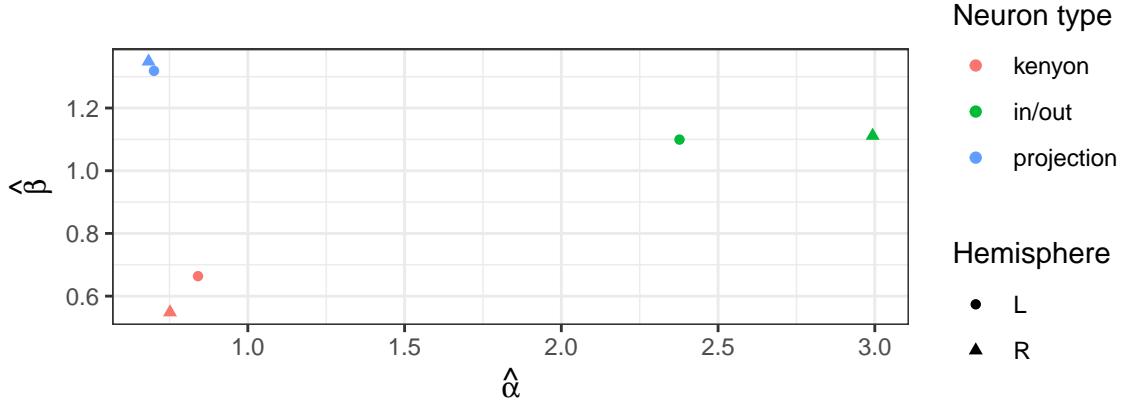


Figure 7: Beta parameter estimates for each curve and hemisphere.

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

In this dataset, there are $L = 516$ 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 8).

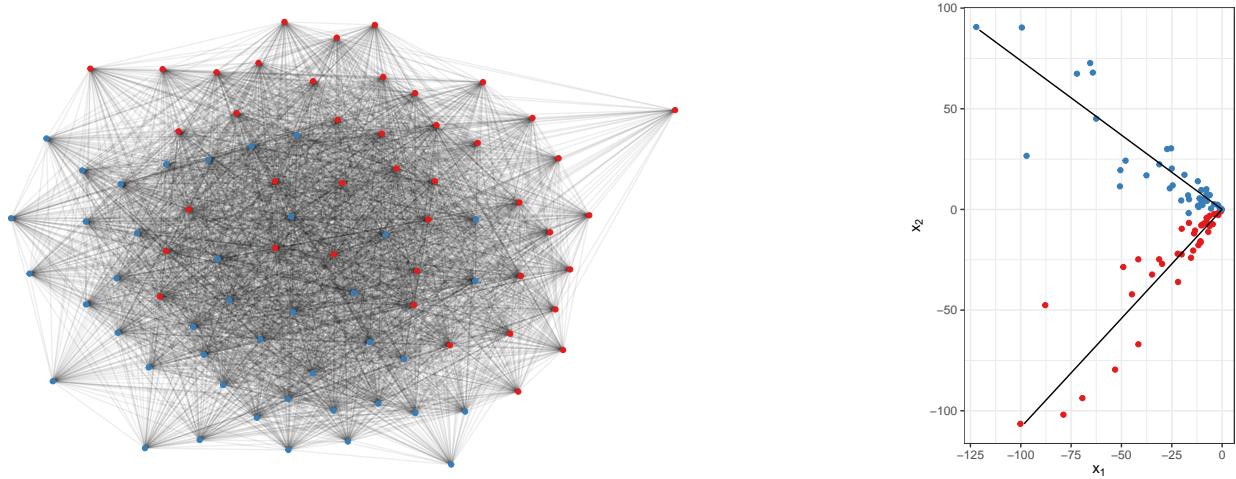


Figure 8: One graph from the HCP dataset (left) and its ASE (right). In the ASE, the lines are fitted via K -curves clustering using $\text{degree} = 1$. The outputted clusters correspond exactly to the left (red) and right (blue) hemispheres.

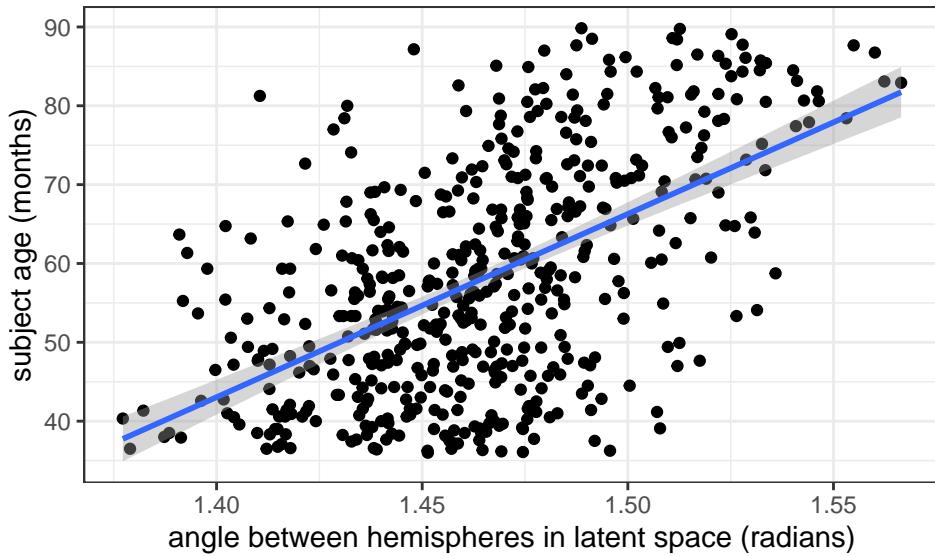


Figure 9: Scatterplot between the subject age (in months) vs. the fitted angle between hemispheres of the brain in the latent space.

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 9). 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 1. 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.

Table 1: Correlation between age and various graph metrics.

Metric	Correlation	95% conf. int.
Angle between hemispheres	0.558	(0.496, 0.615)
Degree within hemisphere	0.436	(0.363, 0.503)
Degree between hemispheres	-0.499	(-0.561, -0.431)
Modularity w.r.t. hemisphere	0.434	(0.362, 0.502)
Joint Embedding	0.556	(0.493, 0.613)

5 Discussion

Our main contribution is applying the latent structure model, a specific form of the random dot product graph, to multiple graphs for regression and classification problems. We show that if the response variable is connected to the distribution parameter of a LSM, we can construct a consistent estimator for the parameter and then use that estimator to construct consistent supervised learning algorithms to estimate the responses. Since LSMs are not necessarily linear in the latent space, we show that our approach can accommodate nonlinear structures in the latent space, which cannot be adequately captured by other multiple graph models such as the multilayer DCBM or COSIE graph model. Furthermore, we remove the restriction of the graphs sharing the same vertex set, although the MLSM can still apply to such scenarios.

Appendix A: Proofs of Theorems

Lemma 1. Suppose $A \sim \text{LSM}(p, F_\theta)$ such that $p : [0, 1] \rightarrow \mathbb{R}^d$ 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 $n \rightarrow \infty$, $\|X - \tilde{X}W\|_{2,\infty} \xrightarrow{p} 0$ 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 [13]. Since the ASE is consistent, the embedding vectors will converge in probability to a Bezier curve, so $\|\hat{E}W\|_{2,\infty} \xrightarrow{p} 0$. \square

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] Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C. E., and Vogelstein, J. T. (2020). Inference for multiple heterogeneous networks with a common invariant subspace.
- [4] 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.
- [5] Athreya, A., Tang, M., Park, Y., and Priebe, C. E. (2020). On estimation and inference in latent structure random graphs.
- [6] 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*.
- [7] Gallier, J. (1999). *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [8] Gilbert, E. N. (1959). Random Graphs. *The Annals of Mathematical Statistics*, 30:1141 – 1144.
- [9] Jones, A. and Rubin-Delanchy, P. (2021). The multilayer random dot product graph.

- [10] Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1).
- [11] 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.
- [12] Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1):49–80.
- [13] 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.
- [14] Nielsen, A. M. and Witten, D. (2018). The multiple random dot product graph model.
- [15] Passino, F. S. and Heard, N. A. (2020). Bayesian estimation of the latent dimension and communities in stochastic blockmodels. *Statistics and Computing*, 30(5):1291–1307.
- [16] 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)*.
- [17] Sánchez-Reyes, J. (2022). The uniqueness of the rational bézier polygon is unique. *Computer Aided Geometric Design*, 96:102118.
- [18] 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.

- [19] 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.
- [20] Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.
- [21] 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.
- [22] Xie, F. (2021). Entrywise limit theorems of eigenvectors for signal-plus-noise matrix models with weak signals.
- [23] 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.