Estimation and Inference for Network Data

Shane Lubold

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Tyler McCormick, Chair

Yen-Chi Chen

Abel Rodriguez

Program Authorized to Offer Degree:
Department of Statistics

University of Washington

Abstract

Inference and Estimation for Network Data

Shane Lubold

Chair of the Supervisory Committee:
Tyler McCormick
Department of Statistics

Networks play a key role in many scientific domains. In this thesis, we analyze several important questions in network analysis. The first question we analyze concerns how to understand latent structure in networks. Specifically, we propose a method that estimates the latent type, dimension, and curvature of the latent space model. The second problem we consider concerns network data collection. Collecting full network data is often prohibitively expensive and time-consuming. A common form of cheaper network data, known as Aggregated Relational Data (ARD), asks respondents "How many people do you know with trait X?" for various pre-determined traits. In the second project, we show that ARD is sufficient to recover many statistics of the unobserved graph, which shows that researchers can simply collect ARD instead of full network data. The third question we analyze concerns model selection for network data. We derive a testing procedure that allows researchers to select the most appropriate model from a collection of candidate models, using the eigenvalues of the normalized adjacency matrix. We also show how this testing method is applicable to

cases where the researcher only has access to ARD. Finally, we consider the problem of obtaining low-dimensional representations of objects from dissimilarity data. We propose a Bayesian procedure that uses dissimilarity data to obtain a representation of the objects in a Hyperbolic space, and show that this procedure obtains useful representations for the objects in several down-stream tasks in gene expression data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGEMENTS

I am very grateful for all of the advice and help from my advisor Tyler McCormick, from my collaborators, and from my cohort at the University of Washington. I would first like to thank Tyler for his help these past 5 years. He has helped me to identify important problems and communicate my work effectively. This thesis would not be possible without his constant feedback and encouragement. I would also like to thank Arun Chandrasekhar, who has been an excellent collaborator over these past 5 years. He has helped me to reason intuitively about problems, how to argue technical details properly, and how to identify important problems. He has been an extremely accessible and helpful collaborator, and I thank him for always being available to talk. I would like to thank my committee members Yen-Chi Chen and Alan Griffith for providing feedback on my progress in graduate school and for attending my exams.

I would like to thank all of the students in my cohort and in the department, including Sarah Teichman, Anna Neufeld, Alan Min, Anupreet Porwal, Michael Pearce, Steve Wilkins-Reeves, Jess Kunke, and Sara Laplante for always being happy to help with coursework or research and for being great friends. I'd also like to thank the great professors in the statistics and biostatistics departments, including Michael Pearlman, Yen-Chi Chen, and Amy Willis for teaching courses that taught me a great deal. I'd also like to thank Ellen Reynolds, Tracy Pham, and Kristine Chan for always being helpful with department administrative tasks. Outside of UW, I'd like to thank Horst Thieme and John Fricks at ASU for teaching excellent courses that instilled in me a passion for math and statistics, and I'd like to thank Clark Taylor at the Air Force Research Laboratory for his support and guidance. I'd like to thank my parents Marie

and Rory and my siblings Rory Dean, Jessie, Maddie and Aubyn for their support and love. Finally, I want to thank Eli Cole, who always believed in me and has always been my strongest supporter. I would not have been able to do this without him.

DEDICATION

to Eli

Chapter 1

INTRODUCTION

Networks play a key role in many areas of social science. A network consists of a collection of nodes, which can correspond to people, organizations, or businesses, and edges between the nodes. There are many application areas for network analysis, including in social media [149, 126, 42], sexual health [78], and politics [51].

Building accurate and reliable models for network data is a crucial and challenging question. Many models have been proposed, including the stochastic block model [88], the latent space model [82], and the exponential random graph model [63, 161]. In the first chapter of this thesis, we focus on an important problem in network inference. Latent space models are a generative model for network data in which nodes are assigned positions in a latent (or unobserved) social space, and the probability of connection between two nodes increases as the distance between the nodes decreases [85]. There are three common latent space types commonly used - Euclidean, spherical, and hyperbolic - and the choice of the latent space is known to have a large impact on the properties of the networks generated from the model [11, 10, 160]. For example, spherical latent spaces tend to produce highly clustered networks, whereas hyperbolic latent spaces tend to produce tree-like networks. Despite the impact that the latent space type has on the properties of the resulting network, there has been little work on estimating the properties of the latent space from network data. In Chapter 2, we provide a method that uses network data to estimate the type, dimension, and curvature of the latent space from a fully observed graph. This is joint work with Tyler McCormick and Arun Chandrasekhar and has been accepted for publication at the *Journal of the Royal Statistical Society Series B*.

In Chapter 3, we focus on using sampled network data to understand properties of an unobserved network. Specifically, we focus on survey methods that reduce the cost of network data collection, which is often expensive and time-consuming [27]. Aggregated Relational Data (ARD) is an alternative to full network data that asks respondents questions of the form "How many people with trait X do you know" for various pre-selected traits. Instead of collecting individual edges between nodes, as is done when collecting complete network data, ARD only collects information about how many connections an individual has to people of given traits. [28] uses a field survey from rural India to show that ARD can reduce the cost of network data collection by 70 percent. In the second chapter, we provide a methodology to estimate network statistics of unobserved networks using just ARD. We provide conditions under which this methodology returns consistent estimates of these statistics, as the network size grows. We also discuss estimation of network-level regression coefficients when the researcher only has access to ARD and not full network data. In this setting, we also provide conditions under which the OLS estimates, computed using only the ARD, are consistent for the true regression parameters as the graph size grows. This is joint work with Emily Breza, Arun Chandrasekhar, Tyler McCormick, and Mengjie Pan, and it has been accepted for publication at the *Proceedings of the National Academy of Sciences.*

In Chapter 4, we consider the problem of model goodness of fit for network data. Selecting an appropriate model for network data is a vital question. There are many methods for doing model selection for network data. One such method, often used when selecting the form of the exponential random graph model, involves comparing certain statistics from the network (such as the degree distribution or number of triangles) against the simulated statistics, computed by generating networks from a candidate network model. If the model is correct, then the simulated statistics should match the observed statistics. In the third chapter, we show how to do model selection for network data using the eigenvalues of the adjacency matrix. [110] proposed a

method to estimate the number of communities in a stochastic block model using the largest and smallest eigenvalues of the (normalized) adjacency matrix of a graph. Building on this idea, we show how to use this method to derive a general goodness-of-fit procedure for any parametric network model. In particular, our proposed method is applicable whenever the researcher is able to estimate the parameters of the network model. In order for this method to return a consistent classifier of the network model, the estimation of the network model parameters must be sufficiently fast [110]. We also show how to extend the ideas in [110] to perform model selection when the researcher only has access to ARD from the network and does not have access to full network data. This is joint work with Bolun Liu and Tyler McCormick, and is currently under reivew.

In Chapter 5, we consider the problem of obtaining a low-dimensional embedding of these objects from dissimilarity data. In such an embedding, objects with high similarity (or low dissimilarity) would be closer together, and objects with high dissimilarity would be far apart. Many approaches exist to answer this question, with multi-dimensional scaling (MDS) being a common approach [107, 23, 48, 45]. In this chapter, we build upon the work of [131], which builds a Bayesian model for obtaining low-dimensional embeddings of objects in a Euclidean space using noisy dissimilarity data. Specifically, we propose a Bayesian for obtaining low-dimensional embeddings of objects in a hyperbolic space using noisy dissimilarity data. We show how to estimate the curvature of the hyperbolic space using the dissimilarity data, and run simulations to understand when our proposed Bayesian procedure outperforms state-of-the-art methods, such as those proposed in [99]. Finally, we use our proposed Bayesian procedure to analyze gene expression data. This is joint work with Adrian Raftery, Bolun Liu, and Tyler McCormick, and it is currently under review.

In Chapter 6, we provide concluding remarks and discuss future work.

Chapter 2

IDENTIFYING LATENT SPACE GEOMETRY OF NETWORK FORMATION MODELS VIA ANALYSIS OF CURVATURE

2.1 Introduction

Social, economic, biological, and technological networks play a crucial role in a myriad of environments. Job referrals [74, 31, 16, 80], neurological function [113], epidemics [82, 15, 157], social media [149, 126, 42], informal insurance [6, 30], education decisions [32], sexual health [78], financial contagion [66, 55, 2], international trade [36], and politics [51] are among the many settings in which networks play a major role. Modeling network formation is, therefore, essential for both descriptive and counterfactual analyses.

Constructing such models is challenging from a statistical perspective since networks typically feature higher-order dependence between the connections. Phenomena such as transitivity are common and mean that standard regression approaches, which assume independence across connections, are not appropriate. A common approach for modeling this dependence structure is the latent space model, introduced by [82]. One estimates a probability distribution over graphs that is consistent with the single, observed graph. The model assigns each node in the network to a position on a low-dimensional manifold. Likelihood of a connection is inversely proportional to distance between actors on a manifold with a pre-specified dimension and geometry. Connections are assumed independent conditional on the latent positions. Standard practice in this area assumes a manifold class beforehand.

The choice of latent manifold is extremely consequential for both the interpretation

and its theoretical properties. In enumerating these properties, it is first critical to distinguish between the intrinsic and extrinsic geometries. Our focus is on the former, which embodies the fundamental properties of the geometric space. Modeling in a geometry manifold and inheriting the distance measure of that geometry (e.g., using a 2-dimensional sphere means points exist on the surface of the sphere and distances are measured by arc length).

Moving now to the connection between geometry and networks, we first note that, holding constant the distribution of points in the latent space, the choice of the geometry in particular determines the nature of network structure captured by the latent space. For a simple example, consider a two dimensional Euclidean space (a plane). Here it is not possible to place four nodes in such a way that they are equidistant from one another, meaning that it isn't possible to represent groups of four such that, holding constant node effects, each node has the same likelihood of interacting with any other. Another way to see the impact of geometry is through triangles. Since a sphere has bounded area, there is an upper bound to how far apart nodes can be from one another before they start getting closer together. Positive curvature also encourages the formation of triangles and communities. Additionally, certain networks, such as a network of neurons or a network exchange built along a supply chain, may have a tree-like structure. Trees are difficult to embed in spherical or Euclidean space but fit more naturally in hyperbolic space. Recent work on statistical modeling has also shown the importance of modeling networks using non-Euclidean latent representations. For instance, [125] model latent space as a sphere and [106] and [11] use hyperbolic space. [175] explores both spherical and Euclidean representations in a Bayesian model for spatial voting patterns. [168] examines the relationship between the latent space curvature and graph motifs. [170] propose a test of the assumption that the latent space has constant curvature which uses the clique structure in the graph. Finally, [160] provides a comprehensive review of the implications and consequences of the choice of geometry.

A second consequence of the choice of geometry arises in the theoretical properties of latent space model estimates. Consistency of the estimates of the individual locations on the unobserved manifold is the subject of recent work by [158]. Since the distribution of the network formation process depends on the manifold itself, the key open question is whether a researcher can consistently estimate the latent space. After all, the network formation process is sensitive to the geometry inclusive of its curvature and dimension.

It is currently common practice to assume the latent dimension and manifold type. Our approach provides a data-driven alternative. It also contrasts with cross-validation based selection procedures that are sometimes used, in particular to estimate the dimension. These approaches subsample connections and then use either model fit diagnostics or out-of-sample prediction metrics. Our approach avoids a critical issue with these approaches, namely subsampling can fundamentally alter graph properties in unpredictable ways [35], calling into question the relevance of the subsampled distribution. We approach the question from a fundamentally different perspective than currently available alternatives that use the likelihood or penalized likelihood to estimate model fit. First, rather than characterizing fit or predictive accuracy with a particular dataset, our approach takes a more classical hypothesis testing perspective. Comparing (for example) an information metric across a model fit with a spherical or hyperbolic latent space is fundamentally characterizing the congruence between the embedding for a given dataset and the spaces under consideration. Uncertainty in this framework arises from sampling, but also from potential model mis-specification. A likelihood based metric for a spherical space with small curvature will likely perform quite well for a graph generated from Euclidean embeddings, for example. We isolate uncertainty to only sampling error by using a test for isometric embeddings of distances into the space under consideration. We conceptualize variability in the observed distance matrix as representing expected noise due to sampling realizations of a graph of a given size. Second, we isolate the test to

uniquely distance, rather than to the model as a whole, as would be the case with a likelihood-based measure. A likelihood ratio test for whether or not the curvature of the space is zero, for example, may seem to be an appealing alternative to our approach. Such a test would, however, confound changes in the latent geometry with changes in the fixed effects. To see this, recall that the surface area of the sphere changes as a function of the curvature. To preserve the overall density of the graph, therefore, the individual effects must change when the curvature changes. We sidestep this issue by leveraging the structure of the network formation model to isolate the test as specific to the latent geometry. Constructing an appropriate likelihood-based test, in contrast, would require marginalizing over the individual effects, which would be computationally intensive and require specifying distributions for the individual effects (which we do not require).

We address the question of how to choose the manifold $\mathcal{M}^{p^\star}(\kappa^\star)$, meaning the manifold class ($\mathcal{M}^\star$), curvature ($\kappa^\star$), and dimension ($p^\star$) of the latent space. We present a hypothesis testing framework which connects distances in the latent space to feasible embeddings on simply connected, complete Riemannian manifolds with constant curvature. Rather than relying on likelihood or cross-validation techniques, we directly leverage geometric results that provide necessary and sufficient conditions to embed points into particular manifolds, given pairwise distances between the points.

Our main insight is as follows. The manifolds we consider in this work all come equipped with a metric—an inner product which is used in calculation of distances between points. The metric uniquely identifies the manifold. Given a distance matrix D between K points, even without knowing where the points are located, we can check if the points can be isometrically embedded in the manifold. It can be embedded if and only if the distance matrix is compatible with the candidate manifold's metric. Specifically, given D, we can construct a test matrix $W_{\kappa^\star}(D)$ and check that the eigenvalue spectrum has the same signature as the manifold's metric. The manifold

detection problem is therefore reduced to testing the spectrum of $W_{\kappa^\star}(D)$.

We make two contributions. Our first contribution is in statistical geometry (Theorem 2.1.1). We show how to estimate the manifold consistently when observing a noisy estimate $\hat{D}$ of the true distances. For this, we must estimate the signature of the spectrum of $\hat{W}_{\hat{\kappa}}(\hat{D})$, the empirical test matrix. The logic relies on Weyl's inequality, which places bounds on the change in eigenvalues due to perturbations of a matrix. This provides an avenue for consistent estimation of $\mathcal{M}^{p^\star}(\kappa^\star)$.

Our second contribution is to then extend the argument to the latent space network model. The main idea is that we take the observed graph G on n nodes and construct some distance matrix $\hat{D}(G)$ among $K << n$ points and then apply our statistical geometry result. We define distance based on interaction rates between K *groups* of nodes. Using the observed graph, we define the distance between two cliques based on the probability of an edge between a node in clique C_1 and a node in clique C_2, which we can calculate using the definition of the latent space model plus the fraction of realized links between cliques. We estimate the matrix of cross-clique edge probabilities, and then use the latent space model to estimate the pairwise distances between cliques. We then leverage our statistical geometric result to estimate the manifold.

In the remainder of this section, we formally define the two problems we address and provide an overview of our approach. Specifically, we address (i) the general problem of estimating geometry from a noisy distance matrix and (ii) estimating network geometry from latent space models using cliques. Using cliques represents the most challenging case for our method since we expect that in many settings cliques may be relatively small. Next, in Section 2.2 we review key geometric concepts crucial for our testing procedure. Section 2.3 covers the general geometry problem in detail and develops our general method of estimating geometry from a noisy distance matrix. We turn to studying the estimation of the latent space in general using cliques in Section 2.4. In Section 2.5 we present simulation experiments that explore

the efficacy of our approach. We apply our results to two empirical examples in Section 2.6. The first empirical example considers data from 75 Indian village social networks, comprised of informal finance, information, and social links. We study the financial flows by geometry and also how the introduction of microfinance impacts geometry. The second example focuses on the neural network of the *C. Elegans* worm. Section 2.7 concludes. All proofs are in Appendix A.1 unless otherwise noted.

2.1.1 Statistical Geometry Problem

The estimation methods we propose are quite general. They apply broadly to a large set of problems (Section 2.3.1 provides examples) in which the researcher observes a noisy distance matrix $\hat{D}$ and wishes to estimate the properties of the underlying space. We make the following assumption about the latent space $\mathcal{M}^p(\kappa)$. After that we provide a broadly applicable classification theorem of $\mathcal{M}^p(\kappa)$ from an estimator $\hat{D}$ of distances computed between points in $\mathcal{M}^p(\kappa)$.

ASSUMPTION **2.1.1.** *$\mathcal{M}^{p^\star}(\kappa^\star)$ is a simply connected, complete Riemannian manifold of constant sectional curvature $\kappa^\star$, with $p^\star \in \mathbb{Z}$ with known upper bound and $\kappa^\star \in [-b,-a] \cup \{0\} \cup [a,b]$ with $a > 0, b > 0$.*

The technical geometric definition of simply connected, complete Riemannian manifolds is provided in a self-contained manner in Appendix A.8. By [100], Assumption 2.1.1 means that the manifold must be Euclidean, spherical, or hyperbolic with a bounded dimension and curvature value in some compact set. We emphasize that $a > 0$ means that in the curved cases, the geometry is not arbitrarily close to a Euclidean space ($\kappa = 0$). All such Riemannian manifolds are locally Euclidean, by definition, so this is required to be meaningful. We also emphasize that by dimension $p^\star$ we mean the minimum such dimension, as one can clearly embed $\mathcal{M}^{p^\star}$ in $\mathcal{M}^{p^\star+h}$ for $h > 0$.

Algorithm 2 takes in $\hat{D}$ and returns consistent estimates of the manifold type,

Algorithm 2: Consistent Estimation with a Noisy Distance Matrix

Input: noisy $K \times K$ distance matrix $\hat{D}$.

1. Estimate the curvature of the manifold for each of the curved cases $(\hat{\kappa}_S, \hat{\kappa}_H)$, from (2.7).
2. For each of the three candidate geometries, calculate the test matrix W_κ using the corresponding curvature estimate from step (1), with projection $J = I_K - 1_K 1_K^T / K$:

$$W_\kappa\left(\hat{D}\right) := \begin{cases} \frac{1}{\kappa}\cos\left(\sqrt{\kappa}\hat{D}\right) & \text{for } \kappa \neq 0 \\ -\frac{1}{2}J\hat{D} \circ \hat{D}J & \text{for } \kappa = 0. \end{cases}$$

3. Construct $\hat{\mathcal{M}}$, the estimate the manifold class, using Proposition 2.3.4.
4. Given $\hat{\mathcal{M}}$, select $\hat{p}$, the estimate of its dimension using Proposition 2.3.5.

curvature, and dimension. We use $\circ$ to denote the Hadamard product and 1_K denotes a vector of ones of length K.

THEOREM **2.1.1** (Consistent estimation with a noisy distance matrix)**.** *Let $\mathcal{M}^{p^\star}(\kappa^\star)$ be a latent space of unknown manifold class, dimension, and curvature that satisfies Assumption 2.1.1. Fix a set of $K > p^\star$ locations on $\mathcal{M}^{p^\star}(\kappa^\star)$ and suppose they uniquely identify the latent space. Suppose there is a sequence of $K \times K$ matrices $\hat{D}_T$, indexed by $T \in \mathbb{N}$, such that $\hat{D}_T \xrightarrow{p} D$ as $T \to \infty$. Under these assumptions, the estimators produced by Algorithm 2 are consistent as $T \to \infty$. That is, $\mathbb{P}(\hat{\mathcal{M}}^{\hat{p}} \neq \mathcal{M}^{p^\star}) = o(1)$ and $\hat{\kappa} - \kappa^\star = o_P(1)$.*

2.1.2 Latent Space Model

Having proposed a consistent classification method of the manifold class, dimension, and curvature from noisy distances (see Theorem 2.1.1), we now turn to the latent space model. Consider a graph $G = (V, E)$ where V are nodes and E are edges (also called links or connections), with $|V| = n$. For simplicity, we assume throughout that the graph is un-directed, all connections are symmetric, and unweighted, all connections are either present or absent. Our methods readily extend to the weighted and directed case, though it increases the complexity in terms of both notation and exposition. We assume that edges in G are drawn independently according to

$$\mathbb{P}\left\{G_{ij} = 1 \mid \nu^\star, z^\star, X_{ij}^\star, \mathcal{M}^{p^\star}(\kappa^\star)\right\} = \Lambda\left(\nu_i^\star + \nu_j^\star - d_{\mathcal{M}^{p^\star}(\kappa^\star)}(z_i^\star, z_j^\star)\right) , \tag{2.1}$$

for some increasing, invertible link function Λ. We represent $\nu = (\nu_1, \ldots, \nu_n)$ as the vector of individual effects, restricted to lie in some set to ensure (2.1) is a probability value in $[0, 1]$.[1] These are independent effects that encode individual gregariousness, and are related to the total number of connections [39, 73]. The $d_{\mathcal{M}^p(\kappa)}(z_i, z_j)$ terms represent the distance on the manifold $\mathcal{M}^p(\kappa)$, with dimension p and curvature κ, between locations z_i and z_j. Most of our analysis is done on the model above in (2.1) using an exponential link function, so $\Lambda(x) = \exp(x)$, but this is mostly out of convenience. In Appendix A.10, we show how to handle other common link functions (e.g., logistic link, [82]) or how to handle node- and pair-level covariates effects [73].

We now discuss how Assumption 2.1.1 applies to our network problem. Simple connectedness and completeness are innocuous and constant curvature provides a place to start and nests all manifolds used in the literature, but rule out inhomogeneous latent spaces entirely such as those with "structural holes" in the manifold, like the torus. Nevertheless, these three types of manifolds span a large and usable set of

[1] One way to model a directed graph is to allow each node to have two different fixed effects, ν_i and χ_i, one playing a role when i is the sender of the link (G_{ij}) and the other when i is the receiver (G_{ji}).

empirically relevant networks. With zero curvature, we model networks that allow for many paths where following them along nodes takes one increasingly far from nodes in other directions, while preserving local clustering. So while there is clustering, a flat space models a sort of vastness. Meanwhile, a sphere which has constant positive curvature does force such behavior. Following friends of friends of friends and so forth typically leads to encountering some distant friends in common at a much higher rate. Therefore there is a sort of cloistering in addition to clustering. Finally, hyperbolic spaces in contrast naturally embed trees or hierarchical networks or any context where expansiveness is a key feature. Intuitively this is because any set of initially parallel lines spread apart. Figure 2.2.2 presents intuitions. [160] provides a comprehensive discussion on the relationship between network properties and the latent space.

Assumption 2.1.2 is a mild assumption that ensures that (2.1) produces probabilities.[2]

ASSUMPTION **2.1.2.** *Every node i has a fixed-effect $\nu_i^\star$ i.i.d. from a distribution F_ν. The support of F_ν is required to be such that (2.1) always returns values in* $[0, 1]$.

If the link function is exponential, Assumption 2.1.2 requires that $\text{support}(F_\nu) \subseteq (-\infty, 0]$, but for the logistic link function Assumption 2.1.2 is satisfied for any distribution. Recall that our goal is to apply Theorem 2.1.1 to identify the manifold properties from just one network. We need to identify a set of K locations on the manifold and a sequence of estimators $\hat{D}_T$ that satisfy the assumptions in Theorem 2.1.1.

Our approach, which we study in detail in Section 2.4, exploits the clique structure in the network. Because of (2.1), nodes in even modest sized cliques (e.g., 5 nodes) are very likely to be close in the latent space. In other words, we can imagine that nodes in a clique are at the same location on the manifold. Finding K disjoint cliques

[2]Equivalently if one defined $\theta_i = \exp(\nu_i)$, then these fixed effects are simply multiplicative factors on the linking probability due to distances.

in the network therefore gives us K distinct points in the latent space. By counting the number of edges between pairs of these K cliques, we can therefore estimate the probability that nodes on the latent space connect. Since (2.1) relates distances in the latent space and edge probabilities, we can therefore use the estimated probability of connections to estimate distances between these K points.[3] We use the notation $C(\ell)$ to denote a clique of size ℓ in a graph, and $C_1(\ell), \ldots, C_K(\ell)$ refers to a collection of K cliques each of size ℓ.

Algorithm 3: Estimating Geometry of Latent Space Network Model using Cliques

1 Input: graph G.

1. Construct $\hat{D} = \hat{D}(G)$.

   (a) Identify K disjoint ℓ-cliques $C_1(\ell), \ldots, C_K(\ell)$.

   (b) Estimate the cross-clique linking probability

   $$\hat{P}_{kk'} = \frac{1}{\ell^2} \sum_{i \in C_k(\ell)} \sum_{j \in C_{k'}(\ell)} G_{ij}$$

   (c) Calculate $\hat{D}_{kk'} = -\log(\hat{P}_{kk'}) + \log(\hat{\gamma})$, where $\hat{\gamma}$ is an estimate of $E\{\exp(\nu)\}^2$.

2. Apply Algorithm 2 to $\hat{D}$ to construct estimator $\hat{\mathcal{M}}^{\hat{p}}(\hat{\kappa})$.

We require an assumption to ensure that observed cliques are likely to be comprised of nodes that are near each other in the latent space. Assumption 2.1.3 sets out a general requirement for F_z which makes explicit the condition that is required for our proofs: proportionally most of the cliques in the graph are comprised of nodes

[3] This makes clear that if the researcher observed weighted graph data, since G_{ij} is now a smooth function of distance and fixed effects, they can dispense with the clique approach altogether, since they directly observe a transformation of distances between specific points. The problem is easier.

that are proximate in latent space. This assumption captures a typical feature of latent space models and empirical data.

Let G be a graph drawn from (2.1). For an arbitrary set of nodes $V_0 \subseteq V$, let G_{V_0} denote the sub-graph induced by these nodes. If $|V_0| = \ell$, we use $\{G_{V_0} \in C(\ell)\}$ to denote the event that G_{V_0} is an ℓ-clique; that is, G_{V_0} is a complete graph on ℓ nodes.

ASSUMPTION **2.1.3.** *Every node i resides at a location $z_i^\star$ that is drawn independently and are identically distributed from a distribution F_z on manifold $\mathcal{M}^{p^\star}(\kappa^\star)$. The latent location distribution must satisfy two properties:*

(a) Identifiability: *The support of F_z must consist of at least $K > p^\star$ distinct points that uniquely identify the manifold.*

(b) Local cliques: *For any collection V_0 of ℓ nodes with locations drawn i.i.d. from F_z we have for all $\delta > 0$, $\mathbb{P}\{\max_{ij \in V_0} d(z_i, z_j) \leq \delta | G_{V_0} \in C(\ell)\} \to 1$, as $n, \ell \to \infty$.*

Part (a) states that we need K to be larger than the true dimension and that we need there to be only one latent space in which we can embed these points isometrically (Section 2.2.1 contains definitions of these terms). Part (b) states that given a clique of size ℓ, the probability that nodes in this clique are close to each other goes to 1 as the clique size and graph size grow. Aside from this condition, we impose no restrictions on the distribution of F_z. This allows for continuous, discrete, or mixed distributions as well as dependence on n. In Section 2.4.4 we provide a discussion on these two conditions in the context of the network model and provide high-level conditions which imply Assumption 2.1.3. We prove that these high-level conditions hold in common cases, such as when the node locations are drawn from a lattice model, Gaussian mixture model, or uniformly over a bounded but expanding region.

Before continuing, we emphasize a few key points. First, Assumption 2.1.3(b) is written with $n, \ell \to \infty$. Let $\ell = \ell(n)$ depend on the graph size. In order for there to

be cliques of size ℓ as $n \to \infty$, we need $\ell(n)$ to grow slowly, usually $\ell(n) \propto \log(n)$. Appendix A.11 shows that cliques of size $\log(n)$ exist with high probability as n grows for many common location distributions. Second, the existence of cliques is guaranteed by the latent space model under our assumptions as the number of nodes increases. The conditional independence relation that is key to the latent space model requires an assumption of exchangeability.[4] The Aldous-Hoover Theorem implies that exchangeable sequences of nodes correspond to dense graphs in the limit [4, 135], which implies that cliques are present in the limit. We also examine the existence of cliques using our empirical and simulated examples. We find that the number and size of cliques in our empirical examples is sufficient to match settings in simulations where the method controls Type 1 error and has high power.

THEOREM **2.1.2** (Estimating geometry via cliques)**.** *Let $\mathcal{M}^{p^\star}(\kappa^\star)$ be a latent space of unknown type, dimension, and curvature that satisfies Assumption 2.1.1. Consider a sequence of graphs on n nodes drawn from the distribution in (2.1), satisfying Assumptions 2.1.2-2.1.3. Under these assumptions, the estimators produced by Algorithm 3 are consistent as $n, \ell \to \infty$. That is, $\mathbb{P}(\hat{\mathcal{M}}^{\hat{p}} \neq \mathcal{M}^{p^\star}) = o(1)$ and $\hat{\kappa} - \kappa^\star = o_P(1)$.*

2.2 Overview of geometry and embedding conditions

In this section, we provide a brief overview of the three manifold types we consider in this work: Euclidean, spherical, and hyperbolic space. As noted above, these spaces span a class of empirically relevant manifolds on which much of the latent space network literature focuses [100]. We then provide necessary and sufficient conditions on an arbitrary distance matrix that ensures the points from which the distances are computed can be embedded isometrically into one of these three geometries.

[4] Large graphs with exchangeable nodes are only dense if the graph is a subgraph of an infinitely exchangeable graph. If the distribution of the graph changes with n, the limiting graph need not be dense.

2.2.1 Candidate geometries

In order to study the candidate manifold $\mathcal{M}^p(\kappa)$, we embed them in $\mathbb{R}^{p+1}$. Clearly the Euclidean case is trivial. In the spherical case we embed it in Euclidean space ($\mathbb{R}^{p+1}$ with the usual metric) and in the hyperbolic case we use Minkowski space ($\mathbb{R}^{1,p}$). Note that the only difference is that the bilinear form of the space, denoted by Q below, varies in signature described below.

The model for each is constructed by looking at a locus of points in the ambient space in which it is embedded:[5]

$$\mathcal{M}^p(\kappa) := \left\{x \in \mathbb{R}^{p+1} : \ Q(x,x) = \kappa^{-1}\right\}.$$

This implies a way of calculating distances between points on the manifold. Specifically,

$$d_{\mathcal{M}^p}(x,y) = \frac{\arccos\left\{\kappa Q(x,y)\right\}}{\sqrt{\kappa}}.$$

Let us turn to our candidate cases. The Euclidean space $\mathbb{R}^p$ is the p-dimensional Euclidean space with the usual Euclidean metric. In the case of the sphere $\mathbf{S}^p$, we have the usual Euclidean inner product $Q_{\mathbb{R}^{p+1}}(x,y) := \sum_{i=1}^{p+1} x_i y_i$. The locus of points and distances between two points $x, y \in \mathbb{R}^{p+1}$ for the embedding is

$$\mathbf{S}^p(\kappa) := \left\{x \in \mathbb{R}^{p+1} : \ Q_{\mathbb{R}^{p+1}}(x,x) = \kappa^{-1}, \ \kappa > 0\right\} \text{ and } d_{\mathbf{S}^p}(x,y) = \frac{\arccos\left\{\kappa Q_{\mathbb{R}^{p+1}}(x,y)\right\}}{\sqrt{\kappa}}.$$

Hyperbolic space $\mathbf{H}^p$ is embedded in Minkowski space, $\mathbb{R}^{1,p}$ which is $\mathbb{R}^{p+1}$ equipped with the Minkowski bilinear form: $Q_{\mathbb{R}^{1,p}}(x,y) := -x_0 y_0 + \sum_{i=1}^{p} x_i y_i$. The important point is that the signature is distinguished from the Euclidean space which will play a

[5]For the hyperboloid $x_0 > 0$ is an additional restriction.

key role in distinguishing the geometries. The locus of points and distances are given by

$$\mathbf{H}^p(\kappa) := \left\{x = (x_0, x_{1:p}) \in \mathbb{R}^{1,p} : \ Q_{\mathbb{R}^{1,p}}(x,x) = \kappa^{-1}, \ x_0 > 0, \ \kappa < 0\right\}$$

and

$$d_{\mathbf{H}^p}(x,y) = \frac{\arccos\left\{\kappa Q_{\mathbb{R}^{1,p}}(x,y)\right\}}{\sqrt{\kappa}}.$$

2.2.2 Isometric Embedding Conditions

Equipped with a notion of how distances are calculated between points in our candidate manifolds, we briefly review the conditions to check if a collection of K points can be isometrically embedded in each of the manifolds.

Let D be a known distance matrix from K points given by $Z = \{z_1, \ldots, z_K\}$. We say that Z can be *isometrically embedded* in manifold $\mathcal{M}^p(\kappa)$, written as $Z \overset{\text{isom}}{\to} \mathcal{M}$, if there exists an isometry ϕ such that for all $l, l' \in \{1, \ldots, K\}$, $d_{\mathcal{M}}(\phi(z_l), \phi(z_{l'})) = d_{ll'}$.

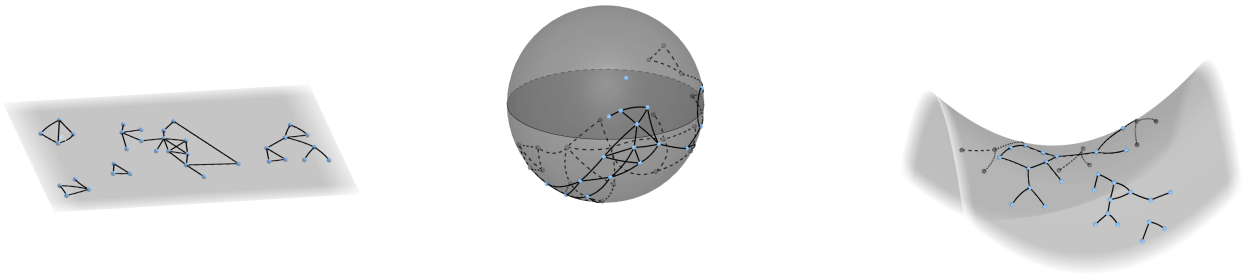
Given D, we define the $K \times K$ matrix $W_\kappa(D)$ which will allow us to determine whether an isometric embedding is possible in one of the three candidate geometries. To do this, we choose the matrix to correspond to the bilinear form $Q(\cdot,\cdot)$ above. For the Euclidean case we need a matrix $J := I_K - \frac{1}{K} 1_K 1_K'$. Our test matrix is given by

$$W_\kappa(D) := \begin{cases} \frac{1}{\kappa}\cos\left(\sqrt{\kappa}D\right) & \text{for } \kappa \neq 0 \\ -\frac{1}{2}JD \circ DJ & \text{for } \kappa = 0, \end{cases} \tag{2.2}$$

where we apply the cosine operation element-wise, as before. We write $W_\kappa = W_\kappa(D)$, suppressing the dependency on D unless otherwise noted. By using a Taylor series of $W_\kappa(D)$ around $\kappa = 0$, one can see the relationship between the expression of $W_\kappa(D)$ for $\kappa > 0$ and $W_0(D)$.[6]

The following lemma characterizes the conditions for isometric embedding and is a concise restatement of classical results: [155] Theorem 1 (which we include as part

[6] We would like to thank Gabriel Caroll for pointing this out to us.



(a) Flat $\kappa^\star = 0$ (b) Positive $\kappa^\star > 0$ (c) Negative $\kappa^\star < 0$

Figure 2.2.1: How curved geometries affect network embeddings where each displayed graph has 36 nodes.

(1) of our Lemma 2.2.1) and [18] Theorem 1 (which we include as parts (2-3) of our Lemma 2.2.1). See [19] for an overview of related topics in distance geometry. The *signature* of a square matrix A is a triple $\text{sig}(A) = (a, b, c)$, where a, b, and c are respectively the number of positive, zero, and negative eigenvalues of A. A positive semi-definite matrix has $c = 0$. Throughout the chapter, we use the convention that $\lambda_{\max}(A) := \lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_K(A) =: \lambda_{\min}(A)$ are the eigenvalues of the $K \times K$ matrix A sorted in decreasing order.

LEMMA **2.2.1.** *Let* $p_{\min}$ *be the minimum dimension for which* $Z \overset{isom}{\to} \mathcal{M}^p(\kappa)$*, and assume* $a > 0$*.*

1. $Z \overset{\text{isom}}{\to} \mathbb{R}^p$ *for some* p *if and only if* $sig(W_0) = (a, K - a, 0)$*. Further,* $p_{\min} = a$*.*
2. $Z \overset{isom}{\to} \mathbf{S}^p(\kappa)$ *if and only if* $\text{sig}(W_\kappa) = (a, K - a, 0)$*. Further,* $p_{\min} = a - 1$*.*
3. $Z \overset{isom}{\to} \mathbf{H}^p(\kappa)$ *if and only if* $\text{sig}(W_\kappa) = (1, K - a - 1, a)$*. Further,* $p_{\min} = (K - a) - 1$*.*

The result above tells us, for example, that $\lambda_{\min}(W_0) \geq 0$ when D is computed from points in $\mathbb{R}^p$. This allows us to then phrase the problem of geometry identification,

where we do not observe the manifold, into a problem about eigenvalues of W, which we do observe. This re-framing of the statistical geometry problem is the main insight behind our geometry classification procedure in Theorem 2.1.1. The lemma also allows us to estimate the dimension of the latent space through the rank of W_κ, which provides the basis of our dimension estimation procedure in Section 2.3.6.

2.3 Testing geometry from an arbitrary distance matrix estimate $\hat{D}$

This section addresses how to test whether a set of K points can be isometrically embedded into a candidate manifold out of the set we consider, given only a consistent estimator $\hat{D}$ of the pairwise distance matrix D between them. We develop the constituent pieces for our main result, Theorem 2.1.1.

This section, therefore, is not about networks exclusively but rather about a general statistical geometry problem that we outlined in Theorem 2.1.1. We show below that the general statistical geometry problem outlined in Section 2.1.1 is applicable in a wide range of relational data examples, extending the scope of our work beyond the binary adjacency matrix case we consider in Section 2.4. Each of the following examples will produce an estimator $\hat{D}$ which approximates some unknown matrix D, which contains pair-wise distances between K objects (people, nodes, firms, etc.). For the asymptotic results presented below, index $\hat{D}$ by some T such that $\hat{D} = \hat{D}_T \xrightarrow{p} D$ as $T \to \infty$. Informally, T maybe be thought of as the sample size. In our main application to networks, T is a function of network size n. However, the general statistical geometry problem may generate an estimator of distances in other ways. For example, longitudinal data where T indexes time in an international trade example or number of samples in a neuroscience example. For convenience, we often drop the notation $\hat{D}_T$ and instead simply write $\hat{D}$.

Because the results of this section are more general than the network model studied in equation (2.1), our main application, we provide a few examples to develop an intuition for other potential applications. These examples deal with a general set of

problems where we observe weighted relationships between pairs of nodes, and these relationships are formed based on distances or dissimilarities between nodes. The latent space model is a special case where the relationship is binary. The geometric results in this section are not confined to even these applications – in fact, our estimation procedure in Theorem 2.1.1 only requires a consistent estimate of distances along the latent space.

2.3.1 Examples of Relational Data with Distances, $\hat{D}$

We provide four motivating examples for $\hat{D}$.

EXAMPLE **1** (A Single Large Network)**.** *The researcher only observes a single large network, G, drawn from the distribution specified in equation (2.1). We study it in detail in Section 2.4.*

EXAMPLE **2** (Relational data, [138, 1, 69, 174, 153])**.** *There are K units, such as individuals, neurons, sensors, or firms. The researcher observes an outcome of an interaction between two units i and j, given by $f_{ij,t}$ at time $t = 1, \ldots, T$ with some disturbance $\epsilon_{ij,t}$. For instance, $f_{ij,t} = \Lambda\left\{d_{\mathcal{M}^{p^\star}(\kappa^\star\}}\left(z_i^\star, z_j^\star\right) + \epsilon_{ij,t}\right)$ where Λ is a bijective function. This may be binary or continuous, such as an instance of a signal being transmitted between neurons or sensors, some financial flows between individuals, or some transactions between firms. We can then compute $\hat{D}$ with entries $\hat{d}_{ij} = \frac{1}{T}\sum_t \Lambda^{-1}\left(f_{ij,t}\right)$ and under regularity conditions (such as differentiability of Λ^{-1}) our results will follow.*

A particularly relevant case is the following. The researcher observes T networks $G_1, \ldots, G_T$, where T could represent the number of observations of one network with a fixed set of nodes, or it could represent the number of observed networks, each with a potentially distinct set of nodes. Here we write $p_{ij} := \mathbb{P}(g_{ij} = 1 | z_i^\star, z_j^\star)$ is the probability that nodes i and j connect, given their latent space locations. This term depends on $d_{\mathcal{M}}(z_i^\star, z_j^\star)$. We write this as $p_{ij} = H_{ij}\{d_{\mathcal{M}}(z_i^\star, z_j^\star)\}$ for some invertible function H_{ij}.

Here we suppose that the generative model for the networks is constant across all T networks. We can then estimate $\hat{D}_{ij} = H^{-1}(\hat{P}_{ij})$, where $\hat{P}_{ij} = T^{-1}\sum_{i=1}^{T} G_{ijt}$ is the number of observed edges between nodes i and j, normalized by the number of observations T.

EXAMPLE **3** (Trait Groups, [101, 123])**.** *The n nodes each have one of K traits and the number with a trait k is given by n_k. Locations in the latent space are determined uniquely by a node's trait, and nodes with the same trait share the same location on the latent space, with $z^{\star}_{k_i}$ denoting the common location of nodes with trait k_i. An interaction between nodes follows $f_{ij} = \Lambda\left\{d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z^{\star}_{k_i}, z^{\star}_{k_j}\right) + \epsilon_{ij}\right\}$ where again $\Lambda(\cdot)$ is bijective. In this example, the researcher can construct $\hat{D}$ with entries*

$$\hat{D}_{kk'} = \frac{1}{n_k n_{k'}} \sum_{i,j} \Lambda^{-1}\left(f_{ij} \cdot \mathbf{1}\left\{k_i = k, k_j = k'\right\}\right) \ ,$$

where n_k is the number of nodes with trait k, which can be obtained from census or surveys.

2.3.2 Perturbation

In Section 2.2.2 we saw that the isometric embedding conditions related the manifold class, curvature, and dimension to the spectrum of a test matrix $W_\kappa(D)$. In practice, since we observe $\hat{D}$, we must construct $\hat{W}_{\hat{\kappa}}(\hat{D})$ and study its spectrum. The main idea of our approach to estimating the manifold class, curvature, and dimension comes from Weyl's inequality, which says that the eigenvalues of the estimated test matrix, $\hat{W}_{\hat{\kappa}}(\hat{D})$ are very close to those of the target $W_\kappa(D)$ if the estimators of the distance matrix and curvature are consistent. Under our assumptions, we are able to bound how the estimated spectrum may deviate from the true spectrum.

PROPOSITION **2.3.1.** *Suppose that D is a $K \times K$ distance matrix from K points on $\mathcal{M}^{p^\star}(\kappa^\star)$ satisfying Assumption 2.1.1, with K chosen such that the $\mathcal{M}^{p^\star}(\kappa^\star)$ is uniquely identified. Assume further that there are estimators $\hat{D}$ and $\hat{\kappa}$ such that*

$\hat{D}_T - D \xrightarrow{p} 0$ and $\hat{\kappa} \xrightarrow{p} \kappa^\star$ as $T \to \infty$. Let θ_α be defined as the αth quantile of the distribution of $\|\hat{W}_{\hat{\kappa}} - W_\kappa\|_F$. Then, for every $k \in \{1, \ldots, K\}$,

$$\mathbb{P}\left\{|\lambda_k(\hat{W}_{\hat{\kappa}}) - \lambda_k(W_\kappa)| < \theta_\alpha\right\} \leq \alpha\,. \tag{2.3}$$

Owing to this result, we can study the estimated spectrum in order to look at the metric signature and therefore estimate the manifold class, curvature, and dimension.

2.3.3 Hypothesis Tests of Geometry

We now frame the problem of classifying the geometry of D into three hypothesis tests. We will then combine the results of these three tests into a classifier of the geometry. To determine the geometry of $\mathcal{M}$, we use Lemma 2.2.1 and develop testable statements about the spectrum of the test matrix W_κ.

For the Euclidean case, we can test positive semi-definiteness of the test matrix as

$$H_{0,e} : \lambda_K(W_0) \geq 0,\;\; H_{a,e} : \lambda_K(W_0) < 0. \tag{2.4}$$

Using the same reasoning, for the spherical case we can test the hypothesis that the embedding space is spherical for some $\kappa > 0$ as

$$H_{0,s} : \lambda_K(W_\kappa) \geq 0, \quad H_{a,s} : \lambda_K(W_\kappa) < 0\,. \tag{2.5}$$

since the test matrix must be positive semi-definite.

Finally, to determine if the embedding space is hyperbolic for some $\kappa < 0$, we want to test

$$H_{0,h} : \lambda_2(W_\kappa) = 0, \quad H_{a,h} : \lambda_2(W_\kappa) \neq 0 \tag{2.6}$$

since the signature switches sign and that the matrix does not have full rank under the assumption on dimension implies that there are zeros in the spectrum.[7]

[7]By Lemma 2.2.1, failing to reject in the hyperbolic case $\lambda_2(W_\kappa) = 0$ is not enough to conclude that D is hyperbolic, since we must test the first is positive and smallest eigenvalue is negative as well. In practice, however, we found that testing only one eigenvalue was sufficient and, thus, use this simpler test. Clearly, it would also be possible to test all three eigenvalues using an intersection test, which we leave to future work.

A natural first step in deriving a geometry classifier would be to use Proposition 2.3.1. While this method produces a type-1 error that is below α, the power of the method may be low. So as a classifier, that procedure can be improved upon. In Section 2.3.5 we derive more powerful tests by approximating the distribution of the eigenvalues under the assumption that the distances are computed along one of the three geometries (recall from Lemma 2.2.1 that the eigenvalues of W_κ tell us the underlying geometry type).

To derive tests of the three geometry hypotheses, we first need to estimate the curvature of $\mathcal{M}$, which is computed assuming the geometry of $\mathcal{M}$ is curved (i.e., not Euclidean). We use these estimated curvature values to then identify the geometry type. For this reason, we start our discussion by studying curvature.

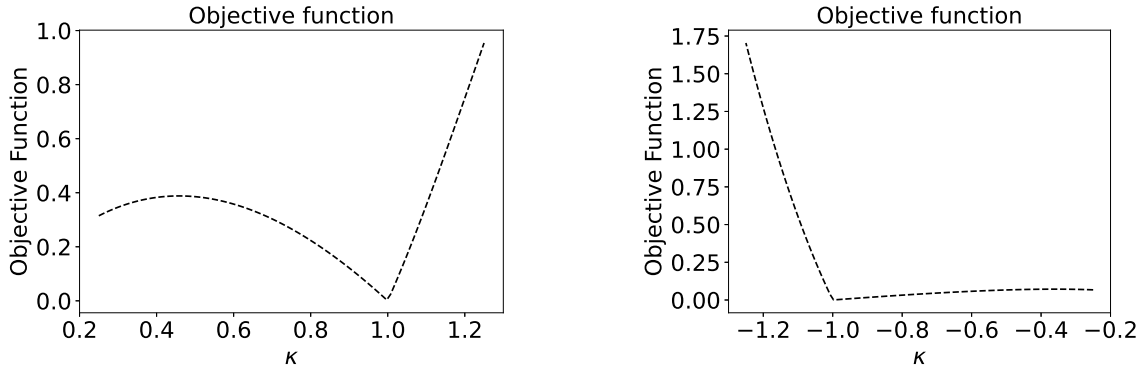
2.3.4 Estimating Curvature $\kappa^\star$

We begin assuming that the researcher has a consistent estimator $\hat{D}_T \xrightarrow{p} D$ (which we will develop below). Equipped with $\hat{D}_T$, we construct a consistent estimate of $\kappa^\star$. The core observation comes from Lemma 2.2.1. Namely, by looking at $\text{sig}(W_\kappa)$, we can use the fact that certain eigenvalues must *exactly* be zero under the various geometries. For instance, both $\lambda_K(W_0(D)) = 0$ and $\lambda_K(W_{\kappa^\star}(D)) = 0$ for the Euclidean and spherical cases, respectively, as both are positive semi-definite, provided $p^\star < K$. A similar phenomenon is true for $\lambda_2(W_{\kappa^\star}(D)) = 0$ for the hyperbolic case.

This observation leads to the following estimators of the curvature:

$$\hat{\kappa}_S := \underset{\kappa\in[a,b]}{\arg\min} \left|\lambda_1\left\{\kappa W_\kappa(\hat{D}_T)\right\}\right|, \qquad \hat{\kappa}_H := \underset{\kappa\in[-b,-a]}{\arg\min} \left|\lambda_2\left\{\kappa W_\kappa(\hat{D}_T)\right\}\right| \tag{2.7}$$

for some $0 < a < b$. In Appendix A.3, we discuss how to pick a and b in practice. The subscript S indicates that this estimate is used when testing if the manifold is spherical. Similarly, the subscript H indicates that this estimate is used when testing if the manifold is hyperbolic. As $T \to \infty$, the estimates approach the true curvature under the correct geometry.



(a) Spherical

(b) Hyperbolic

Figure 2.3.1: Plot of the objective function from (2.7) when D corresponds to 15 points in $\mathbf{S}^2(1)$ (left) and $\mathbf{H}^2(-1)$ (right). We plot the curvature κ against the value of the function $\kappa \mapsto \left|\lambda_1\left(\cos(\sqrt{\kappa}D)\right)\right|$ (left) and of the function $\kappa \mapsto \left|\lambda_2\left(\cos(\sqrt{\kappa}D)\right)\right|$ (right). We see that at the true κ, the objective function is minimized.

PROPOSITION **2.3.2** (Consistency of curvature estimates)**.** *Suppose that D is a $K \times K$ matrix containing pairwise distances between points in either $\mathbf{S}^p(\kappa^\star)$ or $\mathbf{H}^p(\kappa^\star)$, where $|\kappa^\star| > 0$. Let Z denote the collection of these K points. Suppose there is an estimate $\hat{D}_T$ such that $\hat{D}_T \xrightarrow{p} D$ as $T \to \infty$. Finally, suppose that there is either*

1. *a unique $\kappa^\star \in [a, b]$ such that $Z \overset{isom}{\to} \mathbf{S}^p(\kappa)$ for some p. Set $\hat{\kappa}_T = \hat{\kappa}_S$.*

2. *or a unique $\kappa^\star \in [-b, -a]$ such that $Z \overset{isom}{\to} \mathbf{H}^p(\kappa)$ for some p. Set $\hat{\kappa}_T = \hat{\kappa}_H$.*

In cases (1) and (2), $\hat{\kappa}_T \xrightarrow{p} \kappa^\star$ as $T \to \infty$.

The estimators we propose for curvature are similar to those proposed in [171], though [171] does not prove that these estimators are consistent.

Before continuing, we want to discuss the requirements in Proposition 2.3.2. First, Proposition 2.3.2 requires that $\kappa^\star$ is bounded away from zero (meaning that the space

is not flat and hence has non-zero curvature). In other words, the manifold is either spherical or hyperbolic, and a and b must be chosen to include the true curvature value $\kappa^\star$. We also require that there is a unique curvature in which we can find an isometric embedding to ensure that the statement that $\hat{\kappa}_T \xrightarrow{p} \kappa^\star$ makes sense. We also want to emphasize that $\kappa^\star$ is fixed and again is assumed to be non-zero. The case where $\kappa^\star$ changes with T is an interesting and challenging problem that we leave to future work.

2.3.5 Estimating Geometric Class $\mathcal{M}^\star$

Again, we suppose the researcher has access to a noisy distance matrix $\hat{D}_T$ that approximates an unknown distance matrix D of interest. The matrix D consists of pair-wise distances between K objects along the surface of $\mathcal{M}$.

Consistent Tests for Latent Geometry

We begin by showing a consistent testing framework and then give a bootstrap method for implementation in Algorithm 7.

To test $H_{0,e}$, first, define a rejection region $\mathcal{R}_T = (-\infty, \delta_T]$ for some real-valued sequence $\delta_T \in (-\infty, 0]$ and we define our test $\phi_T(\hat{W}_0) \in \{0, 1\}$ as

$$\phi_T(\hat{W}_0) = \begin{cases} 0, & \lambda_K(\hat{W}_0) \in \mathcal{R}_T \\ 1, & \lambda_K(\hat{W}_0) \notin \mathcal{R}_T \ . \end{cases} \tag{2.8}$$

If the test is 0, this indicates that we fail to reject the null hypothesis $H_{0,e}$ while if the test is 1, we reject the null hypothesis $H_{0,e}$. We use this notation throughout the chapter when discussing the output of a hypothesis test.

When testing if D is spherical, let $\hat{\kappa}$ denote an estimate of κ, defined in Proposition

2.3.2. We define our rejection region of $H_{0,s}$ as $\mathcal{R}_T = (-\infty, \delta_T]$ and our test as

$$\phi_T(\hat{W}_{\hat{\kappa}}) = \begin{cases} 0, & \lambda_K(\hat{W}_{\hat{\kappa}}) \in \mathcal{R}_T, \\ 1, & \lambda_K(\hat{W}_{\hat{\kappa}}) \notin \mathcal{R}_T \ , \end{cases} \tag{2.9}$$

which is a similar test of positive semi-definiteness.

Finally, when testing if D is hyperbolic, let $\hat{\kappa}$ denote an estimate of $\kappa < 0$, defined in Proposition 2.3.2. We define our rejection region of $H_{0,h}$ as $\mathcal{R}_T = [\delta_T, \infty)$ and define our test as

$$\phi_T(W_{\hat{\kappa}}) = \begin{cases} 0, & \lambda_2(\hat{W}_{\hat{\kappa}}) \in \mathcal{R}_T, \\ 1, & \lambda_2(\hat{W}_{\hat{\kappa}}) \notin \mathcal{R}_T \ , \end{cases} \tag{2.10}$$

which looks to reject positivity of the second eigenvalue as per the metric signature.

We now study what conditions must hold on this sequence δ_T in order for the three tests to be consistent, by which we mean that the probability the test rejects the null goes to 1 under the alternative hypothesis and that the probability it fails to reject the null goes to 1 under the null.

PROPOSITION **2.3.3.** *Let $\delta_T = o_P(1)$ be a random or deterministic sequence and let Assumption 2.1.1 hold. Let $\hat{D}_T \xrightarrow{p} D$ as $T \to \infty$. Then,*

1. *If $\delta_T \in (-\infty, 0]$, $\delta_T = o_P(1)$ and $\mathbb{P}\left\{\lambda_K(\hat{W}_0) \leq \delta_T\right\} = 1 - o(1)$, then the test for $H_{0,e}$ in (2.4) with rejection region $\mathcal{R}_T := (-\infty, \delta_T]$ is consistent.*

2. *If $\delta_T \in (-\infty, 0]$, $\delta_T = o_P(1)$ and $\mathbb{P}\left\{\lambda_K(\hat{W}_{\hat{\kappa}}) \leq \delta_T\right\} = 1 - o(1)$ with $\hat{\kappa} \in [a, b]$, then the test for $H_{0,s}$ in (2.5) with rejection region $\mathcal{R}_T := (-\infty, \delta_T]$ is consistent.*

3. *If $\delta_T \in [0, \infty)$, $\delta_T = o_P(1)$ and $\mathbb{P}\left\{\lambda_2(\hat{W}_{\hat{\kappa}}) \geq \delta_T\right\} = 1 - o(1)$ with $\hat{\kappa} \in [-b, -a]$, then the test for $H_{0,h}$ in (2.6) with rejection region $\mathcal{R}_T := [\delta_T, \infty)$ is consistent.*

In order to combine these tests into a single estimate of the latent space geometry, we suggest using an ordered test. There are 6 possible orderings of such a test (e.g,

Euclidean, then spherical, then hyperbolic). The proof is simple and uses the fact that each of the three geometry tests is consistent, under suitable assumptions on the threshold sequence δ_T.

PROPOSITION **2.3.4** (Consistent estimation of geometry type)**.** *Under the assumptions on the sequence δ in Proposition 2.3.3, any of the 6 ordered tests return a consistent estimate of the latent space geometry.*

We omit the proof of Proposition 2.3.4 which follows immediately from Proposition 2.3.3. We have shown that we can use the observed distance matrix $\hat{D}_T$ to test the hypotheses that the latent space is Euclidean, spherical, or hyperbolic. From these tests we define $\hat{\mathcal{M}}^{\hat{p}}(\hat{\kappa})$ as the intersection of the three tests. That is, the estimated latent geometry based on $\hat{D}_T$ is defined by the result of three hypothesis tests in equations (2.8), (2.9), and (2.10). More specifically, we can select any estimator that preserves consistency. As noted, for example, we can use an ordered test to estimate the geometry type. Thanks to Proposition 2.3.3, with sufficiently large T the probability that more than one of these tests will fail to reject the null goes to zero, leading to a consistent test.

2.3.6 Estimating Dimension $p^\star$

Given $\hat{D}$, $\hat{\kappa}$, and $\hat{\mathcal{M}}$, we develop a consistent estimate of $p^\star$, the minimal dimension of the manifold class in which the points can be embedded. We focus on the minimum dimension since, trivially, if one can embed in p one can embed in $p' > p$ for all p'. As we noted in Lemma 2.2.1, we see that $p^\star$ relates to the rank. So we proceed by estimating the rank of $W_\kappa(D)$.

We present two approaches. The first continues our use of the logic of Weyl's inequality to propose a consistent estimate as $T \to \infty$. The second uses the laddle plot method of [120]. We did not verify the required assumptions in [120] so we do not claim their estimator is consistent in our problem, but we find in practice (as do

they) that the estimator performs well, so we suggest practitioners actually use this.

Spectral estimate of dimension

We are interested in finding the rank of W_κ using $\hat{W}_{\hat{\kappa}}$. To do this, we let ϵ_T be a (potentially random) sequence such that ϵ_T goes to zero slower than any $\lambda_j(\hat{W}_{\hat{\kappa}})$ for which $\lambda_j(W_\kappa) = 0$. In other words, we select ϵ_T to go zero slower than the slowest zero eigenvalue of the test matrix W_κ.

Since the rank of a matrix is the number of non-zero eigenvalues, we will estimate the number of non-zero eigenvalues of W_κ. To do this, we define a rejection region $\mathcal{R}_T = (-\epsilon_T, \epsilon_T)$. For any index T, define the estimated rank to be

$$\widehat{\text{rank}}(W_\kappa) = \#\{j = 1, \ldots, K : \lambda_j(\hat{W}_{\hat{\kappa}}) \notin \mathcal{R}_T\} \, . \tag{2.11}$$

Our estimate of the rank is then the number of observed eigenvalues that are sufficiently far away from zero, as measured by the threshold sequence ϵ_T.

Clearly, the performance of this estimator depends on the choice of the threshold sequence ϵ_T. If ϵ_T does not converge to zero (in probability), then this estimate cannot be consistent, since it will eventually start to count zero eigenvalues of W_κ as being non-zero. Hence convergence to zero is a necessary condition. It must also converge fast enough to zero. For example, if $\lambda_j(W_\kappa) = 0$ and we have access to $\lambda_j(\hat{W}_{\hat{\kappa}}) = 1/T^2$ (i.e., a deterministic sequence), and we use $\epsilon_T = 1/T$, then we will under-count the rank of W_κ because we will classify the eigenvalue $\lambda_j(W_\kappa)$ as non-zero at every T.

We therefore must pick an ϵ_T that goes to zero slower than all estimates of eigenvalues for which their true counterpart is zero. From Weyl's inequality, we know what such sequences look like. For any index k for which $\lambda_k(W_\kappa) = 0$,

$$|\lambda_w(\hat{W}_{\hat{\kappa}})| = |\lambda_k(\hat{W}_{\hat{\kappa}}) - \lambda_k(\hat{W}_{\hat{\kappa}})| \leq ||\hat{W}_{\hat{\kappa}} - W_\kappa||_F \, ,$$

where the inequality is due to Weyl's. By defining $r_T := ||\hat{W}_{\hat{\kappa}} - W_\kappa||_F$, from the conditions in Theorem 2.1.1, we know that $r_T = o_P(1)$. We need to select $\epsilon_T \to 0$ with $r_T/\epsilon_T \to 0$, which means that ϵ_T goes to zero slower than r_T does.

PROPOSITION **2.3.5** (Consistency of minimum dimension estimate)**.** *Choose* $\epsilon_T \to 0$ *such that* $r_T/\epsilon_T \to 0$*. If the assumptions in Theorem 2.1.1 hold, then* $\widehat{\text{rank}}(W_\kappa) \xrightarrow{p} \text{rank}(W_\kappa)$ *as* $T \to \infty$*. Using Lemma 2.2.1, we can therefore consistently estimate* $p^\star$ *as* $T \to \infty$*.*

While the rank estimator above leads to a consistent estimate for suitable chosen ϵ_T, choosing a sequence that satisfies these conditions in practice is challenging. We therefore recommend in practice to estimate the rank with a different estimator.

Rank estimator from [120]

Recent work by [120], however, has been shown to have more appealing finite sample performance and in Appendix A.5, we provide the algorithm to estimate the rank with this method. The intuition for their approach is as follows. Looking at the scree plot (related to our above approach) is consistent. And a bootstrap procedure, leveraging the fact that the eigenvectors corresponding to indices beyond the rank will be uncorrelated in a bootstrap) also performs well. They note that under certain regularity conditions (a sufficiently fast estimate of the matrix W_κ, a self-similar bootstrap estimator), the combination of these two—a scree plot together with a bootstrap evaluation of eigenvector uncorrelatedness—performs better than either.

[120] prove that in a number of problems, their rank estimate is consistent. We are not able to verify these conditions in practice, since they assume that their data consists of i.i.d. data. However, in our case, our data is independent but *not* identically distributed. Therefore, we do not make a claim about the consistency of this approach when applied to our problem. We do, however, note that in simulations this approach has better finite sample properties.

To summarize, through Propositions 2.3.2, 2.3.4, and 2.3.5 we have established that the procedure in Algorithm 2 is consistent for its estimands, as claimed in Theorem 2.1.1.

Returning to our four examples, it is easy to see that Examples 2 and 3 immediately satisfy the assumptions in Theorem 2.1.1, and as a consequence in each of those cases we can consistently recover the geometry. The situation is more subtle for Example 1, where rather than directly observing K units and noisy distances among them, the researcher observes some graph G is more subtle. From the graph, distances must be constructed and then geometry estimated. This is the subject of Sections 2.4.

2.4 Identifying the latent space using only graph data

Having developed statistical tests to estimate the geometry behind a collection of points when we observe and arbitrary noisy distance matrix (the statistical geometry problem from Section 2.1.1), we now turn the original network problem. Specifically, our goal is to use a graph drawn from the latent space model in (2.1) and estimate the type, curvature, and dimension of the latent space. Theorem 2.1.2 presents the answer to this problem and demonstrate how the manifold can be consistently estimated. The proof of the theorem is provided in Appendix A.1. In this section, we lay out the ingredients: how one constructs the noisy distance matrix and adjusts for fixed effects. With these we are able to prove Theorem 2.1.2, which states that Algorithm 3 returns a consistent estimate of $\mathcal{M}^{p^\star}(\kappa^\star)$.

Importantly, we also provide a number of examples of node location distributions, motivated by empirically-relevant models, to demonstrate that our core Assumption 2.1.3 for the clique-based method holds (Section 2.4.4). We conclude by discussing some practical choices for implementation.

2.4.1 Estimating Distances using Graph Data

In order to apply the geometric test, we use the graph G to estimate a set of distances on $\mathcal{M}^{p^\star}(\kappa^\star)$. Since the probabilities defined by the latent space model in (2.1), our approach is to use estimated linking frequencies in order to identify a system of implied

distances between K points in the manifold.

To motivate our approach to constructing a distance matrix, we consider a simplification of the main model in (2.1). We make two assumptions for illustration, which are subsequently relaxed in the main analysis. First, suppose there are no individual effects (so $\nu_i^\star = 0 \ \forall i$). Second, suppose that nodes are assigned to one of K distinct points in the latent space, which we denote by $\zeta_1^\star, \ldots, \zeta_K^\star$. Define $V_k = \{j \in \{1, \ldots, n\} : z_j^\star = \zeta_k^\star\}$ to be the set of nodes at location $z_k^\star$. Under this simplification, we can write the distance between points $z_k^\star$ and $z_{k'}^\star$ using the definition of the latent space model in (2.1) as

$$d_{k,k'} = -\log(p_{k,k'}) \tag{2.12}$$

where $p_{k,k'} := \mathbb{P}(G_{k,k'} = 1 | z^\star)$ is the probability that nodes at locations $z_k^\star$ and $z_{k'}^\star$ connect for any $k, k' \in \{1, \ldots, K\}$.[8] Then, we can estimate the probability $p_{k,k'}$ by

$$\hat{p}_{k,k'} := \frac{1}{|V_k||V_{k'}|} \sum_{(i,j) \in V_k \times V_{k'}} G_{ij} \ . \tag{2.13}$$

In words, this estimator counts the number of observed edges between $z_k^\star$ and $z_{k'}^\star$ and divides by the number of possible edges, given by $|V_k||V_{k'}|$. Since $G_{ij} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p_{k,k'})$ for $(i,j) \in V_k \times V_{k'}$, this estimator is unbiased for $p_{k,k'}$. In addition, supposing that $|V_i| \to \infty$ as $n \to \infty$, the weak law of large numbers implies that $\hat{p}_{k,k'} - p_{k,k'} \xrightarrow{p} 0$.

By (2.12), we can estimate a $K \times K$ distance matrix $\hat{D} = \{\hat{d}_{kk'}\}$ comprised of entries with distances between ζ_k and $\zeta_{k'}$ by

$$\hat{d}_{kk'} = -\left[\log(\hat{p}_{k,k'})\right]_{k,k'} .$$

We can then apply the continuous mapping theorem to show that $\hat{d}_{k,k'} - d_{k,k'} \xrightarrow{p} 0$. That is, the assumption in Theorem 2.1.1 that we have access to a consistent estimate of distances along the latent space holds. Here, the sample size T is the number of

[8] Clearly in this simplified model this corresponds to a stochastic block model with K communities with a linking probability law $p_{kk'}$ that satisfies a geometric restriction.

edges between the K points on the latent space, given by $|V_k||V_{k'}|$ for every pair of points $\zeta_k^\star$ and $\zeta_{k'}^\star$.

To summarize, in this example we have used the edges in G to estimate a distance matrix $\hat{D}$ between K points in the unobserved latent space $\mathcal{M}^{p^\star}(\kappa^\star)$. From this, we can apply Theorem 2.1.1 to consistently estimate the geometry. However, this example made two simplifying assumptions: no fixed effects and every node is located on exactly one of K finite points on the manifold. Our more general result, which we now describe, relaxes these assumptions considerably.

In (2.1), nodes have individual fixed effects $\nu_i^\star$ as well as latent positions $z_i^\star$. The individual effects describe heterogeneity in the propensity for an individual to form connections and are not directly related to which connections will form, which is what the latent space captures. We would prefer to estimate the distances used to test hypotheses about latent geometry without potential confounding by individual effects not specifically related to the geometry. We accomplish this by marginalizing over the individual effects in (2.1). Recalling that the support of $\nu_i^\star$ is $(-\infty, 0]$, we integrate out the node effects to find that

$$\mathbb{P}\left\{G_{ij} = 1 | z^\star, \mathcal{M}^{p^\star}(\kappa^\star)\right\} = E\{\exp(\nu_i)\}^2 \exp\{-d(z_i, z_j)\} \, .$$

Solving for the distance $d(z_i^\star, z_j^\star)$, we have

$$d(z_i^\star, z_j^\star) = -\log(p_{i,j}) + 2\log[E\left\{\exp(\nu_i)\right\}] \, . \tag{2.14}$$

Note that if $\nu_i = 0$ with probability 1, which we assumed in the simplified model above, then $E\{\exp(\nu)\} = 1$, so that (2.14) becomes (2.12). Here we use properties of the exponential function to isolate the node effect term $E\{\exp(\nu)^2\}$. See Appendix A.10 for a discussion on how to marginalize out the node effects when other link functions are used. We must now estimate (i) the term $p_{k,k}$ and then (ii) the term $\log[E\{\exp(\nu)\}]$.

Our strategy therefore has two components. The first is to identify some $K \times K$ distance matrix D among K points on the manifold that could be used to test the

geometry and then develop a consistent estimator of the pair-wise distances between these K points. By showing that the regularity conditions for the geometric result (Theorem 2.1.1) are met under the network formation model (Assumptions 2.1.1-2.1.3), we can identify the geometry. The second (lesser) component is to recognize and adjust for the fact that in going from linking probabilities to distance matrix estimation, we need to adjust for the nuisance of the fixed effects parameters.

2.4.2 Estimating probability of edges

We begin by estimating the term $p_{kk'}$. The approach exploits the clique structure in the network. Cliques are useful because under regularity conditions that are useful in applications, a clique of size ℓ tells us that all nodes in that clique are extremely likely to be very close in the latent space. Therefore, we can treat them as if they are all located at the same point in the manifold. Of course, this is not exactly true, but for sufficiently large clique size ℓ, this approximation strategy becomes more and more accurate. If we find K disjoint cliques in the graph, then we have identified K distinct points on the latent space. Then, by counting the number of edges between cliques, we can approximate all pairwise distances between these cliques, as we did above in (2.13). So by looking at the clique structure, we can compute an estimate $\hat{D}$ which approximates distances along the surface of the latent space. We now describe this approach more formally.

Let $C_1, \ldots, C_K$ denote K disjoint cliques of size ℓ. We can take ℓ to grow like $\log(n)$, and such cliques will exist with probability tending to 1 in the graph (see Appendix A.11 for a discussion). Let ζ_k be the Fréchet mean of the locations of the nodes in clique k:

$$\zeta_k^\star(\ell) = \underset{\zeta \in \mathcal{M}^{p^\star}(\kappa^\star)}{\operatorname{argmin}} \sum_{i \in C_k}^{\ell} d^2_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z_i^\star, \zeta\right) \ .$$

Informally, this point represents the average of all node locations in the latent space.

We then define a $K \times K$ distance matrix $D = \{d_{kk'}\}$ with entries

$$d_{kk'} := d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(\zeta_k^\star(\ell), \zeta_{k'}^\star(\ell)\right) \ .$$

Note that $\zeta^\star$ and D are indexed by the clique size ℓ and therefore n. Conditioned on seeing cliques of size ℓ, we show that the terms $\zeta_k^\star(\ell)$ each converge to some fixed point $\zeta_k^\star$ on the latent space under general and relevant conditions (see Assumption 2.1.3 and Section 2.4.4) . Thus, the matrix containing the pairwise distances is the distance matrix we wish to estimate. Recall that this is the distance term on the left hand side of (2.14).

To estimate the probability on the right hand side of (2.14), we compute

$$\hat{p}_{kk'} = \frac{1}{\ell^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G_{ij} \mathbf{1}\left\{i \in C_k, j \in C_{k'}\right\}. \tag{2.15}$$

2.4.3 Adjusting for the Expected Fixed Effect

For notational convenience let us denote $\tau := E\{\exp(\nu_i^\star)\}$. The main observation here is that if we focus on nodes that are close together in the latent space, since the distance term in such cases will be (nearly) zero in (2.1), it we will be able to estimate and account for the individual effects.

Naively, since we use cliques to estimate $p_{k,k'}$, we may consider using these nodes to also estimate τ. We could, for example, compute the number of edges in the cliques out of the number of possible edges. However, this estimate will be 1, since by definition all edges exist between nodes in the same clique. Therefore, we define a closely related idea, which we call the "almost-clique."

Fix an ℓ-clique C_k and a number $t < \ell$. We define an "almost-clique" $I_k(t)$ by

$$I_k(t) := \left\{j : j \notin C_k, |C_k| > \sum_{i \in C_k} G_{i,j} \geq t\right\}$$

to be the set of nodes not in C_k that connect to at least t nodes in C_k. The intuition behind this definition is that if $t \approx \ell$, then the distance between nodes in $I_k(t)$ should be close to zero, but since they are not in the clique not all connections will be realized.

We can estimate the probability that nodes in the sub-graph induced by $I_k(t)$ connect,

$$\hat{E}(t,k) = \binom{|I_k(t)|}{2}^{-1} \sum_{(i,j)\in I_k(t)\times I_k(t)} G_{i,j} \ .$$

To estimate $E(\exp(\nu))$, we average the above term over all cliques, leading to an estimate

$$\hat{E}(\exp(\nu)^2) := \frac{1}{K}\sum_{k=1}^{K} \hat{E}_\nu(t,k) \ .$$

This approach suffers from selection bias when the clique size is large. That is, by Assumption 2.1.2 all individuals have independent and identically distributed $\nu_i^\star$ terms. Conditional on being part of a large clique, however, an individual is likely to be on the right tail of the v_i distribution. We could adjust for this bias by, for example, assuming a parametric model for ν_i. If we made such an assumption we could compute a correction for the selection bias based on the tail of the assumed distribution. In practice, we found our non-parametric estimator worked sufficiently well without such a correction. We suggest taking t to be large, for example $t = \ell - 1$ because our simulations suggest that large values of t reduce the selection bias and therefore increases the accuracy of our method.

2.4.4 Examples Satisfying Assumption 2.1.3

Part (a) of Assumption 2.1.3 is innocuous. It simply requires that K points identify the manifold. Note that the existence of K points out of $n \to \infty$ random points with continuous distribution over any ball on the manifold will identify it outside a measure zero event in the usual measure. To see this, it is easy to observe that with continuous mass on any patch of a sphere, K points will exist that are not simply forming an arc or are arbitrarily local with probability tending to one.

Part (b) of Assumption 2.1.3 requires that for sufficiently large ℓ, nodes in a clique are at approximately the same location on the latent surface, which allows us to conclude consistency and asymptotic normality of our distance estimates. We

provide a general set of assumptions that are sufficient and show several models used in applied work are covered.

ASSUMPTION **2.4.1.** *Let* $\Omega_n = \left[0, A_n^{1/p^\star}\right]^{p^\star} \subset \mathcal{M}^{p^\star}(\kappa^\star)$ *be such that* $A_n = o(n)$ *is growing. Assume either*

1. Bounded Support: Ω_n *is the support of the distribution* $F_n(z)$ *of locations, or*

2. Thin Tails: $F_n(z)$ *satisfies, for some constant* $b > 0$,

$$\mathbb{P}\left(\max_{i,j} d_{\mathcal{M}^{p^\star}(\kappa^\star)}(z_i, z_j) > A_n^{1/p^\star}\right) \leq \exp\left(-bA_n^{1/p^\star} + \log n\right) \to 0.$$

Obviously (1) implies (2), but (2) allows for sub-Gaussian tails with unbounded support. The next assumption is natural and general, holding as long as distributions are not taking mass only on some sub-manifolds. It says that the odds that all ℓ independent draws are within δ-distance of each other is inversely related to the volume of that ball. We verify this for examples below.

For any set of ℓ points in the latent space, define the event

$$\mathcal{E}_\delta := \left\{\max_{1 \leq i < j \leq \ell} d_{\mathcal{M}^{p^\star}(\kappa^\star)}(z_i, z_j) < \delta\right\}.$$

In words, $\mathcal{E}_\delta$ is the event that the largest distance between these ℓ points is less than δ. We omit the dependence on ℓ when writing E_δ for convenience.

ASSUMPTION **2.4.2.** *The location distribution sequence satisfies, for any* δ *and* ℓ,

$$\mathbb{P}(\mathcal{E}_\delta) = \frac{a(\delta)}{A_n^\ell}(1 + o(1)).$$

for a positive constant $a(\delta)$*, which can depend on* δ.

Let $\mu_{d,n} := E\left\{d_{\mathcal{M}^{p^\star}(\kappa^\star)}(z_i, z_j)\right\}$, where z_i is independent of z_j, be the expected distance between two location draws.

ASSUMPTION **2.4.3.** *For any $\delta > 0$, assume a sequence of clique sizes $\ell_n \to \infty$ satisfying*

$$\ell_n \geq 2\frac{\log A_n}{a(\delta)(\mu_{d,n} - \delta)} + 1$$

for all n sufficiently large.

The above assumption is written for any $\delta > 0$, but in fact we are only concerned with $\delta < \mu_{d,n}$, the average distance between two arbitrary points drawn on the latent surface. For $\delta > \mu_{d,n}$, it is unlikely that two nodes that are at least $\mu_{d,n}$ apart would form an edge. Therefore, we are only interested in checking the condition for sufficiently small δ.

We show that Assumptions 2.1.3 hold under the above assumptions.

PROPOSITION **2.4.1.** *Let Assumptions 2.1.1, 2.1.2, 2.4.1, 2.4.2, and 2.4.3 hold. Then Assumptions 2.1.3 holds.*

We now consider three common ways of modeling node locations in latent space. For each model, we show that Assumptions 2.1.1, 2.1.2, 2.4.1, and 2.4.2 hold.

These three examples cover a wide span of location distribution models. The first, a lattice, is a stylized example that simply corresponds to a community block model with a geometric structure, since all nodes exactly live at one of several locations. The second, a uniform distribution, is the other extreme where nodes have no bias towards any specific locations that help organize cliques. This is the most adversarial case. The Gaussian Mixture Model lives in-between and interpolates between these models and is frequently used in practical statistical modeling. It functions much like the uniform if the dispersion of nodes about their type-centers tends to infinity and is similar to the lattice if the dispersion tends to zero.

EXAMPLE **4** (Lattice)**.** *Every node is a member of some community $t_i \in \{1, \ldots, T_n\}$. Node locations on the manifold are determined as follows. Let a lattice $\Lambda_n = \{0, \ldots, T_n^{1/p^\star}\}^{p^\star} \subset \Omega_n$ be a list of coordinates which serve as the support for node placements with a spacing of 1 between points along any axis. Node locations z_i are placed i.i.d. across these*

T_n points on the manifold, i.e., on the lattice Λ_n, so Assumption 2.4.1 holds and t_i corresponds to the location drawn. Assumption 2.4.2 holds for any $\delta < 1$, with $A_n = T_n$, since every location $t \in \{0, \ldots, T_n\}$ is equally likely: $\mathbb{P}(\mathcal{E}_\delta) = (1/T_n)^\ell$.

To understand the growth-rate restrictions on $\ell(n)$, recall that $\ell(n)$ has to grow slowly enough so that there are cliques of size ℓ with probability 1. Consider the lattice above. Let us assume A_n is fixed at $A > 0$. Then, at each location in the lattice, the nodes connect independently with some fixed probability determined by the fixed effects, so these nodes are in an Erdos-Renyi model. So we need $\ell(n) < \log(n)$ in order for there to be an ℓ-clique in the graph with probability approaching one (recall Appendix A.11). And according to Assumption 2.4.3, we need $\ell > \log(A_n) = \log(A)$, a constant. So taking $\ell(n) \to \infty$ to be growing slower than $\log(n)$ satisfies Assumption 2.4.3 for sufficiently large n.

The next example is in some sense the most adversarial model for our method. By placing nodes uniformly over the support, the odds of accumulation, and therefore clique formation, at any given point are minimized. Still, the result holds.

EXAMPLE **5** (Uniform Distribution)**.** *Node locations $z_i \sim U(\Omega_n)$ are assumed to be drawn independently, uniformly over Ω_n such that Assumption 2.4.1 holds. It immediately follows by a calculation that $\mathbb{P}(\mathcal{E}_\delta) = \left(\delta^{p^\star}/volume(\Omega_n)\right)^\ell$, so Assumption 2.4.2 holds as well.*

The final example interpolates between the extremes of the lattice and uniform models.

EXAMPLE **6** (Gaussian Mixture Model)**.** *As in the lattice example, every node is a member of some community $t_i \in \{1, \ldots, T_n\}$. Node locations on the manifold are determined as follows. Let $\Lambda_n \subset \Omega_n$ be a lattice of T_n points, which designate community centers, randomly drawn as follows. There is a support $\Omega_n' \subset \Omega_n$, with $\Omega_n' = \left[0, B_n^{1/p^\star}\right]^{p^\star}$ and the community centers $\zeta_t \sim U(\Omega_n')$ are drawn independently, uniformly over this support.*

Given the community centers, every node conditional on the community it is assigned to has a location that is dispersed about the center

$$z_i | t_i = t \sim F_n\left(z; \zeta_t, \sigma_t^2\right)$$

independently. Examples include Gaussian in Euclidean space, the von Mises-Fisher on the sphere, and the wrapped-normal distribution on hyperbolic space. Note that community centers reside within $\Omega_n' \subset \Omega_n$ *but given the distribution* F_n*, the random variable* z_i *may have full support over* $\mathcal{M}^{p^\star}(\kappa^\star)$*.*

The distance between the outer boundary of Ω_n' *and* Ω*, denoted by* $\Delta_n := A_n^{1/p^\star} - B_n^{1/p^\star}$ *is assumed to be growing at a sufficiently fast, possibly sub-logarithmic rate* $\Delta_n = \omega\left(\sqrt{\log n}\right)$*. Then one can calculate that Condition 2 of Assumption 2.4.1. The function of this is to ensure that even if the* z_i *have full support, they are going to essentially all live within* Ω_n*.*

In this setup, we can calculate that

$$\mathbb{P}\left(\mathcal{E}_\delta\right) = \left(\frac{\delta^{p^\star}}{A_n}\right)^\ell \times \left(1 + o\left(1\right)\right).$$

so that Assumption 2.4.2 holds too. Complete calculations are in the Appendix A.1.

2.4.5 Consistent Estimates of Location and Fixed Effects

Suppose that the researcher has access to estimates of the node locations fixed effects $(\hat{z}_i, \hat{\nu}_i)$. Our result is agnostic to how these estimators are constructed and allow any consistent ones from the literature.

COROLLARY **1.** *Let* $\hat{\mathcal{M}}^{\hat{p}}(\hat{\kappa})$ *denote the estimate of the geometry from Algorithm 3. Let* $\hat{z}_i(\mathcal{M}^{p^\star}(\kappa^\star))$ *and* $\nu_i(\mathcal{M}^{p^\star}(\kappa^\star))$ *be any set of consistent estimators, computed using the assumed geometry* $\mathcal{M}^{p^\star}(\kappa^\star)$*. Then,* $\hat{z}_i(\hat{\mathcal{M}}^{\hat{p}}(\hat{\kappa}))$ *and* $\hat{\nu}_i(\hat{\mathcal{M}}^{\hat{p}}(\hat{\kappa}))$ *are consistent.*

The proof is straightforward consequence of Theorem 2.1.2, so we omit it. The mode of convergence in Corollary 1 depends on how the estimates of node locations

and effects converge. For example, if conditioned on the right geometry, $\max_{1\leq i\leq n}|\hat{\nu}_i - \nu_i^\star| \xrightarrow{p} 0$, then this same convergence holds in Corollary 1.

2.4.6 Practical Implementation of Manifold Hypothesis Tests via Bootstrap

Following Proposition 2.3.4, any arbitrary implementation of hypothesis testing for (2.8), (2.9), and (2.10) would suffice. For instance, following the logic of Weyl's inequality and Proposition 2.3.1, since our estimate of W is asymptotically normally distributed, one can readily develop an analytic conservative test. Since Weyl's inequality is often a loose upper bound, we suggest take a different approach and provide a specific method which is extremely fast, easy-to-implement, and effective in simulations and data work via a sub-sample bootstrap.

There are two non-standard features of our problem that make classical bootstrapping challenging. First, W_0 does not have full rank, meaning that $\lambda_K(W_0) \leq 0$; under the null $\lambda_K(W_0) = 0$ meaning the parameter is on the boundary of the parameter space. Classical bootstrap is not valid in such a case [7]. Second, W_κ has repeated eigenvalues at zero under the null for both curved spaces, which again excludes the classical bootstrap [54].

We adapt the sub-sampling method from [140] which is valid both with parameters on the boundary and with repeated eigenvalues. Algorithm 7 in the appendix presents the method. It uses sub-sampling to generate a distribution $\{D_b^\star\}_{b=1}^B$ of B bootstraps of the $K \times K$ distance matrices. Then given this distribution, the method constructs the corresponding distribution of the eigenvalue of interest, $\lambda_{\tilde{k}}\{W_\kappa(D_b^\star)\}$, which is then used to test the null hypothesis for the original data. In Appendix A.11, we investigate another way of testing geometry via the Cayley-Menger determinant. We show numerically that this procedure has lower power than the bootstrap procedure discussed above, so in the following simulations we use the bootstrap procedure.

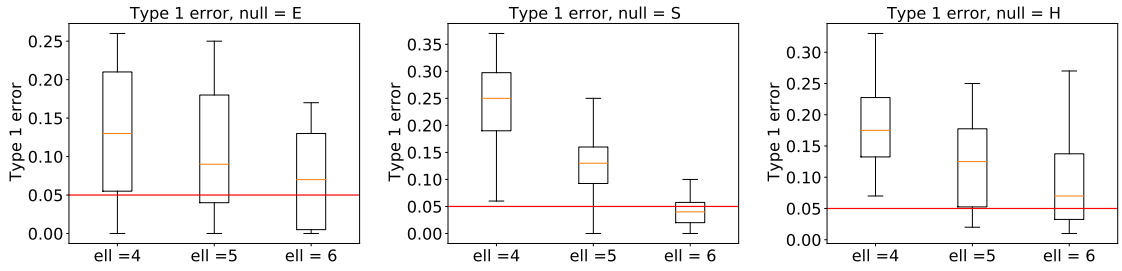
2.5 Simulation evaluation

In this section, we examine the performance of our proposed method on simulated data from each of the three candidate geometries. The goal is to understand how well the methods perform in a setting where we know the (simulated) true geometry. We first examine the Type 1 error and power of the tests to select manifold class and then show the performance of our algorithm for estimating the latent dimension. We provide additional simulation results, including results for estimating curvature, in Appendix A.7.

We discuss the Type 1 error and power of our proposed tests under various values for the clique size (ℓ) and the number of cliques we select for our estimation (K). In all cases, we simulate graphs in the following way. First, we generate a set of groups centers randomly in the latent geometry and dimension to be tested. For the type 1 error, we generate graphs on $n = 200$ nodes and use $K = 5$ and clique sizes that are realistic in the data: $\ell \in \{4, 5, 6\}$. When generating the type 1 error figures, the graph statistics are as follows. For the Euclidean graphs, the average degree is 41 and the average clustering coefficient is 0.26. For the spherical graphs, the average degree is 56 and the average clustering coefficient is 0.35. For the hyperbolic graphs, the average degree is 40 and the average clustering coefficient is 0.26.

For the power simulations, we generated 25 sets of latent positions and then, for each set of latent positions, constructed 100 graphs. When comparing across values of K and ℓ, we use the same graphs for all comparisons (e.g., when comparing $K = 10$ vs $K = 5$, the cliques in the $K = 5$ set are randomly selected from among those in $K = 10$). We provide specific values we used for simulations and additional results in Appendix A.2.

Figures 2.5 and 2.5.2 show results for Type 1 error and power of the tests we propose using the simulation procedure described above. Each point in the boxplot is the fraction of rejections out of 250 graphs for a given set of latent space positions.

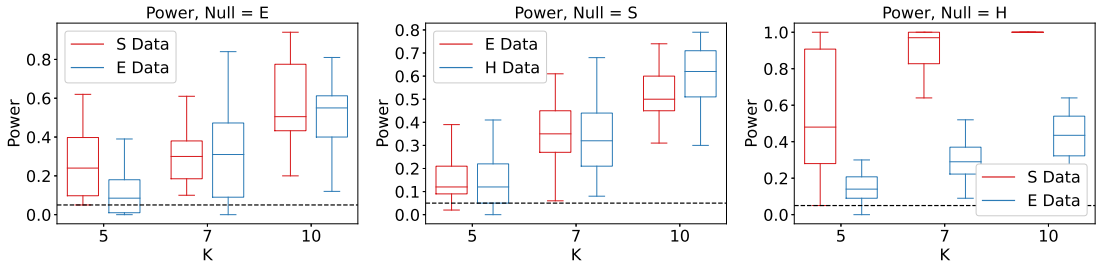


(a) (b) (c)

Figure 2.5.1: Estimated type 1 error. For each set of LS positions, we perform the test 100 times and plot the average rejection probability.

The variation in the boxplot, therefore, represents heterogeneity across latent space locations that are consistent with the true underlying geometry and the simulation procedure we use. Figure 2.5 shows boxplots of the Type 1 error for each of the three null hypotheses for three values of ℓ. We focus on variation in the Type 1 error across values of ℓ to see whether the properties of the [140] bootstrap procedure are preserved empirically. We see that, in all three cases, the Type 1 error decreases as the clique size increases. Further, for the Euclidean and hyperbolic cases the Type 1 error tends to be below the nominal level of five percent, but the spherical type 1 error is higher than five percent. In Figure 2.5.2 we see that the power increases as we increase K for all three geometries. In these simulations we use $\ell = 5$. Recall that all of our manifolds are locally Euclidean—indeed that is part of their definition. So, it is unsurprising, if not expected, that power against Euclidean alternatives rises more slowly than power against alternatives of the opposite curvature. Appendix A.9 contains more simulations.

Moving now to the estimates of the minimal dimension, we consider $p \geq 2$ and take $\max(2, \hat{p})$ as our estimate of the dimension of $\mathcal{M}^{p^\star}(\kappa^\star)$. In Table 2.5.1 we give our estimates of the dimension for the three geometries.



(a) Euclidean null (b) Spherical null (c) Hyperbolic null

Figure 2.5.2: Estimated power using simulated LS positions. For each set of LS positions, we perform the test 100 times and plot the average rejection probability.

Table 2.5.1: Average probability of correctly predicting the dimension of the latent space, averaged across 25 different sets of $n = 200$ latent space positions. We use $K = 7$ and $\ell = 4$. For each set of latent space positions, we generate 50 networks and predict the dimension.

	True Geometry		
	$\mathbb{R}^4$	$\mathbf{S}^3(1)$	$\mathbf{H}^3(-1)$
$\mathbb{P}(\hat{p} < p^\star)$	0.02	0.03	0.06
$\mathbb{P}(\hat{p} \geq p^\star)$	0.98	0.97	0.94

2.6 Examples from economics and biology

In this section we demonstrate the performance of our method settings with the complexity of observed data. We demonstrate that, in vastly distinct contexts, our approach captures features of the underlying geometry that provide contextually salient insights. We begin by offering guidance on choices a practitioner would make when implementing the method, then provide examples from three contexts. Access to data from [14] is restricted, however, a similar dataset is freely available here: https://doi.org/10.7910/DVN/U3BIHX. Data from our neural network example is available here: https://www.dynamic-connectome.org/?page_id=25. Replication code is available here: https://zenodo.org/record/7474776#.Y6TryS-B2fU.

2.6.1 Choices for Implementation of Algorithm 3

A key decision for implementation is how to identify and then select cliques for a given graph. Overall, there are several considerations.

Choosing Clique Size

Algorithm 3 requires identifying K cliques of size ℓ (again recalling that we only assume for simplicity that they are of the same size ℓ), with ℓ growing slowly. K is assumed to not grow with n, although more points will raise the power of our tests. This means that the researcher *does not* need to identify all ℓ-cliques in the graph, only K of them which will typically be a small number (e.g., 8-12) and are easily able to be found in our empirical applications, which our simulations show leads to high power. In Appendix A.11 we show that taking $\ell < \log(n)$ is often sufficiently slow to guarantee that cliques of size ℓ exist in the network. Of course, one can take ℓ to grow slower than that, but a higher ℓ results in a lower Type 1 error, since larger cliques gives better estimates of the distances in the latent space. First, we would like to take the number of cliques K and the size of the cliques ℓ of cliques to both be as large

as possible. In practice, we use the networkX command `enumerate_all_cliques` in Python, which uses a clique finding algorithm from [108]. While identifying numerous cliques can be challenging, given the modest size of the cliques and that we only need a small, fixed number of cliques, applied researchers can easily implement our technique.

Choosing Cliques

As ℓ increases, the variance of estimates of $\hat{D}$ decreases, and the power of the test increases as K increases, since we have more distances between points on the manifold. Figure 2.5.2 from our simulations shows that as K increases, the power of our tests increase. Second, we need cliques that are well-separated on the manifold, but connected in the graph. Since we use cliques as "points" on the manifold to measure distance, the cliques should ideally not have nodes in common, since if two cliques do overlap, its possible these two "points" on the manifold are close together or even the same point. Third, if two cliques have no edges between them, then our estimate of the distance between the two points is $+\infty$, which contains no information about the geometry.

Motivated by these three considerations, our goal is to solve

$$\begin{aligned} \hat{C}_1, \ldots, \hat{C}_K \in & \underset{C_1,\ldots,C_K}{\operatorname{argmin}} \sum_{i,j}^{K} |C_i \cap C_j| \\ \text{such that} \quad & |C_i| = \ell \text{ for each } i \text{ and } \hat{P}(C_1, \ldots, C_K) \text{ does not contains a } 0. \end{aligned} \tag{2.16}$$

In practice, we set K and ℓ by first looking at the number of cliques of various sizes in the graph and choosing and ℓ that is close to the size of the largest cliques in the graph, but where there are still enough cliques of that size to find K and are well-separated. We then take random draws from the (very large) set of possible cliques and evaluate the objective function in (2.16). Searching over the set of possible cliques is a well-studied (NP-hard) problem in computer science and graph theory, however, we found that our relatively simple approach yielded high quality cliques

after around 10^6 draws from the clique distribution. We evaluate the quality of the cliques we select by running the optimization independently several times. A stable objective function value across the runs indicates high quality cliques. In the data from [14], we take K as either 7 or 10. The value we choose is based on how easy it is to find appropriate cliques in a given network using the problem formulation in (2.16). In the Indian village sample, the average number of cliques of size $c_i - 1$ is 80, where c_i is the size of the largest clique in network i, known as the clique number. It takes on average 0.005 seconds to find all cliques of size $c_i - 1$ over the 75 networks. It takes 0.004 seconds in the *C. elegans* sample to find the 29 cliques of size 5 in the network.

To select ℓ, we use the size of the largest clique found in the graph minus one. In most of the villages, choosing ℓ in this way resulted in dozens of possible cliques to choose from. We present more details about cliques in the [14] data in Appendix A.6. For the *C. elegans* data, we select $K = 12$ and set $\ell = 5$, which is the size of the largest clique in the graph. We reiterate that for our approach we need only K cliques and do not need to enumerate all ℓ cliques in the graph.

2.6.2 Village Risk-sharing Networks and the Introduction of Microfinance

We begin by studying the underlying geometries of Indian village networks. We use the Wave II village network data of [14], in part collected by one of the authors of the present chapter. This consists of a collection of graphs for each of 75 villages in Karnataka, India constructed by surveying 89% of all households in each village, thereby generating a 99% edge sample for the resulting undirected graph. There are a total of 16,451 households in the sample. In every village we have relationship data between households on each of 12 dimensions: 5 social dimensions, 4 financial dimensions, and 3 information sharing dimensions. See [14] for more details including descriptive statistics. The links across these dimensions line up for the most part, consistent with a theory of multiplexed incentives to form links, so we study the

undirected, unweighted graph following the prior literature using this data [94, 13, 26, 14].

The social networks literature has long been interested in excess closure [43]. Friends of friends tend to be friends more than one might expect and this is particularly true if network relationships substitute for formal institutions. A literature focusing on equilibrium informal financial networks, which facilities the sharing of risk between households in a village, describes why the equilibrium network shapes exhibit excess closure (e.g., [6, 93]). The idea is that in order to maintain cooperation, when individuals can renege on their promises to aid each other in times of need, it is useful to have friends in common to amplify punishment, thereby maintaining a good equilibrium.

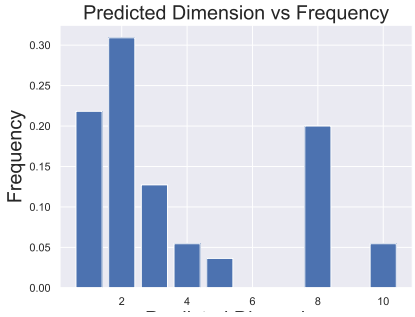
From the perspective of a latent space model, this means that we might expect excess closure in the village. There are incentives by households to "curve" the space, so friends of friends and so on are much more likely to themselves link, discussed in greater depth below. A natural hypothesis, therefore, is that village networks for the most part not be hyperbolic. Rather, they may be more likely to be spherical or, perhaps, Euclidean.[9]

Our proposed method gives hypothesis tests (and corresponding p-values) for each

[9]We briefly note that common modeling assumptions in the socio-economic literature imply constant curvature from the perspective of our model (2.1), though certainly there are perspectives that would violate constant curvature which would require future work. To see this, consider two examples. First, imagine a model in which nodes have some random locations. They can choose their efforts to link and the value of their links depends on the number of their friends who are themselves friends in expectation. There is a parameter that governs the value of closure among one's friends which can be positive, zero, or negative, which may depend on the socio-economic context. In such a model, this parameter exactly maps to curvature. Second, one can imagine a model in which agents can take an action to influence the extent to which their neighbors know each other. For instance, the action could be imagined as throwing parties (selecting positive curvature) or the opposite and ensuring "worlds do not collide" (selecting negative curvature) [47]. In such a model, if we study the symmetric equilibrium, then the equilibrium choice of the extent of forced socialization or barred socialization among one's friend exactly maps to constant curvature. Both of these examples also illustrate the limitations of such models. While these examples demonstrate how conventional assumptions map to constant curvature, certainly more complex models with heterogeneity would require modeling manifolds with non-constant curvature, which we leave to future work.

Geometry	$\mathbb{R}^p$	$\mathbf{S}^p$	$\mathbf{H}^p$	N/A
% Classification	26%	36%	12%	25%

(a) Geometry Estimates



(b) Predicted Dimensions

Figure 2.6.1: Predicted geometries and dimensions for the [14] village networks.

of the three candidate geometries. As a descriptive summary, we "classify" each of the villages into one of the geometry types using the following procedure. For villages where at least one village has a p-value over .05, we consider, for the purposes of summarizing our results, the manifold type that has the largest p-value. If all three geometries reject the null at the .05 level, then we say that the village cannot be classified. This outcome could mean a number of things, ranging from a false-rejection by chance to a village whose underlying geometry is not captured by one of the three candidates (e.g., curvature may be nonconstant). Figure 2.6.2, Panel A presents classification results for the 75 villages using this descriptive approach. We see that we are able to classify 75% of the villages, despite the fact that N/A was a possibility. Classification was not forced. Further, the results are consistent with the socio-economic hypothesis on villages needing closure. 48% of the classified networks are spherical, 35% are Euclidean, and only 16% are hyperbolic.

We also examine the estimated dimension of the latent space. Figure 2.6.2 presents the estimated dimensions, which irrespective of curvature is important to know the minimal dimension of the space required to model location decisions by agents.

We now explore the relationship between the latent geometry and socio-economic phenomena. This is an observational, not causal, analysis. First, we look at how the volume of informal financial transactions vary with network geometry. Specifically, we are interested in how the volume of informal loans that a household has with network neighbors (e.g., friends or members of their rotating, savings, and credit associations) varies with geometry. Both the theoretical and empirical economic literatures suggest that it is ex ante ambiguous as to the relationship between the amount of network financial flows and curvature. For example, [104] study how informal financial flows efficiently allocate credit to households that experience negative shocks in the network. Theory suggests that such flows are more efficient in more expansive networks, which require negative curvature [6]. At the same time, as discussed above, the ability to facilitate informal financial transactions may increase in the importance of closure, and therefore require positive curvature. Which force dominates is an empirical question.

To study this, we estimate the following regression:

$$\text{Network Loan Amount}_i = \alpha + \beta_E \mathbf{1}\{\hat{\mathcal{M}}_i^{\hat{p}} = E\} + \beta_H \mathbf{1}\{\hat{\mathcal{M}}_i^{\hat{p}} = \mathbf{H}\} + \beta_N \mathbf{1}\{\text{N/A}_i\} + \epsilon_i$$

where i indexes the village. Network Loan Amount$_i$ is the average volume of loans from either friends or rotating savings and credit association members that a household has in the village. The loan amount is presented in INR (USD 1 $\approx$ INR 73.5). Here, the omitted category (α) corresponds to the loan amount for a sphere.

The leftmost panel of Figure 2.6.2 presents the results. We find that a Euclidean village relative to a spherical one has INR 3940 or 24% ($p = 0.098$) more informal network loans. Further decreasing curvature, we compare hyperbolic villages to spherical ones and find that hyperbolic villages have INR 5865 or 35% ($p = 0.034$) more in informal network loans. These increases are extremely large in real economic terms: the difference in credit between the hyperbolic and spherical geometries corresponds to an individual in the hyperbolic geometry receiving additional credit worth 20 days of wages. Taken together, we have seen greater financial flows precisely in geometries

that permit more expansive network topologies.

Second, having studied how informal financial transaction patterns are associated with geometry, we now turn to studying determinants of geometry. Our primary interest is in whether the introduction of a formal credit market (microfinance) to a setting otherwise dominated by only informal financial transactions changes the network structure by changing the geometry. In our setting, as described below, microcredit was introduced to only some villages, allowing us to compare the impact of access to microfinance on network structure.

In addition to microcredit access, we focus on three other determinants: wealth, inequality, and caste fractionalization. It is ex ante not obvious as to how any of these might correlate with geometry and is therefore an important empirical question. For example, wealthier villages may have a reduced need to sustain informal insurance—their worst case scenario is better off than their poorer counterparts—and as a consequence may require less positive curvature. Or, in contrast, wealthier villages may be able to take on greater entrepreneurial risk as they can sustain losses, and such endeavors require group cooperation and therefore closure. Similarly, within-village wealth inequality can change incentives for triadic closure, as can ethnic fractionalization [46]. Ultimately, the empirical correlations are of interest. The most important relationship to study is how the introduction of formal credit to villages that otherwise used informal network transactions affects geometry. From 2007, a microfinance institution entered 43 of the 75 villages studied here and the network data we utilize is taken after the intervention [13, 12]. This allows us to study the effect of the introduction of microcredit on network geometry as a way to understand whether credit access differentially changes the need for ones' friends to maintain relationships with each other. Note that this is different from clustering or other measures of closure per se, which are also affected by the locations z and fixed effects ν. So we can specifically address that, all things being equal, whether the demand for one's friends to themselves be linked increases, decreases, or is unchanged when the village now

has access to formal financial instruments. It is a priori not obvious. On the one hand, the new credit opportunity may encourage re-lending or joint business ventures among clients of microcredit, increasing the need for closure and generating positive curvature. On the other hand, the new credit opportunity may reduce reliance on informal financial relationships with others in the village and push towards negative curvature. In either case, the answer as to how a large credit intervention may affect geometry is of empirical interest.

To study the determinants of geometry, we estimate a multinomial regression:

$$\frac{\mathbb{P}(\hat{\mathcal{M}}_i^{\hat{p}} = m)}{\mathbb{P}(\hat{\mathcal{M}}_i^{\hat{p}} = \mathbf{S})} = \exp(\delta_m + \beta^m_{\text{MFI}}\text{MFI}_i + \beta^m_{\text{W}}\text{Wealth}_i + \beta^m_{\text{I}}\text{Inequality}_i + \beta^m_{\text{F}}\text{Frac}_i)$$

where $m \in \{E, \mathbf{H}, \text{N/A}\}$. MFI_i denotes whether the microfinance institution entered village i. Wealth_i denotes a wealth index measure.[10] Inequality_i is within-village standard deviation of wealth.[11] Finally, $\text{Frac}_i = \alpha_U(1-\alpha_U)$ where α_U is the share of households that are of upper caste. The score is zero if society is perfectly homogenous and 1/4 for an even split.

The right three panels of Figure 2.6.2 present the results. We begin by looking at microfinance. We estimate $\hat{\beta}^{\mathbf{H}}_{\text{MFI}} = -1.40$ ($p = 0.093$). This means that a village receiving microcredit is associated with an 8.8% decline in the probability of being hyperbolic relative to spherical. In other work, [12], we have shown that introducing microcredit has decreased density and also the number of triads in the network. Our analysis here demonstrates that the fundamental value of having friends in common itself *increased* suggesting that the effects documented in our prior work came from shifts in node locations (z_i) and efforts of socializing (ν_i) in the latent space, rather than changes in the relative value of closure which appears to have increased.

[10] The [14] dataset does not have consumption nor expenditure measures. So we utilize the score constructed from the first principle component of a number of household features that correlate with wealth in the village. This consists of access to private electricity, home ownership, quality of roofing material, and number of rooms in the household.

[11] Specifically, we take the score from the first principle component of the within-village standard deviation of each of the constituent wealth measures.

We also find that wealthier villages are less likely to be hyperbolic relative to spherical. We estimate $\hat{\beta}^{\mathbf{H}}_{\mathrm{W}} = -1.02$ ($p = 0.098$). This corresponds to a 8.4% decline in the relative probability of being hyperbolic as compared to spherical. We do not find any significant relationship between wealth inequality nor caste fractionalization and geometry.

Taken together, we have shown the empirical content of the estimation of the latent geometry. We can classify the vast majority of villages (despite allowing for N/A) and they are predominantly spherical. We find informal financial loans are higher in villages that exhibit negative curvature. Finally, and importantly, villages where microcredit was introduced tend to have more spherical structure. This can perhaps be interpreted as showing that access to microcredit generates, *ceteris paribus*, demand for greater triadic closure.

2.6.3 Network of Neurons

Our second setting looks at a network of neurons. There is a neuroscience literature that is interested in documenting regularities in network structure as well as modeling network structure through statistical network formation models.

The first strand of the literature looks at how patterns of the graph of neurons relate to neurological mechanisms [98]. For instance, these networks exhibit short path lengths–disparate regions of the human brain are connected by a few steps. Further, the degree distribution reflects thick tails: certain nodes have numerous connections. Moreover, the network is dynamic: early in age the network exhibits high amounts of homophily whereas as the individual ages this declines.

The second strand develops low dimensional statistical representations of the neural networks since this allows for interpretability, counterfactuals, and deals with the fact that otherwise there is a litany of statistics that can be used to correlate with biological outcomes without any interpretable control [49, 147]. To this end, conditional edge independence models, scale-free models, block models, and latent space

models have been explored [165, 98].

Third, and particularly relevant for latent space models, is the concept of the functional graph of neurons rather than the structural graph of neurons [138, 1]. The idea is that while a graph can be drawn of the phyiscal links between all nodes, predominantly the graph that is able to be activated–the functional network–is a network that is distinct. Much like individuals who reside in geographic space but functionally interact in a network that can be thought of as in a latent space, the functional network perspective presents an opportunity leverage latent space models.

Our specific application is to a network of neurons of *Caenorhadbitis elegans*, which are soil-dwelling roundworms. There is a long history of using *C. elegans* as a model organism for studying nervous systems of animals. In fact neurons of *C. elegans* are extremely similar to that of humans [113]. For our example, we use the *C. elegans* neuron data of [97], which has been used a number of times in order to model neural network structure. There are several goals in modeling neural network structure. The relative location distribution, how distance affects linking rates, and the geometry all inform how signals could be passed across nodes. Moreover, though beyond the scope of our knowledge, there may be interpretations to the distribution of fixed effects—latent heterogeneity in the propensity for certain neurons to systematically link to others.

A priori it is unclear what the right latent geometry ought to be. For instance, if the network of neurons ought to have a high degree of expansiveness, it ought to be embedded in hyperbolic space. In contrast, if it ought to reflect strong, localized redundancies, or a high degree of homophily it may be better modeled as being embedded in a spherical geometry.

The dataset contains a neural network from a single *C. elegans*, consisting of a connected graph of 131 neurons, with 764 edges, and a clustering coefficient of 0.245. The clique number of this graph is 6, but it has only one clique of size 6, but it has 29 cliques of size 5, so we use $\ell = 5$. We find $K = 12$ cliques using the

problem formulation in (2.16) and then take a maximally disjoint clique set which is sufficient for our test. We use Algorithm 3 and compute the p values for the Euclidean, spherical, and hyperbolic geometries and find these values are: $p_E = .378$, $p_S = 0.05$, and $p_H = 0.267$, so we reject the spherical hypothesis (noting that we can do so despite there being a high level of clustering). The *C. elegans* network of neurons, therefore, is inconsistent with a latent space with positive curvature, where neurons are excessively likely to exhibit triadic closure relative a flat benchmark. We can only say that there is no or negative curvature, but the data are not sufficiently powered to allow us to distinguish this.

2.7 Conclusion

Latent space models are widely used in network analysis across numerous disciplines including, but not limited to sociology, economics, biology, and computer science. The predominant approach is to assume a Euclidean latent space, though there is current discussion about adopting a hyperbolic space in certain contexts. Nonetheless, the current methods employed do not provide a way to estimate the geometry itself. Unfortunately, incorrect embedding spaces can deliver misleading results and while there may be convergence to pseudo-true values, counterfactual analysis will be affected.

We develop methods which the researcher can apply in order to consistently estimate the latent space geometry from network data. Our core observation is that the observed network data encodes information on the distance between nodes in latent space. That is, a finite sample network corresponds to a noisy set of distances. So we transform our network problem to a statistical geometry problem.

In our first result we study a more general problem: whether an observed estimate of a distance matrix among K points contains enough information to consistently estimate the unobserved manifold in which the K nodes can be isometrically embedded. We answer this in the affirmative: the spectrum of a distance matrix encodes the

manifold's metric and therefore the manifold class, rank, and curvature. Leveraging results on eigenvalue perturbations, we prove the result. Our second result applies to the network setting. By looking at cross-clique link frequencies, one can construct a noisy distance matrix and therefore estimate the latent manifold consistently.

An important advantage of our approach is that, unlike other strategies, we do not need to estimate the fixed effects or the locations in a candidate manifold (nor integrate them out) in the estimation procedure. Instead, by focusing on a strategy that exploits the fundamentals of geometry, we directly checks isometric embeddings, so we can estimate the geometry without ever estimating the numerous other parameters and only move to them after having estimated the geometry.

We also demonstrate the the empirical content of estimating the latent geometry which is novel in the literature. Strikingly, even though N/A is a possibility, we were able to classify (75%) of villages, indicating the empirical relevance of our methods. Further, consistent with theory, we show Indian risk sharing villages are often spherical. Additionally, villages that are more expansive are associated with a greater flow of informal financial loans through the network. Finally, the introduction of microcredit is associated with a shift to positive curvature: the relative value of having triadic closure increases when villages have access to formal credit.

A number of future steps come to mind. While our assumptions on geometry—that it is a simply connected, complete Riemannian manifold—are parsimonious and natural, they are also limited. They nest the current assumptions in the literature (we know of no empirical research that assumes a torus of genus two for instance in the networks literature) but they are still admittedly lacking. [179] has shown a relationship between the torus and the sphere that might allow us to apply our current spherical methodology to the torus. We speculate that there may be strategies to use local structures in the network to patch together some more global structure. That is, for instance, if it can be arranged into a pseudo-block diagonal structure, perhaps in each block there is room for a different geometry and then these can be stitched

together. See [77], among others, for related work on this topic. Additionally, extending our results to settings where the full graph is not observed, such as Aggregated Relational Data [125, 28], would allow researchers who do not have resources to collect data on all edges to leverage insights about underlying geometry. Individual node covariates could also be leveraged to form trait groups in settings without complete network data. Finally, an interesting area of future research involves exploring how to optimally combine the results from multiple tests of the three geometries, where each test is computed using a different set of cliques. Of course the tests based on different cliques would likely be correlated, and so an important question would be to understand how to use the correlation to increase the power of the combined test.

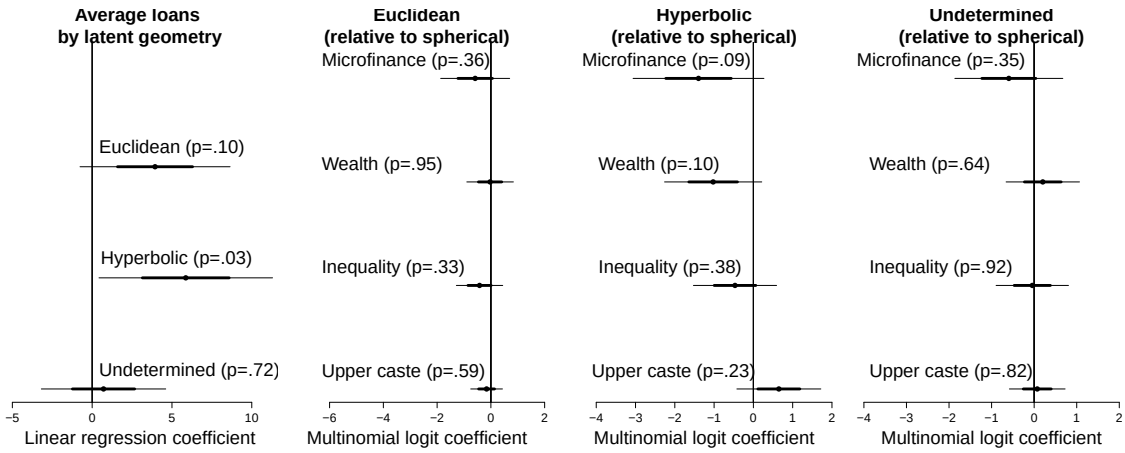


Figure 2.6.2: Regression coefficients showing the determinants of geometry. In the left figure, each line in the plot corresponds to the coefficient in a multivariate linear regression where the outcome is the average amount of loans (in thousands of INR) and the predictors are geometry types (with spherical as the reference). The wide bars correspond to one standard error and the narrow bars represent two standard errors. The reference value for spherical is 16.71 (again in thousands of INR). Plots 2-4 show the coefficients from a multinomial logistic regression where the outcome is the predicted geometry type for each village. Each panel shows all coefficients for a particular geometry (with spherical as the reference). Each line in the plot corresponds to an estimated coefficient. The wide bars correspond to one standard error and the narrow bars represent two standard errors. The constant values for the Euclidean, hyperbolic, and undetermined comparisons are .005, -.69, and -.01, respectively.

Chapter 3

CONSISTENTLY ESTIMATING NETWORK STATISTICS USING AGGREGATED RELATIONAL DATA

The empirical study of social networks has grown rapidly across a variety of disciplines, including but not limited to economics, psychology, public health, sociology, and statistics. The aim ranges from researchers trying to understand features of network structure across populations, to parameters in models of network formation, to how network features affect socio-economic behavior, to how interventions can affect the structure of the social network. In Chapter 2, we introduced the latent space model and described consistent estimators for key parameters in this model. In this chapter, we consider a different key question in network inference: the problem of studying networks when it is too difficult or too costly to collect full network data. Studying network structure and its relationship to other phenomena can be demanding particularly in contexts where survey-based research methods are used: obtaining high quality network data from large populations can be expensive and often infeasible for cost, privacy, or logistical reasons. The challenges associated with collecting complete network data mean that researchers must choose to either (i) reuse one of a handful of existing full graph datasets, likely not designed with their particular research goals in mind or (ii) postpone their research agenda while raising sufficient capital.

One recent approach to address these issues is known as Aggregated Relational Data (ARD), which solicit summaries of respondents' connections by asking for the number of people a respondent knows with a given trait. ARD questions take the form "How many people with trait k are you linked to?" and can be integrated into

standard probability-based survey sampling schemes because they do not directly solicit any connections in the graph [28]. One major advantage of collecting ARD over more traditional network surveys is the reduced cost. In the context of one large-scale randomized controlled trial studying the relationship between network structure and household finance in 60 villages, [28] showed that ARD implementation is shorter (3 versus 8 months) and cheaper ($34,000 versus $189,000) compared to full network enumeration and yet delivered the same economic conclusions that would have been obtained using the full network data. Because it is cheaper to collect, ARD also enables practitioners to collect panel data across multiple networks.

ARD was originally proposed to estimate the size of hard to reach populations, such as the number of HIV-positive men in the U.S. [103, 156, 95]. Since then, the use of ARD has expanded significantly, particularly in the social sciences [52, 60, 112] where ARD enable researchers to estimate core features of respondent networks (e.g., a respondent's centrality or the extent of clustering). In terms of methodology for analyzing ARD, [125] connects a model for ARD responses to network models of the fully observed graph. Specifically, [125] established a connection between ARD and the latent distance model, a common statistical approach for modeling fully observed network data. The key result is that ARD are sufficient to identify parameters in a generative model for graphs, allowing inference about the distribution of graphs that plausibly correspond to the ARD. [28] exploit this connection to generate a distribution over network statistics, such as the centrality of an individual or the average path length of the graph, and show examples where using statistics generated from ARD gives similar results to using statistics from the completely observed graph. ARD has also been used to estimate common econometric models and outcomes, such as the linear-in-means model [24], choosing the optimal seeding for maximal information flow [152], and can be used to assess network model goodness-of-fit [117].

Despite its increasingly widespread use, there is still little understanding of when or why ARD contains sufficient information to estimate model parameters or estimate

network properties of the unobserved network. We provide such a characterization in two steps. First, we show we can consistently estimate the parameters of a rich class of generative network models using only ARD. This fact relies on a simple but powerful observation that if the cross-type link probabilities allow us to identify the model parameters, then ARD is sufficient for consistent estimation. Critically, this insight allows us to sidestep maximizing the complicated log-likelihood directly and instead solve a system of equations based on the cross-trait linking probabilities. We show that three common generative models fall into this class.

Next, we provide sufficient conditions to consistently estimate features of the underlying, observed network using ARD. The intuition is that, for sufficiently large graphs, some statistics of graphs converge to their expected value, where the expectation is taken over graphs from the same generative process. In such cases, ARD suffices to recover the value of the graph statistics, so long as the statistics are not too sensitive to error introduced by using estimates for the generative network models parameters. In such cases, the information in ARD is sufficient to consistently estimate generative model parameters as well as graph statistics of interest.

We investigate this both theoretically and empirically in two settings. The first is when researchers can consistently estimate features of the underlying, unobserved network structure itself. Examples include centrality or clustering measures for nodes. This analysis studies the case of a single large network. The second is when researchers can consistently estimate response functions of or by the network. That is, how do changes in network features correspond to changes in socio-economic outcomes or how might an intervention affect the structure of the network. This analysis studies the case of many networks.

3.1 Aggregated relational data

We begin by defining ARD formally. Take an undirected, unweighted graph, $g = (V, E)$ with vertex set V and edge set E. There are $n = |V|$ nodes, so we sometimes

write g_n to emphasize the graph size, and $g_{ij} = \mathbf{1}\{ij \in E\}$ denotes that i and j are connected in the graph. Suppose each node has one of K traits, where K is fixed and $K > 3$. Let G_k denote the nodes with trait k, for $k = 1, \ldots, K$, where $n_k = |G_k|$ is the number of nodes with trait k. We write $t_i^\star = k$ to denote that node i has trait k. We suppose that the traits are binary and mutually exclusive, so every node has one of K traits. This is not a prohibitive assumption. To see this, imagine there are L characteristics (e.g., rural/urban or college educated/not) and for simplicity assume that these are binary. Then it is clear that we can always construct K traits, mutually exclusive, with $K = 2^L$. The extension to multi-valued characteristics is straightforward. Additionally, traits constructed through intersecting characteristics (e.g., men with a given occupation below a particular age) also reduces the size of the target population, which can limit recall bias [124].

To collect Aggregated Relational Data (ARD), the researcher asks m randomly chosen nodes "How many people with trait k are you linked to?" for each of these K traits. Linking is typically defined as knowing a potential connection (e.g., having interacted with the person in the past 2 years or recognizing the person if passing on the street). [61] provide an extensive discussion and experimental evidence regarding the definition of linking. To simplify exposition, we will set $m = n$, meaning we have ARD from all nodes. Our results also apply when $m << n$, as is common in practice. In such cases, we would either need to impute parameters for nodes without ARD data (see [28], for example) or make an assumption about node equivalence. For example, under a stochastic block model, all nodes in a given community have the same linking behavior. So by collecting ARD from at least one node in each community, we can then estimate the parameters of all nodes.

Let y_{ik} denote node i's response to this question about trait k, with $y_{ik} = \sum_{j \in G_k} g_{ij}$. Critically, when collecting ARD, the researcher does not observe any edges, just how many edges are present between a node i and people of a given trait. We use $\mathbf{y}$ to denote the $m \times K$ matrix of ARD responses.

Since the K traits are mutually exclusive, ARD responses count distinct alters across each trait group. Otherwise, if trait group A and trait group B overlap, then a person in both groups would be counted twice, once in response to the ARD question about trait A and once about B.

To model the network, we consider a general graph model $\mathbb{P}(g_n|\theta^\star)$, where edges form independently in the network, conditional on the unknown parameter vector $\theta^\star$. We call such models conditional edge-independent graph models. The number of elements in the vector $\theta^\star$ can depend on the graph size n (to accommodate node-level heterogeneity parameters, for example) but we omit this dependence to simplify the notation ($\theta^\star = \theta^\star_n$). In most settings, each component $\theta^\star_i$ is independently and identically drawn from F. In other cases, sometimes the distribution of $\theta^\star_i$ depends on the traits that node i possesses, which we write as $\theta^\star_i|t^\star_i = k \sim F_k$. This conditional independence representation relies on exchangeability amongst nodes and, thus, implies that the resulting asymptotic sequence of networks generated by these models are dense, meaning that the average degree for a given n is a constant times n [135].

We define $p_{ij} = p_{ij}(\theta^\star)$ to be the probability that i and j connect, given the model parameters. The ARD response $y_{ik} = \sum_{j \in G_k} g_{ij}$ is then is a sum of independent, but not identically distributed, Bernoulli(p_{ij}) random variables, which in various disciplines is known as either the Poisson's Binomial random variable or the Poisson Binomial random variable [166, 150, 151]. The probability mass function of y_{ik} given θ in conditional edge-independent models is

$$f_{ik}(y|\theta^\star) := \sum_{A \subseteq \mathcal{A}_y} \prod_{j \in A} p_{ij}(\theta) \prod_{j \in A^c} \{1 - p_{ij}(\theta)\} \ ,$$

where $\mathcal{A}_y$ is the set of subsets of $\{1, \ldots, n_k\}$ that contain exactly y elements. In the case where $p_{ij} = p$, then this expression simplifies to the probability mass function of the Binomial(n_k, p) random variable. An obvious first approach to modeling the ARD is to analyze the likelihood of the data $\mathcal{L}_n(\mathbf{y} \mid \theta) = \prod_{i=1}^n \prod_{k=1}^K f_{ik}(y_{ik} \mid \theta)$. Here, we have used the assumption of mutually exclusive traits, which allows us to

write the likelihood of observing $(y_{i1}, \ldots, y_{iK})$ as $\prod_{k=1}^{K} f_{ik}(y_{ik}|\theta)$. The conditional independence of edges, given θ, allows us to write the joint distribution of ARD responses over all individuals as a product, which does not depend on whether traits are mutually exclusive.

Proving consistency of $\hat{\theta}_n := \arg\max_\theta \mathcal{L}_n(\mathbf{y} \mid \theta)$ is challenging due to the complex nature of the log-likelihood, since each θ_i appears in n terms of the likelihood. We explore a different approach to estimate $\theta^\star$. Instead of looking at $f_{ik}(y_{ik}|\theta)$, where θ includes the parameters of node i as well as those of all other nodes with trait k (which are not observed with ARD), we look at the probability that node i connects to an arbitrary node with trait k, P_{ik}, which is

$$P_{ik} := \mathbb{P}(g_{ij} = 1|\theta_i^\star, j \in G_k) = \int_{\Theta_k} \mathbb{P}(g_{ij} = 1|\theta_i^\star, \theta_j) dF_k(\theta_j) \ ,$$

where again $\theta_j \sim F_{\theta,k}$ for nodes i with trait k, and Θ_k denotes the support of F_k. In the latent space model, Θ_k might be p-dimensional Euclidean, spherical, or hyperbolic space, and node locations are drawn according to a mixture model along the surface of the latent space [82, 79, 115]. In the beta-model, Θ_k is a subset of the real line.

To understand the utility of analyzing P_{ik}, rather than the full log likelihood $\mathcal{L}_n(\mathbf{y} \mid \theta)$, note that for any node i,

$$\frac{y_{ik}}{n_k} = \frac{1}{n_k} \sum_{j \in G_k} g_{ij} \xrightarrow{p} P_{ik} \ , \tag{3.1}$$

as $n_k \to \infty$, where P_{ik} is again the probability that node i connects with someone of trait k and n_k is the number of nodes with trait k. Here we have assumed that the weak law of large numbers applies to the average y_{ik}/n_k, as is the case for conditionally edge-independent graphs. In the conclusion we discuss extensions for settings where edges could be correlated or where edge probability scales with the graph size.

Supposing that (3.1) holds, we can then equate the vector of normalized ARD responses with their respective edge probabilities $P_{ik}(\theta^\star)$, and use an estimating equation approach to estimate the model parameters. Supposing that this system has a

unique solution in θ (or unique up to an isometry, as in the latent space model), this general approach allows us to derive estimators of model parameters and prove uniform convergence of these estimators in a host of rich and frequently used network models. When this system does have a unique solution, we say informally that such a model "identifies" the model parameters.

By equating observed ARD responses and the probability of connection between a node and nodes in a given trait group, we can invert that equation to solve for the parameters $\theta_i^\star$. In the next three sections, we consider three common generative network models and derive consistent estimates of the parameters in each model using this intuition.

3.2 Beta-model

We first consider the generalized beta-model [39, 73]. The original version of this model states that an edge forms between nodes i and j with probability $\text{expit}(\nu_i^\star + \nu_j^\star)$ for some sequence of parameters $\nu_1^\star, \ldots, \nu_n^\star$ that encode the popularity of nodes. Here, $\text{expit}(x) = \exp(x)/(1+\exp(x))$. The generalized beta-model includes a term that measures the effect of dyad-level covariates $X_{ij} \in \mathbb{R}^p$ on linking probability, so

$$\mathbb{P}(g_{ij} = 1 | \nu_i^\star, \nu_j^\star, \beta^\star) = \text{expit}(\nu_i^\star + \nu_j^\star + \beta^\star X_{ij}) \; .$$

[39, 73] propose estimates of the parameters using a fixed point procedure using the full network data. This procedure only requires the degree of a node. Suppose that we observe ARD about a collection of traits that are mutually exclusive and exhaustive. Then the degree of node i is $d_i = \sum_{k=1}^K y_{ik}$. Let $\hat{\nu}_i$ and $\hat{\beta}$ denote the fixed-point estimates of ν_i and β from [39, 73] computed using the ARD, which is by the preceding comments equivalent to the estimate computed from the full network data.

In the theorem below, we require that the support of the parameters in the beta-model be compact subsets of $\mathbb{R}$. This regularity condition was also imposed in [73].

Theorem 3.2.1. *Suppose the support of each node effect $\nu_i^\star$ and of $\beta^\star$ are compact subsets of $\mathbb{R}$. Then, with probability $1 - O(1/n^2)$,*

$$\max_{1 \le i \le n} |\hat{\nu}_i - \nu_i^\star| \le C\sqrt{\frac{\log(n)}{n}}$$

for some constant C that does not depend on n. In addition, $\hat{\beta} \xrightarrow{p} \beta$ as $n \to \infty$.

Here, we have not made any assumption about the relationship between traits and the distribution of the node parameters.

In cases where ARD is collected at the characteristic level and not at the trait level which creates a mutually exclusive partition, or when the mutually exclusive trades do not exhaust the space, then $\sum_{k=1}^K y_{ik}$ does not need to equal d_i, the degree of node i. In these cases, we can estimate degree of a node via other methods. One such example is the network scale-up method [103], which assumes that given a node's degree, ARD responses are modeled as $y_{ik}|d_i \sim \text{Binomial}(d_i, \frac{n_k}{n})$. This leads to the so-called "ratio of sums" estimator $\hat{d}_i = n \sum_{k=1}^K y_{ik} / \sum_{k=1}^K n_k$, where n_k is the size of group k and n is the total size of the population [103, 177]. Typically, these ARD questions are based on characteristics with known group sizes, so that each n_k is known. We can then plug in $\hat{d}_i$ in place of d_i in the estimation procedures from [39, 73] to estimate the model parameters.

3.3 Stochastic block model

We consider a generalized version of the stochastic block model (SBM), in which observable traits are dependent on, but potentially distinct from, latent community structure. Edges are determined by community structure. This setting corresponds to a case where nodes belong to unobserved communities and a researcher observes traits that are (imperfectly) associated with community membership. We show that ARD allows us to use links to observable groups to infer latent community membership.

We describe this model more formally. We first assign node communities $c_i^\star$ independently with probabilities $\pi_1, \ldots, \pi_C$. Conditioned on these parameters, edges are

generated independently with probabilities

$$\mathbb{P}(g_{ij} = 1 | c_i^\star = c, c_j^\star = c') = P_{cc'} \ ,$$

where P is a $C \times C$ matrix of within- and cross-community edge probabilities. Here we suppose that the graph is undirected, so that P is assumed to be symmetric. The intuition for this model is that the probability two nodes connect depends only on their latent group membership. In many cases of interest, the community structure is unknown *a-priori* and unobservable but traits are observable. We let the $C \times K$ matrix Q encode the probability of having trait k, given that a node is in community c, so

$$\mathbb{P}(t_i^\star = k | c_i^\star = c) = Q_{ck} \ . \tag{3.2}$$

In the case of mutually-exclusive traits, each node is assigned exactly one of the K traits with probabilities in (3.2). The intuition behind this model is that nodes with the same traits form edges in a similar way.

We suppose that the ARD we have access to is about these K traits and not about the unobserved community structure. To estimate the parameters in the SBM, we begin by making the following assumption, which allows us to consistently cluster the ARD to estimate community structure in the unobserved graph. Specifically, we assume that no two communities have the same linking pattern to all other traits, which is clearly required for identification.

Assumption 3.3.1. *The following condition holds:*

$$\min_{c,c'} ||Z_c - Z_{c'}|| > 0 \ ,$$

where $Z_c := (\tilde{P}_{c1}, \ldots, \tilde{P}_{cK})$ *and* $\tilde{P}_{ck} := \mathbb{P}(g_{ij} = 1 | c_i = c, t_j = k)$ *is the probability that a node in community* c *connects to a node with trait* k*:* $\tilde{P}_{ck} = \left(\sum_{\ell=1}^{C} Q_{\ell k} \pi_\ell \right)^{-1} \sum_{\ell'=1}^{C} P_{c\ell'} Q_{\ell' k} \pi_{\ell'}$.

To understand this assumption, let us consider a simple case when $C = K = 2$.

Assumption 3.3.1 then requires that

$$\begin{pmatrix} \tilde{P}_{11} \\ \tilde{P}_{12} \end{pmatrix} \neq \begin{pmatrix} \tilde{P}_{21} \\ \tilde{P}_{22} \end{pmatrix} .$$

We now analyze when these equalities do not hold. If the probability of belonging to community 1 and 2 are equal ($\pi_1 = \pi_2$), the first inequality is then equivalent to requiring that

$$(P_{11} - P_{21})Q_{11} + (P_{12} - P_{22})Q_{21} \neq 0 .$$

If $P_{11} = P_{12} = P_{22} = P_{22}$, which corresponds to no community structure in the model, then Assumption 3.3.1 is not satisfied for any Q matrix. If $Q_{12} = Q_{21}$, which means that there is no relationship between traits and community membership, then Assumption 3.3.1 is satisfied whenever $P_{11} - P_{21} \neq P_{22} - P_{21}$, which occurs in undirected networks whenever $P_{11} \neq P_{22}$. In this case, even if there is no relationship between traits and network structure, Assumption 3.3.1 is satisfied provided that communities behave differently in the network (i.e., there is meaningful community structure).

We now provide a classification algorithm to estimate the community membership of nodes. This procedure does not require us to know the number of communities. We initialize $W = V$, the set of nodes in the sample, so $|W| = n$. Let $\tilde{y}_i = (y_{i1}/n_1, \ldots, y_{iC}/n_C)$. While $W \neq \emptyset$, do the following:

1. Select a node i randomly from W. Set $W = W \setminus \{i\}$.

2. For any $j \in W$: If $||\tilde{y}_i - \tilde{y}_j||^2 \leq n^{-1}\log(n)$, assign node j to be in the same community as i, and second set $W = W \setminus \{j\}$.

This procedure returns a consistent estimate of the community membership *and* the number of communities. The distribution of ARD responses for people in a given community c collapses to a point mass as the sample size grows, and so clustering in

our problem is easier than clustering in general clustering problems, where the distribution of data does not need to change with the sample size. We therefore propose the algorithm above, over more standard clustering algorithms, because our clustering algorithm lends itself easily to concluding the uniform consistency in Theorem 3.3.1 that we need later in Theorems 3.5.1 and 3.5.2.

We prove in Theorem 3.3.1 that this classification algorithm returns consistent community labels. Given the community memberships $\hat{\mathbf{c}}$, let $\hat{C}_c$ denote the set of nodes in our sample that are estimated to be in community c, under the membership vector $\hat{\mathbf{c}}$, with $|\hat{C}_c| =: m_c(n)$. We can estimate $P_{cc'}$ with

$$\hat{P}_{cc'} = \begin{cases} \frac{1}{m_c(n)} \sum_{i \in \hat{C}_c} \frac{y_{ic'}}{n_{c'}}, & c \neq c' \\ \frac{1}{m_c(n)} \sum_{i \in \hat{C}_c} \frac{y_{ic'}}{n_{c'}-1}, & c = c' \end{cases} .$$

where again y_{ic} is the ARD response from node i about trait c. We can estimate Q_{ck} with

$$\hat{Q}_{ck} = \frac{1}{m_c(n)} \sum_{i \in \hat{C}_c} \mathbf{1}\{t_i = k\}.$$

where t_i is the observed trait of node i and we can estimate π with entries $\hat{\pi}_c = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{\hat{c}_i = c\}$.

Theorem 3.3.1. *Suppose Assumption 3.3.1 holds. Then, up to a permutation on the community labels, the estimated community membership vector $\hat{c}$ satisfies*

$$\max_{1 \leq i \leq n} \mathbf{1}\{\hat{c}_i \neq c_i^\star\} \xrightarrow{p} 0 ,$$

as $n \to \infty$. The estimated number of communities $\hat{C}$ as well as $\hat{P}$, $\hat{Q}$, and $\hat{\pi}$ are all consistent as $n \to \infty$.

3.4 Latent space model

We consider a broad class of latent space models. Broadly speaking, each node has a position in a latent (or unobserved) space, and the closer two nodes are in this space,

the more likely they are to connect. Each node also has a gregariousness parameter, which controls the baseline edge probability for that node [82, 79, 115, 10, 158].

We formally define one variant of the latent space generative model, which we study in this work. We draw the gregariousness parameter $\nu_i^\star$ from a distribution F_ν with compact support in $(a, 0)$ for some $a < 0$. We draw traits $t_i^\star \in \{1, \ldots, K\}$ independently with probabilities $\pi_1, \ldots, \pi_K$. Conditioned on traits, we also draw node positions $z_i^\star | t_i^\star = t \sim F_t$, where F_t is some distribution over the latent surface $\mathcal{M}^p(\kappa)$. Here, $\mathcal{M}^p(\kappa)$ is a complete simply connected Riemannian manifold with constant curvature κ, which means by the Killing-Hopf theorem that it is either Euclidean, spherical, or hyperbolic space of dimension p and curvature κ [100]. We suppose that F_t is a symmetric distribution over $\mathcal{M}$ and is uniquely determined by its mean μ_t and variance σ_t^2. Some examples of this include the Gaussian distribution over $\mathbb{R}^p$ and the von-Mises Fisher distribution over the p-sphere. In words, the node positions $z_i^\star$ is drawn from a mixture distribution on $\mathcal{M}^p(\kappa)$, with weights determined by $\pi_k = \mathbb{P}(t_i^\star = k)$. Conditioned on these parameters, we draw edges independently with probability

$$\mathbb{P}(g_{ij} = 1 | \nu_i^\star, \nu_j^\star, z_i^\star, z_j^\star) = \exp\{\nu_i^\star + \nu_j^\star - d(z_i^\star, z_j^\star)\} \, . \tag{3.3}$$

Again, we suppose that we only have access to ARD about these K traits. We write $\eta = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2)$ to refer to the "global" parameters. To build our estimators $\hat{\nu}_i, \hat{z}_i$, and $\hat{\eta}$, we proceed in two steps: (a) estimate the global parameters and (b) use them as a plug-in to estimate the node parameters. Our proof is based mainly on the following calculation. Consider the marginal probability of a connection between person i and group k, P_{ik}. The form of P_{ik} comes from integrating across all individuals in group k in (4.11), which is consistent with the information in ARD since no individual connections are observed. Further, following [82] and [125], we can model $y_{ik} \mid \nu_i^\star, z_i^\star \sim \text{Binomial}(n_k, P_{ik})$, where P_{ik} is the probability that i connects to a member of group k (the explicit form is derived in the Supplementary Materials

and is a function of $\nu_i^\star$ and $z_i^\star$) and n_k is the number of nodes in group k.

In step (a), we derive the estimators for the global parameters, $\hat{\eta}$, in Section S.1 of the Supplementary Materials, but provide the intuition here. If we consider the probability of an arbitrary link between two members of the same group k, it does not depend on μ_k but only on the variance σ_k^2 and the expected shift in linking probability due to node effect ν_i. Similarly, if we consider the probability of an arbitrary link across two groups k, k' knowing the variance terms, then this provides information on centers μ_k, μ_k'. We can therefore equate the probability of connection between traits with the observed number of traits and solve for the parameter η. Given estimates of the global parameters η, we now estimate the node locations and fixed effects. Since $E(y_{ik}) = P_{ik}/n_k$, we construct a system of equations by equating the ratio of the marginal probability of connection for person i in group k to that in group k' ($P_{ik}/p_{ik'}$) to the ratio of sample averages ($y_{ik}n_{k'}/y_{ik'}n_k$), which does not depend on the fixed effect of node i. This allows us to estimate the locations of all nodes, up to a global isometry in the latent space. We then similarly estimate the node fixed effects, once we have estimated the node locations and global parameters, by equating y_{ik} and p_{ik}. In summary, we construct Z-estimators of the global parameters, the node locations, and the node fixed effects by constructing 4 systems of equations, which allows us to consistently estimate all of the parameters in the latent space model. Equivalently, one can interpret the moments based estimators for the location and fixed effects parameters as coming from maximizing a pseudo likelihood, which we describe in the Supplementary Materials.

We now state the assumptions for consistency of these estimators. $E_{kk'}[\exp\{-d(z,z')\}]$ denotes the expectation of $\exp\{-d(z,z')\}$, where $z \sim F(\mu_k^\star, \sigma_k^\star)$ is independent of $z' \sim F(\mu_{k'}^\star, \sigma_{k'}^\star)$.

Assumption 3.4.1. *For each k, μ_k is in a compact subset of $\mathcal{M}^p(\kappa)$ and σ_k is in a compact subset of $(0, \infty)$.*

Assumption 3.4.2. *The node effects* $\nu_i^\star \overset{iid}{\sim} H$ *satisfy* $E\{\exp(\nu_i^\star)\} < \infty$.

Assumption 3.4.3. *The distribution* F *is a symmetric distribution on* $\mathcal{M}^p(\kappa)$ *that is completely characterized by its mean and variance and satisfies the following two conditions. The function* $z_i \mapsto E_k[\exp\{-d(z_i, z)\}]$ *is Lipschitz for every* $k \in \{1, \ldots, K\}$ *and* $z_i \mapsto E_k[\exp\{-d(z_i, z)\}]/E_{k'}[\exp\{-d(z_i, z')\}]$ *has a pseudo-inverse that is Lipschitz.*

Assumption 3.4.4. *Define the function* F_1 *by*

$$F_1 : (z_i, \sigma_k, \sigma_{k'}) \mapsto E_k[\exp\{-d(z_i, z)\}]/E_{k'}[\exp\{-d(z_i, z')\}].$$

The inverse function F_1^{-1} *is continuous in* σ *and for every* k, k', ℓ, *and* ℓ', *the following two functions are Lipschitz:*

$$\eta \mapsto \frac{E_{kk'}[\exp\{-d(z, z')\}]}{E_{\ell\ell'}[\exp\{-d(z, z')\}]}, \quad \eta \mapsto \frac{E_{kk'}[\{\exp(-d(z, z')\}]^2}{E_{\ell\ell'}[\{\exp(-d(z, z')\}]^2} .$$

Assumptions B.2.3-B.2.4 ensure that the probabilities from (4.11) vary smoothly with changes in the distribution of points on $\mathcal{M}^p(\kappa)$. In the Supplementary Materials, we verify that common distributional choices (e.g., Gaussian in Euclidean space or von Mises Fisher on the hypersphere) satisfy these assumptions and discuss the pseudo-inverse defined in the assumptions above. For simplicity, we suppose that $n_k = n/K$ for each trait, so that traits are evenly divided amongst the nodes, and write $\tilde{n} = n/K$.

Theorem 3.4.1. *Suppose Assumptions B.2.1, B.2.2, B.2.3, and B.2.4 hold. The estimators* $\hat{z}_i$ *and* $\hat{\nu}_i$ *computed from equating the ARD responses and the marginal probability of connections, as well as* $\hat{\eta}$ *(defined in the Supplementary Materials) are consistent for* $z_i^\star, \nu_i^\star$, *and* $\eta^\star$ *as* $m, n \to \infty$, *up to isometry on* $\mathcal{M}^p(\kappa)$ *and satisfy*

$$\max_{1 \le i \le m(n)} d_{\mathcal{M}^p(\kappa)}(\hat{z}_i, z_i^\star) \le \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}},$$

$$\max_{1 \le i \le m(n)} |\hat{\nu}_i - \nu_i^\star| \le \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}},$$

with probability $1 - O(m/\tilde{n}^3)$.

The proof of Theorem 3.4.1 and associated simulations are in the Supplementary Materials.

3.5 A taxonomy for estimating graph statistics

We assume that data arise from one of three models considered in the previous work (beta-model, stochastic block model, or latent space model) and that ARD allows us to estimate the model parameters $\theta_n^\star$. We leverage Theorems 3.2.1, 3.3.1, and 3.4.1 and assume throughout the rest of this work that that the researcher has access to an estimator $\hat{\theta}_n(\mathbf{y})$ of $\theta^\star$. Here, $\theta^\star$ denotes the true parameters of one of the three models, and $\hat{\theta}_n(\mathbf{y})$ denotes the estimates of the model parameters from Theorems 3.2.1, 3.3.1, and 3.4.1. We separate our discussion into two cases: (1) the researcher has a single large network with n nodes; (2) the researcher has many independent networks. We recall for convenience that the user has access to an ARD survey from $m \leq n$ nodes.

3.5.1 Single large network

Starting with the first case, assume the researcher is interested in estimating a network statistic, $S_i\left(g_n^\star\right)$ for node i computed on the graph $g_n^\star$. For simplicity we write this as a function of a single node, though it can easily be extended to functions of multiple nodes. For the purposes of this argument, there is one actual realization of the graph, g_n^*. This is what we would have observed if we had collected information about all actual connections between members of the population, rather than collecting ARD. Importantly, the researcher collecting ARD cannot observe g_n^*. This actual network realization does, however, come from a generative model with parameters that can, by Theorems 3.2.1, 3.3.1, and 3.4.1, be estimated from ARD.

In the following results, we characterize settings where network statics can be consistently estimated using only the $n \times K$ matrix of ARD, $\mathbf{y}$. For simplicity we set $m = n$, though our results hold when $m < n$ as well, though a researcher would need to sample a sufficiently large fraction of the graph to capture the structure of interest

[35]. Based on observing ARD, we compute $E\{S_i(g_n) \mid \hat{\theta}(\mathbf{y})\}$, where $\hat{\theta}(\mathbf{y})$ is the estimator from Theorems 3.2.1, 3.3.1, or 3.4.1 using the ARD $\mathbf{y}$. We are interested in when $E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\}$ is a good estimator of $E\{S_i(g_n) \mid \theta_n^\star\}$ and therefore of $S_i(g_n^\star)$.

There are two general conditions require to consistently estimate graph parameters from ARD. First, the statistic of interest must be one that is relatively stable between draws from the graph generating process. This condition is required since our estimators in the previous section concern parameters of the network formation model, but the goal is to estimate a statistic for a particular draw from this generating process, g_n^*. Second, we require that these estimates of generating model parameters are sufficiently precise and the form of the statistics is such we can control the variation in the estimated network statistic in the presence of small variance in the estimated model parameters. We formalize these conditions in the following theorem. We use the notation $\theta_{j,n}^\star$ to refer to the jth entry of the vector of true parameter values $\theta_n^\star \in \mathbb{R}^n$. Finally, let the partial derivative with respect to the ith component be denoted by $\partial_i E\{S_i(g_n) \mid \theta_n\}$.

Theorem 3.5.1. *Let $g_n^\star$ denote the graph of interest drawn from a conditional edge-independent graph models with parameters $\theta_1^\star, \ldots, \theta_n^\star$, and let $\hat{\theta}_n$ denote estimates of these parameters. Suppose that*

1. $1/n \sum_j |\hat{\theta}_{j,n} - \theta_{j,n}^\star| \xrightarrow{p} 0$,

2. $|E\{S_i(g_n) \mid \theta_n^\star\} - S_i(g_n^\star)| \xrightarrow{p} 0$, *and*

3. *the function $\theta_n \mapsto E\{S_i(g_n) \mid \theta_n\}$ is differentiable and*

$$\max_j \sup_{\theta_n} \partial_j E\{S_i(g_n) \mid \theta_n\} \leq C/n$$

for some finite constant $C > 0$.

Then, $|E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\} - S_i(g_n^\star)| \xrightarrow{p} 0$ as $n \to \infty$.

We provide a proof of Theorem 3.5.1 in the Supplementary Materials. The proof relies on a Taylor series approximation of the network statistic $E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\}$. In particular, we require that the approximation term due to the estimation of $\theta_n^\star$ with $\hat{\theta}_n(\mathbf{y})$ disappear as $n \to \infty$. One sufficient condition for this to occur is given in Conditions 1-3 of Theorem 3.5.1.

Condition 1 of Theorem 3.5.1 requires that the average estimation error goes to zero in probability as the graph size grows. The estimators from Theorems 3.2.1, 3.3.1, and 3.4.1 satisfy Condition 1 of Theorem 3.5.1, since the average estimation error is always upper-bounded by the maximum estimation error. Thus, Theorem 3.5.1 implies that the researcher can use $E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\}$ to estimate $S_i(g_n^\star)$, provided the network statistic $S_i(g_n^\star)$ satisfies Conditions 2 and 3.

Condition 2 of Theorem 3.5.1 equires that $|E\{S_i(g_n) \mid \theta_n^\star\} - S_i(g_n^\star)| \xrightarrow{p} 0$, which must be true regardless of the estimator used to estimate $\theta_n^\star$. Many network statistics are an average of terms, such as the clustering coefficient or the centrality coefficient, and so this condition holds for many statistics of interest. Condition 3 of Theorem 3.5.1 requires that changing the graph model parameters slightly does not change the value of $E\{S_i(g_n) \mid \theta_n\}$ too much. For many common network statistics, this condition is true, as we show in Corollary 3.5.2.

To clarify when the conditions of Theorem 3.5.1 hold and when they fail, we provide several pedagogical examples. Our first example is an obvious failure of the second condition. Specifically, we show the statistic from a given realization does not converge to its expectation, then even after more nodes are observed, there is no increasing information, and the mean-squared error of the estimate should not go to zero. Let $p_{ij}(\theta^\star)$ denote the probability that nodes i and j connect.

Corollary 3.5.1. *Consider a sequence of distributions of conditional edge-independent graphs $\mathbb{P}(g_n|\theta^\star)$ on n nodes, where $\theta^\star$ is known. Given an (unobserved) graph of interest, g_n^*, and $0 < p_{ij}(\theta^\star) < 1$, then the mean squared error for $E\{S_i(g_n)\} = E(g_{ij})$,*

the expectation of a draw from the distribution of any single link g_{ij}, is

$$E[\{E(g_{ij}) - g_{ij}^*\}^2] = p_{ij}(\theta^\star)\{1 - p_{ij}(\theta^\star)\} \ .$$

When a link exists, the mean squared error is $\{1 - p_{ij}(\theta^\star)\}^2$ and when a link does not, it is $p_{ij}(\theta^\star)^2$. In edge-independent models, node-level exchangeability ensures that $p_{ij}(\theta^\star)$ does not vanish with n, which means that the mean squared error cannot go to zero as $n \to \infty$. However, for graph models in which p_{ij} tends to zero, then Condition 2 does hold.

However, for many commonly used and non-trivial network statistics, the conditions of Theorem 3.5.1 do hold. By verifying the conditions of Theorem 3.5.1, we have the following result.

Corollary 3.5.2. *Suppose $g_n^\star$ is drawn from either the β-model, stochastic block model, or latent space model and $\hat{\theta}_n$ is computed from Theorems 3.2.1, 3.3.1, and 3.4.1, respectively. For the following statistics $S_i(g_n)$, we have that $|E\{S_i(g_n^\star) \mid \hat{\theta}_n(\mathbf{y})\} - S_i(g_n^\star)| \xrightarrow{p} 0$.*

1. *Density (normalized degree): The density of node i is $S_i(g_n) = \sum_j g_{ij}/n$.*

2. *Diffusion centrality (nests eigenvector centrality and Katz-Bonacich centrality): Define $S_i(g_n) = S_i(g_n, q_n, T) = \sum_j \{\sum_{t=1}^{T} (q_n g_n)^t\}_{ij}$ for some $q_n = C/n$ and any T.*

3. *Clustering: Let $N(i) = \{j : g_{ij} = 1\}$ denote the neighbors of node i. The clustering coefficient is defined as $S_i(g_n) = \sum_{j,k \in N(i)} g_{jk}/(|N(i)||N(i) - 1|)$.*

Diffusion centrality is a more general form which nests eigenvector centrality when $q_n \geq 1/\lambda_1^n$, and because the maximal eigenvalue is on the order of n, this meets our condition. Here λ_1^n is the largest eigenvalue of the adjacency matrix of g_n. It also nests Katz-Bonacich centrality. In each of these, $T \to \infty$. It also captures a number

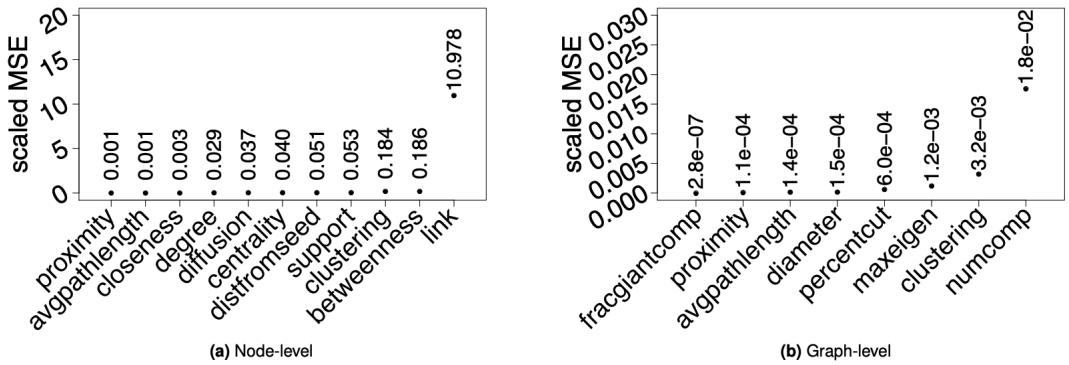


Figure 3.5.1: Scaled mean squared error of node-level and graph-level network features. These results corroborate the theoretical intuition we developed. Specifically, we show in Corollary 3.5.1 that the mean squared error should be large for a single link and in Corollary 3.5.2 that the mean squared error should diminish for (normalized) degree and diffusion at the node level and clustering at the graph level.

of other features of finite-sample diffusion processes that have been used particularly in economics [14, 13]. These notions each relate to the eigenvectors of the network—objects that are ex-ante not obviously captured by the ARD procedure but ex-post work since in this model statistics converge to their expectations.

These results give two practical extreme benchmarks. ARD should not perform well for estimating a realization of any given link in the network. In contrast, it should perform quite well for statistics such as density or eigenvector centrality. Other statistics may fall somewhere in the middle of this spectrum. For example, whether a notion of centrality such as betweenness - which relies on the specifics of the exact realized paths in the network - works well may depend on the specific statistic and network distribution. We explore these predictions empirically in Figure 3.5.1.

3.5.2 Many independent networks

Consider the setting where the researcher has R networks each of size n_r, and the networks are over disjoint sets of nodes. We use the terminology independent networks to refer to such a collection of networks For each network r we observe ARD $n_r \times K$ matrix $\mathbf{y}_r$. We take $n_r = n$ for simplicity, but our results do not require this. Also, we drop the dependence on n in the notation g_r. Every network is generated from a network formation process with true parameter $\theta_r^\star$. In this case of many networks, we consider how well the ARD procedure performs when the researcher wants to learn about network properties, aggregating across the R graphs. This is the case in a large literature [29, 17, 25].

Let $S_r^* = S(g_r^*)$ be a network statistic from the R unobserved graphs generating the ARD. For any given graph from the data generating process, define $S_r = S(g_r)$. For notational simplicity, we consider network-level statistics, but the argument can easily be extended to node, pair, or subset-based statistics. We use the notation $\theta_{i,n,r}^\star$ to denote the ith entry of the vector of parameters $\theta_{n,r}^\star \in \mathbb{R}^n$ for network r. We use similar notation for the estimator $\hat{\theta}_{i,n,r}$.

We consider two regression problems. In the first problem, the goal of the researcher is to estimate the model

$$O_r = \alpha + \beta S_r^* + \epsilon_r \quad r = 1, \ldots, R \,,$$

where O_r is some socio-economic outcome of interest and and the parameter of interest is β. As before, S_r^* is unobserved because g_r^* is unobserved and the researcher only has ARD, $\mathbf{y}_r$. The researcher instead estimates the expectation of the statistic given using ARD, $\bar{S}_r = E\{S_r \mid \hat{\theta}_{n,r}\}$. The regression becomes

$$O_r = \alpha + \beta \bar{S}_r + u_r. \tag{3.4}$$

and $\hat{\beta} = \hat{\beta}_{n,R}$ is the ordinary least squares (OLS) estimator of β from (3.4). Critically, $\hat{\beta}$ depends on the size of each network n and the number of networks R.

In the second regression model we consider, the network feature is an outcome that responds to an intervention, T_r:

$$S_r^* = \alpha + \gamma T_r + \epsilon_r.$$

We let $\hat{\gamma}_{n,R}$ denote the OLS estimator of γ from the regression

$$\bar{S}_r = \alpha + \gamma T_r + \epsilon_r. \tag{3.5}$$

Theorem 3.5.2. *Let $\hat{\beta}_{n,R}$ denote the OLS estimate from (3.4) and let $\hat{\gamma}_{n,R}$ denote the OLS estimate from (3.5). Suppose that*

1. *the estimators of the parameters for the rth network, denoted by $\hat{\theta}_r(n)$, satisfy*

$$\max_{1\le r\le R} \frac{1}{n}\sum_{i=1}^{n} |\hat{\theta}_{i,n,r} - \theta^\star_{i,n,r}| \xrightarrow{p} 0 \text{ as } n, R \to \infty$$

2. *the functions $\theta_n \mapsto E\{S_r \mid \theta_{r,n}\}$ is differentiable for each network r and each network size n. Suppose also that*

$$\max_{1\le r\le R} \max_j \sup_{\theta_{n,r}} \partial_j E\{S_r \mid \theta_{n,r}\} \le \frac{C}{n},$$

for some finite constant $C > 0$.

If $E(\epsilon_r \mid S_r^\star) = 0$ and the design matrix has full rank, then $|\hat{\beta}_{n,R} - \beta| \xrightarrow{p} 0$ and $|\hat{\gamma}_{n,R} - \gamma| \xrightarrow{p} 0$ as $n, R \to \infty$.

The following theorem shows that the three conditions from Theorem 3.5.2 hold.

Theorem 3.5.3. *Suppose that each network $g^\star_{n,r}$ is known to be drawn from either the beta-model, stochastic block model, or latent space model and $\hat{\theta}_n$ is computed from Theorems 3.2.1, 3.3.1, and 3.4.1, respectively. If $S_r^\star$ is the density, centrality, or clustering of a node in network r, as defined in Corollary 3.5.2, then Conditions 1 and 2 of Theorem 3.5.2 hold if $Rn/\exp(n) \to 0$.*

In words, Theorem 3.5.3 states that a researcher is able to run the regression in (3.4) using the estimators in Theorems 3.2.1, 3.3.1, and 3.4.1 to consistently estimate β, the true effect of the network statistics on the observed socio-economic outcomes.

Take the most extreme example of a single link, where we know its presence cannot be identified in a single large network. Even if we were interested in a regression of $y_{12,r} = \alpha + \beta g_{12,r} + \epsilon_r$, where whether nodes 1 and 2 are linked affects some outcome variable of interest across all R networks, we can use $E\{g_{12,r} \mid \hat{\theta}(y_r)\}$ in the regression to consistently estimate β. Here, nodes 1 and 2 refer to arbitrarily labeled nodes and can be different across the R networks. In contrast to the single network case, where the mean squared error of the estimate of $g_{12,r}$ does not tend to zero as n grows, here simply having the conditional expectation is enough to estimate the slope of interest, β. Therefore, with many graphs, the ARD procedure works well under weaker conditions on the network statistics. However, despite the generality of Theorem 3.5.2, Condition 2 of Theorem 3.5.2 still must hold. Some statistics are more sensitive to the input parameters and thus might not satisfy Condition 2. For example, the number of connected components has a higher mean squared error than the other statistics, which suggests that this statistic might lead to poor OLS estimators in (3.4) and (3.5).

3.6 Simulation results

3.6.1 Single large graph

We explore the results for a single large graph through simulation exercises. We first generate 250 graphs from the generating process in (4.11) then randomly assign each node to one of K traits. Each network consists of 250 nodes, similar to the size of villages from in [13]. We then draw a sample of nodes from the graph and construct ARD using traits. Our simulation does not reflect error in the ARD, which may arise if, for example, a person is a member of a group but the respondent does not have

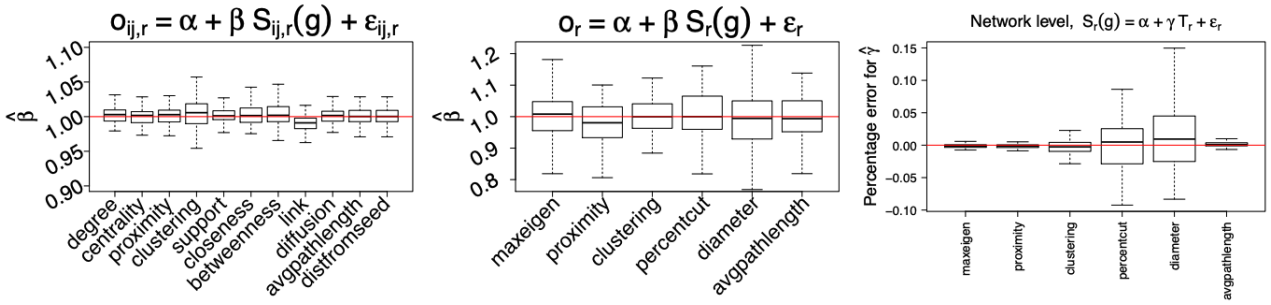


Figure 3.6.1: Boxplots for the simulation experiments with multiple independent networks. In the left figure, we consider a regression where the node-level network statistics determine outcomes on one network. In the middle figure, we consider a regression where network-level statistics determines outcomes on multiple networks. In the right figure, we consider a regression where a treatment determines a network-level statistics. On the x-axis we provide the network statistics used and the y-axis represents the value of the regression coefficients estimators. The red line indicates the true value of the regression coefficients. These results corroborate the theoretical intuition developed in Theorems 3.5.1 and 3.5.2.

this information (e.g., [102], [177], [59], or [61]). We then estimate graph statistics using the procedure outlined in [28].

Figure 3.5.1 plots the mean squared errors of our estimation procedure across a range of common network statistics. These mean sqaured errors reflect uncertainty in estimation of the model parameters and in the underlying network statistics. In order to make the mean squared errors comparable across statistics, we scale by $1/E(S_i)^2$. Subfigure (a) in Figure 3.5.1 focuses on node level statistics. We compute ten node level statistics: (1) proximity (average of inverse of shortest paths); (2) average path length; (3) closeness centrality (the average inverse distance from i over all other nodes); (4) degree (the number of links); (5) diffusion centrality (as defined in [13] – an actor's ability to diffuse information through all possible paths); (6) eigenvector

centrality (the ith entry of the eigenvector corresponding to the maximal eigenvalue of the adjacency matrix for node i); (7) the average distance from a randomly chosen seed (as in a diffusion experiment where the seed has a new technology or piece of information); (8) support (as defined in [92] – whether linked nodes ij have some k as a link in common); (9) clustering (the share of a node's links that are themselves linked); (10) betweenness centrality (the share of shortest paths between all pairs j and k that pass through i); (11) whether link ij exists. The results from the simulation, ordered in terms of scaled mean squared error in the figure, are consistent with the theoretical results. Statistics such as density and centrality take values for each realization that are nearly their expectation, meaning that we can recover the statistics with low mean squared error. For a single link this is not the case and, correspondingly, the simulations show higher error.

Subfigure (b) of Figure 3.5.1 focuses on graph-level statistics. The graph level statistics are as follows: (1) share of nodes in the giant component; (2) average proximity (average of inverse of shortest paths); (3) average path length; (4) diameter; (5) the share of links across the two groups relative to within the two groups where the cut is taken from the sign of the Fiedler eigenvector (this reflects latent homophily in the graph); (6) maximal eigenvalue; (7) clustering; (8) number of components. All network statistics, with the exception of the number of components one, have small scaled mean squared error. This reflects the intuition of Corollary 3.5.2. ARD recovers statistics that converge to their expectations, such as density, and might fail to recover statistics that do not.

We also evaluate our approach using observed, fully-elicited graphs. We use data from [13], which consists of completely observed graphs from 75 villages in rural India. In each village, about one-third of respondents were asked ARD questions. [28] compare statistics estimated with ARD (using estimated formation model parameters) from these graphs with the same statistics calculated using the complete graph. We leverage these results and present a different aspect: how the mean squared error

changes as the size of the graph grows. We present results for individual-level statistics from these graphs and compute mean squared error across individuals. Our results using graphs with real-world complexity and properties (e.g. density and community structure) confirm the results from our simulation experiments. These results are presented in Figure S2 of the Supplementary Materials.

3.6.2 Many independent networks

Multiple independent networks often arise in experiments, so we simulate a setting where we assign graph level treatment randomly to half of the graphs. Graphs in the control group have expected degree generated from a normal distribution with mean 15 and variance 25, while graphs in the treatment group are generated from a normal density with mean 25 and variance 25. Each graph has 250 nodes. All graphs have a minimum expected degree of 5 and a maximum expected degree of 35. Due to the association between density and treatment, we expect treatment effects on graph-level statistics, such as average path length and diameter. The average sparsity over all graphs is 20/250=0.08, which is a value similar to Karnataka data discussed in [26]. For individual measures, 50 actors are randomly selected in each network. For links measured between actors, 1000 pairs are randomly selected in each network. For network level measures, there is one measure per network, so the regression consists of R data samples, where R is the number of networks.

Figure 3.6.1 shows the simulation exercise with multiple independent networks. We use formation model parameters, $\theta^{\star}$, to get $\bar{S}_{ij,r}$ or $\bar{S}_r$ and include results using estimated model parameters in the Supplementary Materials (Figures S3, S4, and S5). We present results with $R = 200$ ($R = 50, 100, 200$ are in the Supplementary Material). ϵ_r comes from a normal distribution with zero mean, and $\text{var}(\epsilon_r) = \text{var}(S^*_{ij,r})$ to maintain a 0.5 noise to signal ratio.

The first two panels in Figure 3.6.1 show the distribution of the estimate of β in a regression where the network statistic predicts an outcome of interest. The middle

line of each boxplot is the median $\hat{\beta}$, and the borders of boxes denote first and third quartiles. All boxplots have outliers removed. The leftmost panel gives results for individual level measures while the center panel gives network level measures. Among the node level statistics we see that all estimated $\hat{\beta}$s are close to the simulation value of one. The individual link measure, though empirically similar, is not centered around the true simulated value. The downward bias is an example of attenuation bias or regression dissolution, since there is variability in the network statistic acting as the covariate. The indicator of the presence/absence of a single link is the most variable of the network measures and, thus, bias persists for the link measure when it does not for the others. For graph level measures, all estimated coefficients are centered around the generated values.

The rightmost panel in Figure 3.6.1 shows results for the case where the network statistic is the outcome and is predicted by another covariate, in this case treatment status. The percentage error is defined as $(\hat{\gamma} - \gamma)/\gamma$. Percent cut and diameter has large variation of percent errors than the other measures. This is due to the fact that the treatment effect, density differences between treatment and control, has a smaller effect on percent cut and diameter than on other measures. The average percent of variation explained by treatment in S_r for percent cut and diameter is around 0.3, while it is around 0.5 for other measures.

3.7 Discussion

Collecting full network data in large networks (e.g., a city) or across many networks (e.g., villages or schools) requires enumerating all egos and alters and therefore can be prohibitively expensive, logistically hard, or face privacy concerns. The use of ARD allows the researchers to overcome these problem by fitting frequently used and rich generative models, which can be then used to estimate socio-economic quantities and parameters of interest. This can include features of the network, but also responses in network structure to interventions as well as how socio-economic outcomes are

affected by network structure.

In this work, we first demonstrated that by using ARD we are able to consistently estimate parameters in several families of frequently used generative network models, including ones where the number of parameters grows as the graph size grows. Second, we provided a taxonomy to describe when we may expect to estimate socio-economic features consistently using ARD. Together, our theoretical results and supportive simulations using empirical data, present new insights into settings where researchers can count on ARD to reliably estimate socio-economic quantities of interest. This makes the study of socio-economic networks much more accessible to a wide set of researchers; in our own setting using ARD delivers the same economic conclusions as the full network data deoes but at 80% less cost [28].

There are several promising avenues for future work. First, the techniques studied here are likely more relevant for networks of the scale of villages or cities counties but certainly not necessarily things like large social media networks. It is true that when the number of nodes is very large, one needs many more traits K to exceed the number of latent communities C (since presumably a large C is needed to fit the network well). Note that geography can be included, to some degree, in a reasonably natural way. After all, one can imagine carving out a set of locations (as set if L regions) and now a "type" K is the sub-trait (e.g., caste) crossed with the location. So $K = T \times L$ and we would use $K > C$ in this way. This is not the only approach, but we leave a complete exposition of this strategy to future work. Second, we demonstrate consistent estimation for edge-independent network models. Extending these results to a broader class of models, particularly those that are asymptotically sparse or which have correlated edges would extend the reach of our work and we believe much of the infrastructure we developed around the necessary properties of network statistics would still apply [137, 167, 161]. Third, a natural question to ask is whether other data collection strategies might be more useful to deliver consistent estimates for quantities that fall outside of the taxonomy of statistics that are estimable with ARD.

Chapter 4

SPECTRAL GOODNESS-OF-FIT TESTS FOR COMPLETE AND PARTIAL NETWORK DATA

4.1 Introduction

In this chapter, we turn out attention to the problem of assessing model goodness-of-fit for network data. Networks consist of connections, also known as edges or ties, between individual actors or nodes. Such data are common in a variety of settings in the social, economic, and health sciences. The simplest model is the Erdös-Rényi model [57], in which edges form independently with the same probability. More complex models have been developed, such as stochastic block models (SBM) and degree-corrected variants [88, 3, 148, 173], latent space models [85, 84, 158, 116], exponential random graph models (ERGMs) [89, 90], and many more. Among others, [71] provides a survey of common network models.

Given the multiple available models, a natural question in practice is how to choose a model that is appropriate for a particular dataset. Broadly speaking, there are two common approaches to this problem currently in the literature. A first approach leverages the fact that the problem has a "parametric null." Since many network models are also generative, one common strategy is to estimate parameters of the model in question, then simulate a series of graphs. Statistics from the fitted model should resemble the observed statistics [136, 41, 159, 68]. A potential issue with these methods is that, in many setting, there is limited information available to a practitioner to decide which statistics to use for comparison. In some cases, the researcher can choose statistics that are important to their application, but it might not always be possible to select *a-priori* which statistics will be the most important in future analyses. This

method also requires taking multiple samples from the generative process, which can be cumbersome in high dimensional settings.

A second common strategy for assessing goodness-of-fit (GoF) involves using the penalized likelihood methods, such as the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC). For example, AIC or BIC could be used to select the dimension of the latent space in latent space models, but as we show with simulations in Appendix C.2, the BIC approach leads to poor dimension estimates. In fact, the manual for one of the most common software packages for fitting parametric models, `latentnet`, states "*It is not clear whether it is appropriate to use this BIC to select the dimension of latent space ....*" This issue has also been documented previously [132, 79, 145, 111, 72]. By contrast, the goodness-of-fit test we propose here does not use a penalized likelihood to select dimension. Instead, it uses the eigenvalues of a random matrix that measures how well an assumed model fits the data. This procedure, as we show in this work, outperforms BIC and similar metrics when applied to selecting the dimension of the latent space.

In this work, we present a novel goodness-of-fit test to assess model fit when the network of interest is undirected or directed. Our method also accommodates partial network data, which is a vital part of modern network analysis [20, 124, 28, 27, 5], but are not easily handled in existing goodness-of-fit tools. Our goal is to derive a testing framework for the hypotheses

$$H_0 : G \sim F_\theta, \quad H_a : H_0 \text{ is false} , \tag{4.1}$$

where G is a random network of interest and F_θ is a parametric network model with an (unknown) parameter vector represented by θ. In words, we have an assumed parametric network model, F_θ, and our goal is to test whether G could be drawn from F_θ. Throughout this work, we will use graph and network interchangeably.

A critical aspect in the setup of the tests above is that they assess goodness-of-fit for the entire model simultaneously. In many settings, this means that the

contribution of individual parameters to the fitness measure may not be separately identified. Take, as an example, the latent distance model discussed above. This model has both individual effects for each respondent and latent distances for each pair. A common question, as described above, involves testing for the dimension of the latent space. To test exclusively for the dimension, we would need to either marginalize over or condition on potential values for the additional model parameters (see the discussion in [132], for example, which makes this point in the related setting of multidimensional scaling). In the latent space model this is particularly challenging since both the latent distances and individual effects impact overall graph properties, such as the density (see, for example, [116] for further discussion). In our approach, we ask a related, but distinct question from the literature that tests for specific model parameters. In the case of the latent space model, for example, our test asks whether a model, overall, could have plausibly generated a given set of data, rather than attempting to identify a single "true" latent dimension. Despite this, in our simulations, we see however that this approach tends to find the true dimension with high probability.

The motivation for our test statistic is taken from a result that has been used before in community detection [110, 21] and two-sample tests for networks [40]. The result, which goes back to [58], states that if A is a $n \times n$ random symmetric matrix with (i) $E(A_{ij}) = 0$ and (ii) $\sum_{j \neq i} \mathrm{Var}(A_{ij}) = 1$ for each i, then $n^{2/3}(\lambda_{\max}(A) - 2)$ converges in distribution to a random variable with a Tracy-Widom distribution, where $\lambda_{\max}(A)$ is the largest eigenvalue of the matrix A. The same argument shows that the smallest eigenvalue of A, which we denote by $\lambda_{\min}(A)$, satisfies a similar central limit theorem.

Leveraging this result, we propose a two-step procedure to test the hypothesis in (4.1). First, we compute an estimate $\hat{\theta}$ of θ and estimate $\hat{P}_{ij} := P(G_{ij} = 1|\hat{\theta})$, where $G_{ij} = 1$ if person i and person j are observed to be connected in the network, and P_{ij} is the probability of such a connection (as defined by the assumed parametric model).

Second, we define the random matrix A by, for $i \neq j$,

$$A_{ij} = \frac{G_{ij} - \hat{P}_{ij}}{\sqrt{(n-1)\hat{P}_{ij}(1-\hat{P}_{ij})}}, \tag{4.2}$$

and $A_{ii} = 0$. Under H_0, we expect that a reasonable estimator for $\hat{P}_{ij}$ should approximate $P(G_{ij} = 1|\theta)$, so A from (4.2) should approximately satisfy conditions (i) and (ii). We can then compare the largest and smallest eigenvalues of A against quantiles of the Tracy-Widom distribution to construct a test of H_0 in (4.1).

This chapter contributes to the literature on testing goodness-of-fit for network models in three ways. First, we expand work by [110], which estimates the number of communities in a stochastic block model, to accommodate a variety of common parametric network models. Second, we develop a test for directed data by introducing a similar central limit theorem for eigenvalues of non-symmetric matrices from [96] and [33]. Third, we show how to test (4.1) when the researcher only has access to partial network data, such as Aggregated Relational Data. Along with asymptotic arguments we also present a bootstrap procedure which improves performance of our hypothesis tests in finite samples.

The chapter is structured as follows. First, we review relevant literature in the remainder of this section. Next, we introduce the construction of the Tracy-Widom distribution and asymptotic arguments for undirected, directed, and partial network data in Section 4.2. In Section 4.3, we discuss a bootstrap correction algorithm to improve finite sample properties. We then present a series of network models that are compatible with our method in Section 4.4 along with simulation results. Lastly, in Section 4.5, we analyze several observed networks using a latent distance model [85, 83, 84], which assumes that relationships in the network depend on the positions of actors in latent "social space" of low but unknown dimension. Our goal with these data is to test for the minimal latent dimension. The R code for the simulations and to implement the method can be found in https://github.com/slubold/Network_GOF.

4.1.1 Literature Review

Goodness-of-fit methods for dyadic data generally address the question "Does the proposed model fit my network data well?" One reason this problem is challenging, among others, is that there is often only one network of interest. In other words, we cannot access more draws from the distribution that generated the observed network. Many goodness-of-fit methods for dyadic data try to use Monte Carlo methods to simulate network statistics, such as average degree or average path length. If the simulated values match the observed values, then one might claim that the fitted model is adequate. See, for example, [136] which derives a test for an Erdös-Rényi network using the degree distribution. [67] looks at using small graph statistics, such as the number of triangles or edges, to determine if there is community structure in a network. Each of these methods, generally speaking, requires a new derivation of a central limit theorem for the network statistic of interest under a suitable null hypothesis, which makes a general method hard to derive.

[159] proposes a general goodness-of-fit method based on resampling the graph Laplacian's eigenvalues and constructing confidence intervals based on these values. Their null model is always the Erdös-Rényi model, which may not reflect the complexity of observed data. Similarly, [114] proposed a way to do cross validation with network data. In terms of more specific methods, For example, [172] derives a test of the form of an ERGM (defined formally in Section 4.4) using kernel stein discrepancy. To the best of our knowledge, network goodness-of-fit tests using partial data are not well-studied. Recent work on modeling partial network data, such as [20], [28], [27], [5], and [37], consider model adequacy using out of sample prediction, which may be appropriate in some circumstances but asks a fundamentally different question than goodness-of-fit.

Finally, our methods draw on results from random matrix theory and its applications. See, among others, [163] and references therein, for an introduction. Our

method builds on the method presented in [110] and [21], which also use spectral properties to estimate the number of communities in a stochastic block model. See [58] and [65] and their references for recent work on related central limit theorems for eigenvalues. [40] proposes a two-sample test for network using the Tracy-Widom distribution, which is the same distribution that motivates our goodness-of-fit test statistics.

4.2 Methodology

We now outline the goodness-of-fit problem. Let Y be an $n \times n$ matrix containing relationships between actors or nodes, which we label from 1 to n. In this work, we usually only consider Y to be binary-valued, so Y represents a network on n nodes. We suppose that Y is drawn from some distribution F. The network goodness-of-fit question then asks whether a given set of observed data Y could plausibly have been generated by F. In many cases, it is possible to index F by some parameter θ. We are therefore interested in testing the GoF hypothesis in (4.1). When $F = F_\theta$, we write $P_{ij} = P(Y_{ij} = 1|\theta)$ to mean the probability that nodes i and j connect, given the parameter θ.

The methodology we present in this work to test (4.1) requires an estimate of θ. We can estimate θ via maximum likelihood estimation (MLE), for example. Assuming we have estimated θ with $\hat{\theta}$, we can use the parametric form of F_θ to obtain a fitted distribution $F_{\hat{\theta}}$. From this distribution, we can estimate $\hat{P}_{ij} = P(Y_{ij} = 1|\hat{\theta})$, which is the probability that nodes i and j connect, given the parameter $\hat{\theta}$. In the next section, we derive a testing framework to test the hypothesis in (4.1). We discuss the cases of undirected, directed, and partially-observed networks in their own sections, since each case requires a different approach.

4.2.1 Undirected Networks

We first consider the case where Y corresponds to an undirected binary matrix. The following result, which motivates our test statistic for (4.1), states that the eigenvalues of a transformation of the adjacency matrix satisfy a central limit theorem. See [58], [109], [65], [110], and [169] for related results and discussion. Formally, this result combines results from [58] and [109] and is formulated in Lemma A.1 of [110], among other works.

Theorem 4.2.1 (Lemma A.1 of [110]). *Let Y be the adjacency matrix of a random graph on n nodes with edges drawn independently with probability P_{ij}. Define the $n \times n$ random matrix A with entries*

$$A_{ij} := \frac{Y_{ij} - P_{ij}}{\sqrt{(n-1)P_{ij}(1-P_{ij})}}, \quad A_{ii} = 0 \, .$$

Then, as $n \to \infty$,

$$t_1 := n^{2/3} \left(\lambda_{\max}(A) - 2 \right) \xrightarrow{d} TW_1, \tag{4.3}$$

$$t_2 := n^{2/3} \left(-\lambda_{\min}(A) - 2 \right) \xrightarrow{d} TW_1 \tag{4.4}$$

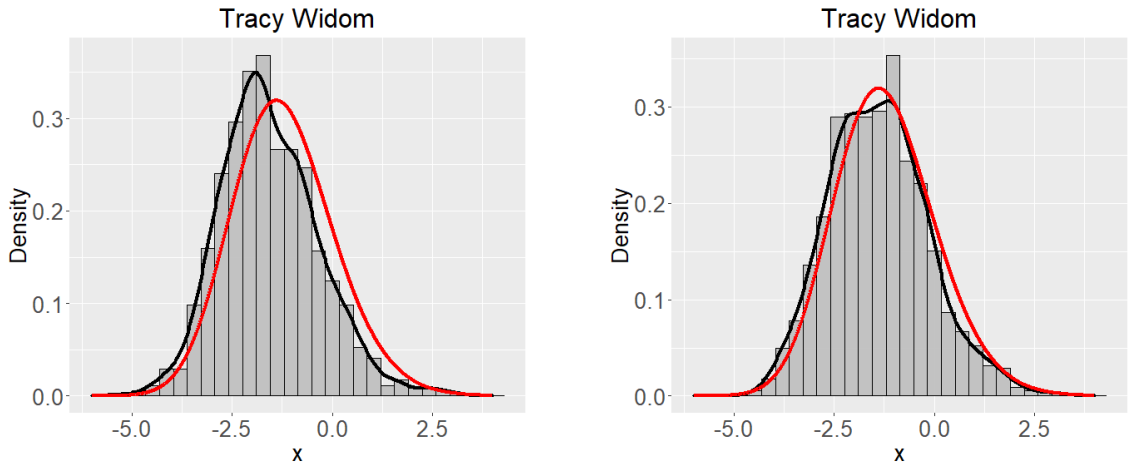
where TW_1 is the Tracy-Widom distribution with parameter 1.

In words, this result states the largest and smallest eigenvalues of the matrix A satisfy a central limit theorem. In Figure 4.2.1 we plot the TW_1 distribution as well as $n^{2/3}(\lambda_{\max}(A) - 2)$ to illustrate this theorem.

In general when testing (4.1), we do not know θ, but as mentioned in the previous section, we do have an estimate $\hat{\theta}$. We therefore plug in $\hat{P}$ in place of P, where $\hat{P}_{ij} = P(G_{ij} = 1 | \hat{\theta})$. We then define

$$\hat{A}_{ij} := \frac{Y_{ij} - \hat{P}_{ij}}{\sqrt{(n-1)\hat{P}_{ij}(1-\hat{P}_{ij})}}, \quad \hat{A}_{ii} = 0 \, .$$

This suggests a test of (4.1) based on both $\lambda_{\max}(\hat{A})$ and $\lambda_{\min}(\hat{A})$, using the statistics $\hat{t}_1$ and $\hat{t}_2$, where the hat indicates that we replace the unknown P_{ij} with the estimate $\hat{P}_{ij}$.



(a) (b)

Figure 4.2.1: Distribution of statistic in Theorem 4.2.1 for $n = 50$ (left) and $n = 1000$ (right), where the red curve corresponds to the Tracy-Widom distribution with $\beta = 1$. The difference in the distributions decreases as n increases, but the convergence is slow. This motivates the bootstrapping correction algorithm, given in Algorithm 4.

We reject H_0 in (4.1) when

$$\max\{\hat{t}_1, \hat{t}_2\} > TW_1(1-\alpha/2) \quad \text{or} \quad \min\{\hat{t}_1, \hat{t}_2\} < TW_1(\alpha/2)\,, \tag{4.5}$$

where t_1 and t_2 are the test statistics from (4.3) and (4.4), $TW_1(\alpha/2)$ and $TW_1(1-\alpha/2)$ are the $(\alpha/2)\%$ and $(100-\alpha/2)\%$ quantile of the TW_1 distribution, respectively. If we instead use t_1 and t_2, this test has size α by a union bound argument. In practice, the test that uses $\hat{t}_1$ and $\hat{t}_2$ is size α if the eigenvalues of $\hat{A}$ converge quickly enough to the eigenvalues of A in probability. The next result, which we do not believe has previously been reported in the literature, gives a rate at which this happens.

Theorem 4.2.2. *If* $n^{2/3}(\lambda_{\max}(\hat{A}) - \lambda_{\max}(A)) = o_P(1)$, *then* $n^{2/3}(\lambda_{\max}(\hat{A}) - 2) \xrightarrow{d} TW_1$. *Furthermore, the test in (4.5) has size* α *as* $n \to \infty$.

Theorem 4.2.2 states that if we want to show that the test based on $\hat{P}$, rather than the unknown P, has size α as $n \to \infty$, we must prove that $\lambda_{\max}(\hat{A})$ converges fast enough to $\lambda_{\max}(A)$. This is problem specific and depends on the complexity of the graph distribution. [110] shows under certain constraints, the conditions of Theorem 4.2.2 hold in the case of a stochastic block model. To the best of our knowledge, there is no work that verifies this condition in other, more complicated models, such as the latent space model. In the simulations in this work, we assume that this condition holds and see that in many models, we achieve an approximately size α test as $n \to \infty$. This suggests that in these models, the condition in Theorem 4.2.2 holds, but we do not have a formal proof that Theorem 4.2.2 holds in these models.

Before continuing, we comment on the term $P(G_{ij} = 1|\hat{\theta})$. In many models, such as the stochastic block model (SBM), this term is available in closed form in terms of $\hat{\theta}$. In other cases, such as exponential random graph models, this is not the case, since the graph model asserts a joint distribution over all pairs of edges. We are not aware of a formula for marginal probability of a single edge in terms of the graph model. In these cases, we need to estimate the marginal probability matrix P. We present a simple method in Algorithm 6 to do this.

4.2.2 Directed Networks

In the case where A is the adjacency matrix of a directed network, then Theorem 4.2.1 will not be applicable, since the eigenvalues of A are not guaranteed to be real. To test (4.1) in the case of directed networks, we therefore introduce two central limit theorems for the singular values of non-symmetric random matrices, which always exist and are real. Both of these results assume a matrix with independent entries with (1) mean zero and (2) variance 1. Note that this differs slightly from the undirected case, where we required that the sum of the variance of entries in each row was 1. To satisfy conditions (1) and (2) in the directed case, we define the random matrix $\hat{A}$

with entries

$$\hat{A}_{ij} := \frac{Y_{ij} - \hat{P}_{ij}}{\sqrt{\hat{P}_{ij}(1 - \hat{P}_{ij})}}, \quad \hat{A}_{ii} = 0 \, . \tag{4.6}$$

where again $\hat{P}_{ij} = P(G_{ij} = 1|\hat{\theta})$. Notice that there is no $(n-1)$ in the denominator of the expression for $\hat{A}$.

Theorem 4.2.3 (Theorem 1.1 of [96])**.** *Let A be a $m \times n$ standard Gaussian random matrix such that*

$$A_{ij} \sim_{iid} \mathcal{N}(0,1) \quad \textit{for all } 1 \leq i \leq m,\ 1 \leq j \leq n.$$

Let $s_{\max}(A)$ be the largest singular value of A. Then, if $m = m(n) \to \infty$, with $m \leq n$, and $\lim_{n\to\infty} m(n)/n = \gamma \in (0, 1]$,

$$\frac{s_{\max}(A)^2 - \mu_{1,n}}{\sigma_{1,n}} \xrightarrow{d} TW_1 \, ,$$

where $\mu_{1,n} = (\sqrt{n-1} + \sqrt{m})^2$ and $\sigma_{1,n} = \sqrt{\mu_{1,n}}(1/\sqrt{n-1} + 1/\sqrt{m})^{1/3}$. Moreover, if $\gamma > 1$, then the result remains true up to the swap of the roles of m and n in the formulas.

Theorem 4.2.3 requires that the entries of A be Gaussian, which is not the case when A is the (binary) adjacency matrix of a random network. In Figure 4.2.3, we show that the convergence claim in Theorem 4.2.3 still holds reasonably well when the entries of A follow a Poisson binomial distribution. This suggests that we can use Theorem 4.2.3 to construct a test when the entries of A are not Gaussian.

For directed networks, Theorem 4.2.3 suggests that we take our test statistic to be $(s_{\max}(\hat{A})^2 - \mu_{1,n})/\sigma_{1,n}$ with $m = n$ and the rejection region to be $\{x : TW_1(\alpha/2) < x < TW_1(1-\alpha/2)\}$, which is identical to the undirected case. Moreover, it is not necessary to restrict $m = n$, which indicates such a test statistic is also applicable on directed networks or networks for which we only have partial network data. We will elaborate on these ideas in a later section.

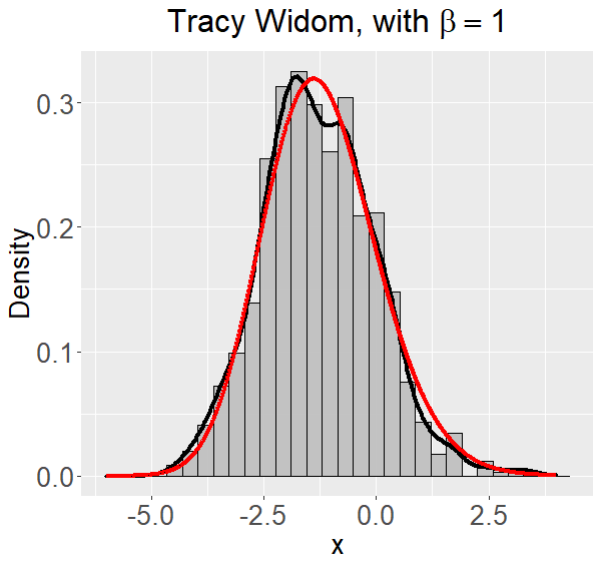


Figure 4.2.2: Distribution of the test statistics for networks with size $n = 1000$ in Theorem 4.2.3. The red curve in the figure corresponds to the Tracy-Widom distribution with parameter $\beta = 1$. Overall, the convergence to the Tracy-Widom distribution is good enough and the theoretical Tracy-Widom distribution can be used for constructing our test statistic.

Our second result states that the scaled singular of a non-symmetric random matrix, when suitably transformed as in (4.6), converge to an exponential-type distribution.

Theorem 4.2.4 (Theorem 2.4 of [33]). *Suppose that A is a random $n \times n$ non symmetric matrix whose entries have mean zero and variance 1. Then, if $s_{\min}(A)$ denotes the smallest singular value of A,*

$$\lim_{n\to\infty} \mathbb{P}\left(\sqrt{n}s_{\min}(A) \geqslant t\right) = \exp\left(-\frac{1}{2}t^2 - t\right).$$

Theorem 4.2.4 suggests that we take our test statistic to be $\sqrt{n}s_{\min}(A)$ and the rejection region to be $\{x : x > q_{\mathrm{E}}(1-\alpha)\}$ where $q_{\mathrm{E}}(1-\alpha)$ is the $(1-\alpha)100\%$ percentile of the distribution in Theorem 4.2.4.

To summarize, in this section we provided two central limit theorems for the singular values of random, non-symmetric matrices, which require that the entries of the random matrix have mean zero and variance 1. We discussed how to use the observed adjacency matrix Y to construct such a matrix and to derive a test statistic for the GoF hypothesis in (4.1). In Section 4.4.5, we discuss the performance of these two test statistics.

4.2.3 Partial Network Data

Suppose our goal, as it was above, is to test whether a graph G is drawn from a particular model. That is, we want to test the hypothesis in (4.1). In many applications, complete network data is not available, is too expensive to collect, or cannot be collected for privacy-related reasons. A common form of partial network data, particularly in economics, is Aggregated Relational Data (ARD) [28, 27, 5]. In this work, we focus on using ARD to test the goodness-of-fit hypothesis in (4.1), but we believe our framework can be extended to other data types too.

To describe what ARD is, suppose that we can partition the nodes of the network into K categories $G_1, \ldots, G_K$. These categories correspond to different covariates, so for example all nodes in G_1 have black hair and all nodes in G_3 are left-handed. We then ask $m \leq n$ nodes how many people they know with trait j for $j = 1, \ldots, K$. In summary, by collecting ARD of a network with size n, we actually collect $\{Y_{ij} : i = 1, \ldots, m, j = 1, \ldots, K\}$, with

$$Y_{ij} := \sum_{k \in G_j} G_{ik} \ . \tag{4.7}$$

Recall that G_{ik} represents whether an edge exists between nodes i and k, so Y_{ij} represents how many people node i knows with trait j. We show in this section how to use ARD to test some of the network models previously mentioned. We assume, as is common in applications, that $|G_j|$ is known. Such information can come from census data or similar data sources.

To illustrate ARD, we consider a simple example. Suppose that we consider $K = 2$ and $m = 4$ and we then collect the following data Y, written in matrix form as

$$Y = \begin{pmatrix} 3 & 10 \\ 1 & 7 \\ 0 & 3 \\ 1 & 5 \end{pmatrix}. \tag{4.8}$$

This means, for example, that the first person we surveyed knows 3 people with trait 1 and the fourth person we surveyed knows 5 people with trait 2. In the above example, m does not have to equal K (and usually does not in practice), so Y is often not square. This means that we cannot apply Theorem 4.2.1 to test (4.1). Instead, we test the hypothesis with Theorem 4.2.3, which is applicable for non-square matrices.

One challenge is to estimate θ, given just the ARD. In this work, we consider a simple test of whether there is degree heterogeneity, which is equivalent to testing if the underlying model is an Erdös-Rényi model. Other, more complicated methods exist for estimating the parameters using only ARD in more complex models, such as those given in [5] or [28].

Before continuing, we discuss whether the assumptions in Theorem 4.2.3 hold for ARD. We discuss three assumptions. First, recalling the notation from Theorem 4.2.3, this result requires that $m \leq n$ so the matrix is "long" rather than "tall". In practice, the number of traits is smaller than the number of nodes we survey, so Y is often "tall", as it is in (4.8). This does not pose a problem since the singular values for Y and Y^T are the same. Second, Theorem 4.2.3 requires that the $m/n \to \gamma \in (0, 1]$, which in the ARD context requires that the number of traits grows with m. Previous work on the large sample properties of ARD estimators has either taken the number of traits as fixed in [27] or growing slowly, like $K = O(\sqrt{n})$ as in [5]. Despite the assumption that K grow with the sample size, our simulations in Section 4.4.4 show that this result of Theorem 4.2.3 still hold reasonably well. Lastly, in Theorem 4.2.3, A is required to be a Gaussian random matrix with continuous entries. ARD are, however, counts.

If the number of ARD responses are relatively large, then the counts may appear reasonably normally distributed. In most cases, however, we expect that the counts for most categories will be small. Despite these potential violations of the required assumptions, Figure 4.2.3 shows that the approximation works well, at least visually, when the entries of the random matrix are not Gaussian and the underlying data come from a skewed distribution of counts, as would be the case in ARD. We give simulation evidence in Section 4.4.4 that the approximation is sufficiently accurate to achieve favorable performance in hypothesis tests.

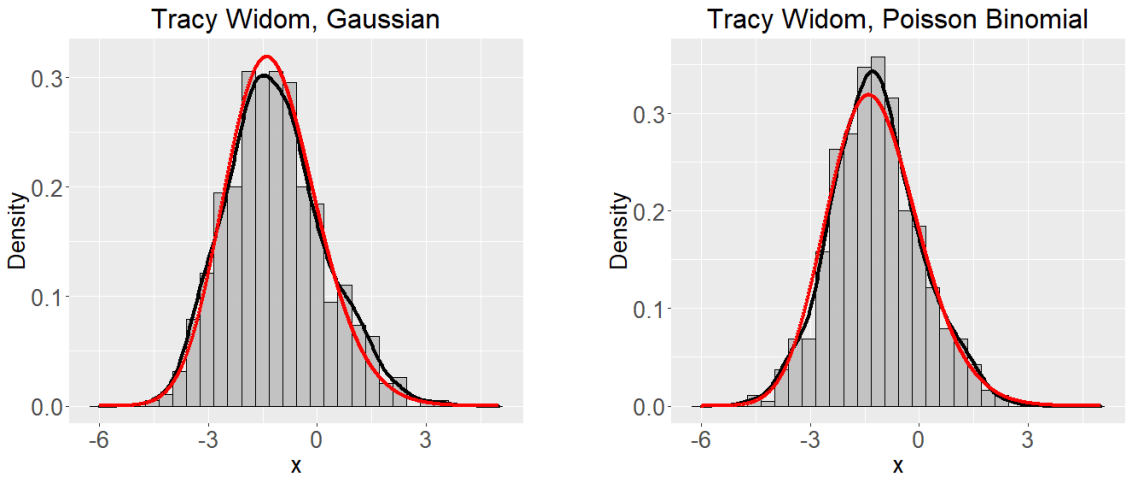
4.3 Bootstrap correction

As we saw previously, our asymptotic results can require a large sample size in practice. We derive a bootstrapping correction method, similar to the one presented in [110]. We note that our algorithm generalized the one in [110]. Algorithm 4 contains the algorithm for the undirected case, and Algorithm 9 contains the algorithm for the directed case.

Before continuing, we make a few remarks. First, the intuition behind this method is as follows: the distribution of $t' \equiv (\lambda_1(\hat{A}) - \mu_{\max})/s_{\max}$ is approximately TW_1 except that the mean and variance are incorrect, but by scaling t' by s_{TW} and then shifting t' by μ_{TW} we obtain a better approximation of a TW_1 distribution. Second, we know that both $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ have TW_1 distributions, and since we are taking a max over these two quantities, we want to use the $\alpha/2$ quantile of TW_1. This follows from a simple application of Bonferroni and leads to an α-size test. Finally, in Appendix C.1 we give a similar bootstrap correction algorithm for directed network data.

4.4 Models

In this section, we demonstrate how our method can be used to perform model selection on a broad class of network models. We consider the following problems:



(a) (b)

Figure 4.2.3: Left: Distribution of Tracy-Widom test statistics with parameter $\beta = 1$ and A to be a 600×800 standard Gaussian random matrix. Right: Distribution of Tracy-Widom test statistics with parameter $\beta = 1$ and A is a 600×800 random matrix re-centered with mean 0 and variance 1 from G where G_{ij} follows a Poisson binomial distribution with probability vector $p_i \sim_{i.i.d.} Unif(0, 0.3), i = 1, \cdots, 25$. The convergence of the distribution of Poisson binomial random matrix is almost as good as the Gaussian one, indicating Theorem 4.2.3 is compatible with non-Gaussian data, like Poisson binomial data.

Algorithm 4: Bootstrap correction of Undirected Tracy Widom statistic

Input: Observed sociomatrix: G; Estimated probability matrix $\hat{P}$; Bootstrap iterates: B; TW_1 mean: μ_{TW}; TW_1 standard deviation: s_{TW}; Significance Level: α.

Compute

$$\hat{A}_{ij} = (G_{ij} - \hat{P}_{ij})/\sqrt{(n-1)\hat{P}_{ij}(1-\hat{P}_{ij})} \ ;$$

for $b = 1$ **to** B **do**

  Sample $G_b^\star \sim F_{\hat{\theta}}$;

  Compute

$$[A_b^\star]_{ij} = ([G_b^\star]_{ij} - \hat{P}_{ij})/\sqrt{(n-1)\hat{P}_{ij}(1-\hat{P}_{ij})} \ ;$$

  Set $\lambda^\star_{b,\max} = \lambda_{\max}(A_b^\star)$ and $\lambda^\star_{b,\min} = \lambda_{\min}(A_b^\star)$;

end

Set $\mu_{\max}$ to be the sample mean of $\{\lambda^\star_{b,\max}\}_{b=1}^B$ and $s_{\max}$ to be the sample standard deviation of $\{\lambda^\star_{b,\max}\}_{b=1}^B$. Set $\mu_{\min}$ and $s_{\min}$ similarly;

Compute the test statistic t

$$t := \mu_{TW} + s_{TW} \cdot \max\left(\frac{\lambda_{\max}(\hat{A}) - \mu_{\max}}{s_{\max}}, -\frac{\lambda_{\min}(\hat{A}) - \mu_{\min}}{s_{\min}}\right) \ ;$$

if $TW_1(\alpha/2) < t < TW_1(1-\alpha/2)$ **then**

  Do not reject t and set Rej = FALSE;

else

  Reject t and set Rej = TRUE.

end

Output: Rejection of bootstrap statistic: Rej.

1. Testing the link function in the β-model (Section 4.4.1).

2. Comparing latent space models with different dimensions (Section 4.4.2).

3. Comparing exponential random graph models with different forms (Section 4.4.3).

4. Testing degree heterogeneity using Aggregated Relational Data (Section 4.4.4).

5. Testing Community Structure in Directed Networks (Section 4.4.5).

4.4.1 Testing the Link Function in β-Model

In this generative model, each node has a node effect β_i which controls the probability it connects with other nodes. Let $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{R}^n$ denote the vector of node effects. Then, conditioned on β, edges form independently in the undirected network with probability

$$\mathbb{P}(G_{ij} = 1|\beta) = \Lambda(\beta_i + \beta_j) \ , \tag{4.9}$$

for some link function $\Lambda : \mathbb{R} \to [0, 1]$. Common examples of the link function include the expit and exp. [38] provides a fixed point method to compute the MLE of this model when the link function is the expit. We use this method to compute $\hat{\beta}$ in this work. The details of the MLE method can be found under Theorem 1.4 of [38] and also in our supplementary R code. Note that many model selection tools, like BIC or AIC, would not be applicable here because the GoF question here is between two equally complex models because the only difference is in the link function. Our method therefore has the advantage of being applicable to link function tests.

In our simulations, we consider three different cases. In the first case, we are interested in testing if G is drawn from a β model with the expit link function, or a Sigmoid function, such that $\text{expit}(x) = 1/(1 + e^{-x})$. The hypothesis can be rewritten

as

$$H_0 : G \sim \beta\text{-model with } \Lambda(x) = \text{expit}(x), \quad H_a : H_0 \text{ is false} . \tag{4.10}$$

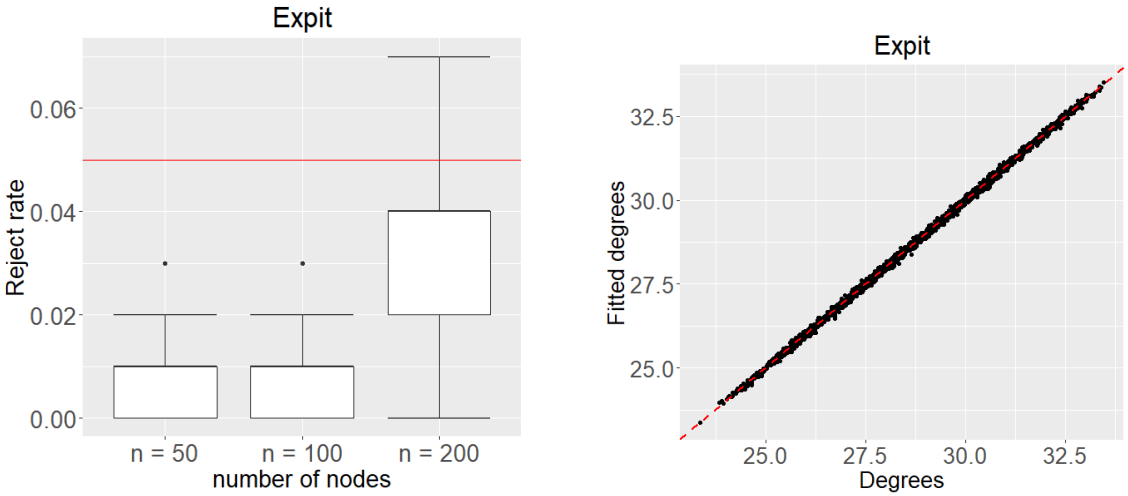
To test this hypothesis, we generate a set β of node specific effects on n nodes and form a network with probabilities from (4.9), with $\Lambda(x) = \text{expit}(x)$. We compute the MLE as described in [38] and form the $n \times n$ matrix $\hat{P}_{ij} = \text{expit}(\hat{\beta}_i + \hat{\beta}_j)$. We then compute

$$\hat{A}_{i,j} = (G_{ij} - \hat{P}_{ij})/\sqrt{(n-1)\hat{P}_{ij}(1-\hat{P}_{ij})} .$$

for $i \neq j$ and $\hat{A}_{ii} = 0$. Using the bootstrapping algorithm from Section 4.3, we record the number of times that we reject H_0. We repeat this process 100 times and plot the Type I error for these 100 simulations in Figure 4.4.1 for $n \in \{50, 100, 200\}$ with $\beta_i \overset{\text{i.i.d.}}{\sim} \text{Unif}(-2, 0)$.

In our second set of simulations, we want to determine the power of our method for the hypothesis in (4.10) when H_0 is false. In particular, we consider two reasons why H_0 is false. The first is that $\Lambda(x) = \exp(x)$, that is, the link function is incorrectly assumed. As before, we generate $\beta_i \overset{\text{i.i.d.}}{\sim} \text{Unif}(-2, 0)$ and generate the graph according to (4.9) using $\Lambda(x) = \exp(x)$. In Figure C.4.1 we plot the rejection rates. We see that while the rejection rate is higher than in Figure 4.4.1 (that is, when the null hypothesis is true), the average rejection rates for all n values is below 10%. Therefore our method suggests that while the model is incorrectly specified, it is not a bad fit to the data.

In the third set of simulations, we generate P with $P_{ij} \sim \text{Uniform}(0, 0.1)$ for $i < j$ drawn independently. We then generate an undirected network on n nodes, where nodes i and j connect with probability P_{ij}. In other words, there is no "structure" to the matrix P, as there was when the matrix P was formed according to (4.9). In Figure C.4.2, we plot the rejection rates for (4.10). We see that the power is much higher in this case than it was in the previous two simulations. In Figure C.4.3 we plot the number of triangles in the three fitted models versus the number of observed



(a) (b)

Figure 4.4.1: Left: Type I error for the null hypothesis in (4.10) for $n = 50, 100, 200$. As n increases, the Type I error increases to $\alpha = 0.05$. Right: On the x-axis we plot the average degree in networks of size $n = 200$, and on the y-axis we plot the average fitted degree across 50 simulations. We see that most points lie on the diagonal, which suggests that the expit model is a good fit. This is consistent with the left figure, which shows that our method is not rejecting (4.10) often.

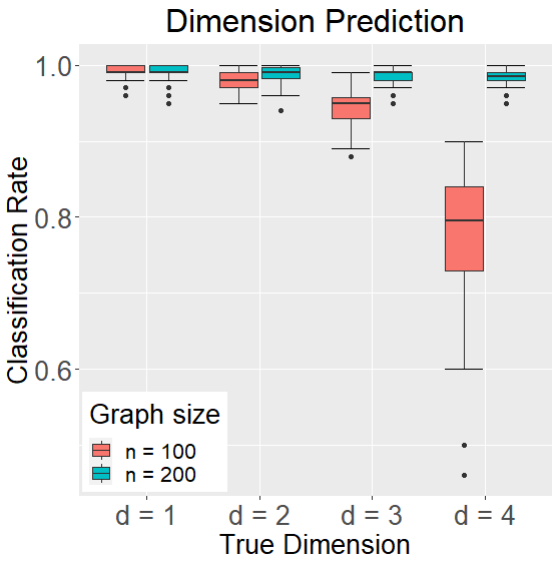


Figure 4.4.2: Correct classification rate for $n = 100, 200$ for the dimension of the latent space in Section 4.4.2. For a fixed n, increasing the dimension makes the problem harder and so the classification rate falls. However, the classification rate improves as we increase n from 100 to 200.

triangles. We see that the data drawn from the third simulation, where P_{ij} are drawn uniformly, the simulated values do not match the observed values, but in the first two simulations we see a much closer fit.

4.4.2 Testing Latent Space Models with Different Dimensions

The latent space model, originally proposed in [85], is a generative network model that asserts that each node in a network has a position on some latent space. The closer two nodes in this latent space are, the more likely they connect. There is a large literature on latent space models. See, for example, [79], [84], [9], [132], [158] and their references. In many cases, the user is interested in testing the dimension of the latent space as well as the geometry type. [116] shows how to, among other things, estimate the dimension of the latent space by using the clique structure in the

network. In this work, we take a different approach and use the entire network to estimate the fit of the model to a hypothesized latent space dimension.

Let G denote the adjacency matrix with observed covariate matrix X. One form of the latent space model [84, 121] asserts that conditioned on the network parameters, edges form independently with probability

$$\begin{aligned} P(G_{ij} = 1|\theta) = P_{ij}, \qquad \text{where} \\ \text{logodds}(P_{ij}) = \alpha_i + \alpha_j + \beta X_{ij} + \langle z_i, z_j \rangle, \end{aligned} \tag{4.11}$$

where $\text{logodds}(x) = \log(x/1-x)$, $\{\alpha_i\}_{i=1}^n$ are the parameters modeling degree heterogeneity, β is the coefficient scaling the observed covariate X, $\langle z_i, z_j \rangle$ are the inner products between latent positions with $z_i \in \mathbb{R}^d$, and d is the dimension of the latent space model. We let $\theta = (\alpha_1, \ldots, \alpha_n, z_1, \ldots, z_n, \beta)$ denote the collection of all the model's parameters.

Let G be an observed network drawn from the model in (4.11), where $z_i \in \mathbb{R}^{d_{\text{true}}}$. We develop a GoF procedure in Algorithm 5 via the Tracy-Widom statistic and a fast MLE method via non-convex projected gradient descent, described in [121]. The details can be found in Algorithm 1 and 3 in [121] and also in the supplementary R code. We now give a motivation for our algorithm. For any hypothesized dimension d, we can fit the model in (4.11) and check if we reject the hypothesis that this dimension fits the network well. With no covariates or node effects, we expect to reject the null hypothesis for $d < d_{\text{true}}$, since a lower dimensional embedding should fail to accurately model the network structure, whereas dimensions equal to and higher than d_{true} will capture the structure well and so we expect to fail to reject the corresponding hypotheses. This suggests that we should take the predicted dimension to be the smallest dimension for which we fail to reject the corresponding hypothesis. As we mentioned in the introduction, the node effects have a confounding effect on the estimation procedure, so that model fits from two distinct dimensions, and their corresponding node effect estimates, might lead to equally good model fits. Even with

the confounding issue, our simulations show that our procedure finds that the true dimension is often the smallest one that fits the model well.

In the following simulations, we will focus on the inner product model without covariate components. However, our algorithm can be generalized to any inner product models with "simple" covariates" as described in [121] by following an almost identical methodology. For $n = \{100, 200\}$, we generated 50 sets of $\{\alpha, z\}$ with $d = \{1, 2, 3, 4\}$ respectively, where $\alpha_i \sim_{iid} \text{Unif}(-2, -1) \times 10^{-2}$ and $z_i \sim_{iid} \text{N}(0, I_d)$. For each combination of $\{n, d, \alpha, z\}$, 100 networks are drawn from the corresponding generated models and predicted with Algorithm 5. The classification rates for each set of parameters are recorded and shown in Figure 4.4.2. We notice two trends. First, as n increases, the probability of correct dimension classification increases. Second, for a fixed n, larger dimensions are harder to classify correctly. This makes intuitive sense since higher dimensions often correspond to more complex latent space relationships, and so it takes more data to model these relationships well.

4.4.3 Comparing exponential random graph models with different forms

Exponential random graph models (ERGMs) are a common choice to model complex network data. To perform inference on these models, one must estimate an often intractable normalizing constant, which makes inference challenging. Some authors have presented maximum pseudo-likelihood [91] and Monte Carlo estimation methods. In this section, we show how to apply Theorem 4.2.1 to test the form of an ERGM.

We now briefly review the form of ERGMs. This model asserts that a random graph G arises through the model

$$P(G = g \mid \theta) = \frac{1}{c(\theta)} \exp\left(\sum_{i=1}^{K} \theta_i h_i(g)\right) \tag{4.12}$$

where $h_1, \ldots, h_K$ are functions of the graph g and $c(\theta)$ is the normalization constant. The user specifies the functions h as well as the value K. Some examples of h include

Algorithm 5: Dimension prediction for Latent Space model

Input: Observed sociomatrix: G.

Set $d_{\text{fit}} = 0$ and $T = 0$;

while $T = 0$ **do**

  Update $d_{\text{fit}} = d_{\text{fit}} + 1$;

  Compute the estimate $\hat{\theta}$ via Projected Gradient Descent algorithm with d_{fit}. Use $\hat{\theta}$ and the model in (4.11) to compute $\hat{P}$;

  Use $\hat{P}$ in the bootstrap algorithm (Algorithm 4) to determine if the null hypothesis $H_0 : d_{\text{true}} = d_{\text{fit}}$ is rejected;

  if H_0 *is rejected* **then**

    Set $T = 1$;

  else

    Remain $T = 0$;

  end

end

Output: Predicted latent dimension: d_{fit}.

$h(g) = \sum_{i<j} g_{ij}$, the number of edges in g, and

$$h(g) = \sum_{i,j,k}^{n} g_{ij} g_{jk} g_{kj} \ ,$$

the number of triangles in g. Except in simple cases, the MLE for θ, denoted by $\hat{\theta}$, is not available in closed form. We compute the MLE using the `ERGM` package in `R`.

Having estimated θ, we now need to estimate the $n \times n$ matrix P, where $P_{ij} = P(G_{ij} = 1|\theta)$. In most models, there is a clear correspondence between θ and P. For example, in a latent space model without covariates, once we estimate $\theta = (z_1, \ldots, z_n, \alpha_1, \cdots, \alpha_n, \beta)$, we can simply use the graph model in (4.12) to estimate P. But for ERGMs, the model in (4.12) asserts a model for the entire network G all at once, rather than specifying individual edge probabilities. To simulate P from $\hat{\theta}$, we therefore propose to simulate from the fitted model and record the number of edges between pairs of nodes across B simulations. We present this simple procedure in Algorithm 6.

Algorithm 6: Given sociomatrix G, simulate $\hat{P}$

Input: Observed sociomatrix: G; Bootstrap iterates: B.

1 Compute an estimate of θ, denoted by $\hat{\theta}$;

2 **for** $b = 1$ **to** B **do**

3 Sample $G_k^* \sim F_{\hat{\theta}}$;

4 Set A_k^* to be the $n \times n$ adjacency matrix for the graph G_k^*;

5 Record A_k^*;

6 **end**

7 For all i, j, compute

$$\hat{P}_{ij} = \frac{1}{B} \sum_{k=1}^{B} [A_k^*]_{ij} \ ;$$

Output: Estimated probability matrix: $\hat{P}$.

Having now described how to estimate P from an estimate of the ERGM parameter, we now consider an ERGM model and show how to test the significance of its parameters. Consider the model

$$P(G = g) \propto \exp\left(\theta_1 \cdot \text{edges} + \theta_2 \cdot \text{triangle} + \theta_3 \cdot \text{kstar}(2)\right) \ , \tag{4.13}$$

where edges counts the number of edges in g, triangles counts the number of triangles, and k-star(2) counts the number of 2-stars, which is a triangle with one edge missing.

Suppose that we are interested in testing whether $\theta_3 = 0$. In other words, we believe that the model above is correctly specified, with the exception that we do not know if $\theta_3 \neq 0$. Writing this as a hypothesis testing problem, we want to test the hypothesis

$$H_0 : \theta_3 = 0, \quad H_a : \theta_3 \neq 0 \ . \tag{4.14}$$

To test this, we fit our data to the model in (4.13) with $\theta_3 = 0$. That is, we estimate (θ_1, θ_2) in the model $P(G = g) \propto \exp\left(\theta_1 \cdot \text{edges} + \theta_2 \cdot \text{triangle}\right)$. Let $(\hat{\theta}_1, \hat{\theta}_2)$ denote these estimates. We then simulate $\hat{P}$ using Algorithm 6. With these estimates, we can then form the matrix $\hat{A} = (G - \hat{P})/\sqrt{(n-1)\hat{P}(1-\hat{P})}$, with $\hat{A}_{ii} = 0$. We test H_0 using Algorithm 4. In Figure 4.4.3, we plot the power function for the hypothesis. We see that near $\theta_3 = 0$, the power is roughly equal to the Type I error $\alpha = 0.05$. As $|\theta_3|$ becomes larger, the power increases. We also see that the power increases for all $\theta_3 \neq 0$ as n increases.

4.4.4 Testing Degree Heterogeneity Using ARD

In this section, we show an interesting application of our method to the case of partial network data. We focus on a particular type of partial network data known as Aggregated Relational Data (ARD). This type of data is often cheaper to collect and can still be used to perform inference. For example, [27] showed that the maximum likelihood estimate (MLE) for the latent space model, computed using only ARD instead of the entire network, is consistent as the graph size grows.

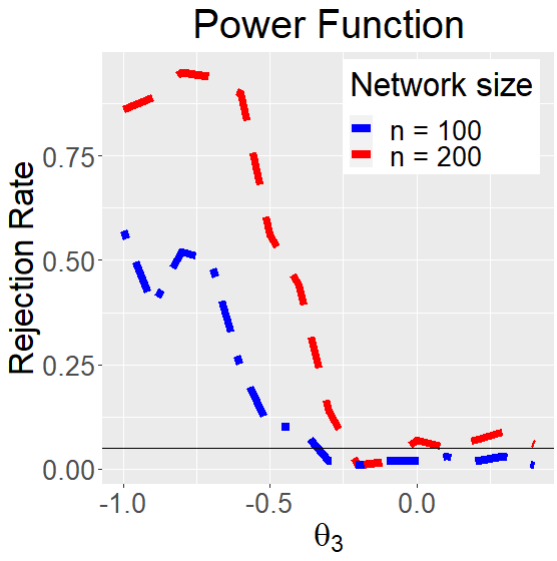


Figure 4.4.3: Power function for the hypothesis in (4.14). The null hypothesis is $\theta_3 = 0$. The black horizontal line represent the $\alpha = 0.05$ threshold.

Let G denote a network of interest on n nodes and suppose that we want to test if there is degree heterogeneity in the network. One way to model this question is through the following:

$$H_0 : g \sim \text{ER}(p^\star) \text{ for some } p^\star, \quad H_a : \ H_0 \text{ is false.} \tag{4.15}$$

where $\text{ER}(p^\star)$ denotes an Erdös-Rényi model with unknown parameter $p^\star$. Suppose that instead of observing the whole network g, we instead observe Aggregated Relational Data (ARD).

Under the null hypothesis, each $Y_{ij} \sim \text{Binomial}(n_j, p^\star)$, where $n_j = |G_j|$ is the size of group G_j. So if we define an $m \times K$ matrix A with

$$A_{ij} = \frac{Y_{ij} - n_j p^\star}{\sqrt{n_j p^\star (1 - p^\star)}} ,$$

then A is a $m \times K$ random matrix with mean zero and variance 1. Note that unlike in previous forms of A, in this case the diagonal of A is not set to be zero.

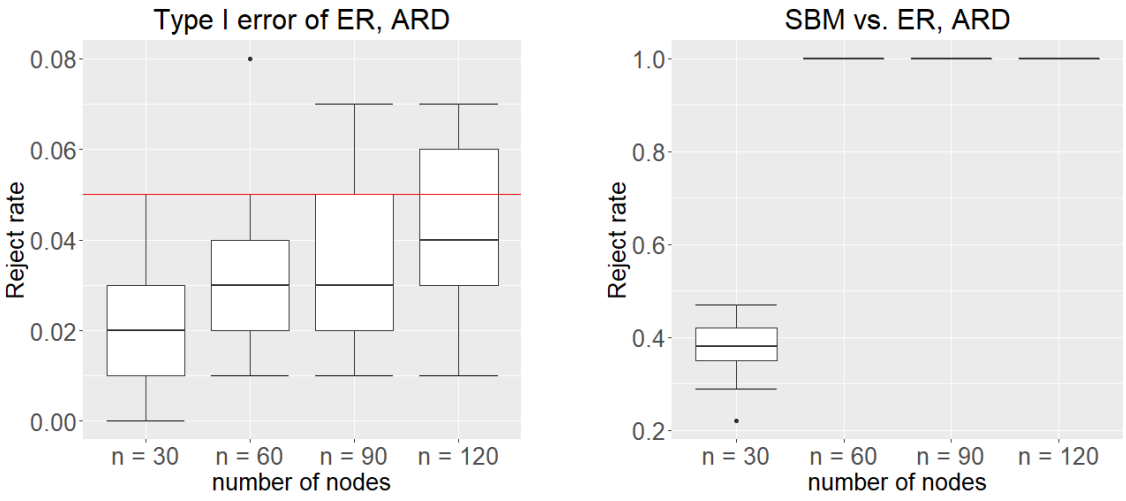
In general, we do not know $p^\star$ but given ARD, we can estimate $p^\star$ with

$$\hat{p} = \frac{1}{mK} \sum_{i=1}^{m} \sum_{j=1}^{K} \frac{Y_{ij}}{n_j} .$$

Under H_0, since $E(Y_{ij}/n_j) = p^\star$, it follows that $\hat{p} \xrightarrow{p} p^\star$ as $m \to \infty$. Here we consider K fixed; see the discussion at the end of Section 4.2.3. We can therefore define

$$\hat{A}_{ij} = \frac{Y_{ij} - n_j \hat{p}}{\sqrt{n_j \hat{p}(1-\hat{p})}} ,$$

We can use Theorem 4.2.3 to construct a test statistic for the null hypothesis. Our test statistic is the largest singular value of the matrix $\hat{A}$. Our rejection region for the null hypothesis is based on the quantiles of the Tracy-Widom distribution, as indicated in Theorem 4.2.3.



(a) (b)

Figure 4.4.4: Left: Type I error of ER model via ARD. Right: Power of fitting SBM ARD to ER model. When the hypothesis model is correct, we observed a Type I error centered around the level of testing $\alpha = 0.05$. When the ARD of a more complex model is fitted to a simple hypothesis model (i.e. ER is a special case of SBM with one community), we will observe a very high power which grows with network size n.

We first consider the Type I error of this method. For $n \in \{30, 60, 90, 120\}$, we draw an Erdös-Rényi graph with $m = \gamma_m n$ and $K = \gamma_K n$, where $\gamma_m = 1/3$ and $\gamma_K = 1/10$. We divide nodes equally into each of the K categories. Given a graph G, we define Y_{ij} as in (4.7). Our goal is to test whether G is drawn from an ER model. We plot our results in Figure 4.4.4.

Of course, more complicated testing problems can be used, but we leave that to future work. The goal of this section is to simply show how our method might be used to analyze network goodness-of-fit in cases where only partial network data is available.

4.4.5 Directed Network Case

In this section, we show how to test (4.1) when the network is directed. Recall that Theorem 4.2.3 and 4.2.4 tell us the distribution of singular values and so they provide us with test statistics.

Suppose that we are given a directed graph g. We are interested in testing whether g is drawn from a directed Erdös-Rényi model. By this, we mean a directed graph whose directed edges form independently with probability $p^\star$. Our goal is to test

$$H_0 : G \sim \text{DER}(p^\star) \text{ for some } p^\star, \quad H_a : H_0 \text{ is false.} \tag{4.16}$$

where the notation DER(p) stands for a directed ER model. Theorems 4.2.3 and 4.2.4 give us test statistics to test this hypothesis. We start with the statistic from Theorem 4.2.3. This theorem states, informally, that the singular values of X, once rescale and re-centered, converge to a Tracy Widom distribution. As in the undirected case, this convergence can be slow, so we use the bootstrap correction algorithm in Algorithm 9. Theorem 4.2.4 also provides a test statistic to test (4.16). This theorem states, informally, that n times the largest singular value of a random matrix converges to an exponential random variable.

Using these two theorems, we can test H_0 in (4.16). In Figure 4.4.5, we plot the

Type I error in the first row for the "bootstrap" method from Algorithm 9 and the "exponential" method. The second row plots the power of our method when g is drawn from a directed stochastic block model with two communities. We see that both methods have a good control on the Type I error at $\alpha = 0.05$, but only the "bootstrap" method is able to distinguish between a DER and a directed stochastic block model.

4.5 Community detection with latent space models

In this section, we analyze three data sets that are studied in [121]: the Political Blog data, the Simmons College data, and the Caltech data. [121] fits these data sets to the latent space model in (4.11) without covariate components.

The authors computed the estimates of the latent space positions $\{z\}$ with the projected gradient descent methods, then applied a simple k-means clustering on the estimated latent positions for community detection. In Table 1 of their work, [121] compares the clustering results with the community membership provided in the original data set, and reported the mis-classification rate between the estimated clustering and the true network clustering. In this analysis, the fitted latent dimensions are set to either K or $K+1$, where K is the known number of clusters in the data. We observed that fitting these datasets to different dimensions changed the mis-classification rate, which suggests that choosing an optimal latent dimension is crucial for community detection.

We made three major adjustments based on their evaluating procedure. First, instead of directly setting the latent dimension as K or $K+1$, we fit the data sets with Algorithm 5 and used the resulting d_{fit} as the fitted dimension. Second, the k-means method produces different clustering results even with the `replicate` command, so to avoid bias, we run the k-means clustering function 200 times in MATLAB and select the set of positions with the best fit. We then repeat this process 100 times and return the average mis-classification rate across the 100 simulations. Lastly, instead

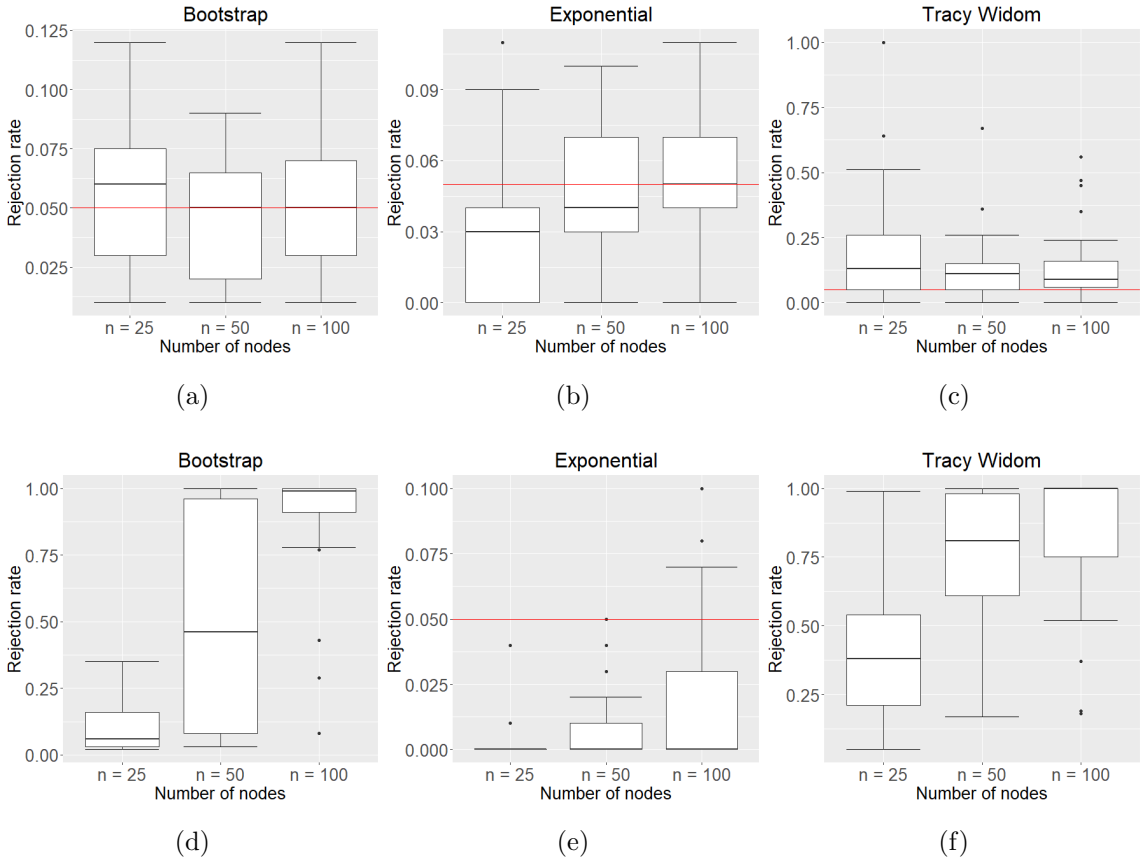
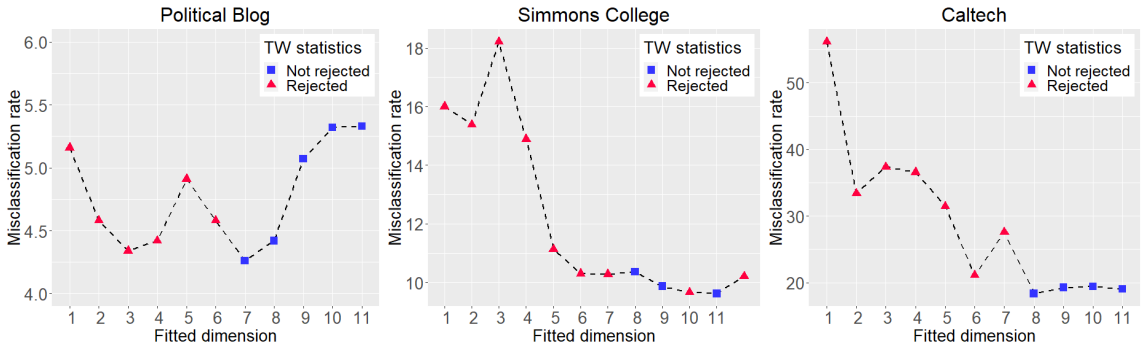


Figure 4.4.5: Type I error and rejection rates for directed network data. The first row corresponds to the case of a directed Erdös-Rényi model. In the top left figure, we plot the average rejection rate over 50 sets of simulations for $n = 25, 50, 100$ using the bootstrap test from Section C.1 . In the top middle, we plot the average rejection rate using the exponential test statistic in Theorem 4.2.4. In the top right, we plot the average rejection rate using Tracy Widom test statistic in Theorem 4.2.3. In the second row, we plot the average rejection rates using a directed stochastic block model (DSBM) with 2 communities and distinct cross community probabilities. We see that bootstrap and exponential methods have good Type I error, yet that of Tracy Widom statistics are relatively larger. In terms of power against DSBM, bootstrap and Tracy Widom obtain good power, but the Exponential does not. Overall, the bootstrap statistic has a better performance in general.

of using only the first k eigenvectors of $\hat{z}$ as in [121], we simply use the estimated positions $\hat{z}$ in the k-means algorithm. Our approach is intuitive, simple, and yields good performance on these three data sets.

We present our results in Table C.2.2 and Figure 4.5.1. For Table C.2.2, in column t_{TW}'s, text labelled with star indicates the Tracy Widom test statistics is not rejected. In column R_{mis}, bold text indicates the optimal classification rate. Figure 4.5.1 gives a visual representation of the misclassification rate over different choices of latent dimensions.

In the Political Blog data and Caltech data, the optimal dimension chosen by our method are 7 and 8 respectively. The test statistics for $d_{\mathrm{fit}} > d_{\mathrm{opt}}$ are also not rejected. This behavior is similar to the behavior we saw in the latent space simulations. The optimal mis-classification rates are also achieved at d_{opt}. Compared to the results in Table 1 of [121], for the Political Blog data, we obtained a better mis-classification rate, from 4.513% (latentnet) to 4.26% (Latent Space based Community Detection (LSCD), $d_{\mathrm{fit}} = 7$). For the Caltech data, we obtained the same optimal rate 18.35% (LSCD, $d_{\mathrm{fit}} = 8$), as our predicted dimension coincides with the number of clusters.



(a) Political Blog data (b) Simmons College data (c) Caltech data

Figure 4.5.1: Mis-classification rates of Political Blog data, Simmons College data, and Caltech data.

These results shows that the latent space model is a good fit of the two data sets, and our method performs well in achieving the optimal mis-classification rate.

In the Simmons College data, the predicted dimension is $d_{\text{opt}} = 8$, with mis-classification rate 10.37%. However, for $d_{\text{fit}} > d_{\text{opt}}$, we still observe that some fitted dimensions, namely $d = 10, 12$, are rejected. Moreover, the result for the Tracy Widom statistics is not as robust as in previous two cases: our algorithm provides different predicted dimensions in different trials, whereas the results are consistent in the previous two data sets. This potentially suggests that the latent space model might not be a good fit for the Simmons College data. Nevertheless, our method still reveals certain natures of the network. The optimal rate is achieved at $d_{\text{fit}} = 11$, which is also substantially larger than the fitted dimension $d_{\text{fit}} = 4$ in [121], at which our test statistic is not rejected. The mis-classification rate is improved from 11.17% (LSCD, $d_{\text{fit}} = K + 1$) to 9.62% (LSCD, $d_{\text{fit}} = 11$).

Our result shows that, based on the behavior of the test statistics with $d_{\text{fit}} > d_{\text{opt}}$, our Algorithm 5 potentially suggests whether the latent space model can be a good fit for the observed network. For networks that fit the latent space model well, our method will choose the optimal latent dimension that minimizes the community detection misclassification rate.

4.6 Conclusion

In this work we proposed a network goodness-of-fit test that uses the eigenvalues of the centered, scaled adjacency matrix. We used recent work in random matrix theory to derive a test statistic that can test whether an observed network is a good fit for common network models. This framework can handle undirected and directed networks, and can also handle cases where the researcher only has access to partial network data. We discussed the performance of this method on several common network models, like the latent space model, and showed that the test has favorable properties in terms of Type I error and power.

There are many avenues of future work. First, we would like to answer more general goodness-of-fit questions when the researcher only has access to ARD. We believe that the estimation methods presented in [5] can be used to estimate the $m \times K$ matrix P under a variety of realistic null hypotheses, which means that we can test the null hypothesis in (4.1) in a variety of more realistic settings. Second, we would like to extend this method to time-varying networks, such as those considered in [154]. Finally, we would like to determine whether other random matrix theory results, such as Theorem 1 in [65], will lead to a test of (4.1) with better properties, like higher power.

Research reported in this publication was supported by the National Institute Of Mental Health of the National Institutes of Health under Award Number DP2MH122405. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Chapter 5

BAYESIAN HYPERBOLIC MULTIDIMENSIONAL SCALING

In this final chapter, we now turn our attention to the problem of obtaining low-dimensional representations of objects from (dis)similarity data. In Chapter 2, we discussed the latent space model and noted that hyperbolic latent spaces produce networks with tree-like structure, a pattern found in many networks of interest [160]. In this chapter, we discuss a procedure to embed objects into a low-dimensional hyperbolic space using multi-dimensional scaling. These objects might represent nodes in a network, but they could also represent words from a piece of text or cell types from a genomic dataset.

Multidimensional scaling (MDS) methods represent high-dimensional data in a low-dimensional space, using dissimilarities between the observations as a means of identifying positions [107]. Observations with small dissimilarities will be placed close together, while those with larger dissimilarities will be placed further apart. A long literature on MDS methodology illustrates the utility of MDS as a means of summarizing complex, dependent data and for downstream applications, such as detecting clusters from the dissimilarities [23, 48, 45].

In many settings, the observed dissimilarities we wish to apply MDS methods to are likely to contain measurement error. These errors could arise from misreporting in the context of the social sciences (e.g. a retrospective behavioral inventory) or from miscalibration of machinery or operator error in industrial settings. Using a probabilistic model is one way to account for this additional uncertainty. [162, 76, 122] and others have proposed maximum likelihood MDS methods for handling measurement

error. The use of these methods relies on asymptotic theory, which might not apply for sample sizes used in applications, and the problem requires solving a non-linear optimization problem where the number of parameters grows quickly as the sample size grows [45]. One potential framework to address these potential issues with MDS is a Bayesian framework. [132] provided a Bayesian procedure to estimate the configuration of objects given (potentially noisy) dissimilarities between objects. Extensions of Bayesian MDS to the case of large datasets were discussed in [87].

Along with the statistical framework, another critical, but often ignored, choice in implementing MDS is the choice of geometry for the low-dimensional manifold. Multidimensional scaling methods often assume the observed dissimilarities are computed using Euclidean distances among objects in a Euclidean space. [132], for example, assume a Euclidean distance model with a Gaussian measurement error and propose a Markov-Chain Monte Carlo algorithm to compute a Bayesian solution for the object configuration. Yet there is a growing literature which shows that representing objects in other embedding spaces might lead to better representations and therefore be more useful in downstream tasks [50, 34, 160, 116, 99]. In particular, hyperbolic spaces, defined in Section 5.2, have been shown to produce embeddings with lower distortion, especially for data that is hierarchical or tree-like.

In this work, we combine the hyperbolic MDS methods with a Bayesian procedure. Specifically, we apply the Bayesian MDS procedure from [132] to a Hyperbolic space. We assume a Hyperbolic distance model with a Gaussian measurement error model. We then derive a Markov-chain Monte Carlo (MCMC) method we use to obtain a Bayesian solution for the object configuration in hyperbolic space.

The organization of the chapter is as follows. In Section 5.2, we formally define the Hyperbolic space model used in this work and posit a model for observed dissimilarities computed from points in a Hyperbolic space. We then discuss a prior distribution over this space and derive an MCMC algorithm to draw samples from the posterior in Section 5.3. Sections B.9 and 5.5 contains simulations and applications of our method

to real datasets in genomics. We conclude in Section 5.6.

5.1 Hyperbolic geometry

We now discuss the mathematical details of the hyperbolic geometry. The hyperbolic geometry is a non-Euclidean geometry that has a constant negative curvature, and is commonly visualized as the upper sheet of the unit two-sheet hyperboliod. There exist multiple equivalent hyperbolic models, such as the Klein model, the Poincaré disk model, and the Lorentz (Hyperboloid/Minkowski) model. We use the Lorentz model to parameterize the hyperbolic geometry, which parallels the representation used in existing hyperbolic MDS algorithms [99, 50]. Additionally, this representation facilitates convenient priors for our Bayesian model in Section 5.2.

To define the Lorentz model, we begin with the definition of the Lorentzian product. For any $\boldsymbol{x} = (x_0, \ldots, x_p)$ and $\boldsymbol{y} = (y_0, \ldots, y_p) \in \mathbb{R}^{p+1}$, the Lorentzian product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}}$ is defined as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}} \equiv -x_0 y_0 + \sum_{i=1}^{p} x_i y_i \ .$$

The p-dimensional Lorentz model with curvature $-\kappa$, which we denote by $\mathbb{H}^p(\kappa)$, can be represented as a collection of coordinates $\boldsymbol{x} \in \mathbb{R}^{p+1}$ with $x_0 > 0$ such that its Lorentzian product with itself is -1 and equipped with the hyperbolic distance proportional to the square root of κ. That is,

$$\mathbb{H}^p(\kappa) \equiv \left\{ \boldsymbol{x} \in \mathbb{R}^{p+1} : x_0 > 0, \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathcal{L}} = -1 \right\}, \quad \kappa > 0 \ ,$$

equipped with the hyperbolic distance

$$d_{\mathbb{H}^p(\kappa)}(\boldsymbol{x}, \boldsymbol{y}) \equiv \frac{1}{\sqrt{\kappa}} \operatorname{arccosh}\left(- \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}} \right) \ , \tag{5.1}$$

which is the geodesic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ on $\mathbb{H}^p(\kappa)$. Specifically, the curvature $-\kappa$ controls the hyperbolicity of the geometry, so that the space becomes more hyperbolic as $-\kappa$ becomes more negative and becomes flatter as κ shrinks to zero (Euclidean geometry has curvature exactly 0).

5.2 Bayesian modeling framework

We now describe our statistical framework for Bayesian Hyperbolic Multidimensional Scaling (BHMDS), which represents the objects of interest by coordinates in a hyperbolic geometry, so that the hyperbolic distances between objects resemble their true dissimilarity measures. We suppose the objects dwell on $\mathbb{H}^p(\kappa)$ with coordinates $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$, and denote by δ_{ij} the dissimilarity measure between object i and object j as well as the hyperbolic distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$\delta_{ij} \equiv d_{\mathbb{H}^p(\kappa)}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{\sqrt{\kappa}} \arccos\left(-\langle \boldsymbol{x_i}, \boldsymbol{x}_j \rangle_{\mathcal{L}}\right), \quad i, j = 1, \cdots, n\,. \tag{5.2}$$

In many settings, the observed dissimilarities contain measurement errors. As in [132], we represent the observed dissimilarity d_{ij} as the true dissimilarity plus a Gaussian error, with the constraint that the observed dissimilarity is always positive. We therefore assume that d_{ij} follows a truncated normal distribution:

$$d_{ij} \sim N\left(\delta_{ij}, \sigma^2\right) I\left(d_{ij} > 0\right), \quad i < j, \quad i, j = 1, \ldots, n\,, \tag{5.3}$$

where δ_{ij} is as defined in (5.2), $\boldsymbol{x}_i$ are unobserved, and σ^2 is the variance of the measurement error.

Given the statistical model, we now specify priors for $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ and σ^2. Using (5.3), the likelihood of the unknown parameters $\boldsymbol{X}$ and σ^2 is

$$l\left(\mathbf{X}, \sigma^2\right) \propto \left(\sigma^2\right)^{-m/2} \exp\left[-\frac{1}{2\sigma^2} SSR - \sum_{i<j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right],$$

where $m = n(n-1)/2$ is the number of dissimilarities, $SSR = \sum_{i<j} \left(\delta_{ij} - d_{ij}\right)^2$ is the sum of squared residuals, and Φ is the cumulative distribution function of the standard normal random variable. The square-root of the SSR term in the likelihood is often referred to as *stress* in the MDS literature, meaning that our approach falls under the umbrella of stress-minimizing approaches to MDS.

Moving now to our prior distribution choices, we use the the wrapped normal distribution on $\mathbb{H}^p(\kappa)$ and centered around the hyperbolic origin [127] as the prior

over the hyperbolic coordinates, $\boldsymbol{X}$. This distribution is a Gaussian-like distribution on hyperbolic geometry that is projected from $\mathbb{R}^p$ to $\mathbb{H}^p$. We leverage this property to define auxiliary variables, $\boldsymbol{v}_i$, which represent coordinates in $\mathbb{R}^p$ and define multivariate Gaussian priors over the auxiliary variables. Specifically, for a prior over the auxiliary variables $\boldsymbol{v}_i$ we use a p-dimensional normal distribution with mean $\mathbf{0}_p$ and a diagonal covariance matrix Λ ($\boldsymbol{v}_i \sim N(0, \Lambda)$, independently for $i = 1, \cdots, n$). At each step in the sampler, we transform from the Euclidean space of auxiliary variables to the Hyperbolic coordinates. We discuss the details of this transformation in the next section.

We next define the prior for the observation error variance, σ^2. We use a conjugate prior $\sigma^2 \sim IG(a, b)$, the Inverse Gamma distribution with mode $b/(a+1)$. For the hyperprior on the diagonal entries of the auxiliary variable variance matrix $\Lambda = \text{Diag}(\lambda_1, \ldots, \lambda_p)$, we also assume a conjugate Inverse Gamma prior, $\lambda_j \sim IG(\alpha, \beta_j)$, independently for $j = 1, \cdots, p$. We will further suppose prior independence among $\boldsymbol{V}$, Λ, and σ^2, i.e., $\pi(\mathbf{V}, \sigma^2, \Lambda) = \pi(\mathbf{V})\pi(\sigma^2)\pi(\Lambda)$, where $\pi(\mathbf{V}), \pi(\sigma^2)$, and $\pi(\Lambda)$.

Often there is little prior information about $\boldsymbol{V}$, Λ, and σ^2. [132] proposed to use preliminary results from a frequentist MDS method for parameter selection in the priors, and we use a similar methodology in this study. Specifically, we use the embedding result $\boldsymbol{X}^{(0)} = \{\boldsymbol{x}_1^{(0)}, \cdots, \boldsymbol{x}_n^{(0)}\}$ from `hydraPlus`, the stress-minimizing hyperbolic MDS algorithm proposed in [99], to choose the parameters of the prior distributions. For the hyperparameters of σ^2, one can set a small a, e.g. $a = 5$, for a vague prior of σ^2, and choose $b = (a-1)SSR^{(0)}/m$, where $m = n(n-1)/2$ and

$$SSR^{(0)} \equiv \sum_{i<j} \left(d_{\mathbb{H}^p(\kappa)}(\boldsymbol{x}_i^{(0)}, \boldsymbol{x}_j^{(0)}) - d_{ij} \right)^2 = \sum_{i<j} \left(\delta_{ij}^{(0)} - d_{ij} \right)^2 \tag{5.4}$$

is the sum of squared residuals of $\boldsymbol{X}^{(0)}$, so that the prior mean matches $SSR^{(0)}/m$. Similarly, for the hyperprior of λ_j, one may choose a small α, e.g. $\alpha = 0.5$, and choose β_j such that the prior mean of λ_j matches the jth diagonal element of the sample covariance matrix $S_v = \sum_{i=1}^n \boldsymbol{v}_i^{(0)\top} \boldsymbol{v}_i^{(0)}/n$, where $\boldsymbol{v}_i^{(0)} = T^{-1}(\boldsymbol{x}_i^{(0)})$ and T^{-1} the inverse

transformation we describe in Section 5.3.1.

After specifying the prior distributions for $\boldsymbol{V}, \sigma^2$, and Λ, the posterior density function of the unknown parameters $\boldsymbol{V}, \sigma^2$, and Λ becomes

$$\pi\left(\boldsymbol{V}, \sigma^2, \Lambda \mid D\right) \propto \left(\sigma^2\right)^{-(m/2+a+1)} \prod_{j=1}^{p} \lambda_j^{-n/2} \times \exp\left[-\frac{1}{2\sigma^2} SSR - \sum_{i<j} \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right) - \frac{1}{2} \sum_{i=1}^{n} \boldsymbol{v}_i^{\top} \Lambda^{-1} \boldsymbol{v}_i - \frac{b}{\sigma^2} - \sum_{j=1}^{p} \frac{\beta_j}{\lambda_j}\right], \tag{5.5}$$

where $D = \{d_{ij}\}_{i,j=1,\cdots,n}$ is the matrix of observed dissimilarities. Due to the complex form of the posterior density function, we use a Markov-chain Monte Carlo sampler to draw from the posterior distribution.

5.3 Posterior computation

After specifying the Bayesian model and prior choices, we use a Markov-chain Monte Carlo algorithm to sample from the posterior distribution. We first discuss the implementation details of the MCMC algorithm in Section 5.3.1. Then, in Section 5.3.2, we present a likelihood approximation based on work by [143] for social networks to accelerate the MCMC with large scale dissimilarity data. Specifically, we leverage the realization that the posterior likelihood can be well approximated using random samples of the objects drawn from a case-control scheme, which reduces the MCMC time complexity from $O(n^2)$ to $O(n)$.

5.3.1 Markov chain monte carlo

Our MCMC sampler requires the hyperbolic dimension p and curvature κ as inputs. For the hyperbolic curvature κ, we can estimate $\widehat{\kappa}$ using a stress minimizing algorithm which we will describe in detail in Appendix D.2. For the dimension, p, we could use a similar stress minimization approach across potential values of p, keeping in mind that the goal is dimension reduction so our prior is that p is much smaller than n.

We could also use a Bayesian model selection approach similar to the one described by **(author?)** [132]. Given p and κ, we initialize the starting values for the MCMC sampler using the output from the `hydraPlus` algorithm. We take $\boldsymbol{X}^{(0)}$, $\boldsymbol{V}^{(0)}$, and $\{\delta_{ij}^{(0)}\}_{i,j=1,\cdots,n}$ as the initial values for $\boldsymbol{X}$, $\boldsymbol{V}$, and $\{\delta_{ij}\}_{i,j=1,\cdots,n}$. Moreover, from $\boldsymbol{X}^{(0)}$, one can compute the initial sum of squared residuals $SSR^{(0)}$ and the sample variance $\sigma^{2(0)} = SSR^{(0)}/m$, which can be used as the initial value of σ^2. In addition, the diagonal elements of the sample covariance matrix of $\boldsymbol{V}^{(0)}$ can be used as initial values for the λ_j's.

At each iteration, we will first propose a new value of λ_j from its conditional posterior distribution given the other unknowns. From (5.5), the full conditional posterior distribution of λ_j is the Inverse Gamma distribution $IG\left(\alpha + n/2, \beta_j + s_j/2\right)$, where s_j/n is the sample variance of the jth coordinates of the $\boldsymbol{v}_i$'s. By Algorithm 1 described in [127], the transformation T from $\boldsymbol{v}_i$ to $\boldsymbol{x}_i$ is

$$\boldsymbol{x}_i = T(\boldsymbol{v}_i) = \cosh\left(\|\tilde{\boldsymbol{v}}_i\|_{\mathcal{L}}\right)\boldsymbol{\mu}_0^p + \sinh\left(\|\tilde{\boldsymbol{v}}_i\|_{\mathcal{L}}\right)\frac{\tilde{\boldsymbol{v}}_i}{\|\tilde{\boldsymbol{v}}_i\|_{\mathcal{L}}}\,, \tag{5.6}$$

where $\tilde{\boldsymbol{v}}_i = (0, \boldsymbol{v}_i) \in \mathbb{R}^{p+1}$ and $\|\tilde{\boldsymbol{v}}_i\|_{\mathcal{L}} \equiv \sqrt{\langle\tilde{\boldsymbol{v}}_i, \tilde{\boldsymbol{v}}_i\rangle_{\mathcal{L}}}$. Correspondingly, the inverse transformation T^{-1} from $\boldsymbol{x}_i$ to $\boldsymbol{v}_i$ is

$$\tilde{\boldsymbol{v}}_i = (0, \boldsymbol{v}_i) = T^{-1}(\boldsymbol{x}_i) = \frac{\operatorname{arccosh}(\alpha)}{\sqrt{\alpha^2 - 1}}(\boldsymbol{x}_i - \alpha\boldsymbol{\mu}_0^p)\,, \tag{5.7}$$

where $\alpha = -\langle\boldsymbol{\mu}_0^p, \boldsymbol{x}_i\rangle_{\mathcal{L}}$. By transformation T, applying a Gaussian prior on $\boldsymbol{v}_i \in \mathbb{R}^p$ is then equivalent to using a hyperbolic wrapped normal prior on $\boldsymbol{x}_i \in \mathbb{H}^p(\kappa)$ with mean $\boldsymbol{\mu}_0^p = (1, 0, \cdots, 0) \subset \mathbb{R}^{p+1}$ and covariance matrix Λ.

With this transformation in mind, we then use a random walk Metropolis-Hasting algorithm to sample the $\boldsymbol{v}_i$'s and σ^2. Since we specify a Gaussian prior on $\boldsymbol{v}_i$, we correspondingly use a normal proposal density. To choose the variance of the normal proposal density, we first write down the full conditional posterior density of $\boldsymbol{v}_i$, that is,

$$\pi\left(\boldsymbol{v}_i \mid \cdots\right) \propto \exp\left[-\frac{1}{2}\left(Q_1 + Q_2\right) - \sum_{j\neq i, j=1}^{n} \log\Phi\left(\frac{\delta_{ij}}{\sigma}\right)\right]\,, \tag{5.8}$$

where $Q_1 = \sum_{j \neq i, j=1}^{n} \left(\delta_{ij} - d_{ij}\right)^2 / \sigma^2$ and $Q_2 = \boldsymbol{v}_i^\top \Lambda^{-1} \boldsymbol{v}_i$. From numerical experiments, we found that $\delta_{ij} = d_{\mathbb{H}^p(\kappa)}\left(T(\boldsymbol{v}_i), T(\boldsymbol{v}_j)\right)$ is approximately of the order of $\sqrt{\kappa}||\boldsymbol{v}_i||$. Consequently, $(\delta_{ij} - d_{ij})^2$ can be approximated by a quadratic form of $\|\boldsymbol{v}_i\|$, so that Q_1 has $n-1$ quadratic forms of $\|\boldsymbol{v}_i\|$ with coefficient $1/\sigma^2$, whereas $Q_2 = \boldsymbol{v}_i^T \Lambda^{-1} \boldsymbol{v}_i$ has only one quadratic form of $\|\boldsymbol{v}_i\|$ with coefficient Λ. Thus, we conclude that Q_1 dominates the full conditional posterior distribution, and we may approximate the full conditional variance of $\boldsymbol{v}_i$ by a constant multiple of $\sigma^2/(n-1)$, which we use as the variance of the normal proposal density in the MCMC sampler.

Finally, from a preliminary numerical study similar to that carried out by [132], we found that the full conditional density function of σ^2 can be well approximated by the density function of $IG(m/2 + a, SSR/2 + b)$. Moreover, when the number of dissimilarities $m = n(n-1)/2$ is large, the Inverse Gamma density function can be well approximated by a normal density. Thus, we use a random walk Metropolis-Hasting algorithm with a normal proposal density and a variance proportional to the variance of $IG(m/2 + a, SSR/2 + b)$ to sample σ^2.

We now summarize our MCMC algorithm. At iteration t:

1. For each $i = 1, \cdots, p$, sample $\lambda_i^{(t)}$ as

$$\lambda_i^{(t)} \sim IG\left(\alpha + n/2, \beta_i + s_i^{(t-1)}/2\right) ,$$

   where $s_i^{(t-1)}$ is the sample variance of the ith coordinates of $\boldsymbol{v}_i^{(t-1)}$'s.

2. For each $i = 1, \cdots, n$, do the following:

   (a) Make a new proposal for $\boldsymbol{v}_i^{(t)}$ such that

$$\boldsymbol{v}_{i,\text{new}}^{(t)} \sim MVN_p\left(\boldsymbol{v}_i^{(t)}, \frac{c{\sigma^2}^{(t-1)}}{n-1} \cdot I_p\right) ,$$

   where I_p is the $p \times p$ identity matrix. In practice, we can simply set the constant multiple $c = 1$.

(b) Set $\boldsymbol{v}_i^{(t)} = \boldsymbol{v}_{i,\text{new}}^{(t)}$ with probability

$$p = \min\left(1, \frac{\pi_{\boldsymbol{v}}(\boldsymbol{v}_{i,\text{new}}^{(t)})}{\pi_{\boldsymbol{v}}(\boldsymbol{v}_i^{(t)})}\right) ,$$

where $\pi_{\boldsymbol{v}}(\cdot)$ is the full conditional posterior density of $\boldsymbol{v}$ in (5.8).

3. Make a new proposal for $\sigma^{2(t)}$ such that

$$\sigma^{2(t)}_{\text{new}} \sim N\left(\sigma^{2(t)}, \frac{c\gamma^{(t)}}{(\omega-1)^2(\omega-2)}\right) ,$$

where $\gamma^{(t)} = (SSR^{(t)}/2+b)^2$ and $\omega = m/2+a$. Set $\sigma^{2(t)} = \sigma^{2(t)}_{\text{new}}$ with probability

$$p = \min\left(1, \frac{\pi_{\sigma^2}(\sigma^{2(t)}_{\text{new}})}{\pi_{\sigma^2}(\sigma^{2(t)})}\right) ,$$

where $\pi_{\sigma^2}(\cdot)$ is the density function of $IG(m/2+a, SSR^{(t)}/2+b)$.

Using the above algorithm, we obtain samples from the full posterior density. The latent embeddings, $\boldsymbol{X}$, however are only indirectly involved in the posterior distribution through the dissimilarity measures, δ_{ij}. The posterior samples of $\boldsymbol{X}$, therefore, are invariant to isometric actions on $\mathbb{H}^p(\kappa)$. Thus in general, $\boldsymbol{X}$ is not identifiable, and we can only recover the relative embedding of the objects instead of their absolute hyperbolic coordinates. [132] suggested post-processing the MCMC samples of $\boldsymbol{X}$ at each iteration of the MCMC via the Procrustes operation, so that the transformed $\boldsymbol{X}'$ has sample mean 0 and a diagonal covariance matrix as specified in the prior. We found in simulations, however, that the MCMC algorithm mixed well without post-processing and returned accurate estimates of the model parameters. We therefore skipped this post-processing step, which has the added benefit of speeding up our algorithm, since the Procrustes transformation involves an eigen-decomposition of a large matrix.

Our solution is to estimate $\boldsymbol{X}$ through the Bayesian estimates of the dissimilarities $\{\delta_{ij}\}$. To estimate $\{\delta_{ij}\}$, we observe that the likelihood in (5.2) dominates the

posterior density in (5.5), and the term involving the SSR in (5.2) dominates the likelihood, so that the posterior mode of $\{\delta_{ij}\}$ can be well approximated by the posterior sample of $\{\delta_{ij}\}$ that minimizes the SSR. Moreover, the $\{\delta_{ij}\}$ that minimizes the SSR also minimize the MDS goodness-of-fit measure stress, as the stress is just the square root of SSR after normalization. We then define the Bayesian estimate of $\{\delta_{ij}\}$ as the posterior stress-minimizing estimate

$$\{\widehat{\delta}_{ij}\} \equiv \arg \min_{\{\delta_{ij}\}^{(t)}} SSR^{(t)} = \arg \min_{\{\delta_{ij}\}^{(t)}} \text{stress}^{(t)} \,, \tag{5.9}$$

where the superscript (t) indicates that the posterior sample is drawn from the tth MCMC iteration. Finally, since each $\{\delta_{ij}\}^{(t)}$ corresponds to an unique $\boldsymbol{X}^{(t)}$ from the posterior, we simply take the posterior sample of $\boldsymbol{X}^{(t)}$ corresponds to $\{\widehat{\delta}_{ij}\}$ as our Bayesian estimate of $\widehat{\boldsymbol{X}}$.

5.3.2 Case-control likelihood approximation

For dissimilarity data with a sample size of around $n < 200$, $20,000$ iterations of the MCMC algorithm take about 300 seconds using a standard personal computer. However, since the proposal of $\boldsymbol{v}_i$ involves calculations across n terms, and for each iteration we need to update $\boldsymbol{v}_i$ n times in total, the time complexity of each iteration is approximately $O(n^2)$. When the sample size n increases, the algorithm quickly become computationally intractable.

We propose a stratified case-control log-likelihood to approximate the original posterior log-likelihood to facilitate computation using larger datasets. Our case-control approach is based on work by [143] for social networks, where they studied a posterior log-likelihood approximation of the latent space model described in [86]. The core intuition is that, for each object i fixed, if we stratify all other objects regarding their dissimilarities to i, then there are many fewer objects that are similar to i than there are objects that are very dissimilar. This imbalance creates an opportunity to borrow ideas from the widely-used case-control technique from epidemiology. The

samples in a case-control study can also be stratified into two distinct groups, where the "case" group has the outcome of interest, but is often rare and hard to collect compared to the "control" group.

This suggests that the statistical approximation technique in case-control studies can be used to approximate the posterior distribution of $\boldsymbol{v}_i$. If we view the similar objects to object i as samples in the case group, and the rest as being in the control group, we can approximate the posterior distribution using all the samples in the case group and a random sample from the control group. Moreover, to increase precision, we further stratify the samples in the control group by their dissimilarities to object i, and randomly sample from each stratum by their weight determined by their contributions to the proposal likelihood change to enhance the accuracy of the approximation. Under the proposed case-control stratification scheme, we accelerate the MCMC time complexity from $O(n^2)$ to $O(n)$.

We will now give details of the stratified case-control log-likelihood. The full conditional log-posterior density of $\boldsymbol{v}_i$ is

$$l_i \equiv \log \pi \left(\boldsymbol{v}_i \mid \cdots \right) \propto - \sum_{j \neq i, j=1}^{n} \left[\frac{(\delta_{ij} - d_{ij})^2}{2\sigma^2} + \log \Phi \left(\frac{\delta_{ij}}{\sigma} \right) \right] - \frac{1}{2} \boldsymbol{v}_i^\top \Lambda^{-1} \boldsymbol{v}_i \ , \quad (5.10)$$

The first step is to divide object $j = 1, 2, \cdots, n, j \neq i$ into M different strata $S_1^{(i)}, S_2^{(i)}, \cdots, S_M^{(i)}$ according to their observed dissimilarities with respect to object i. The partition of the strata can be highly customized, as long as the total number of strata $M \ll n$. We will later describe several partition strategies in detail later in Section 5.5.2 and 5.5.3. Given the strata, we can write the log-likelihood as

$$l_i = - \sum_{k=1}^{M} \sum_{j \in S_k^{(i)}} \left[\frac{(\delta_{ij} - d_{ij})^2}{2\sigma^2} + \log \Phi \left(\frac{\delta_{ij}}{\sigma} \right) \right] - \frac{1}{2} \boldsymbol{v}_i^\top \Lambda^{-1} \boldsymbol{v}_i = \sum_{k=1}^{M} l_{i,k} - \frac{1}{2} \boldsymbol{v}_i^\top \Lambda^{-1} \boldsymbol{v}_i \ . \quad (5.11)$$

where $l_{i,k} = - \sum_{j \in S_k^{(i)}} \left[(\delta_{ij} - d_{ij})^2 / 2\sigma^2 + \log \Phi \left(\delta_{ij} / \sigma \right) \right]$ is the likelihood contribution from stratum $S_k^{(i)}$.

If a stratum $S_k^{(i)}$ belongs in the case group, we will compute its log-likelihood

explicitly. Otherwise, if $S_k^{(i)}$ belongs in the control group, we will randomly sample $n_{i,k}$ objects from the strata $S_k^{(i)}$ and estimate the strata's log-likelihood contribution by

$$\hat{l}_{i,k} = -\frac{N_{i,k}}{n_{i,k}} \sum_{j \in n_{i,k} \text{ samples}} \left[\frac{(\delta_{ij} - d_{ij})^2}{2\sigma^2} + \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right) \right] , \tag{5.12}$$

where $N_{i,k}$ is the number of elements in strata $S_k^{(i)}$. Since the estimator $\widehat{l}_{i,k}$ is based on a random sample from the strata, we always have $\mathbb{E}(\widehat{l}_{i,k}) = l_{i,k}$, so that the log-likelihood estimator is unbiased. For the sake of analysis, we now assume the first C strata are considered as cases, which we denote as $S_1^{(i)}, S_2^{(i)}, \cdots, S_C^{(i)}$, and the rest are controls. Then, the stratified case-control approximate log-likelihood for object i becomes

$$\hat{l}_i = \sum_{k=1}^{C} l_{i,k} + \sum_{k=C+1}^{M} \hat{l}_{i,k} - \frac{1}{2} \boldsymbol{v}_i^\top \Lambda^{-1} \boldsymbol{v}_i \, . \tag{5.13}$$

where $l_{i,k}$ the $S_k^{(i)}$'s likelihood contribution in (5.11) and $\hat{l}_{i,k}$ are the log-likelihood estimators in (5.12).

We now describe how to determine the subsample size $n_{i,k}$. To accelerate the MCMC iteration to approximately $O(n)$, we want object i's overall random sample size $n_i \equiv \sum_{k=1}^{M} n_{i,k} \ll n$. To do this, we pick a moderate control-to-case rate r, and let n_i be r times of the average number of objects in the case group. That is, we set $n_i \equiv r \cdot \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{C} |S_k^{(i)}|$, where $|S_k^{(i)}|$ denotes the number of objects in $S_k^{(i)}$. Given the fixed n_i, we assign $n_{i,k}$ proportional to the stratum $S_k^{(i)}$'s contribution to the log-likelihood change in sampling $\boldsymbol{v}_i$. We conduct the following pilot MCMC to determine the stratum $S_k^{(i)}$'s likelihood contributions. For object i fixed, we first draw a simple random sample over its control group with size n_i, and use them to construct an approximate log-likelihood

$$\tilde{l}_i \equiv \sum_{k=1}^{C} l_{i,k} + \frac{n - \sum_{k=1}^{C} |S_k^{(i)}|}{n_i} \sum_{j \in n_i \text{ samples}} \left[\frac{(\delta_{ij} - d_{ij})^2}{2\sigma^2} + \log \Phi\left(\frac{\delta_{ij}}{\sigma}\right) \right] - \frac{1}{2} \boldsymbol{v}_i^\top \Lambda^{-1} \boldsymbol{v}_i \, , \tag{5.14}$$

which we will use in the pilot MCMC. Once again, since we randomly sample from the population, $\tilde{l}$ is unbiased. Then, at each iteration t, we calculate the log-likelihood change for $\boldsymbol{v}_i$ as

$$\begin{aligned}\Delta\tilde{l}_i^{(t)} &= \tilde{l}_i(\boldsymbol{v}_{i,\text{new}}^{(t)}) - \tilde{l}_i(\boldsymbol{v}_i^{(t)}) \\ &= \sum_{k=1}^{C}\left[l_{i,k}(\boldsymbol{v}_{i,\text{new}}^{(t)}) - l_{i,k}(\boldsymbol{v}_i^{(t)})\right] + \sum_{k=C+1}^{M}\left[\tilde{l}_{i,k}(\boldsymbol{v}_{i,\text{new}}^{(t)}) - \tilde{l}_{i,k}(\boldsymbol{v}_i^{(t)})\right] + \Delta_i \\ &= \sum_{k=1}^{C}\Delta l_{i,k} + \sum_{k=C+1}^{M}\Delta\tilde{l}_{i,k} + \Delta_i \ ,\end{aligned}$$

where $\Delta_i = -(\boldsymbol{v}_{i,\text{new}}^\top\Lambda^{-1}\boldsymbol{v}_{i,\text{new}} - \boldsymbol{v}_i^\top\Lambda^{-1}\boldsymbol{v}_i)/2$. We then define

$$w_{i,k}^{(t)} = \left|\frac{\Delta\tilde{l}_{i,k}}{\sum_{g=C+1}^{M}\Delta\tilde{l}_{i,g}}\right| \quad \text{for } k = C+1, C+2, \cdots, M \ ,$$

and calculate the relative weights as

$$w_{i,k} = \frac{1}{T-1}\sum_{t=1}^{T-1} w_{i,k}^{(t)} \ ,$$

where T is the number of iterations in the pilot MCMC run after burn-in and thinning. Finally, we take the subsample size $n_{i,k}$ as

$$n_{i,k} = n_i \cdot \frac{w_{i,k}}{\sum_{g=c+1}^{M} w_{i,g}} \ ,$$

for $k = c+1, \cdots, M$. To summarize, the algorithm is as follows.

1. For each object $i = 1, \cdots, n$, partition all other objects into M strata via a user-defined, dissimilarity-based strategy.

2. Set the strata defined with small dissimilarities $S_1^{(i)}, S_2^{(i)}, \cdots, S_C^{(i)}$ as cases, and the rest as controls.

3. For each object i, randomly sample n_i objects from the control group.

4. Run a pilot MCMC with the approximate log-likelihood described in (5.14).

5. Record the relative weights $w_{i,k}$ and compute the subsample sizes $n_{i,k}$.

6. For each object $i = 1, 2, \cdots, n$, sample $n_{i,k}$ objects from strata $S_k^{(i)}$ for $k = C+1, C+2, \cdots, M$.

7. Run a full MCMC with the original log-likelihood functions $l_1, l_2, \cdots, l_n$ replaced by the stratified case control log-likelihood functions in (5.13).

In the following sections, we evaluate both of full and approximate strategies for sampling from the posterior using both simulated and observed data.

5.4 Simulation experiments

We conducted simulation experiments to evaluate aspects of the proposed statistical model and algorithm. First, we evaluate BHMDS's element-wise estimation performance for the true dissimilarities, δ_{ij}, and the measurement error variance, σ^2. Then, we evaluate the overall estimation performance using the coverage rate of the posterior credible interval (CI) over all the $\{\delta_{ij}\}$. Lastly, we evaluate the robustness of the algorithm by examining its calibration under a variety of data generating distributions.

To evaluate BHMDS's estimation performance, we wish to test it under different dataset sizes n, hyperbolic dimensions p, and noise levels σ. Throughout our experiments, we fix the hyperbolic curvature $\kappa = 1$, and generate the simulation data as follows:

1. For each combination of $(n, p) \in \{50, 100\} \times \{2, 5\}$, sample $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ from the hyperbolic wrapped normal distribution such that $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \sim_{i.i.d.} \mathcal{G}(\mathbf{0}_p, 3I_p)$, where $\mathcal{G}(\cdot, \cdot)$ is the distribution described in [127] and I_p is the p-dimensional identity matrix. Specifically, the wrapped normal distribution

$\mathcal{G}(\cdot,\cdot)$ samples from a normal distribution on the tangent space at the hyperbolic origin in $\mathbb{R}^p$, projects the tangent space onto the hyperbolic space by the transportation described in (5.6), results in a Gaussian-like distribution on $\mathbb{H}^p$, and generates tree-like, hierarchical data.

2. Compute the true dissimilarities $\{\delta_{ij}\}_{i,j=1,\cdots,n} = d_{\mathbb{H}^p(1)}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Then, for each $\sigma \in \{1, 1.5, 2\}$, generate the corresponding observed dissimilarity matrices $\{d_{ij}\}_{i,j=1,\cdots,n}$, with entries d_{ij} drawn from (5.3).

3. Apply the full BHMDS MCMC to the $\{d_{ij}\}$'s and record the approximate posterior mode estimates $\{\widehat{\delta}_{ij}\}$ described in (5.9), posterior mean $\widehat{\sigma}$, and the matrix-wise converge rate

$$C \equiv \frac{\sum_{i<j} I\left(\delta_{ij} \in \left[q_{ij}^{(\alpha/2)}, q_{ij}^{(1-\alpha/2)}\right]\right)}{m}, \tag{5.15}$$

where $\left(q_{ij}^{(\alpha/2)}, q_{ij}^{(1-\alpha/2)}\right)$ are the $(\alpha/2)$ and $(1-\alpha/2)$ quantiles of the posterior samples of δ_{ij} and $m = n(n-1)/2$.

We plot the simulation results on $\{\widehat{\delta}_{ij}\}$, $\widehat{\sigma}$, and coverage rate in Figure 5.4, 5.4.2, and 5.4.3 respectively. For any combination of (n, p, σ), all the boxplots are concentrated tightly around the red horizontal lines representing the true values in all three plots, indicating that BHMDS obtains precise and robust estimates of δ_{ij}, σ^2, and close to the nominal coverage rate.

We are interested in further testing the robustness of the BHMDS algorithm using dissimilarity data generated via different distributions defined with a variety of dimensions, curvatures, and distribution parameters. For this we use marginal calibration, a criterion which comprehensively assesses the predictive performance of the forecasting distribution. Suppose at times or instances $s = 1, 2, \cdots, S$, nature picks distributions $G_1, G_2, \cdots, G_S$, and we predict them with forecasting distributions $F_1, F_2, \cdots, F_S$. We further let $x_1, x_2, \cdots, x_S$ be observations randomly sampled from $G_1, G_2, \cdots, G_S$

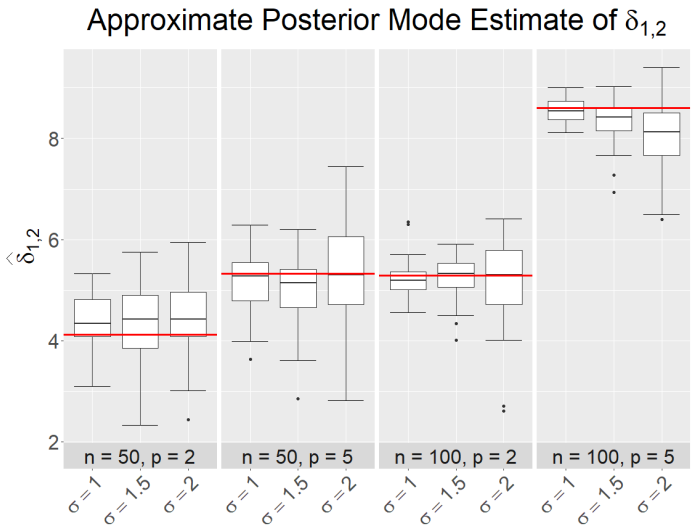


Figure 5.4.1: Simulation results of the BHMDS estimation performance on the true dissimilarity $\delta_{1,2}$. The facet labels, i.e., $n = 50$, $p = 2$, correspond to the sample size n and hyperbolic dimension p of the true dissimilarity data, with $(n, p) \in \{50, 100\} \times \{2, 5\}$. The x-axis labels, i.e. $\sigma = 1$, correspond to the noise level of the observed dissimilarity data, with $\sigma \in \{1, 1.5, 2\}$. For each level of (n, p), we generate a true dissimilarity matrix $\{\delta_{ij}\}$. Then, for each level of σ, we generate 50 sets of noisy dissimilarity matrix $\{d_{ij}\}$ for each $\{\delta_{ij}\}$. We use the proposed BHMDS algorithm to estimate $\{\delta_{ij}\}$. Without loss of generality, we summarized the results on $\widehat{\delta}_{1,2}$, the approximate posterior mode estimate of the dissimilarity between object 1 and 2, in the box plots above. Each box plot corresponds to 50 estimates of $\widehat{\delta}_{1,2}$ at a level of (n, p, σ), and red lines in each facet correspond to the true dissimilarity measures $\delta_{1,2}$ at level (n, p). All box plots closely center around the true values, indicating BHMDS precisely and robustly predicts the true dissimilarity measure.

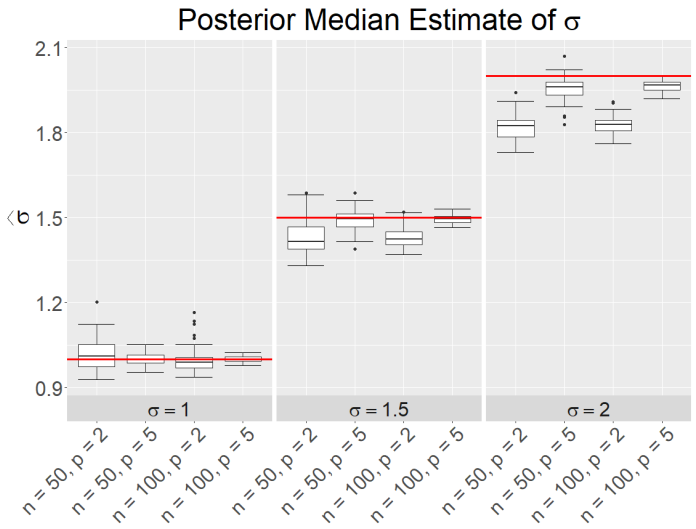


Figure 5.4.2: Simulation result of the BHMDS estimation performance on the true measurement error σ. The facet labels, i.e. $\sigma = 1$, correspond to the noise level of the observed dissimilarity data, with $\sigma \in \{1, 1.5, 2\}$. The x-axis labels, i.e., $n = 50$, $p = 2$, correspond to the sample size n and hyperbolic dimension p of the true dissimilarity data, with $(n, p) \in \{50, 100\} \times \{2, 5\}$. At each level of (n, p, σ), we generate 50 sets of noisy observations of the true dissimilarity matrix, and use the proposed BHMDS algorithm to estimate σ. We summarized the results on $\widehat{\sigma}$, the posterior mean estimate of σ, in the box plots above. Each box plot corresponds to 50 estimates of $\widehat{\sigma}$ at a level of (n, p, σ), and red lines in each facet correspond to the true noise level σ value. We see that as the noise level increases, the accuracy of the estimator $\hat{\sigma}$ decreases, though the difference $|\hat{\sigma} - \sigma|$ decreases as the sample size n increases. In general, all box plots closely center around the true values, indicating BHMDS precisely and robustly measures the amount of uncertainty in data.

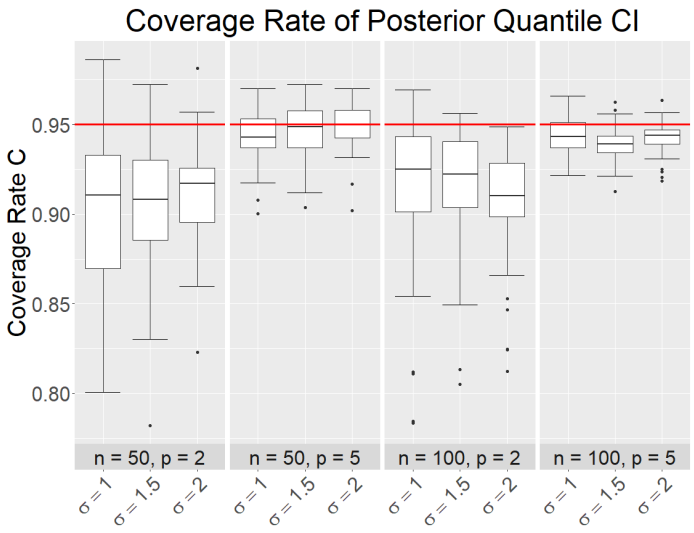


Figure 5.4.3: Simulation result of the BHMDS credible interval's coverage performance. The facet labels, i.e., $n = 50$, $p = 2$, correspond to the sample size n and hyperbolic dimension p of the true dissimilarity data, with $(n, p) \in \{50, 100\} \times \{2, 5\}$. The x-axis labels, i.e. $\sigma = 1$, correspond to the noise level of the observed dissimilarity data, with $\sigma \in \{1, 1.5, 2\}$. At each level of (n, p, σ), we generate 50 sets of noisy observations of the true dissimilarity matrix, and use the proposed BHMDS algorithm to estimate and record the posterior samples of $\{\delta_{ij}\}$. Each box plot contains 50 matrix-wise coverage rates corresponding to the 50 noisy observations, calculated as described in (5.15). The red lines in each facet correspond to the nominal 95% coverage rate. We can observe that BHMDS achieves close-to-nominal coverage rates at all levels of (n, p, σ), and the coverage improves as the sample size n increases.

at time $1, 2, \cdots, S$. We define $\{F_s\}_{s=1,2,\cdots,S}$ as marginally calibrated with respect to $\{G_S\}_{s=1,2,\cdots,S}$ if

$$\bar{G}(x) = \lim_{S\to\infty}\left\{\frac{1}{S}\sum_{s=1}^{S} G_s(x)\right\} \quad \text{and} \quad \bar{F}(x) = \lim_{S\to\infty}\left\{\frac{1}{S}\sum_{s=1}^{S} F_s(x)\right\} \tag{5.16}$$

exist and equal to each other for all x. In practice, we cannot access G_s, but we can always access the empirical cumulative distribution function (CDF) of the observations. Specifically, Theorem 3 of [70] shows that for continuous, strictly increasing G_s and F_s's, $\{F_s\}_{s=1,2,\cdots,S}$ is marginally calibrated with respect to $\{G_s\}_{s=1,2,\cdots,S}$ if and only if

$$\widehat{G}_S(x) = \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}\left(x_s \le x\right) \to \bar{F}(x) \quad \text{almost surely for all } x\ , \tag{5.17}$$

so that the empirical CDF of the observations converges almost surely to the average predictive CDF. [70] suggested plotting the difference between $\widehat{F}_S(x) = \frac{1}{S}\sum_{s=1}^{S} F_s(x)$ and $\widehat{G}_S(x)$ over all x to assess the marginal calibrations of the forecasting distributions, so that the smaller the difference in the plot, the better the calibration.

To evaluate the estimation performance of BHMDS via marginal calibration, we need to first specify the distributions $\{G_s\}_{s=1,2,\cdots,S}$ and $\{F_s\}_{s=1,2,\cdots,S}$. We cannot choose the wrapped normal distribution $\mathcal{G}_s(\boldsymbol{x})$ as G_s, since G_s is required to be univariate, yet $\mathcal{G}_s(\boldsymbol{x})$ is a function of the multivariate random variable $\boldsymbol{x}$, which we wish to vary under different hyperbolic dimensions. Thus alternatively, we choose the distribution of δ_{ij} as G_s, which is the distribution of the hyperbolic distance between coordinates $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ randomly sampled from a given $\mathcal{G}_s(\boldsymbol{x})$, and choose the F_s as the distribution of $\widehat{\delta}_{ij}$ estimated by BHMDS. Choosing a distribution such as G_s is beneficial, as it is inherently univariate and is uniquely determined by the data generating distribution $\mathcal{G}_s(\boldsymbol{x})$, so that if BHMDS estimates the true dissimilarity well, the distribution of $\widehat{\delta}_{ij}$ will resemble the distribution of δ_{ij}. Given G_s's and F_s's, for

each $s = 1, 2, \cdots, 1000$, we randomly generate the parameters

$$
\begin{aligned}
&p^{(s)} \sim \text{Multinomial}(2, 3, 4, 5) \text{ with equal probability} , \\
&\kappa^{(s)} \sim \text{Unif}(0.2, 2) , \\
&\mu_p^{(s)} \sim \mathcal{N}_p(0, 2I_p) , \\
&\Sigma_{ii}^{(s)} \sim_{i.i.d.} \text{Unif}(5, 10), \text{ and } \Sigma_{ij}^{(s)} = 0 \text{ for all } i, j = 1, 2, \cdots, p, i \neq j .
\end{aligned}
$$

For each set of parameters, we sample 20 sets of $\boldsymbol{V}_1^{(s)}, \boldsymbol{V}_2^{(s)}, \cdots, \boldsymbol{V}_{20}^{(s)} \sim \mathcal{N}_p(\mu_p^{(s)}, \Sigma^{(s)})$, each of size $n = 50$, and transform them into hyperbolic coordinates $\boldsymbol{X}_1^{(s)}, \boldsymbol{X}_2^{(s)}, \cdots, \boldsymbol{X}_{20}^{(s)}$ by the transformation in (5.6). From the coordinates, we compute the true dissimilarity matrices and generate their noisy observations at noise level $\sigma = 1$. We then apply BHMDS to the noisy matrices and record the estimated dissimilarities $\{\widehat{\delta}_{ij}\}_1^{(s)}, \{\widehat{\delta}_{ij}\}_2^{(s)}, \cdots, \{\widehat{\delta}_{ij}\}_{20}^{(s)}$, $i, j = 1, 2, \cdots, n$.

Given the distributions G_s at each instance s, we can sample $x_1, x_2, \cdots, x_S$ explicitly and use them to construct $\widehat{G}_S(x)$. On the other hand, we cannot directly access the F_s's, but we can estimate them by the empirical CDFs $\tilde{F}_s$'s from the posterior estimates. To minimize the correlation between the samples, we use the samples $\{\widehat{\delta}_{i,i+1}\}_1^{(s)}, \{\widehat{\delta}_{i,i+1}\}_2^{(s)}, \cdots, \{\widehat{\delta}_{i,i+1}\}_{20}^{(s)}$ for $i = 1, 2, \cdots, n-1$ to compute $\tilde{F}_s$'s. Finally, we plot the difference

$$
\frac{1}{S} \sum_{s=1}^{S} \tilde{F}_s(x) - \widehat{G}_S(x) \tag{5.18}
$$

in Figure 5.4.4 in red. We further evaluate the marginal calibration of the Euclidean `bmds` via the same process and plot the difference in blue. The BHMDS method is much more calibrated than the `bmds` method, as the calibration curve for the BHMDS method is closer to zero for all threshold values on the x-axis. This suggests when the dissimilarity data is hierarchical, tree-like, or has intrinsic hyperbolic property, the proposed BHMDS algorithm yields much more precise estimates of the true dissimilarity compared to Euclidean MDS method.

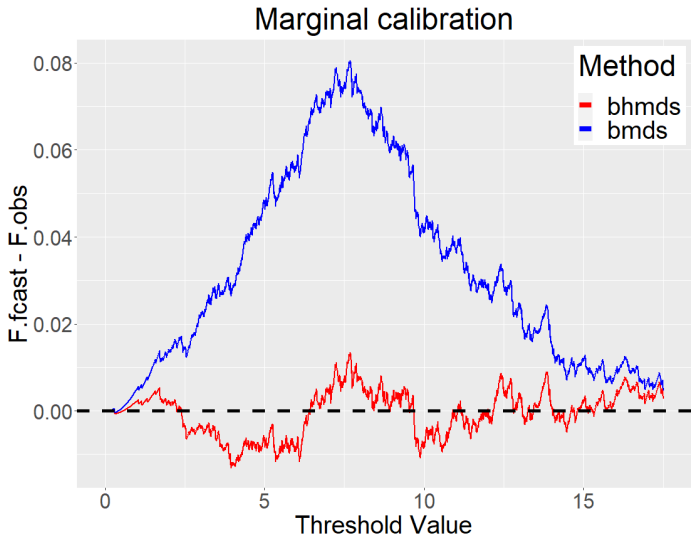


Figure 5.4.4: The marginal calibration result for BHMDS amd `bmds`. The x-axis corresponds to the threshold values of the dissimilarity distribution. The y-axis corresponds to the difference as described in (5.18). The red line corresponds to the marginal calibration plot of BHMDS, the blue line corresponds to the marginal calibration plot of `bmds`, and the black horizontal dashed line corresponds to $y = 0$. We can observe that, when the true dissimilarity are generated from the hyperbolic geometry, BHMDS performs significantly better than `bmds`, with the difference only fluctuating in a small interval around zero, in contrast to the `bmds` difference, where there is a large spike at $x = 7.5$. This suggests that BHMDS outperforms `bmds` in estimating dissimilarities when the data is hierarchical.

5.5 Data analysis

We now apply the proposed BHMDS algorithm to analyze a variety of dissimilarity datasets. First, we cross-compare BHMDS with multiple prevalent MDS algorithms using MDS goodness-of-fit criteria with respect to several hierarchical dissimilarity data collected in social network and Natural Language Processing (NLP) studies. Then, we present a case study of the case-control likelihood approximation on a hierarchical NLP hypernym data. Lastly, we apply BHMDS with a global human gene expression data to investigate the cellular differentiation of different cell types with confidence quantification on their rank statistics.

5.5.1 Comparison with existing MDS approaches

In this section, we apply BHMDS algorithm to several tree-like, hierarchical datasets and evaluate its embedding performance via MDS goodness-of-fit criteria stress and distortion.

We first give the definitions of the MDS goodness-of-fit criteria. Stress is one of the most prevalent criteria in the MDS literature, which measures the L_2 goodness-of-fit of the embedding. Given the observed dissimilarities $\{d_{ij}\}$ and its MDS embedding $\{\widehat{\delta}_{ij}\}$, their stress is defined as

$$\text{stress}\left(\{d_{ij}\},\{\widehat{\delta}_{ij}\}\right) \equiv \sqrt{\frac{\sum_{i<j}\left(d_{ij}-\widehat{\delta}_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}}\ , \quad i,j=1,\cdots,n\ . \tag{5.19}$$

The distortion, on the other hand, measures the L_1 goodness-of-fit of the embedding, defined as

$$\text{Distortion}\left(\{d_{ij}\},\{\widehat{\delta}_{ij}\}\right) \equiv \frac{1}{m}\sum_{i<j}\frac{\left|d_{ij}-\widehat{\delta}_{ij}\right|}{d_{ij}}\ , \quad i,j=1,\cdots,n\ , \tag{5.20}$$

where $m = n(n-1)/2$ is the number of dissimilarities. Both criteria normalize the difference terms by the observed dissimilarities, which enables cross-comparison among datasets with different size and scale.

We consider the following datasets in our numerical study. We first consider the Zachary's Karate Club dataset, a commonly used social network of a university karate club, first described by [176] and studied via the hyperbolic MDS algorithms `hydra` and `hydraPlus` in [99]. Next, we consider a phylogenetic tree dataset which expresses the genetic heritage of mosses growing in urban environments, first described in [81] and studied in [50]. We further consider two tree-like, hierarchical datasets: one is a Computer Science Ph.D. advisor-advisee network, available from [53] and also studied in [50]; the other is the WordNet mammals subtree dataset, a NLP hypernym dataset studied in [130]. All of the four datasets come in the form of undirected graphs, thus we take the shortest path lengths on the graph between objects i and j as the observed dissimilarity measures $\{d_{ij}\}$. Given the observed dissimilarity matrices, we fix the hyperbolic dimension as $p = 2$ and estimate their curvature κ's as described in Appendix D.2, except for the karate dataset, where we set $\kappa = 1$ as in [99]. Given the hyperbolic curvature and dimension, we use our BHMDS algorithm to compute the posterior estimate $\{\widehat{\delta}_{ij}\}$ and the corresponding goodness-of-fit criteria. To compare the embedding performance with existing methods, we included results from common hyperbolic MDS methods using the same hyperbolic curvature and dimension, as well as Euclidean MDS methods with dimension $p = 2$. The embedding results are displayed in Table 5.5.1 and 5.5.2. We can conclude from the tables that the BHMDS algorithm attains optimal or close-to-optimal embedding in terms of both criteria on all of the four datasets. This indicates that, when the dissimilarity data is tree-like or hierarchical, the proposed BHMDS algorithm not only quantifies the uncertainty in the observed dissimilarity data, but also embeds it into hyperbolic geometry with minimal information loss compared to the state-of-the-art MDS algorithms.

5.5.2 Approximated Log-likelihood: A Case Study

In this section, we present a case study to exemplify how to apply the stratified case-control log-likelihood and to evaluate the algorithm's likelihood precision and

Table 5.5.1: Embedding performance of the four datasets in terms of the stress criteria. The red text correspond to the optimal stress values, and the blue text correspond to the second optimal values. We can observe that, the stress-minimizing hyperbolic MDS methods, namely BHMDS and `hydraPlus`, constantly outperforms all other methods, and their embedding results are comparable. This indicates that tree-like, hierarchical data is best represented on hyperbolic geometry in terms of stress. Moreover, it shows that the BHMDS algorithm can be used to optimize the minimal-stress embedding.

	Size n	BHMDS	`hydraPlus`	`hydra`	`bmds`	`smacof`	`cmds`
Karate ($\kappa = 1$)	34	0.1780	0.1727	0.2105	0.2050	0.2112	0.2850
Phylo ($\kappa = 0.14$)	344	0.0413	0.0413	0.2091	0.1601	0.1594	0.2481
CS phd ($\kappa = 0.55$)	1025	0.1469	0.1471	0.2029	0.2281	0.2351	0.3862
Wordnet ($\kappa = 2.06$)	1141	0.0798	0.0807	0.1279	0.2722	0.2725	0.4928

Table 5.5.2: Embedding performance of the four datasets in terms of the Distortion criteria. The red text correspond to the optimal Distortion values, and the blue text correspond to the second optimal values. Again, we can observe that, the stress-minimizing hyperbolic MDS methods, namely BHMDS and `hydraPlus`, constantly outperforms all other methods even though they are not designed to optimize the Distortion, and their embedding results are comparable. This indicates that tree-like, hierarchical data is also best represented on hyperbolic geometry in terms of Distortion. Moreover, it shows that the BHMDS algorithm can be used to optimize the minimal-distortion embedding.

	Size n	BHMDS	`hydraPlus`	`hydra`	`bmds`	`smacof`	`cmds`
Karate ($\kappa = 1$)	34	0.3485	0.3287	0.4278	0.3986	0.3742	0.5234
Phylo ($\kappa = 0.14$)	344	0.1214	0.1206	0.6806	0.3552	0.3516	0.5918
CS phd ($\kappa = 0.55$)	1025	0.2869	0.2868	0.4545	0.4523	0.4510	0.7513
Wordnet ($\kappa = 2.06$)	1141	0.1441	0.1460	0.2242	0.4994	0.4967	0.9670

computational efficiency. We consider the WordNet mammals subtree dataset, a hierarchical NLP hypernym dataset studied in [130], as the input dataset. The WordNet dataset comes in as an undirected graph with $n = 1141$ nodes, and we take the shortest path lengths on the graph between objects i and j as the observed dissimilarities d_{ij}. We set the hyperbolic dimension $p = 2$ and estimate the curvature using the methods described in Appendix D.2.

We now explain how to apply the approximated log-likelihood to the WordNet dataset. For each object i, we first partition objects $j \in \{1, 2, \cdots, n, j \neq i\}$ by their observed dissimilarity d_{ij}'s. Since we are using the shortest path length as dissimilarity, all d_{ij}'s are positive integers ranging from $1, 2, \cdots, \max_j(d_{ij})$ for each i, and objects with the same d_{ij} value are in the same group. That is, we set

$$S_k^{(i)} \equiv \{\text{Object } j : d_{ij} = k, j = 1, 2, \cdots, n, j \neq i\}, \quad k = 1, 2, \cdots, \max_j(d_{ij}), \quad (5.21)$$

so that we collect all objects of distance k to object i in $S_k^{(i)}$.

We consider $S_1^{(i)}$ and $S_2^{(i)}$, the two strata defined with the smallest dissimilarities (neighbours and second-neighbours) as cases and all other $S_j^{(i)}$ for $j > 2$ as controls. We choose a control-to-case rate $r = 5$, so that in the approximated log-likelihood, we explicitly calculate over 7% of the terms in the original one. We run a pilot MCMC with 3000 iterations and 1000 burn-ins to compute the relative weights and samples sizes for each $S_j^{(i)}$ for $j > 2$. We then run the case control approximated log-likelihood MCMC to obtain the posterior estimate $\{\delta_{ij}\}$.

We first evaluate the overall estimation performance of the case-control MCMC algorithm in term of the stress and computation time. We will run both the approximate and full MCMC algorithm with the WordNet dataset, and record the stress value of their posterior estimates as well as the computational times per 100 MCMC iterations. We display the results in Table 5.5.3 below. We observe that the case-control approximate log-likelihood MCMC achieves a comparable stress to the full MCMC but with only half of the computation time, indicating the approximate MCMC achieves

fast and accurate estimates of the true dissimilarities for large datasets.

To evaluate the precision of the case-control likelihood, we compare the case-control log-likelihood change to the full log-likelihood change in the MCMC proposal of $\boldsymbol{v}_i$. If the case-control log-likelihood change approximates the full log-likelihood change well, the case-control MCMC will accept or reject in a similar pattern as the full MCMC. Recall that the change in the log likelihood, as defined in Section 5.3.2, is $\Delta\tilde{l}_i^{(t)} = \tilde{l}_i(\boldsymbol{v}_{i,\text{new}}^{(t)}) - \tilde{l}_i(\boldsymbol{v}_i^{(t)})$. For a valid comparison, it is essential to evaluate both likelihood changes with respect to the same $\boldsymbol{v}_i$ proposal and parameters such as $\{\delta_{ij}\}, \sigma^2, \Lambda, \beta$. To this end, we first run the full MCMC algorithm with the Wordnet dataset, and record 100 sets of parameters $\boldsymbol{\theta}^{(i)} = \left(\{\delta_{ij}\}^{(i)}, (\sigma^2)^{(i)}, \Lambda^{(i)}, \beta^{(i)}\right)$ from 100 MCMC iterations. For computational efficiency, we randomly sample 100 objects from the $n = 1141$ samples for each $\boldsymbol{\theta}^{(i)}$, and make proposal upon the corresponding $\boldsymbol{v}_1^{(i)}, \boldsymbol{v}_2^{(i)}, \cdots, \boldsymbol{v}_{100}^{(i)}$ under $\boldsymbol{\theta}^{(i)}$. We compute the approximate likelihood change and the full likelihood change for each proposal $\boldsymbol{v}_j^{(i)}, i, j = 1, \cdots, 100$, and plot the approximated log-likelihood change values against the full log-likelihood change values in Figure 5.5.1. We observe that the approximate log-likelihood change tightly aligned around $y = x$, with a strongly positive correlation $\rho = 0.82$. This indicates the log-likelihood values computed from the proposed case-control MCMC algorithm are a good approximation to the true log-likelihood values. Furthermore, the case-control MCMC shares a similar accept/reject pattern as the full MCMC and is able to properly sample from the posterior.

5.5.3 Quantifying Uncertainty in Human Gene Expression Data

We now use the proposed BHMDS algorithm to analyze a global human gene expression dataset. [119] integrated microarray data from 5372 human samples representing 369 different cells and tissue types, disease state and cell lines and constructed a global human gene expression map. The original data come in the form of a jointly normalized gene expression matrix of over 22000 probes sets times 5372 genes of

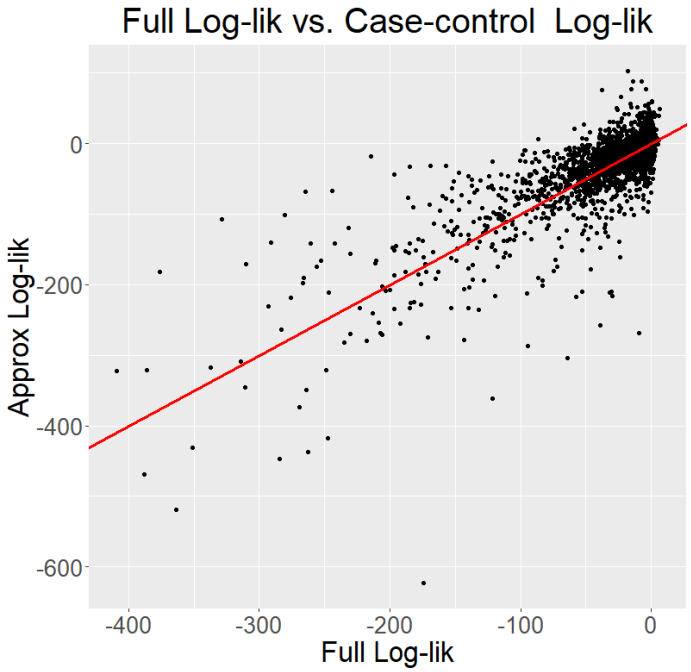


Figure 5.5.1: X-axis: Log-likelihood change values calculated via the full MCMC algorithm. Y-axis: Log-likelihood values calculated via the case-control approximate MCMC algorithm. The red line corresponds to the line $y = x$. We can observe that, the approximated log-likelihood changes tightly aligned around $y = x$ against the full log-likelihood changes, with a strongly positive correlation $\rho = 0.82$. This indicates the log-likelihood values computed from the proposed case-control MCMC algorithm is a good approximation of the true log-likelihood values, and the case-control MCMC share a similar accept/reject pattern as the full MCMC.

Table 5.5.3: Stress values and computational time per 100 MCMC iterations of the WordNet mammal subtree dataset with `hydraPlus`, approximated MCMC, and full MCMC. We can observe that, the approximated case-control algorithm achieves a comparable stress with `hydraPlus` and the full MCMC, indicating it estimates a close-to-optimal hyperbolic embedding of the WordNet dataset. More importantly, the proposed case-control algorithm is approximately twice as fast as the full MCMC algorithm. This will enable the BHMDS framework scalable with dissimilarity data of large sample sizes.

Method	hydraPlus	Approx	Full
Stress	0.081	0.085	0.080
MCMC time (100 iters)	-	28.39s	54.88s

15 human cell types. To construct the dissimilarity matrix, the pairwise Euclidean distances of the gene vectors are taken, which results in a dissimilarity matrix with $n = 5372$. Although the dissimilarities are computed from a Euclidean embedding, [178] proposed that the intrinsic geometry of the human gene expression data is hyperbolic. Moreover, [56], [133], and [146] argued that the gene expression data is likely to contain measurement error. Thus, it is reasonable to apply the proposed BHMDS algorithm with the human gene expression dissimilarity matrix to quantify the uncertainty in analysis. Specifically, we use the case-control approximate MCMC algorithm to compute the Bayesian estimate of the cluster distances between cell type communities, and quantify the uncertainty on the rank statistics of cell types' evolution pseudotimes.

We first elaborate the implementation details of the case-control approximated MCMC. As the dissimilarity matrix is computed from high dimensional Euclidean

metrics, the original human gene expression dissimilarities are large, which leads to overflow issues. Thus, we preprocess the dissimilarity data as follows. We fix the hyperbolic curvature $\kappa = 1$, scale the dissimilarity matrix by a constant, and use `hydraPlus` to compute the corresponding stress with the hyperbolic dimension $p = 5$ as chosen in [178]. We repeat the above process with a grid search over the scaling constants, until we find an optimal constant. Such a process is similar in spirit to the algorithm described in Section D.2, as altering the curvature is roughly equivalent to scaling the distance as observed in (5.1), with the relationships between the dissimilarities unchanged. We found that scaling the dissimilarity by 10 yields the optimal stress at 0.046, resulting a rescaled matrix with an average dissimilarity at 15.12. Since the gene expression dissimilarity is continuous, we partition the rescaled dissimilarity by continuous intervals $[0, 7), [7, 8), [8, 9), \cdots, [24, \infty)$, and consider the dissimilarities that falls within $[0, 7)$ as in the case group. We further choose $r = 10$, so that we will explicitly compute 6% of the likelihood terms.

To compute the Bayesian estimate of the cell type cluster distance, at each MCMC iteration t, we record the cluster distance for cell type community C_i and C_j of size n_i and n_j as

$$d_{\text{cluster}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{(k,l)\in(C_i,C_j)} d_{\mathbb{H}}(\boldsymbol{x}_k, \boldsymbol{x}_l), \quad i \neq j, \quad i, j = 1, 2, \cdots, 15\,, \tag{5.22}$$

where the community membership is given in the original data. We then take the posterior median of each $d_{\text{cluster}}(C_i, C_j), i, j = 1, \cdots, 15$, and plot them on Figure 5.5.2. Our Bayesian estimate identifies that the hematopoietic cells are distinct from all other cells, which is also observed in [119]. We also observe modularity in the neoplasm cells at the upper right corner of Heatmap 5.5.2, indicating their proximity in terms of evolutionary distance.

We further used the BHMDS algorithm to measure different cell types' cellular differentiation using rank statistics with uncertainty quantification, a feature that is not available from previous methods used to analyze these data. Cellular differentia-

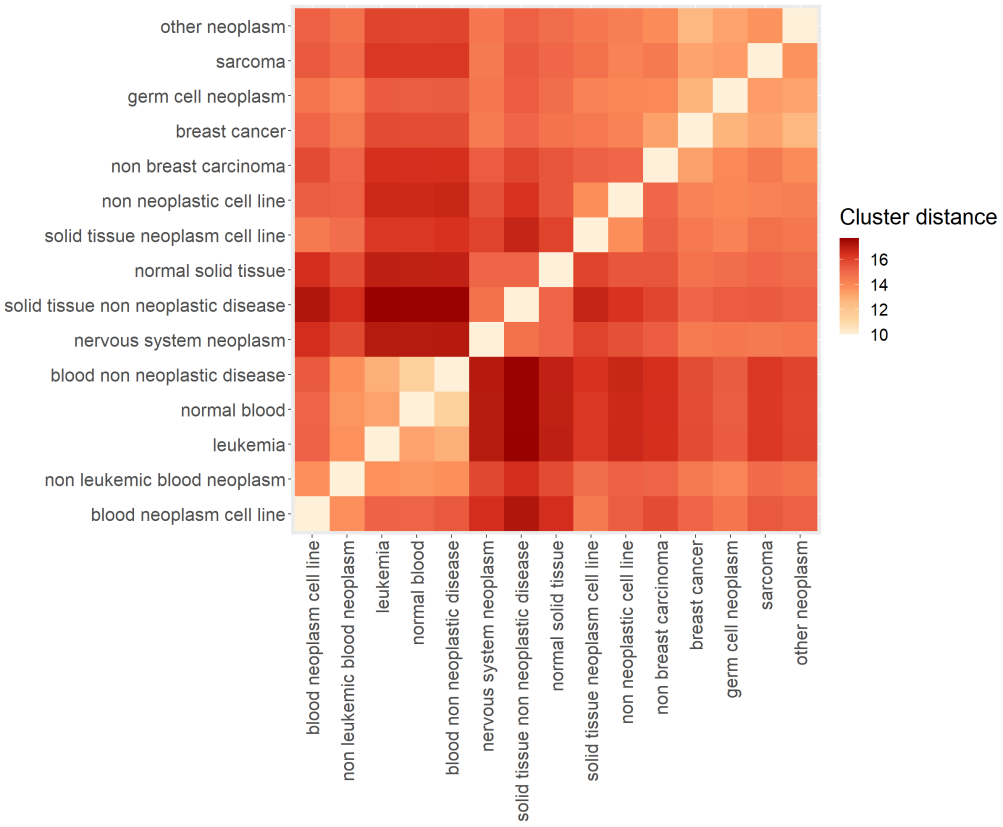


Figure 5.5.2: Heatmap of the cluster distance between cell type communities defined in (5.22). We observe the hematopoietic cells types, i.e., blood neoplasm cell line, non leukemic blood neoplasm, leukemia, normal blood, and blood non neoplastic disease, are distinct with all other cells. We also observed modularity in neoplasm cells, i.e. the clustering of breast cancer cell, germ cell neoplasm, sarcoma, and other neoplasm cells.

tion refers to the transition of immature cells into specialized types, which is a central task in modern developmental biology [105]. Specifically, [105] studied the hierarchy of the single-cell data on hyperbolic geometry. For cells that are less differentiated, or at the beginning of a developmental process, they will be at the root of the evolutionary hierarchy and have relatively equal and small distances with all other cells, so that by the nature of the hyperbolic geometry, they are more likely to be embedded around the hyperbolic origin. Similarly, for cells that are more differentiated, they will have relatively large distances with all other cells, so that they are more likely to be embedded distant from the origin. Thus, [105] proposed that the hyperbolic distance between the origin and the cell's embedded coordinate can be a good measure on the extent of the cell's cellular differentiation, which they denoted as hyperbolic evolution pseudotime. In our study, at each iteration of the MCMC, we record the evolution pseudotime for each gene, and use them to construct the posterior credible intervals of the evolution pseudotime for each gene. To visualize the average evolution pseudotime for each cell type, we summarized the ranks statistics of each cell type as follows. For each cell type community, we randomly sample one of its gene, and draw from its nominal pseudotime confidence interval based on the posterior. Then, we rank the pseudotime drawn from each cell type community and record their rank statistics. We repeat the above process for 10,000 times. We then summarize the rank statistics by frequency in Figure 5.5.3, where the ijth entry of the heatmap represents for the frequency of cell type i being the jth closest to the hyperbolic origin, which indicates it is jth less differentiated. Figure 5.5.3 indicates that cell types within the same cluster in Figure 5.5.2 share similar hierarchy, as the neoplasm cells obtain higher frequency for the higher rank statistics thus are less differentiated, whereas the hematopoietic cells obtain higher frequency for the lower rank statistics thus are more differentiated. Additionally, we observe that the germ cell neoplasm has the smallest evolution pseudotime, which to our knowledge is the first in literature, entailed by breast cancer cell, which is also argued in [178] and [64]. These results further shed

light on the study of cancer stem cells, as short evolution pseudotimes of the two neoplasm cells suggests they are likely to be more de-differentiated, a hallmark often observed in malignant tumors.

5.6 Conclusion

In this work, we proposed a Bayesian approach to multidimensional scaling in hyperbolic space. Using a previously studied generating process for observed dissimilarities from [132], we used prior distributions on hyperbolic space to derive the posterior distribution of the model parameters. We then proposed an MCMC procedure to sample from this posterior and also proposed a quick case-control method to efficiently sample from the posterior when the sample size is large. Finally, we applied our methods to datasets in several domains and showed how our Bayesian procedure leads embeddings with low distortion and allows us to quantify the uncertainty due to noise in the observed dissimilarities. In independent work issued just as we submitted this chapter, [142] also carried out a Bayesian analysis of hyperbolic MDS. The present work differs from that of [142] in several ways. Namely, we use priors for the latent positions in the hyperbolic space whereas [142] assume a Gaussian noise structure on distances. Additionally, we propose using a case-control approximation to address computation whereas [142] use an iterative approach that adds observations in blocks.

There are several avenues of interesting work. First, we have assumed in this work a low dimension for the desired embedding. While this does allow for easy visualization of the resulting embedding, imposing a low dimension on the dissimilarities is likely to lead to higher distortion than larger dimensions. One potential area of future work could derive an information criterion to estimate an optimal embedding dimension given the dissimilarities, as was done in [132]. Second, exploring how our proposed procedure could be adopted into the non-metric multidimensional scaling framework. Finally, while hyperbolic space has received a lot of attention in the past few years, spaces of non constant curvature, as studied in [50] and others, might lead

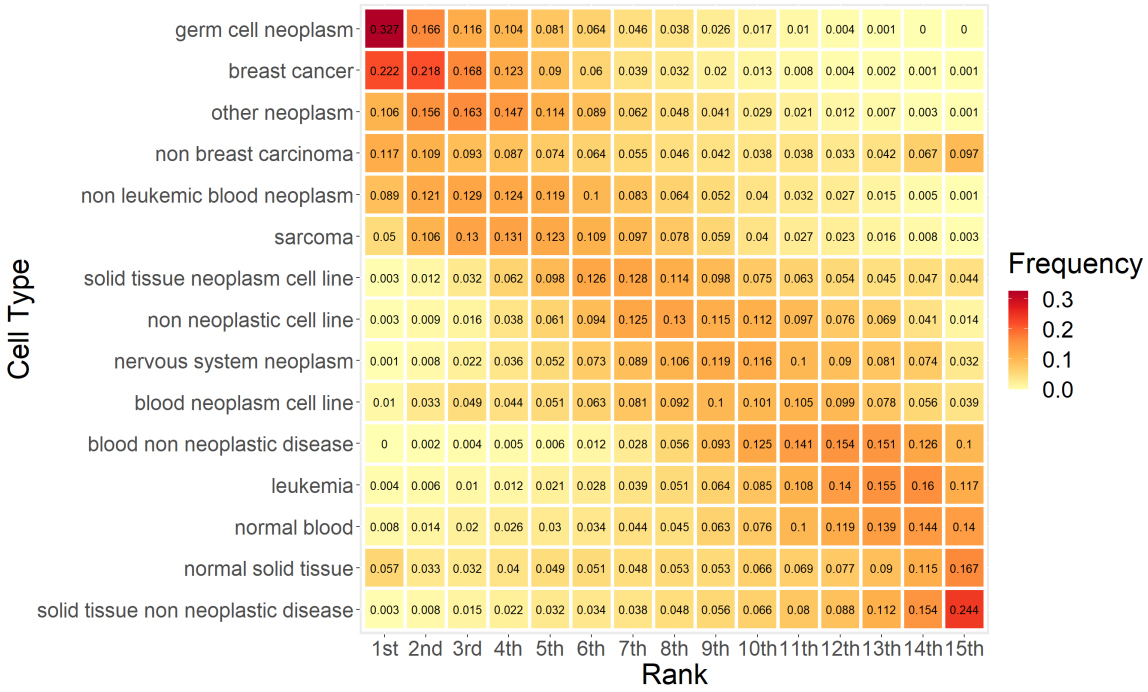


Figure 5.5.3: Heatmap on the frequency of the rank statistics for 15 different human cell type. We observe that cell types within the same cluster (hematopoietic, neoplasm) share similar hierarchy, with the neoplasm cell on the higher hierarchy and the hematopoietic cells on the lower hierarchy. The germ cell neoplasm most frequently attains the smallest evolution pseudotime, entailed by breast cancer cells and other neoplasm cells. This potentially indicates that the neoplasm cells with high frequency in rank statistics are more de-differentiated.

to an embedding with lower distortions. An interesting research question here could focus on proposing Bayesian MDS methods in these spaces.

Chapter 6

CONCLUSION

In this thesis, we focused on several key problems in network inference. In the first chapter, we discussed how to estimate the latent space geometry of the generative latent space network model. Here, as is common in the literature, we have assumed that the curvature of the latent space is constant, which then implies that the latent space is either Euclidean, spherical, or hyperbolic. However, such an assumption might not always be appropriate. See, among many others, [77]. Estimating the properties of the non constant curvature latent space is an interesting area of future work.

The second chapter deals with using ARD to estimate the properties of an unobserved network. We provided conditions under which network statistics can be estimated consistently using ARD. These models assume perfect recall of the network structure, which is unlikely to hold in practice, so an interesting area of future work would involve building statistical models for the ARD responses respondents given to incorporate recall bias. Also, while ARD is a commonly used type of network data, it is not clear that ARD is an optimal data type to collect. An exciting area of future work can try to derive other survey methodology to better estimate properties of unobserved networks.

The third chapter deals with assessing model goodness-of-fit for network data. Here, we show how the largest and smallest eigenvalues of the normalized adjacency matrix can be used to do network model selection. To show that such a procedure returns a consistent classifier of the true network model, one must show that the estimators used to estimate the matrix of node-level edge probabilities converges fast

enough [110]. In future work, we'd like to show that the estimators we derived in chapter 2 of this thesis satisfy this convergence criteria. Finally, in the last chapter we built a Bayesian model for obtaining embeddings of objects in a Hyperbolic space using dissimilarity data. In future work, we'd like to build similar models to obtain embeddings in other spaces, such as those with non-constant curvature [77].

Appendix 1

APPENDIX FOR CHAPTER 2

This chapter contains the proofs for the results in Chapter 2 and additional simulations.

A.1 Proofs

A.1.1 Proofs Required for Theorem 2.1.1 (Section 2.3)

Proof of Proposition 2.3.1. We only prove this claim for the Euclidean case, but the same argument proves the claim for the other two geometries. We have by Weyls's inequality that $|\lambda_1(\hat{W}_0) - \lambda_1(W_0)| \leq ||\hat{W}_0 - W_0||_F$, where $||A||_F$ is the Frobenius norm of A, that is $||A||_F^2 = \sum_{l,l'} a_{ll'}^2$. Then, we have that $\mathbb{P}(|\lambda_1(\hat{W}_0) - \lambda_1(W_0)| < \theta) \leq \mathbb{P}(||\hat{W}_0 - W_0||_F < \theta)$ for all θ. Under $H_{0,e}$, $\lambda_1(W_0) = 0$, so we have that $\mathbb{P}(|\lambda_1(\hat{W}_0)| < \theta) \leq \mathbb{P}(||\hat{W}_0 - W_0||_F < \theta)$. By setting θ to be the α quantile of $||\hat{W}_0 - W_0||_F$, we conclude (2.3). This completes the proof. □

Proof of Proposition 2.3.2. We prove this proposition for the spherical case. The hyperbolic case follows from a similar argument. By Theorem 2.1 in [129], we have consistency if (i) the limit objective function is uniquely maximized at the truth, (ii) the parameter space is compact, (iii) the limit objective function is continuous in the parameter, and (iv) there is uniform convergence of the empirical objective function to its limit. The latter holds if there is point-wise convergence and stochastic equicontinuity. The parameter space is compact and since under the null $W_\kappa(D)$ is positive semi-definite, the minimum eigenvalue is 0 as long as $K > p$. Identification comes from continuity of eigenvalues in parameters of the matrix. Finally we check uniform

convergence. First, note by hypothesis that $\hat{D} \xrightarrow{p} D$ as $T \to \infty$. Since eigenvalues are continuous functions of their matrix arguments, we have by the continuous mapping theorem that $\lambda_1\left(\kappa W_\kappa(\hat{D})\right) \xrightarrow{p} \lambda_1\left(\kappa W_\kappa(D)\right)$ for every $\kappa \in [a,b]$, and so we have pointwise convergence. To complete the proof, we will show stochastic equicontinuity to show uniform convergence. A sufficient condition is a Lipschitz condition (Lemma 2.9, [129]): that for any κ_1, κ_2, $\left|\lambda_1\left(\kappa_1 W_{\kappa_1}(\hat{D})\right) - \lambda_1\left(\kappa_2 W_{\kappa_2}(\hat{D})\right)\right| \leq B_T|\kappa_1 - \kappa_2|$ for some random variable $B_T = O_p(1)$. To do this, fix any $\kappa_1, \kappa_2 \in [a,b]$. By Weyl's inequality,

$$\left|\lambda_1\left(\kappa_1 W_{\kappa_1}(\hat{D})\right) - \lambda_1\left(\kappa_2 W_{\kappa_2}(\hat{D})\right)\right| \leq ||\kappa_1 W_{\kappa_1}(\hat{D}) - \kappa_2 W_{\kappa_2}(\hat{D})||_F.$$

Since $\kappa W_\kappa(D) = \cos(\sqrt{\kappa}D)$ and $\cos(\cdot)$ is Lipschitz continuous with Lipschitz constant 1, we have for each l, l',

$$\left|\cos(\kappa_1^{1/2}\hat{d}_{l,l'}) - \cos(\kappa_2^{1/2}\hat{d}_{l,l'})\right| \leq \hat{d}_{l,l'} \cdot \left|\kappa_1^{1/2} - \kappa_2^{1/2}\right|.$$

For $\kappa_1, \kappa_2 \in [a,b]$,

$$|\sqrt{\kappa_1} - \sqrt{\kappa_2}| = \left|\frac{\kappa_1 - \kappa_2}{\sqrt{\kappa_1} + \sqrt{\kappa_2}}\right| \leq \frac{1}{2\sqrt{a}}|\kappa_1 - \kappa_2|,$$

so for any $\hat{d}_{i,j}$,

$$\left|\cos(\kappa_1^{1/2}\hat{d}_{i,j}) - \cos(\kappa_2^{1/2}\hat{d}_{i,j})\right| \leq \frac{\hat{d}_{i,j}}{2a^{1/2}}\left|\kappa_1 - \kappa_2\right|.$$

Putting this all together, we see that

$$\begin{aligned}
\left|\lambda_1\left(\kappa_1 W_{\kappa_1}(\hat{D})\right) - \lambda_1\left(\kappa_2 W_{\kappa_2}(\hat{D})\right)\right| &\leq \sqrt{\sum_{i,j}\left(\kappa_1 W_{\kappa_1}(\hat{D}) - \kappa_2 W_{\kappa_2}(\hat{D})\right)_{i,j}^2} \\
&\leq \sqrt{\sum_{i,j}\left(\frac{\hat{d}_{i,j}}{2a^{3/2}}\left|\kappa_1 - \kappa_2\right|\right)^2} \\
&= \sqrt{\sum_{i,j}\left(\frac{\hat{d}_{i,j}}{2a^{3/2}}\right)^2}|\kappa_1 - \kappa_2|.
\end{aligned}$$

Since $\sqrt{\sum_{i,j}\left(\frac{\hat{d}_{i,j}}{2a^{3/2}}\right)^2} = O_p(1)$, the desired Lipschitz condition holds, which completes the proof. The hyperbolic case is handled in a similar way. □

Proof of Proposition 2.3.3. We prove the Euclidean case (part a) and note that the proofs of parts b and c (spherical and hyperbolic) are nearly identical. Define $\mathcal{R}_T = (-\infty, \delta_T]$. Let $\mathbb{P}_0(A)$ denote the probability of the event A under the null hypothesis that $\mathcal{M}^p(\kappa)$ is Euclidean. By (2.8),

$$\mathbb{P}_0(\lambda_1(\hat{W}_0) \in \mathcal{R}_T) = \mathbb{P}_0(\lambda_1(\hat{W}_0) \leq \delta_T) = o(1) \ ,$$

by assumption. Under H_1, $\lambda_1(W_0) < 0$ by Lemma 2.2.1. Since $\delta_T = o_P(1)$,

$$\mathbb{P}_1(\lambda_1(\hat{W}_0) \in \mathcal{R}_T) = \mathbb{P}(\lambda_1(\hat{W}_0) \leq \delta_T) = 1 - \mathbb{P}\left(\lambda_1(\hat{W}_0) \geq \delta_T\right) = 1 - o(1) \ .$$

This proves that the test for (2.4) is consistent, as claimed. □

Proof of Proposition 2.3.5. For each index $j = 1, \ldots, K$, we consider two different cases. In case 1, $\lambda_j(W_\kappa) \neq 0$. In this case, know that since $\epsilon_T \xrightarrow{p} 0$, $\mathbb{P}(\lambda_j(\hat{W}_{\hat{\kappa}}) \in \mathcal{R}) \to 0$. Here we just use the fact that $\lambda_j(\hat{W}_{\hat{\kappa}}) \to c \neq 0$, so that eventually this eigenvalue is outside the rejection region. Note that this calculation does not require a particular rate on ϵ; we just need ϵ to go to zero in probability. We now handle case 2, in which $\lambda_j(W_\kappa) = 0$. This is the more subtle case. By definition of the rejection region,

$$\mathbb{P}_1(\lambda_j(\hat{W}_{\hat{\kappa}}) \in \mathcal{R}) = \mathbb{P}_1(|\lambda_j(\hat{W}_{\hat{\kappa}})| \leq \epsilon_T) \ .$$

where the subscript here indicates that the alternative hypothesis that $\lambda_j \neq 0$ is true. By Weyl's inequality, the above probability then becomes

$$\mathbb{P}_1(|\lambda_j(\hat{W}_{\hat{(\kappa}})| \leq \epsilon_T) \leq \mathbb{P}_1(||\hat{W}_{\hat{\kappa}} - W_\kappa||^2 \leq \epsilon_T^2) := \mathbb{P}_1(r_T \leq \epsilon_T) \ .$$

By assumption, we know that $r_T/\epsilon_T \to 0$ in probability. Therefore, for any eigenvalue that is actually zero, for large enough T we will call this eigenvalue zero and it won't count in the estimated rank. This concludes the proof. □

Proof of Theorem 2.1.1. By assumption, we know that $\hat{D} \xrightarrow{p} D$, so by Proposition 2.3.2 we have that $\hat{\kappa} \xrightarrow{p} \kappa$. We will use Proposition 2.3.3 to argue that $\widehat{\mathcal{M}^{\hat{p}}}$ is consistent

for $\mathcal{M}^p(\kappa)$. To do this, note that if $\mathcal{M}^p(\kappa)$ is Euclidean, then by Proposition 2.3.3, $\widehat{\mathcal{M}^{\hat{p}}}$ is consistent. To prove the claim for the spherical case, recall that we define $\phi(\hat{W}_0) = 1$ to mean that we reject the hypothesis that $\mathcal{M}^{p^\star}(\kappa^\star)$ is Euclidean. If $\phi(\hat{W}_0) = 0$ then we fail to reject the hypothesis that $\mathcal{M}^{p^\star}(\kappa^\star)$ is Euclidean. Similar definitions hold for the spherical and hyperbolic cases.

If $\mathcal{M}^{p^\star}(\kappa^\star)$ is spherical, then we have that

$$\begin{aligned}
\mathbb{P}_S(\widehat{\mathcal{M}^{\hat{p}}} = \mathbf{S}^p(\kappa)) &= \mathbb{P}_S(\phi(\hat{W}_0) = 1,\ \phi(\hat{W}_{\hat{\kappa}}) = 0) \\
&= \mathbb{P}_S(\phi(\hat{W}_0) = 1)\mathbb{P}_S(\phi(\hat{W}_{\hat{\kappa}}) = 0) \\
&\to 1 \ ,
\end{aligned}$$

where the notation $\mathbb{P}_S$ indicates that $\mathcal{M}^{p^\star}(\kappa^\star) = \mathbf{S}^p(\kappa)$ and the third line follows from Proposition 2.3.3. A similar argument proves that $\widehat{\mathcal{M}^{\hat{p}}}$ is consistent when $\mathcal{M}^{p^\star}(\kappa^\star)$ is hyperbolic. Therefore, we can conclude that $\hat{p}$ is consistent for the true rank of W_κ. This completes the proof. □

A.1.2 Proofs Required for Theorem 2.1.2 and Section 2.4

Proof of Theorem 2.1.2. In order to show that estimates of distances computed using cliques are consistent, we recall the form of the estimates from (2.14),

$$d(z_i^\star, z_j^\star) = -\log(p_{ij}) + \log(\gamma) \ ,$$

where $\gamma := E\{\exp(\nu)\}^2$ and $p_{ij} = \mathbb{P}(G_{ij} = 1 | z_i^\star, z_j^\star)$. We estimate $d(z_i^\star, z_j^\star)$ with

$$\hat{d}(z_i, z_j) = -\log(\hat{p}_{ij}) + \log(\hat{\gamma}) \ .$$

Under the assumptions of Theorem 2.1.2, we have a consistent estimate $\hat{\gamma} \to \gamma$, so we only focus on estimating the term p_{ij}. We estimate this term using

$$\hat{p}_{kk'} = \ell^{-2} \sum_{i \in C_k(\ell)} \sum_{j \in C_{k'}(\ell)} G_{ij} \ ,$$

where $C_k(\ell)$ is a clique of size ℓ and $C_{k'}(\ell)$ is another clique of size ℓ. Suppose that each node in a clique is at the same location, say ζ_k for clique k. Then, $\hat{p}_{kk'}$ is a consistent estimate of p_{ij}, the probability that node i at location z_k connects to node j at location $z_{k'}$. In practice, the locations in cliques do not fall at exactly the same location, but as $\ell \to \infty$, under Assumption 2.1.3, we do know that $\max_{ij} d(z_i^\star, z_j^\star) \xrightarrow{p} 0$ for nodes i and j in any clique. Since the event that all nodes in a clique fall within δ of each other, for any $\delta > 0$, occurs with probability going to 1, we can condition on the event that nodes in a clique are at the same location. By the preceding argument, we can then conclude that $\hat{p}_{kk'} - p_{ij} \xrightarrow{p} 0$. By the continuous mapping theorem, we can then consistently estimate d_{ij}. We can now apply Theorem 2.1.1. This completes the proof.

□

Proof of Proposition 2.4.1. Before providing the proof, we provide a brief outline of our strategy. We will suppose that each $\nu_i^\star = 0$ to simplify notation, but the general result claimed in the Proposition holds. Our goal in this proof is to show that $\mathbb{P}(\mathcal{E}_\delta \mid \text{clique}) \to 1$ as $\ell \to \infty$. To make the proof easier, we will equivalently show that the ratio

$$\frac{\mathbb{P}\left(\mathcal{E}_\delta \mid clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c \mid clique\right)} \to \infty \, .$$

Showing that this ratio goes to infinity shows the numerator goes to 1 since $x/(1-x) \to \infty$ if and only if $x \to 1$. We now turn to the proof.

We have

$$\begin{aligned}
\frac{\mathbb{P}\left(\mathcal{E}_\delta \mid clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c \mid clique\right)} &= \frac{\mathbb{P}\left(clique \mid \mathcal{E}_\delta\right)}{\mathbb{P}\left(clique \mid \mathcal{E}_\delta^c\right)} \cdot \frac{\mathbb{P}\left(\mathcal{E}_\delta\right)}{1 - \mathbb{P}\left(\mathcal{E}_\delta\right)} \\
&\geq \frac{\mathbb{P}\left(clique \mid \mathcal{E}_\delta\right)}{\mathbb{P}\left(clique \mid \mathcal{E}_\delta^c\right)} \times a(\delta) \cdot \frac{1/A_n^\ell}{1 - 1/A_n^\ell} \left(1 + o\left(1\right)\right)
\end{aligned}$$

for some positive constant $a(\delta)$ by Assumption 2.4.2.

Next, since we have $L := \binom{\ell}{2}$ possible links and δ is the maximal distance between

any two nodes,

$$\mathbb{P}\left(clique \mid \mathcal{E}_\delta\right) \geq \exp\left(-L\delta\right),$$

so we have

$$\frac{\mathbb{P}\left(\mathcal{E}_\delta \mid clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c \mid clique\right)} \geq \frac{\exp\left(-L\delta\right)}{\mathbb{P}\left(clique \mid \mathcal{E}_\delta^c\right)} \times a(\delta) \cdot \frac{1/A_n^\ell}{1-1/A_n^\ell}\left(1+o\left(1\right)\right) .$$

For bounded support, $\mu_d = \mu_{d,n} := E\left\{d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z_i, z_j\right)\right\}$ is finite. Thus, by Lemma A.1.1, we have that

$$\mathbb{P}\left(clique \mid \mathcal{E}_\delta^c\right) \leq \exp\left(-L\mu_d\right).$$

Then, substituting this upper bound on $\mathbb{P}\left(clique \mid \mathcal{E}_\delta^c\right)$ into the ratio of interest we have

$$\begin{aligned}\frac{\mathbb{P}\left(\mathcal{E}_\delta \mid clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c \mid clique\right)} &\geq \frac{\exp\left(-L\delta\right)}{\exp\left(-L\mu_d\right)} \times a(\delta) \cdot \frac{1/A_n^\ell}{1-1/A_n^\ell}\left(1+o\left(1\right)\right) \\ &= \exp\left\{-L\left(\delta-\mu_d\right)\right\} \times a(\delta) \cdot \frac{1/A_n^\ell}{1-1/A_n^\ell}\left(1+o\left(1\right)\right).\end{aligned}$$

We can rewrite this as

$$\exp\left\{L\mu_d - L\delta\right\} \times \exp\left\{\log a(\delta) + \ell \log A_n^{-1} - \log\left(1-A_n^{-\ell}\right) + o\left(1\right)\right\} \tag{A.1}$$

and now need to show that this lower bound goes to infinity. We do this by appealing to Assumption 2.4.3. Specifically, by Assumption 2.4.3 the following condition is true for sufficiently large n

$$L\mu_d - L\delta \geq \ell \log A_n - \log\left(1-A_n^\ell\right) + \log a(\delta).$$

Substituting the above expression into the lower bound in (A.1) we have

$$\frac{\mathbb{P}\left(\mathcal{E}_\delta \mid clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c \mid clique\right)} \geq \exp\left\{L\mu_d - L\delta\right\} \times a(\delta) \cdot \frac{1/A_n^\ell}{1-1/A_n^\ell}\left(1+o\left(1\right)\right) \to \infty,$$

completing the proof in the case where each Ω_n is bounded.

To handle the case where Ω_n is not bounded, such as when F is a Gaussian distribution on $\mathbb{R}^p$, define an event

$$\mathcal{F}_n := \left\{z_i \in \Omega_n, \text{ for all } i = 1, \ldots, n\right\} .$$

In words, $\mathcal{F}_n$ holds whenever all z_i are in the set Ω_n. Recall that under Assumption 2.4.1(1), $\mathbb{P}(\mathcal{F}_n) = 1$, and under Assumption 2.4.1(2) there is an exponentially thin tail. This allows us to conclude that

$$\begin{aligned}\mathbb{P}(clique|\mathcal{E}_\delta^c) &= \mathbb{P}(clique|\mathcal{E}_\delta^c, \mathcal{F}_n)\mathbb{P}(\mathcal{F}_n) + \mathbb{P}(clique|\mathcal{E}_\delta^c, \mathcal{F}_n^c)\mathbb{P}(\mathcal{F}_n^c) \\ &\leq \mathbb{P}(clique|\mathcal{E}_\delta^c, \mathcal{F}_n)\mathbb{P}(\mathcal{F}_n) + \mathbb{P}(\mathcal{F}_n^c)\end{aligned}$$

Since the z_i are i.i.d., we also know that $P(F_n^c) = P(z_i \notin \Omega_n)^n := p(n)^n$. Combining this, we have that

$$\mathbb{P}(clique|\mathcal{E}_\delta^c) \leq \exp(-\mu_d L) + p(n)^n$$

Supposing that $p(n) = \exp(-bA_n^{1/p^\star})$, where again $p^\star$ is the dimension of the latent space, we see that

$$\begin{aligned}\frac{\mathbb{P}\left(\mathcal{E}_\delta \mid clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c \mid clique\right)} &\geq \frac{\exp(-L\delta)}{\exp(-L\mu_d) + p(n)^n} \times a(\delta) \cdot \frac{1/A_n^\ell}{1 - 1/A_n^\ell}\left(1 + o\left(1\right)\right) \\ &= \frac{\exp(-L\delta)}{\exp(-L\mu_d) + \exp(-nbA_n^{1/p^\star})} \times a(\delta) \cdot \frac{1/A_n^\ell}{1 - 1/A_n^\ell}\left(1 + o\left(1\right)\right)\end{aligned}$$

Now, if we can show that $\exp(-nbA_n^{1/p^\star}) \to 0$ is negligible in the limit, then the above inequality reduces to the inequality for the bounded case, for which we proved that taking ℓ to grow faster than the term in this Proposition is sufficient. Since $n >> \ell$, the term $\exp(-nbA_n^{1/p^\star})$ is negligible, which allows us to re-use the proof of the bounded case. This completes the proof.

□

Lemma **A.1.1.** *Consider any distribution of locations from which z_i are drawn i.i.d. on $\mathcal{M}^{p^\star}(\kappa^\star)$ with finite expected distance, $\mu_d := E\left\{d_{\mathcal{M}^{p^\star}(\kappa^\star)}(z_i, z_j)\right\} < \infty$. Then, if again $L = \binom{\ell}{2}$,*

$$E\left[\prod_{i<j} \exp\left\{-d_{\mathcal{M}^{p^\star}(\kappa^\star)}(z_i, z_j)\right\} |\mathcal{E}_\delta^c\right] \leq \exp\left(-L\mu_d\right).$$

Proof. We have

$$\begin{aligned}
E\left[\prod_{i<j}\exp\left\{-d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z_i,z_j\right)\right\}|\mathcal{E}_\delta^c\right] &\leq E\left[\prod_{i<j}\exp\left\{-d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z_i,z_j\right)\right\}\right] \\
&\leq \prod_{i<j}\left(E\left[\exp\left\{-L\cdot d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z_i,z_j\right)\right\}\right]\right)^{1/L} \\
&\leq \prod_{i<j}\left(E\left[\exp\left\{-L\cdot x_{ij}\right\}\right]\right)^{1/L} \\
&\leq \exp\left(-L\mu_d\right)\times\prod_{i<j}\left(E\left[\exp\left\{-L\cdot\eta_{ij}\right\}\right]\right)^{1/L} \\
&\leq \exp\left(-L\mu_d\right)\times 1
\end{aligned}$$

where we (a) unconditioned on the event since the probabilities of linking are higher within $\mathcal{E}$ than $\mathcal{E}_\delta^c$, (b) used Holder's generalized inequality, (c) used $\exp\left(a\right)^L = \exp\left(La\right)$, (d) defined $x_{ij} := d_{\mathcal{M}^{p^\star}(\kappa^\star)}\left(z_i,z_j\right)$, (e) decomposed $x_{ij} = \mu_d + \eta_{ij}$ where $\eta_{ij} := x_{ij} - \mu_d$, and (f) used the boundedness of linking probabilities. □

Proof of Example 6. Let $\Omega = \left[0, B^{1/p}\right]^p \subset \mathbb{R}^p$ so $\mathrm{vol}\left(\Omega\right) = B$ with $B = B_n$. Assume there are $C = C_n$ communities, each with m nodes, distributed uniformly at random in $\Omega' = \left[0, b^{1/p}\right]^p \subset \Omega$ with $\left|B^{1/p} - b^{1/p}\right| =: t$ and $t = \omega\left(\sqrt{\log n}\right)$. That is, the community centers—not members necessarily—reside in a subset within the space of interest with a distance between the boundaries of at least $\sqrt{\log n}$. The extra factor controls for tail events.

Given these community centers ζ_c, we have nodes distributed

$$z_i \sim F_c\left(\zeta_c, \sigma_c^2\right)$$

where F is a Gaussian distribution on $\mathbb{R}^p$ centered at ζ_c with variance σ_c^2.

Note that if $\sigma_c^2 = 0$ then we have an example of an inhomogenous lattice, and with $B = C$ this operates like Example 4 exactly. On the other hand, if $\sigma_c^2 \to \infty$ and we restrict attention only to Ω itself, then we return to the uniform case. In between lies the case of multimodal location distributions with dispersion, governed

by community centers. We will identify similar rates, mildly adjusted for tail events of extreme community or individual locations, though the bounds are not tight.

Define an event

$$\mathcal{F} := \{z_i \in \Omega, \text{ for all } i = 1, \ldots, n\}$$

and observe that the calculations conditional on $\mathcal{F}$ are identical to the lattice and point process cases. We compute

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}_\delta^c | clique\right) &= \mathbb{P}\left(\mathcal{E}_\delta^c | clique, \mathcal{F}\right)\mathbb{P}\left(\mathcal{F}\right) + \mathbb{P}\left(\mathcal{E}_\delta^c | clique, \mathcal{F}^c\right)\mathbb{P}\left(\mathcal{F}^c\right) \\
&\leq \mathbb{P}\left(\mathcal{E}_\delta^c | clique, \mathcal{F}\right)\mathbb{P}\left(\mathcal{F}\right) + \mathbb{P}\left(\mathcal{F}^c\right) \\
&\leq \exp\left(-L\mu_d\right)\left(1 + o\left(1\right)\right)
\end{aligned}$$

where μ_d is the expected distance between two points distributed in Ω from the mixture model.

To bound $\mathbb{P}\left(\mathcal{F}^c\right)$, observe that

$$\mathbb{P}\left(\max_i \text{dist}\left(z_i, \zeta_c\right) > t\right) \leq \exp\left(-\text{const.} \times t^2 + \log\left(n\right)\right) \to 0$$

by the sub-Gaussian distribution of the distance function for normals (the folded normal is sub-Gaussian), the growth assumption on t, that it holds for all communities simultaneously (which is not tight since most nodes will not be within the t-shell of the boundary due to the slow expansion of Euclidean balls). Below we calculate $t > \text{const.} \times \sqrt{LB^{1/p}\sqrt{p}}$, from which the result follows.

By the application of Lemma A.1.1, and a calculation of the expected distance which is $c_2 B^{1/p}\sqrt{p}$,

$$\mathbb{P}\left(E_\delta^c | clique, \mathcal{F}\right) \leq \exp\left(-c_2 L B^{1/p}\sqrt{p}\right),$$

and so

$$\frac{\mathbb{P}\left(\mathcal{E}_\delta | clique\right)}{\mathbb{P}\left(\mathcal{E}_\delta^c | clique\right)} = \exp\left(c_2 L B^{1/p}\sqrt{p} - L\delta\right) \cdot \frac{\mathbb{P}\left(E_\delta\right)}{1 - \mathbb{P}\left(E_\delta\right)}.$$

We therefore have

$$\ell \geq a(\delta)\frac{\log\left(\frac{B}{c_3\delta^p} - 1\right)}{B^{1/p}\sqrt{p} - \delta}\left(1 + o\left(1\right)\right) + 1$$

which gives the growth-rate bound on the clique size. Now recall the restriction on the growth rate of B relative to b. If $b = \alpha B$ for some $\alpha < 1$, then

$$t = B^{1/p} - b^{1/p} = B^{1/p}\left(1 - \alpha^{1/p}\right) = \Theta\left(B_n^{1/p}\right)$$

and so the restriction is immediate if for instance $B_n = \omega\left([\log n]^{p/2}\right)$. This admits many simple rates such as if $C = \log n$, then $m = \frac{n}{C_n} = \frac{n}{\log n}$ or $B_n = \omega\left(\left[\log\left(\frac{n}{\log n}\right)\right]^{p/2}\right)$ which is a slow poly-logarithmic growth rate. In this case the number of communities is smaller than the domain. But if $C = \sqrt{n}$, then $m = \sqrt{n}$ and therefore $B_n = \omega\left(2^{-p/2}\left(\log n\right)^{p/2}\right)$, the number of communities can grow faster than the domain. □

A.2 Generating latent space points

We now describe how we generate our points in the three latent spaces. The basic idea is to generate K group centers. We then call the first n/K nodes to be in group 1, the second n/K nodes to be in group 2, and so on. Let $c_i \in \{1, \ldots, K\}$ denote the group membership of node i. Finally, we distribute the node latent space positions centered at their group locations according to some procedure that is unique for each of the three geometries. To generate the latent space positions in the Euclidean case, we do the following:

1. Generate K group centers $\mu \in \mathbb{R}^p$ distributed according to $\mu \overset{\text{i.i.d.}}{\sim} \mathrm{N}(\mathbf{0}_p, \sigma^2 I_p)$.

2. Then simulate the positions of the nodes as $z_i | c_i \overset{\text{i.i.d.}}{\sim} \mathrm{N}\left(\mu_{c_i}, \frac{\sigma^2}{K} I_p\right)$.

To generate the latent space positions in the spherical case, we do the following:

1. Generate K group centers $\mu \in \mathbf{S}^2(\kappa)$. To do this, we generate two angles: $\theta \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(0, \pi)$ and $\phi \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(0, 2\pi)$. Then compute

$$\mu_i = \kappa^{-1/2}\left(\sin(\theta_i)\cos(\phi_i), \sin(\theta_i)\sin(\phi_i), \cos(\phi_i)\right) \in \mathbb{R}^3 .$$

2. Then simulate the positions of the nodes. To do this, generate two angles $\theta_i \sim \text{Unif}(\theta_{c_i} - \delta, \theta_{c_i} + \delta)$ and $\phi_i \sim \text{Unif}(\phi_{c_i} - \delta, \phi_{c_i} + \delta)$ and compute

$$\mu_i = \kappa^{-1/2} \left(\sin(\theta_i)\cos(\phi_i), \sin(\theta_i)\sin(\phi_i), \cos(\phi_i)\right) \in \mathbb{R}^3 \ .$$

To generate the latent space positions in the Hyperbolic case, we do the following:

1. Generate K group centers $\mu \in \mathbf{H}^2(\kappa)$. To do this, we generate two locations x_i and y_i distributed uniformly on $[-s, s] \times [-s, s]$ and select the third coordinate $z = \sqrt{1/\kappa + x_i^2 + y_i^2}$ so by construction $(x, y, z) \in \mathbf{H}^2(\kappa)$.

2. Then simulate the positions of the nodes. To do this, generate two coordinates x_i and y_i distributed uniformly on $[x_{c_i} - \delta, x_{c_i} + \delta] \times [y_{c_i} - \delta, y_{c_i} + \delta]$ then set $z_i = \sqrt{1/\kappa + x_i^2 + y_i^2}$.

We next present the parameters used for the simulations in Section 2.5. In the table below, κ is the curvature used for the Spherical geometry. The σ parameter determines the spread of the points in the Euclidean geometry. For the Hyperbolic geometry the scale refers to the scale of the first two coordinates of the space. In all of these results, we use rate $= 1/3$. We draw the node effects $\nu_i \overset{\text{i.i.d.}}{\sim} \text{Unif}(\beta, 0)$, where we set $\beta = -0.01$.

A.3 Choosing bounds for curvature estimate

We now discuss a way to pick a, the lower bound in the spherical method to pick κ. Note that the maximum distance between any two points is $r\pi = \pi/\sqrt{\kappa}$, which occurs when the points are antipodal. This shows that for a distance matrix $D = \{d_{ij}\}$, which contains distances between K points on $\mathbf{S}^p(\kappa)$, it must be that

$$\max_{1 \le i,j \le K} d_{i,j} \le \frac{\pi}{\sqrt{\kappa}} \ .$$

Table A.2.1: The parameter values used to make the results in Section 2.5. The rows correspond to the true data generating process and the columns correspond to the null hypothesis being tested.

	E	**S**	**H**
E	$\sigma = 0.5$	$\sigma = 0.8$	$\sigma = 0.8$
S	$\kappa = 0.75$	$\kappa = 1$	$\kappa = 0.75$
H	scale $= 2.5$, $\kappa = 0.75$	scale $= 2.5$, $\kappa = 0.75$	scale $= 2.5$, $\kappa = 1$

By solving for κ, we see that κ satisfies

$$\kappa \leq \left(\frac{\pi}{\max\limits_{1 \leq i,j \leq K} d_{ij}} \right)^2 := b \, .$$

Based on the discussion in [171], we set

$$a := \left(\frac{1}{3 \min_{i,j} d_{i,j}} \right)^2 .$$

The suggestion for a comes from [171], which says that for curvature values less than a, the space is essentially Euclidean. We use the same bounds for the hyperbolic case, but we flip the signs so that $[a, b] \subseteq (-\infty, 0]$. Future work could more thoroughly investigate how to pick the bounds for the hyperbolic case.

Figure 2.3.1a plots the function $\kappa \mapsto \left| \lambda_1 \left(\kappa W_\kappa \right) \right|$ when D corresponds to $K = 15$ points drawn randomly on $\mathbf{S}^2(1)$. Figure 2.3.1b plots the function $\kappa \mapsto \left| \lambda_{K-1} \left(\kappa W_\kappa \right) \right|$ when D corresponds to $K = 15$ points drawn randomly on $\mathbf{H}^2(-1)$. The functions are both minimized at the true curvature ($\kappa_0 = 1$ for the spherical case and $\kappa_0 = -1$ for the hyperbolic case).

A.4 Additional details on the bootstrap procedure

Given n independent and identically distributed data points $X_1, \ldots, X_n$ drawn from a distribution we want to estimate a parameter θ with an estimator $\hat{\theta}_n$. We make the

Algorithm 7: Hypothesis Testing via Sub-sample Bootstrap

1 Input: adjacency matrix G, sub-sample size m, and number of bootstrap samples B, and rate $r \geq 0$.

1. Compute the observed eigenvalue $\lambda_{\tilde{k}}(\hat{W}_{\hat{\kappa}}(\hat{D}))$ for $\tilde{k} = K$ for Euclidean/spherical and $\tilde{k} = 2$ for hyperbolic.

2. Construct a bootstrap distribution of eigenvalues. For $b = 1, \ldots, B$:

   (a) Sample $D_b^\star$

      i. Let $\mathbf{I} := \{I_1, \ldots, I_m\}$ and $\mathbf{J} := \{J_1, \ldots, J_m\}$ be two sets of integers of length m drawn independently and uniformly from $\{1, \ldots, \ell\}$ with replacement.

      ii. Calculate $P_b^\star$ with entries

$$p^\star_{b,kk'} = \max\left\{ \frac{1}{m^2} \sum_{i,j} G_{ij} \cdot \mathbf{1}\left\{ i \in C_k \cap \mathbf{I},\ j \in C_{k'} \cap \mathbf{J} \right\}, \frac{1}{\ell^2} \right\}.$$

      iii. Calculate $D_b^\star = -\log\left(P^\star / \hat{\tau}^2\right)$ where division is component-wise.

   (b) Calculate $W_b^\star = W_{\hat{\kappa}}\left(D_b^\star\right)$ and eigenvalue $\lambda_{\tilde{k}}(W_\kappa(D_b^\star))$.

3. Compute the CDF of the deviation in the bootstrapped and empirical eigenvalue

$$\hat{L}_n(x) := \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\left\{ m^{2r} \cdot \left(\lambda_{\tilde{k}}(W_{\hat{\kappa}}(D_b^\star)) - \lambda_{\tilde{k}}(\hat{W}_{\hat{\kappa}}(\hat{D})) \right) \leq x \right\}, \text{ for any } x \in \mathbb{R}.$$

4. Compute critical values $c_{n,1-\alpha} = \inf\left\{ x :\ \hat{L}_n(x) \geq 1 - \alpha \right\}$.

5. Test hypotheses:

   (a) Reject $H_{0,e}$ and $H_{0,s}$ when, for each of their respective test matrices,

$$\ell^{2r} \cdot \lambda_K(\hat{W}_{\hat{\kappa}}(\hat{D})) < c_{n,\alpha}.$$

   (b) Reject $H_{0,h}$ when

$$\ell^{2r} \cdot \lambda_2(\hat{W}_{\hat{\kappa}}(\hat{D})) > c_{n,1-\alpha}.$$

following assumption about $\hat{\theta}_n$, which appears in [140].

ASSUMPTION **A.4.1.** *There exists a deterministic sequence τ_n such that $\tau_n(\hat{\theta}_n - \theta)$ converges in distribution to some random variable L.*

Suppose that the goal is to construct confidence intervals for θ using $X_1, \ldots, X_n$. To do this, we select a sub-sample rate $m = m(n)$, where $m \leq n$. Then, let $Y_1, \ldots, Y_{\binom{n}{b}}$ be all the subsets of X of size b, and let $\hat{\theta}_{n,i}$ be the estimate of θ using the ith subset Y_i. Using the rate τ_n from Assumption A.4.1, with n replaced by the "sub-sample" size b, we can form the empirical CDF of $\tau_b(\hat{\theta}_{n,i} - \hat{\theta}_n)$,

$$L_n(x) := \frac{1}{\binom{n}{b}} \sum_{i=1}^{\binom{n}{b}} \mathbf{1}\Big\{\tau_b(\hat{\theta}_{n,i} - \hat{\theta}_n) \leq x\Big\} .$$

Intuitively, as n and $b \to \infty$, we expect that L_n converges to the CDF of $\tau_n(\hat{\theta}_n - \theta)$, denoted by L. If this were true, then we could use the quantiles of $\tau_b(\hat{\theta}_{n,i} - \hat{\theta}_n)$ as estimates of the quantiles of $\tau_n(\hat{\theta}_{n,i} - \hat{\theta}_n)$, which would allow us to compute confidence intervals for θ. The following result shows when we can use L_n to construct asymptotically correct confidence intervals for θ.

PROPOSITION **A.4.1** (Theorem 2, (iii) of [140])**.** *Let $c_n(1-\alpha) := \inf\{x : \hat{L}_n(x) \geq 1-\alpha\}$. Similarly, let $c(1-\alpha) = \inf\{x : L(x) \geq 1-\alpha\}$ where L is the CDF of X_1. If the CDF of X_1 is continuous at $c(1-\alpha)$ and $\tau_b/\tau_n \to 0$ and $b/n \to 0$ then*

$$\mathbb{P}\Big(\tau_n(\hat{\theta}_n - \theta) \leq c_n(1-\alpha)\Big) \to 1-\alpha .$$

This proposition allows us to construction asymptotically correct confidence intervals for θ from the sub-sampled data. Note that when n is large, computing all $\binom{n}{b}$ subsets of X is computationally infeasible, so we instead select a collection $\{Y_1, \ldots, Y_s\}$ for some integer $s \leq \binom{n}{b}$, and compute

$$\hat{L}_n(x) := \frac{1}{s} \sum_{i=1}^{s} \mathbf{1}\Big\{\tau_b(\hat{\theta}_{n,i} - \hat{\theta}_n) \leq x\Big\} .$$

According to [140], we have the following result:

PROPOSITION **A.4.2** (Theorem 2, (iii) of [140]). *Let* $c_n(1-\alpha) := \inf\{x : \hat{L}_n(x) \geq 1-\alpha\}$. *Similarly, let* $c(1-\alpha) = \inf\{x : L(x) \geq 1-\alpha\}$ *where* L *is the CDF of* X_1. *If the CDF of* X_1 *is continuous at* $c(1-\alpha)$ *and* $\tau_b/\tau_n \to 0$ *and* $b/n \to 0$, *then*

$$\mathbb{P}\left(\tau_n(\hat{\theta}_n - \theta) \leq \hat{c}_n(1-\alpha)\right) \to 1-\alpha \ .$$

This result allows us to construct confidence intervals for θ. Having described the sub-sampling method from [140], we now return to our original problem and show how to apply this method to our problem. The parameter interest θ is the eigenvalue $\lambda_{k^\star}(W)$. To study this, we will show how to use the [140] method to sub-sample the distance matrix D. Using this sub-sampled distance matrix, we can then compute sub-sampled matrices W_κ and compute their eigenvalues, since W_κ is just a simple transformation of D.

The data in our problem is the adjacency matrix G. More concretely, it is the adjacency matrix for the subgraph with nodes $\bigcup_{k=1}^K C_i(\ell)$, the union of all K cliques. We fix some sub-sample rate m. With the sub-sample rate, we then want to re-sample the entries of D. To do this, we will focus on how to do this for the (k,k') entry of D. This process is repeated for all the entries of D. Let $\tilde{G}_{k,k'}$ denote the adjacency matrix corresponding to the sub-graph induced by the nodes in $C_k(\ell) \cup C_{k'}(\ell)$. For example, if $\ell = 3$, then a potential $\tilde{Y}_{k,k'}$ might take the form

$$\tilde{G}_{k,k'} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \ .$$

This indicates that the first node in C_k connects to the first node in $C_{k,k'}$ but not to the second or third nodes in $C_{k'}$. We then sample two sets of integers of length m, denoted by I_k and $I_{k'}$, independently and uniformly from $\{1,\ldots,\ell\}$, without replacement. These indices will be the re-sampled nodes. We then compute

$$P^\star_{k,k'} = \frac{1}{m^2}\sum [\tilde{G}_{k,k'}]_{ij} \mathbf{1}\{(i,j) \in I_k \times I_{k'}\} \ .$$

Since it is possible that $P^{\star}_{k,k'}$ is zero (meaning that the re-sampled pairs of nodes do not connect), we use $P^{\star}_{k,k'} = \max(1/\ell^2, P^{\star}_{k,k'})$, since we observe at least one edge in $\tilde{G}_{k,k'}$. We repeat this procedure for all pairs of edges (k, k'). We then compute $D^{\star}$ using (2.1). We provide a step-by-step implementation of the sub-sampling method in Algorithm 7.

Recalling that our parameter of interest is the eigenvalue $\lambda_{k^\star}(W)$, we use the above procedure to compute $\lambda_{k^\star}(W^{\star}_b)$ for $b = 1, \ldots, B$. We then define

$$\hat{L}_n(x) = \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\{m^{2r}\left(\lambda_{k^\star}(W^{\star}_i) - \lambda_{k^\star}(\hat{W})\right) \leq x\}, \qquad \text{for any } x \in \mathbb{R}\,.$$

We then perform hypothesis testing. To do this, we let $c_n(1-\alpha) = \inf\{x : \hat{L}_n(x) \geq 1-\alpha\}$ be the $(1-\alpha)100\%$ percentile of $m^{2r}\left(\lambda_{k^\star}(W^{\star}_i) - \lambda_{k^\star}(\hat{W})\right)$. Then, from Proposition 2.3.1, we know that $P(m^{2r}(\lambda_{k^\star}(\hat{W}) - \lambda_{k^\star}(W)) \leq c_n(1-\alpha)) \approx 1-\alpha+o(1)$ for large ℓ. This motivates the bootstrapping method we summarize in Algorithm 7.

A.4.1 Sensitivity of bootstrapping algorithm

We now analyze the sensitivity of the sub-sampling algorithm in 7 to the parameter B, the number of matrices $D^{\star}$ we generate. To do this, we generate a 2-dimensional lattice in $\mathbb{R}^2$ of length 5 and 7 and randomly select 9 points from these C^2 points. Using these 9 points, we generate 50 networks and compute the p-values for the Euclidean, spherical, and hyperbolic geometries using $B = 1000$ and $B = 10,000$. In Figures A.4.1 and A.4.2, we plot the 50 p-values for this simulation, and a diagonal line from $(0, 0)$ to $(1, 1)$. We see that most p-values lie on or very near to the diagonal line, which indicates that the sub-sampling algorithm is not very sensitive to the choice of the parameter B.

A.5 Rank estimator

In Algorithm 8 we formally describe the algorithm and the estimate of the rank of W_κ.

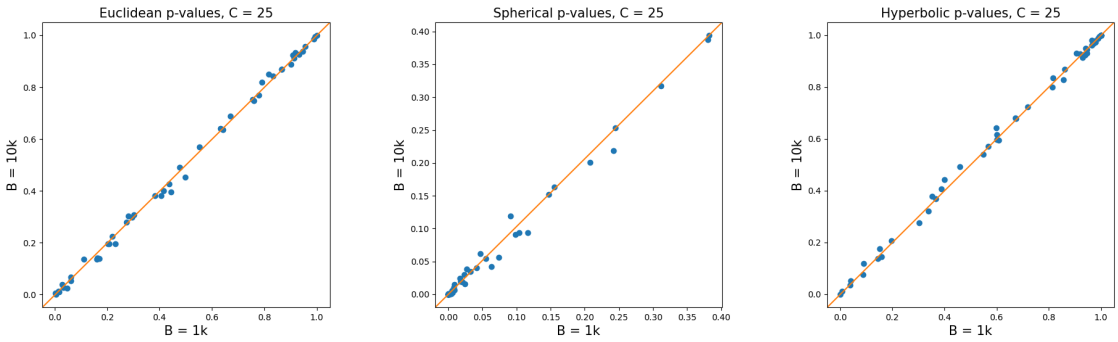


Figure A.4.1: Plot of the p-values for the three geometries (Euclidean, spherical, hyperbolic from left to right). Each point (x, y) corresponds to two p-values. The x coordinate is the p-value computed using $B = 1000$ and the y coordinate is the p-value computed using $B = 10000$. The graph is computed using the graph model in (2.1) and we use 9 latent space positions drawn randomly from a 5×5 lattice in $\mathbb{R}^2$. Since most points fall on or near the diagonal, this is evidence that in this scenario, the sub-sampling algorithm in Algorithm 7 is not very sensitive to the choice of B, the number of distance matrices we sub-sample.

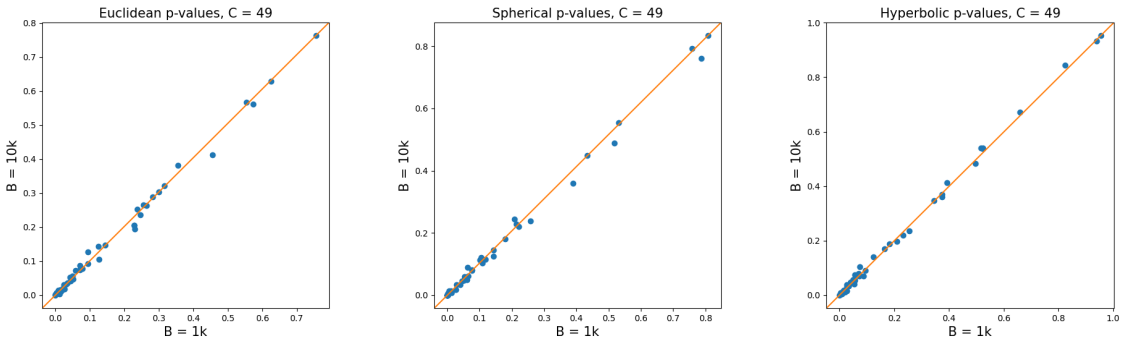


Figure A.4.2: Plot of the p-values for the three geometries (Euclidean, spherical, hyperbolic from left to right). Each point (x, y) corresponds to two p-values. The x coordinate is the p-value computed using $B = 1000$ and the y coordinate is the p-value computed using $B = 10000$. The graph is computed using the graph model in (2.1) and we use 9 latent space positions drawn randomly from a 7×7 lattice in $\mathbb{R}^2$. Since most points fall on or near the diagonal, this is evidence that in this scenario, the sub-sampling algorithm in Algorithm 7 is not very sensitive to the choice of B, the number of distance matrices we sub-sample.

The [120] estimator uses two pieces of information. The first is the scree function, which plots the sample eigenvalues in order from larges to smallest. In Figure A.5 we plot the scree function for a distance matrix computed between $K = 15$ points on a 3-dimensional Euclidean latent space. We see that the scree plot is large but decreasing for the first three eigenvalues but becomes flat after that point. The second piece of information this estimator uses is the variability of the bootstrapped eigenvectors of the matrix W_κ, given in step (4) of Algorithm 8. [120] argues that the for $j < r$, the true rank of W_κ, there is little variation in the term $f_n(j)$ in step (4) but for $j \geq r$, this function increases. We see this behavior in Figure A.5: For $j < 3$, the bootstrap variability is lower than when $j \geq 3$. See [120] for a more thorough explanation of why this phenomenon occurs. Based on these two pieces of information, [120] suggests

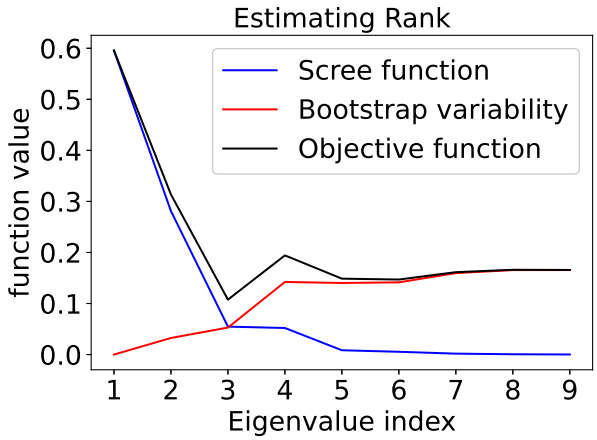


Figure A.5.1: We generate a graph using a 3-dimensional Euclidean latent space with $K = 10$ cliques. We plot the scree function ϕ and the bootstrap variability function f_n defined in Algorithm 8. We also plot their sum, defined as the objective function. The horizontal axis represents the possible ranks of the matrix. We see the objective function has a minimum at 3, so we estimate the rank of the matrix to be 3, which is the true dimension of the latent space.

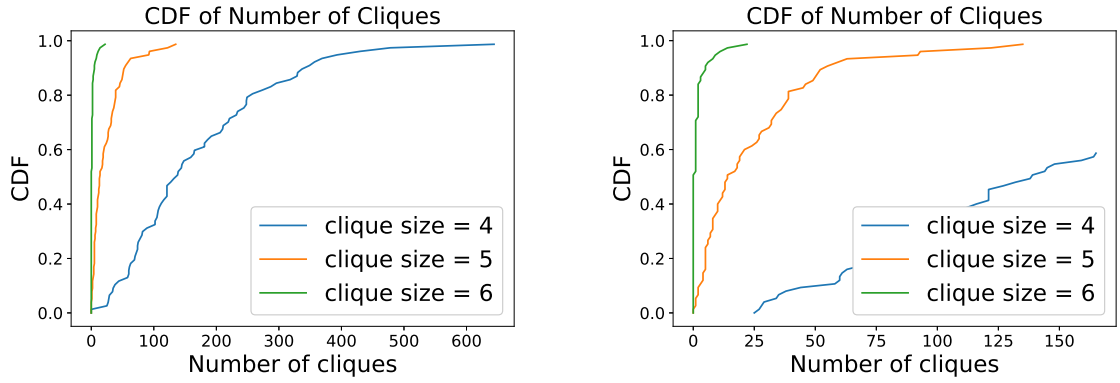
adding the two functions together to produce a final objective function. They claim that this new function has a "ladle" shape. The minimum of this new function is our estimate of the rank of W_κ.

A.6 Additional details for the data from [14]

We show cumulative distribution plots of the number of cliques (sizes 4, 5, and 6) across the 75 villages in Figure A.6.

A.7 Additional simulation results

We plot curvature estimates for 100 simulated graphs using cliques of size $\ell \in \{5, 7, 9\}$. We see that as ℓ increases, the variance and bias of $\hat{\kappa}_S$ decreases in the spherical case



(a) (b)

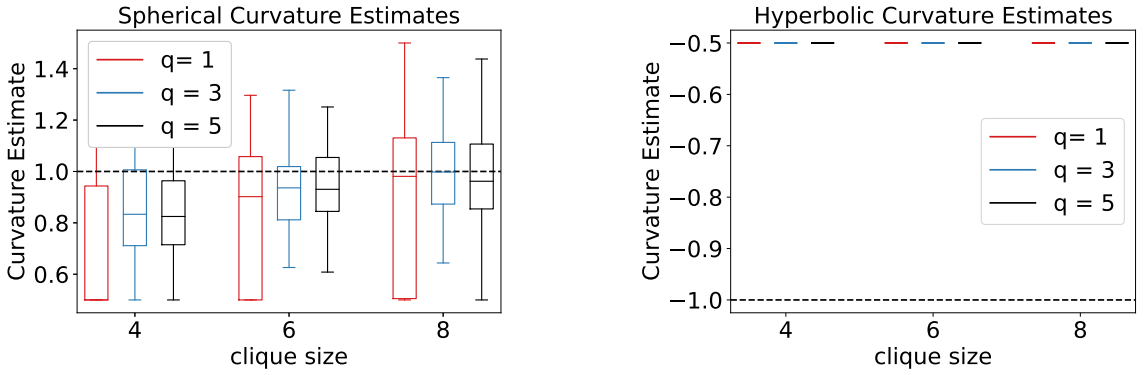
Figure A.6.1: CDF of number of cliques for clique sizes $\ell \in \{4, 5, 6\}$ for the 75 Indian villages.

(Figure A.7).

We now analyze the accuracy of the curvature methods for the spherical and hyperbolic latent space models. The estimator in Proposition 2.3.2 minimizes $\kappa \mapsto \lambda_1(\kappa \hat{W}_\kappa)$. But from Lemma 2.2.1, we in fact know that the first few eigenvalues of $\cos(\sqrt{\kappa}\hat{D})$ are zero, which suggests that we can use the estimator

$$\hat{\kappa}(q) = \frac{1}{q}\sum_{i=1}^{q} \hat{\kappa}_i, \quad \hat{\kappa}_i = \arg\min_{\kappa \in [a,b]} \left| \lambda_i \left(\kappa W(\hat{D})_\kappa \right) \right| \tag{A.3}$$

Assuming that $q << K$, we can reasonably believe that the first through qth eigenvalues of $\cos(\sqrt{\kappa}D)$ are zero. In fact, it is easy to modify the proof of Proposition 2.3.2 to show that $\hat{\kappa}(q) \xrightarrow{p} \kappa$, provided that $q << K$. Taking $q > 1$ does not always reduce the variance of $\hat{\kappa}(t)$, which could be because the $\hat{\kappa}_1, \ldots, \hat{\kappa}_q$ are not necessarily independent. In Figure A.7 we plot 250 estimates of κ when $\mathcal{M}^p(\kappa) = \mathbf{S}^2(1)$ and when $\mathcal{M}^p(\kappa) = \mathbf{H}^2(-1)$ using $K = 10$. Specifically, we generate a network and find K cliques of size 4, 5, 6. We estimate distances between these K groups using the number of cross-clique edges as described in Algorithm 3. Although Proposition 2.3.2



(a) Spherical

(b) Hyperbolic

Figure A.7.1: Left: Curvature estimates for $\mathbf{S}^2(1)$ using $K = 10$ cliques, with clique size $\ell = 4, 6, 8$ on the horizontal axis. We use $q = 1, 3, 5$ where q is defined in (A.3). We plot the true curvature $\kappa = 1$ in the black dashed line. Right: Curvature estimates for $\mathbf{H}^2(-1)$ using $K = 10$ cliques, with clique size $\ell = 4, 6, 8$ on the horizontal axis. In Figure A.7.2, we analyze how large the clique size must be for the hyperbolic curvature estimator to perform better.

says that the estimate is consistent as the sample size grows, we see that the hyperbolic curvature estimate has not reached its asymptotic behavior for cliques of size 4, 5, and 6. In Figure A.7.2 we determine how big the cliques must be in order for the Hyperbolic curvature estimator to be close to the true curvature. However, as we see in Figures 2.5.2, we see that the classification of geometry is still accurate, which is our ultimate goal.

A.8 Sectional curvature definitions

To discuss sectional curvature, some preliminary definitions are required. We review these concepts in a self-contained way. The reader may look to [134] for a more in-depth explanation of these concepts. The *tangent space* at $m \in \mathcal{M}^p$ is denoted

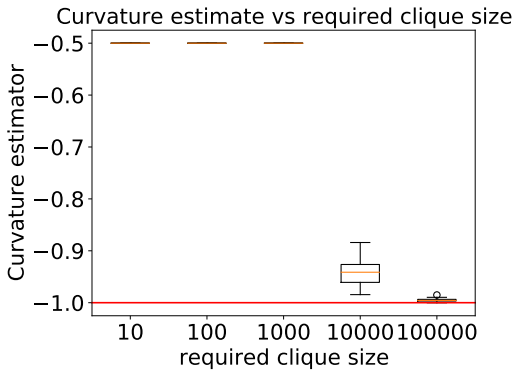


Figure A.7.2: We plot the estimated curvature using distances computed from $K = 10$ points in $\mathbb{H}^2(-1)$ for various clique sizes on the x-axis. To simuluate this figure, we fix a set of $K = 10$ locations on $H^2(-1)$ and compute pairwise distances d_{ij}. We then simulate 50 independent noisy realizations $\hat{d}_{ij} \sim N(d_{ij}, \sigma^2)$ for $\sigma \in \{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. We then compute the estimate of the curvature from (2.7) using $\hat{D} = \{\hat{d}_{ij}\}$. Given a certain noise level, we use that fact that when using cliques to estimate distances, the variance of $\hat{d}_{ij}$ is on the order of $1/\ell^2$. So we equate $\sigma^2 = 1/\ell^2$ to compute an approximate required clique sized required. For example, we require clique sizes of approximately 10^4 or higher to obtain an estimator that does not always select the lower bound a in (2.7).

$T_m(\mathcal{M}^p)$, defined as the set of all tangent vectors to the manifold at m: that is, all real-valued functions v that map any smooth function $f : \mathcal{M}^p \to \mathbb{R}$ to $v(f(m)) \in \mathbb{R}$ that is $\mathbb{R}$-linear and Leibnizian.[1]

A Riemannian manifold $(\mathcal{M}^p, g)$ comes equipped with a *metric tensor* g which at every point $m \in \mathcal{M}^p$ takes two vectors in the tangent space of the manifold at m, $u, v \in T_m(\mathcal{M}^p)$, and maps it to a non-negative number: $g_m(u, v) \mapsto \mathbb{R}_{\geq 0}$ and the map is symmetric, non-degenerate, and bilinear. That is, g defines a scalar product over the manifold; on a smooth manifold the metric tensor smoothly varies over the manifold itself.

To define curvature, we first need to define the *Riemann curvature tensor*, R evaluated at point $m \in \mathcal{M}^p$, which takes three tangent vectors in the tangent space at m—$u, v, w \in T_m(\mathcal{M}^p)$—and returns $R_m(u, v)w \in T_m(\mathcal{M}^p)$[2]

$$R_m(u, v)w := \nabla_{[u,v]}w - [\nabla_u, \nabla_v]w.$$

Here is some intuition. Consider the vector w which is tangent to the manifold at m. Consider the plane defined by u and v which are tangent at m as well. Now take w and parallel transport it, meaning take it along the parallelogram in the u direction and then v direction and compare that to taking the same w along the v direction and then u direction to the same point. The returned vector has entries that describe how much w changes relatively across the two paths. If this is identically zero, this means of course that there was no change in this parallel transportation. Intuitively, if one does this on a flat manifold, for instance $\mathbb{R}^2$ with the usual Euclidean metric, it is clear that the vector w does not change whatsoever. But on a sphere, for instance, the reader can intuit that things change.

Then the *sectional curvature* at m, which we refer to simply as curvature for the

[1] An obvious tangent vector is the directional derivative at a point on the manifold: it maps a smooth function to its derivative in that direction evaluated at that point on the manifold.

[2] Here $[.,.]$ is the Lie bracket.

remainder of this chapter, is given by

$$\kappa_m(u,v) := \frac{g_m(R_m(u,v)v,u)}{g_m(u,u)\cdot g_m(v,v) - g_m(u,v)^2}.$$

It turns out that this is independent of basis u, v whatsoever (see Lemma 39 in [134] for instance) so we can simply write κ_m. That the manifold has constant sectional curvature means that for all $m \in \mathcal{M}^p$, $\kappa_m = \kappa$ and so we simply write $\mathcal{M}^p(\kappa)$.

A.9 Lattice simulations

We now demonstrate the type 1 error and power simulations by drawing points on a lattice in Euclidean, spherical, and hyperbolic space. We present the results in Figures A.9.1-A.9.6.

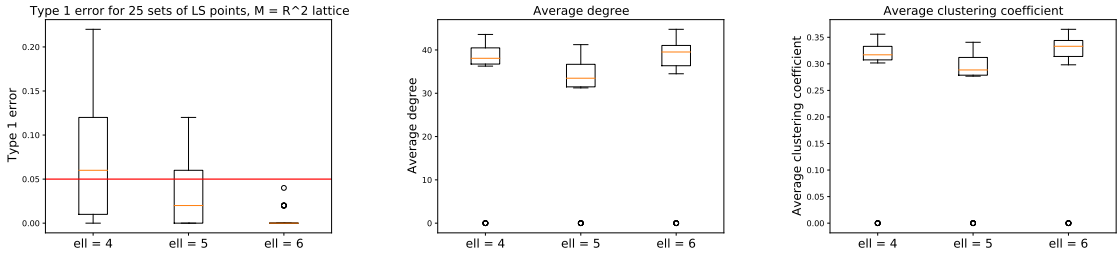


Figure A.9.1: Simulation results for the two-dimensional lattice. Using a 4×4 lattice in $\mathbb{R}^2$, we randomly select 25 sets of 9 latent space positions. For each set of positions, we generate 50 networks from using the graph model in (2.1) and calculate how many of these 50 networks we reject the null hypothesis that $\mathcal{M}$ is Euclidean. We repeat this for all 25 sets of latent space positions and plot the resulting probability of type 1 error for $\ell = 4, 5, 6$. We see that the type 1 error is at $\alpha = 0.05$ or below and decreases as ℓ increases. We also report the average degree (middle figure) and average clustering coefficient for the simulated networks. We use $\tau = 0.4$ and $\beta = -0.6$.

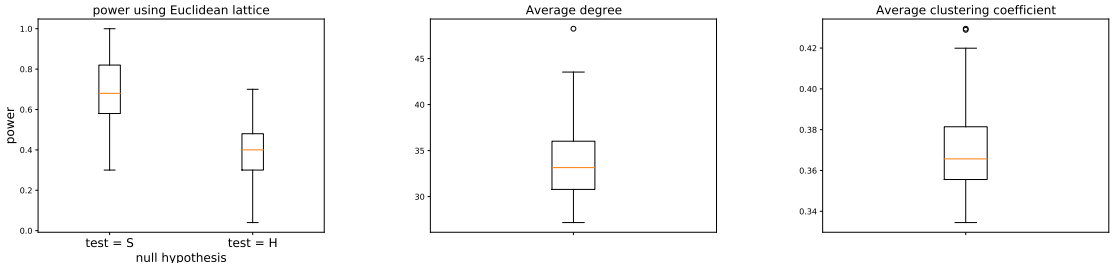


Figure A.9.2: Simulation results for the two-dimensional lattice. Using a 4×4 lattice in $\mathbb{R}^2$, we randomly select 25 sets of 9 latent space positions. For each set of positions, we generate 50 networks from using the graph model in (2.1) and calculate how many of these 50 networks we reject the null hypothesis that $\mathcal{M}$ is spherical or hyperbolic. We repeat this for all 25 sets of latent space positions and plot the resulting power. We also report the average degree (middle figure) and average clustering coefficient for the simulated networks. We use $\tau = 0.4$.

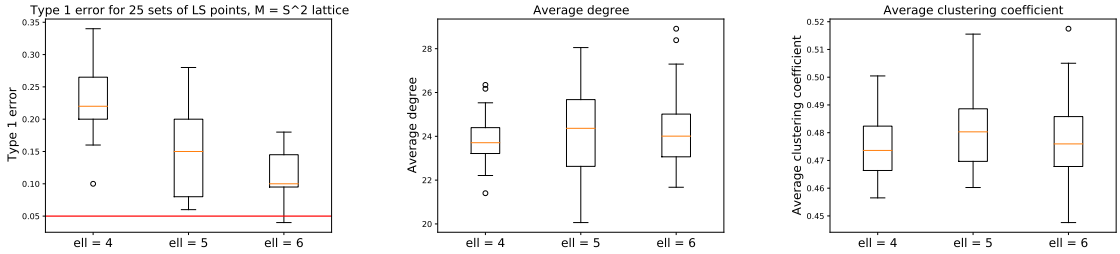


Figure A.9.3: Simulation results for the two-dimensional lattice. Using a 3×3 lattice in $\mathbb{S}^2(1)$, we randomly select 25 sets of 4 latent space positions. For each set of positions, we generate 50 networks from using the graph model in (2.1) and calculate how many of these 50 networks we reject the null hypothesis that $\mathcal{M}$ is Euclidean. We repeat this for all 25 sets of latent space positions and plot the resulting probability of type 1 error for $\ell = 4, 5, 6$. We see that the type 1 error is above $\alpha = 0.05$ but decreases to about 0.1 as ℓ increases. We also report the average degree (middle figure) and average clustering coefficient for the simulated networks. We use $\beta = -0.2$.

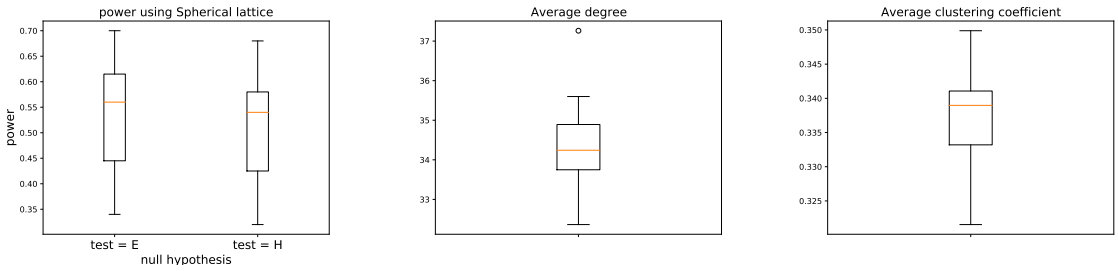


Figure A.9.4: Simulation results for the two-dimensional lattice. Using a 5×5 lattice in $\mathbb{S}^2$, we randomly select 25 sets of 9 latent space positions. For each set of positions, we generate 50 networks from using the graph model in (2.1) and calculate how many of these 50 networks we reject the null hypothesis that $\mathcal{M}$ is Euclidean. We repeat this for all 25 sets of latent space positions and plot the resulting power. We also report the average degree (middle figure) and average clustering coefficient for the simulated networks. We use $\beta = -0.2$.

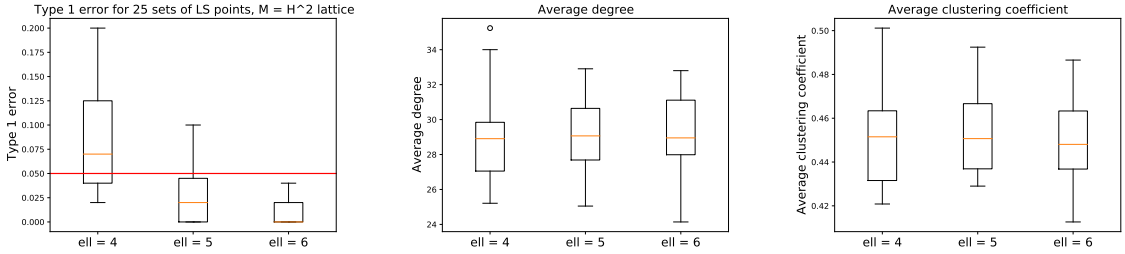


Figure A.9.5: Simulation results for the two-dimensional lattice. Using a 5×5 lattice in $\mathbb{H}^2(-1)$, we randomly select 25 sets of 9 latent space positions. For each set of positions, we generate 50 networks from using the graph model in (2.1) and calculate how many of these 50 networks we reject the null hypothesis that $\mathcal{M}$ is Euclidean. We repeat this for all 25 sets of latent space positions and plot the resulting power. We also report the average degree (middle figure) and average clustering coefficient for the simulated networks. We use $\beta = -0.2$.

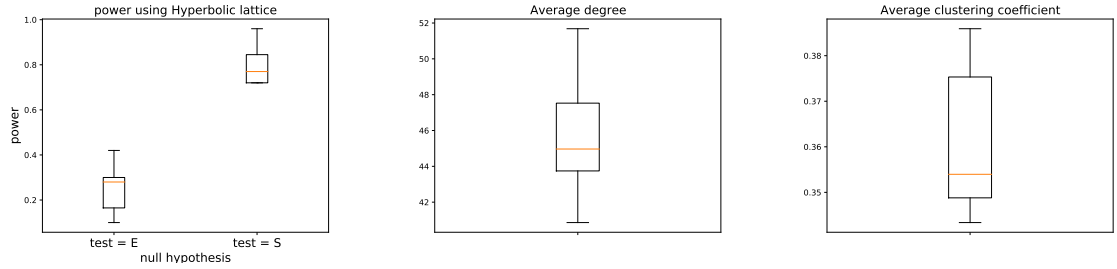


Figure A.9.6: Simulation results for the two-dimensional lattice. Using a 5×5 lattice in $\mathbb{H}^2(-1)$, we randomly select 25 sets of 9 latent space positions. For each set of positions, we generate 50 networks from using the graph model in (2.1) and calculate how many of these 50 networks we reject the null hypothesis that $\mathcal{M}$ is Euclidean. We repeat this for all 25 sets of latent space positions and plot the resulting power. We also report the average degree (middle figure) and average clustering coefficient for the simulated networks. We use $\beta = -0.2$.

A.10 Other graph models

In (2.1), we consider an exponential link function which connects distances in the latent space to the probability that nodes form an edge. The exponential function has the desirable property that $\exp(a+b) = \exp(a)\exp(b)$ which allows us to isolate the node effects and distances:

$$\mathbb{P}(g_{ij} = 1 | \nu_i^\star, \nu_j^\star, z_i^\star, z_j^\star) = \exp(\nu_i^\star + \nu_j^\star) \exp\{-d(z_i^\star, z_j^\star)\} \, .$$

So by integrating out the node effects, we see that

$$\mathbb{P}(g_{ij} = 1 | z_i^\star, z_j^\star) = E\{\exp(\nu)\}^2 \exp\{-d(z_i^\star, z_j^\star)\} \, .$$

We now illustrate how our approach can be applied to other graph models.

Example **7** (Expit link function)**.** *Suppose instead of the exponential link function, we use the expit link function, which was used in [82], among many others:*

$$\mathbb{P}(g_{ij} = 1 | \nu_i^\star, \nu_j^\star, z_i^\star, z_j^\star) = \mathit{expit}\{\nu_i^\star + \nu_j^\star - d(z_i^\star, z_j^\star)\} \, ,$$

where $\text{expit}(x) = \exp(x)/\{1+\exp(x)\}$ *is the expit function.*

The choice of which link function to choose is an important one, which network goodness-of-fit tests can help address [136, 172, 118].

Now, after integrating out the node effects, we have

$$\mathbb{P}(g_{ij} = 1 | \nu_i^\star, \nu_j^\star, z_i^\star, z_j^\star) = \int\int \text{expit}\{\nu_i^\star + \nu_j^\star - d(z_i^\star, z_j^\star)\} dF(\nu_i^\star) dF(\nu_j^\star) \ ,$$

which again assumes that the node effects are drawn independently of each other. Let us define a function $H : [0,\infty) \to [0,\infty)$ with

$$H(x) := \int\int \text{expit}\{\nu_i^\star + \nu_j^\star - x\} dF(\nu_i^\star) dF(\nu_j^\star) \ .$$

Using Monte Carlo integration, we can approximate H to within any desired certainty. Now, suppose that C_k and $C_{k'}$ are two cliques in the graph G drawn from this graph model. Then,

$$\hat{p}_{kk'} := \frac{1}{\ell^2} \sum_{i \in C_k} \sum_{j \in C_{k'}} G_{ij} \approx H\{d(z_k, z_{k'})\} \ .$$

We can then solve for the argument of H that solve the above expression. That is, assuming that H is invertible, we can write $\hat{d}_{kk'} = H^{-1}(\hat{p}_{kk'})$. Assuming H^{-1} is continuous, we can estimate distances consistently.

EXAMPLE 8 (Latent space model with covariates). *Consider the model given in [82]:*

$$\mathbb{P}(g_{ij} = 1 | \alpha^\star, \beta^\star, z_i^\star, z_j^\star) = \textit{expit}\{\alpha^\star + \beta^\star X_{ij} - d(z_i^\star, z_j^\star)\} \ .$$

where $\alpha^\star$ *measures the baseline probability of connecting,* X_{ij} *is a dyad-level covariate,* $\beta^\star$ *measures the effect of this covariate on the probability of edges, and* d *measures distances in the latent space.*

To illustrate how estimate parameters in this model, suppose that X_{ij} is measuring homophily, so that X_{ij} is 1 if nodes i and j share some common trait (like ethnicity, education level, political beliefs, etc) and is 0 otherwise. Suppose also that covariates are observable.

Suppose we observe a clique of nodes with the same trait. Then,

$$\frac{1}{\binom{\ell}{2}} \sum_{i<j} G_{ij} \approx \text{expit}(\alpha^\star + \beta^\star) \ , \tag{A.4}$$

since nodes in the same clique are likely to be close to each other and therefore $d(z_i^\star, z_j^\star) = 0$ for i and j in the same clique (Assumption 2.1.3).

Suppose we also observe an ℓ-clique $C(\ell)$ with nodes that have different traits. Partition these nodes into two groups, $C_0(\ell)$ and $C_1(\ell)$ where the subscript 0 and 1 indicates whether the covariate is 0 or 1, so that $C(\ell) = C_0(\ell) \cup C_1(\ell)$. Then,

$$(|C_0(\ell)||C_1(\ell)|)^{-1} \sum_{i \in C_0(\ell)} \sum_{j \in C_1(\ell)} G_{ij} \approx \text{expit}(\alpha^\star) \ .$$

Thus, we can estimate $\alpha^\star$ by solving the above equation. Using this estimate, we can then plug this value into (A.4) and solve for $\hat{\beta}$.

Given estimates of $\alpha^\star$ and $\beta^\star$, we can now estimate distances between cliques. To do this, we define for any two cliques C_k and $C_{k'}$, where all nodes have the same traits,

$$\hat{P}_{kk'} := (|C_k(\ell)||C_{k'}(\ell)|)^{-1} \sum_{i \in C_0(\ell)} \sum_{j \in C_1(\ell)} G_{ij} \ .$$

where $C_k(\ell)$ and $C_{k'}(\ell)$ are cliques of size ℓ. We can then solve for $\hat{D}_{kk'}$,

$$\hat{D}_{kk'} = \text{logit}(\hat{P}_{kk'}) - \hat{\alpha} - \hat{\beta} \ .$$

A.11 Existence of cliques and locations of nodes in a clique

In Theorem 2.1.2, we used edges between cliques to estimate distances between points on the unknown latent surface $\mathcal{M}$. Theorem 2.1.2 then states that as the graph size n and the clique size ℓ both go to infinity, we can consistently estimate all parameters of the latent space graph model. In particular, we have assumed that as ℓ, n both go to infinity, that we observe cliques of size ℓ. To understand the rate at which ℓ, n can grow, we consider a simplifying example. Let G be an undirected random graph

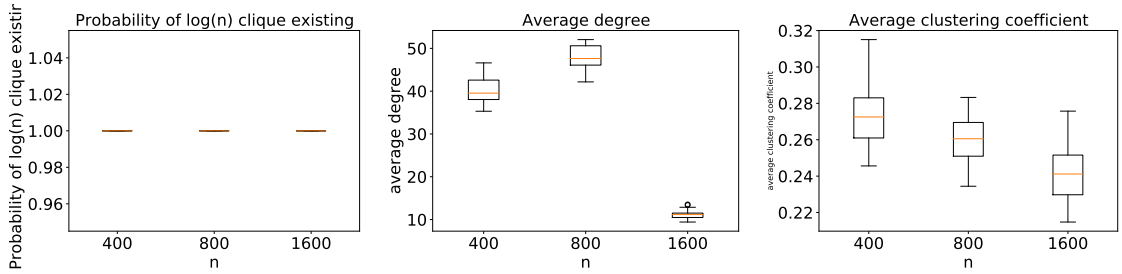
drawn from the Erdos-Renyi model with parameters n and p. That is, edges form independently with probability p. The following result is a well-known result which says that in an ER graph, the clique number grows like $\log(n)$ for large n. It can be found in many places, such as in [75].

PROPOSITION **A.11.1** ([75]). *The clique number of an ER model $Z_{n,p}$ satisfies*

$$\frac{Z_{n,p}}{\log(n)} \to \frac{2}{\log(1/p)} .$$

almost surely.

In other words, the clique number $Z_{n,p}$ grows like $C\log(n)$ for the constant $C = 2\{\log(1/p)\}^{-1}$. So by taking $\ell = C\log(n)$, we will almost surely see an ℓ clique in the graph as $n, \ell \to \infty$. Now let us return to our problem. We observe a graph drawn from (2.1), where the node locations and effects are drawn iid from two distributions. Clearly, the probability of observing an $\ell = \ell(n)$ clique depends on the distributions of the points and node effects. To our knowledge, there are no results like Proposition A.11.1 that hold for arbitrary graph models. However, we would like to investigate the behavior of the clique number for three common ways of assigning points in the latent space. In particular, we want to determine if there are cliques of size $\log(n)$ in graphs where the node locations are drawn according to a lattice mode, Gaussian mixture model, and uniform model. These three models are the models we study in Section 2.4.4. We give these results in Figures A.11, A.11, and A.11. We see cliques of size $\log(n)$ with probability going to 1 as $n \to \infty$. The models are listed in increasing "difficulty," meaning that it should be less likely for there to be clique when points are uniformly drawn (C) than when they are drawn from a GMM (B). And, when nodes are at the same location (A), it should be even more likely that a clique exists. We see this pattern in our simulations; for any n, the GMM in (B) has a higher probability of containing an $\log(n)$ clique for every n. But, for sufficiently large n, all models contain cliques of size at least $\log(n)$ with relatively high probability.



(a) Estimated probability of $\log(n)$ clique (b) Average degree (c) Average clustering coefficient

Figure A.11.1: We generate 25 sets of n latent space positions using the lattice model. For each set of LS positions, we generate 50 graphs and count the number of times a clique of size $\log(n)$ exists in the graph. For each set of LS positions, we record the average degree for the 50 graphs and plot the 25 average values in (B). Similarly, in (C) we plot the average clustering coefficient.

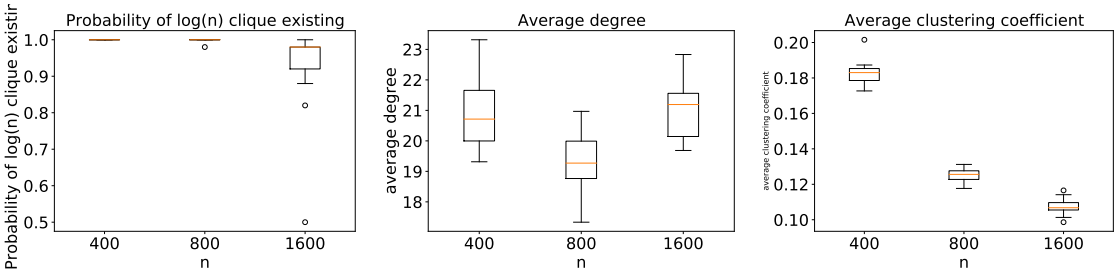
We also verify in Figure A.11 that as the size of a clique goes up, the probability that node locations in the latent space are close to each increases

A.12 Testing geometry using the Cayley-Menger determinant

For $K := 4$ points with pairwise distances $D = \{d_{ij}\}$, define the *Cayley-Menger* determinant to be the determinant of the matrix

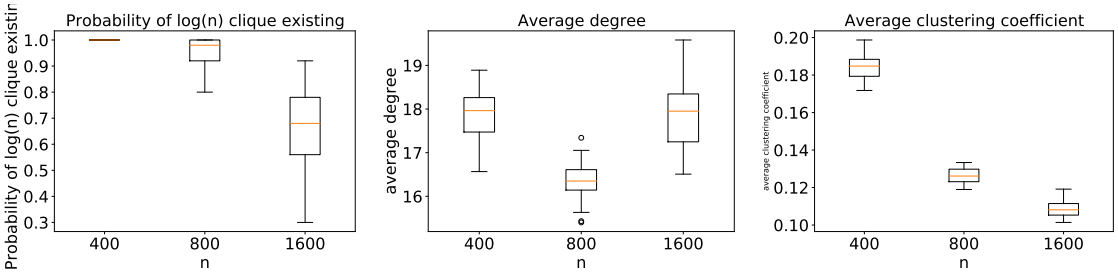
$$CM := \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & d_{12}^2 & d_{13}^2 & d_{14}^2 \\ 1 & d_{21}^2 & 0 & d_{23}^2 & d_{24}^2 \\ 1 & d_{31}^2 & d_{32}^2 & 0 & d_{34}^2 \\ 1 & d_{41}^2 & d_{42}^2 & d_{43}^2 & 0 \end{pmatrix}.$$

When these points are in $\mathbb{R}^2$, then the determinant of the matrix CM is 0. The goal of this section is to derive an estimator of geometry that uses this idea. Similar results



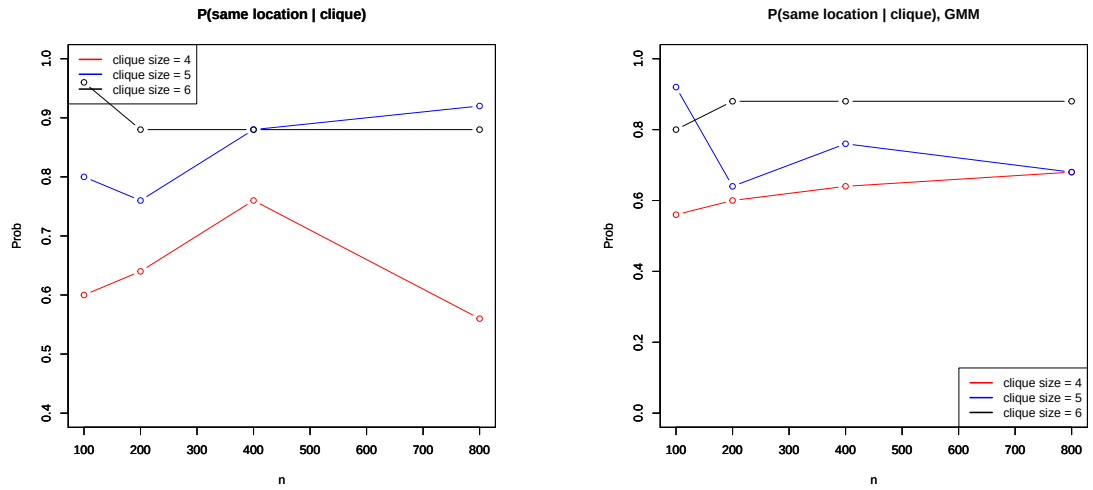
(a) Estimated probability of $\log(n)$ clique (b) Average degree (c) Average clustering coefficient

Figure A.11.2: We generate 25 sets of n latent space positions using the Gaussian mixture model. For each set of LS positions, we generate 50 graphs and count the number of times a clique of size $\log(n)$ exists in the graph. For each set of LS positions, we record the average degree for the 50 graphs and plot the 25 average values in (B). Similarly, in (C) we plot the average clustering coefficient.



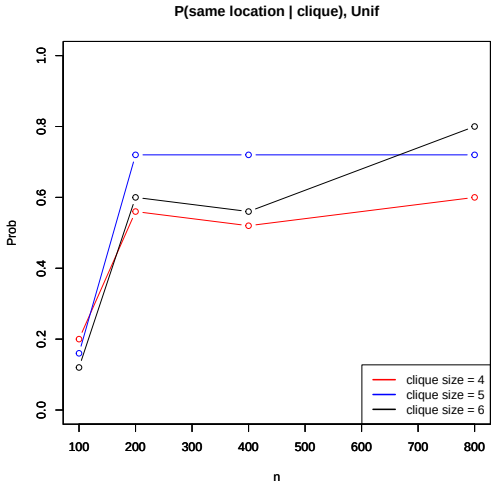
(a) Estimated probability of $\log(n)$ clique (b) Average degree (c) Average clustering coefficient

Figure A.11.3: We generate 25 sets of n latent space positions from a uniform distribution. For each set of LS positions, we generate 50 graphs and count the number of times a clique of size $\log(n)$ exists in the graph. For each set of LS positions, we record the average degree for the 50 graphs and plot the 25 average values in (B). Similarly, in (C) we plot the average clustering coefficient.



(a)

(b)



(c)

Figure A.11.4: We generate 25 networks on n nodes for $n \in \{100, 200, 400, 800\}$. Using $C = \log(n)$, we generate C communities in a lattice model (A), a GMM (B), or a uniform model (C). We check how many times out of 25 nodes in an arbitrary clique are at the same location. We plot the corresponding probabilities above for clique sizes 4, 5, 6.

hold for points from a spherical space or hyperbolic space, with slightly modified matrices.

We now propose a test of geometry using this method. To ensure that this is a reasonable comparison to our previous results, we generate a stochastic block model with locations $z_1, \ldots, z_K$, and assign edges based on $P(G_{ij} = 1|z_i, z_j) = \exp(-d(z_i, z_j))$. These locations correspond to K cliques that we might find in a network. We then compute the determinant of $\widehat{CM}$, which is defined as above except that we use $\hat{d}_{ij}^2$ in place of d_{ij}^2. Since the distribution of the random variable under $H_0 : CM = 0$ is hard to derive, let us do a parametric bootstrap. For $b = 1, \ldots, B$, we repeat the following steps:

1. For each $i < j$, draw with replacement ℓ^2 edges from the set of edges that exist between locations z_i and z_j. This is the re-sampling step.

2. Compute the average number of re-sampled edges between i and j for any $i < j$. Call this number $p_{ij}^\star$ and set $d_{ij}^\star = -\log(p_{ij}^\star)$.

3. Compute $CM_b^\star = \det(CM^\star)$, where $CM^\star$ is the matrix above with d_{ij} now replaced by $d_{ij}^\star$.

If 0 falls outside of the interval

$$(2\det(\widehat{CM}) - q_{1-\alpha/2}\{\det(CM)^\star\}, 2\det(\widehat{CM}) - q_{\alpha/2}\{\det(CM)^\star\})$$

then we reject H_0, where $q_{\alpha/2}\{\det(CM)^\star\}$ is the $\alpha/2$ quantile of the empirical distribution $\{\det(CM)^\star\}$.

We plot the power of this method using spherical data in Figure A.12.1. We see that when we use $K = p + 2$, then the method works its best, but it still performs relatively poorly. So this seems to suggest that only at this value of K does it work well.

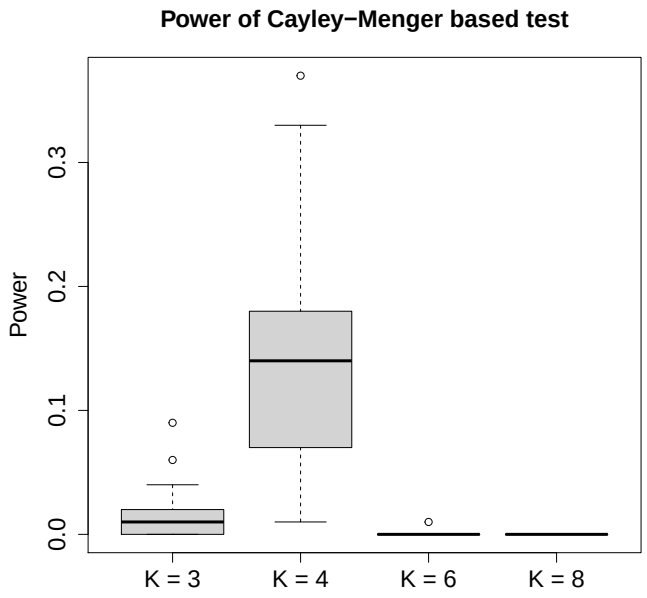


Figure A.12.1: Power of the Cayley-Menger based test of the Euclidean hypothesis. On the x-axis we plot the number of points K that were used. On the y-axis we plot the average number of rejections (out of 100) for 25 sets of K locations drawn from the sphere $\mathbf{S}^2(1/2)$.

Algorithm 8: Estimating Rank of W_κ

1. Compute the scree function $\phi_T(j) := \frac{\hat{\lambda}_{K-j-1}}{\sum_{i=1}^{K} \hat{\lambda}_i}$ for $j \in \{0, 1, \ldots, K-1\}$.

2. Sample B bootstrapped $D_1^\star, \ldots, D_B^\star$ matrices from Algorithm 7 and use them to compute $W_1^\star, \ldots, W_B^\star$.

3. For $j \in \{0, 1, \ldots, K-1\}$,

   (a) Define $\hat{A}_j \in \mathbb{R}^{K \times j}$ with $\hat{A}_j = (\hat{v}_{K-j+1}, \ldots, \hat{v}_K)$.

   (b) Let $v_1^\star, \ldots v_K^\star$ denote the eigenvectors of $W_i^\star$ corresponding to its eigenvalues $\lambda_1^\star \leq \ldots \leq \lambda_K^\star$.

   (c) Set $A_{j,i}^\star \in \mathbb{R}^{K \times j}$ with $A_{j,i}^\star = (v_{K-j+1}^\star, \ldots, v_K^\star)$.

4. Compute
$$f_n^0(j) = 1 - \frac{1}{B} \sum_{i=1}^{B} |\det(\hat{A}_j^T A_{j,i}^\star)|$$

5. Compute
$$f_n(j) = \frac{f_n^0(j)}{\sum_{i=0}^{K-1} f_n^0(i)} \, .$$

6. The estimate $\hat{r}$ of the rank of W_κ is
$$\hat{r} = \underset{j \in \{0,1,\ldots,K-2\}}{\arg\min} \; (\phi_T(j) + f_n(j)) \; . \tag{A.2}$$

Appendix 2

APPENDIX FOR CHAPTER 3

This chapter contains the supplementary materials for Chapter 3

We now outline the main parts of the supplementary materials. In Section B.1, we provide proofs of Theorems 3.2.1 and 3.3.1 in the main chapter, which deal with consistency in the beta-model and the stochastic block model (SBM), respectively. We then move to proving Theorem 3.4.1, which deals with consistency in the latent space model. First, Section B.2 defines the estimates of the node locations and effects, and in Section B.2.1, we prove Theorem 3.4.1 in the main chapter, which deals with the consistency of the estimates of the node locations and effects. The proof of Theorem 3.4.1 relies on proving consistency of the estimates of the global parameters, which we do in Section B.2.2. Section B.2.3 discusses the assumptions made in Theorem 3.4.1 in the main chapter and demonstrates that several conventional distributions used in the literature satisfies these assumptions. Section B.3 contains the proof of Theorem 3.5.1 in the main chapter. Section B.4 provides proofs of the other theorems in the main chapter. Section B.5 contains the proof of Theorem 3.5.2 and Section B.6 contains the proof of Theorem 3.5.3. Sections B.7 and B.8 provide additional simulations. Section B.9 provides simulations to verify the consistency of the claims made in Theorem 3.4.1. Section B.10 contains additional lemmas and results we use in the supplementary materials.

In the proofs, we use C to refer to constants or sequences of constants that can change from line to line, but critically these constants never depend on the graph size n nor the number of nodes with trait k, n_k.

B.1 Consistency of beta-bodel and SBM parameters (Theorems 3.2.1 and 3.3.1)

We begin with the beta-model. Before providing specifics, we first introduce the main ideas of the proof of Theorem 3.2.1, which shows that the estimators, computed using just ARD, proposed in [73] are consistent for the parameters of the beta-model. To do this, we first recall that [73] proposes a fixed point estimator $\hat{\nu}_i$ that satisfies $\hat{\nu}_i(t+1) = \phi(\hat{\nu}_i(t))$ for some known function ϕ, which depends only on the degree sequence. They also propose a consistent estimator of the parameter β, which also only depends on the degree of the nodes. Since ARD allows us to recover the degree of nodes in the survey, we can then directly apply the results of [73] to conclude Theorem 3.2.1. Before getting to the proof of Theorem 3.2.1, we now re-state Theorem 3 of [73], which we use in our proof of Theorem 3.2.1.

PROPOSITION **B.1.1** (Theorem 3 of [73]). *The fixed point estimator, as described in equations 17-18 of [73], satisfies*

$$\max_{1\leq i\leq \hat{n}} |\hat{\nu}_i - \nu_i^\star| \leq C\sqrt{\frac{\log(n)}{n}}$$

with probability $1 - O(1/n^2)$ *for some constant* $C > 0$. *In addition, we have that* $\hat{\beta} \xrightarrow{p} \beta$ *as* $n \to \infty$.

Proof of Theorem 3.2.1. In the case of mutually exclusive and exhaustive traits, $d_i = \sum_{k=1}^{K} y_{ik}$. Since the fixed point estimation procedure proposed in [39, 73] depends only on the degree of each node, which we are able to estimate with ARD, we can then apply Theorem 3 of [73] to conclude Theorem 3.2.1 of the main chapter. Theorem 3 of [73] requires several conditions (Conditions 1, 2, 3, and 5 of [73]), which are all satisfied under the assumptions of Theorem 3.2.1 of the main chapter.

□

We now give a brief overview of the proof of Theorem 2. The intuition is that the the ARD responses $\tilde{y}_i = (y_{i1}/n_1, \ldots, y_{iC}/n_c)$ converge, by the weak law of large

numbers, to $Z_i = (\tilde{P}_{i1}, \ldots, \tilde{P}_{iC})$ at an exponentially fast rate in n. See Figure B.1.1 for an illustration of this fact. Therefore, two nodes in the same community will be classified together with probability going to 1, and since the by assumption the Z_i are distinct, two nodes in different communities will eventually be classified into different communities. We want to emphasize again the differences between the problem we are studying here and classic clustering problems or community detection problems. Compared to classic clustering problems, in which the distribution of data does not change as the sample size grows, the data we are analyzing here, y_{ik}/n_k, is converging to its expectation at an exponentially fast rate. Therefore, as our sample size grows, it becomes easier to correctly cluster the ARD responses and therefore to correctly classify nodes into the right communities. Second, compared to more standard community detection problems, we do not observe the graph but instead observe ARD about the nodes [22]. This ARD, because it is a sample average, converges exponentially fast to its mean, which allows us to perform fast community detection.

Proof of Theorem 3.3.1. To begin, we pick a node randomly from V. Let c_i denote its community membership. For any j, since y_{jk}/n_k is a sum of (conditionally) independent random variables, by Hoeffding's inequality we have that $\mathbb{P}(|y_{jk}/n_k - p_{jk}| > \epsilon_n) \leq C\exp(-\epsilon_n^2 n)$ for some constant C. By recalling that $\tilde{y}_i = (y_{i1}/n_1, \ldots, y_{iC}/n_C)$ is the normalized ARD response with mean $\tilde{p}_i = (\tilde{P}_{i1}, \ldots, \tilde{P}_{iC})$, we can conclude by a union bound that

$$\mathbb{P}(\max_{j:c_j=c_i} ||\tilde{y}_j - \tilde{p}_j|| > \epsilon_n) \leq nC\exp(-\epsilon_n^2 n) \; .$$

By taking $\epsilon_n^2 = \log(n)/n$, we see that $\mathbb{P}(\max_{j:c_j=c_i} \mathbf{1}\{\hat{c}_j \neq \hat{c}_i\} > 0) \leq 1/n$. In addition, since $\Delta > 0$, which gives us well-separated clusters, and $\epsilon_n \to 0$, we have that $\mathbb{P}(\max_{j:c_j\neq c_i} \mathbf{1}\{\hat{c}_j \neq \hat{c}_i\}) \to 1$ for any j with $c_j \neq c_i$. By definition of the classification algorithm, we can conclude that $\mathbb{P}(\max_{j:c_j=c_i} \mathbf{1}\{\hat{c}_j \neq \hat{c}_i\} > 0) \leq \mathbb{P}(\max_{j:c_j=c_i} ||\tilde{y}_j - \tilde{p}_j||)$.

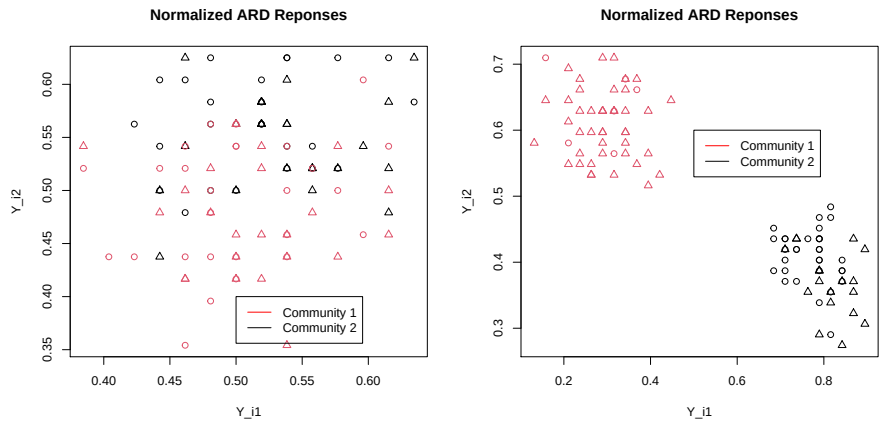


Figure B.1.1: Comparison of ARD responses in two different scenarios. On the left, we generate traits using the matrix $Q = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$. In this case, traits have no relationship with the community membership. In the left figure, we plot the normalized ARD responses, Here red indicates community 1, black indicates community 2, circles indicate trait 1, and triangles indicate trait 2. On the right, we repeat the simulation but using $Q = \begin{pmatrix} 7/10 & 3/10 \\ 1/10 & 9/10 \end{pmatrix}$. Here, there is a strong relationship between traits and community membership, and so K-means returns the correct clustering of the data.

Since the algorithm assigns nodes j that are within ϵ_n away from i into the same category, we see that the probability of any incorrect classification goes to zero for this community. The same argument applies to the second community, when looking at the set $V \setminus \hat{C}_i$. We then repeat this argument until all nodes are classified.

Given a consistent estimate of the community membership vector, it follows from the weak law of large numbers that $\hat{Q}, \hat{P}$, and $\hat{\pi}$ are consistent for Q, P and π, where

$$\hat{Q}_{ck} = \frac{1}{m_c(n)} \sum_{i \in \hat{C}_c} \mathbf{1}\{t_i = k\}$$

$$\hat{P}_{cc'} = \begin{cases} \frac{1}{m_c(n) m_{c'}(n)} \sum_{i \in \hat{C}_c} Y_{ic'}, & c \neq c' \\ \frac{1}{m_c(n)(m_{c'}(n)-1)} \sum_{i \in \hat{C}_c} Y_{ic'}, & c = c' \end{cases},$$

and $\hat{\pi}_c = \frac{1}{m_c(n)} \sum_{i=1}^{n} \mathbf{1}\{\hat{c}_i = c\}$, and $m_c(n)$ is the number of nodes that we estimate to be in community c under the estimated community membership vector $\hat{\mathbf{c}}$.

□

B.2 Consistency of latent space model parameters (Theorem 3.4.1)

We now define the estimates of the node locations and the node effects. In the estimates provided below, we assume that we have estimates of the global parameters, which we denote by $\eta^\star = (\mu_1^\star, \ldots, \mu_K^\star, \sigma_1^\star, \ldots, \sigma_K^\star, E\{\exp(\nu^\star)\})$. In Section B.2.2, we provide estimates of $\eta^\star$ based on method-of-moment estimators.

Recall that the ARD data y_{ik} satisfies $y_{ik} \mid \nu_i^\star, z_i^\star, \eta^\star \sim \text{Binomial}(n_k, p_{ik})$ where n_k is the size of group k and p_{ik}, which we now define. With ARD data we do not observe any connections in the graph directly. It is possible, though unlikely as long as the sample size is small compared to the population size when using simple random sampling, that we might observe an alter of one of the surveyed respondents. That is, if person i reports knowing 5 people named Michael, one of those people named Michael might also be in the survey. Even in the unlikely event that this happens, we do not have access to this information through ARD since we do not observe any links.

When considering the Binomial representation, therefore, we are making a statement not about the connections between any two individuals (which we do not observe) but instead about marginal connections between a person and a population. Respondent i is almost certainly more likely to know some members of the group k than others, but since ARD does not provide information on edges there is no way to specify that heterogeneity. Instead, we focus on an aggregate summary of the relationship between respondent i and members of group k which does not differ between members of the group because ARD, unlike the complete graph, does not contain sufficient data to do so. The power of our approach, however, is that, even under this limited information setting we still recover consistent estimates of model parameters.

Conditioned on node i's effect $\nu_i^\star$ and latent space location $z_i^\star$, the probability node i connects to an arbitrary node j in group k, written as is $\mathbb{P}(g_{ij} = 1 \mid \nu_i^\star, z_i^\star, \eta^\star) := p_{ik}$,

$$\begin{aligned} p_{ik} &= \int_V \int_Z \exp\{\nu_i^\star + \nu_j - d(z_i^\star, z_j)\} f_k(z_j) f_V(\nu_j)\ d\nu_j\ dz_j \\ &= \exp(\nu_i^\star) E\{\exp(\nu)\} \int_Z \exp\{-d(z_i^\star, z_j)\} f_k(z_j)\ dz_j\ . \end{aligned} \tag{B.1}$$

Here, we use the notation $\nu_i^\star$ to refer to a fixed but unknown parameter of interest, whereas ν_j represents a dummy variable that is integrated out. Note here we have used the property that $\exp(a+b) = \exp(a)\exp(b)$. By assuming the link function is exponential, we can easily separate the terms in the expression for $\mathbb{P}(g_{ij} = 1 \mid \nu_i, z_i, \eta)$. We believe we can extend these ideas to other link functions, as was done in [115], but we leave that to future work.

We now motivate and then formally describe these method-of-moment estimators (or equivalently, Z-estimators). Since the ARD is Binomial, we can estimate p_{ik} by equating p_{ik} with y_{ik}/n_k. This then allows us to solve for the parameters ν_i and z_i since p_{ik} depends on these two parameters (and η, which we can consistently estimate). In total, we create two systems of equations (one for the node locations and one for the fixed effects). This section assumes that we know the true parameters $\eta^\star$, but in Section B.2.2 we show how to estimate the parameters $\eta^\star$.

We start with estimating the node locations. To do this, we note that the ratio $y_{ik}/y_{ik'}$ converges in probability, by the weak law of large numbers, to the ratio

$$E_{\sigma_k}[\exp\{-d(z_i, z)\}]/E_{\sigma_{k'}}[\exp\{-d(z_i, z')\}] \ ,$$

which depends only on the variances of the distributions of node locations $\sigma_1, \ldots, \sigma_K$ and the node location z_i, where we define the notation $E_\sigma[\exp\{-d(z, z_i)\}]$ to mean that the expectation is taken with respect to σ. Note that critically, in the ratio $p_{ik}/p_{ik'}$, the terms involving the node effects and $E\{\exp(\nu)\}$, which are all unknown at this point, cancel out. This is the reason we look at the ratio of two ARD responses. Here we also make the simplifying assumption that $n_k = n_{k'}$, although the results do not change significantly if we remove this assumption. This suggests that we should take our estimate of the node location, denoted by $\hat{z}_i$, to be the value of z_i such that $y_{ik}/y_{ik'}$ is equal to the ratio $E_{\sigma_k}[\exp\{-d(z_i, z)\}]/E_{\sigma_{k'}}[\exp\{-d(z_i, z')\}]$.

More formally, we define the function $G_1 : \mathcal{M} \times (0, \infty)^2 \to \mathbb{R}$ by

$$G_1(z_i; \sigma_k, \sigma_{k'}) = \frac{E_{\sigma_k}[\exp\{-d(z_i, z)\}]}{E_{\sigma_{k'}}[\exp\{-d(z_i, z')\}]} \ . \tag{B.2}$$

We drop the dependence on k and k' for simplicity and just write G_1 without any mention of k or k'. This function, when viewed as a function of z_i for a fixed $\sigma_k, \sigma_{k'}$, is not always invertible, but we can define a pseudo-inverse by $G_1^{-1}(x) = \{m \in \mathcal{M} : G_1(m) = x\}$. In the following calculations, we will take the inverse to be chosen in a fixed way from this set. We discuss this condition further and give examples in Section B.2.3. Our estimate of the node location, $\hat{z}_i$, solves $\log\{G_1(\hat{z}_i; \hat{\sigma}_k, \hat{\sigma}_{k'})\} = \log(y_{ik}/n_k) - \log(y_{ik'}/n_{k'})$ for two arbitrary and distinct entries k, k'. In practice, the user selects the values of k and k'. The user can estimate a location using each pair of indices $k \neq k'$. Taking an average (or the Fréchet mean more generally) would improve the accuracy of the resulting estimate. Note that the log transformation simplifies the analysis of this estimator and allows us to use a proof technique that is similar to the one used to prove Theorem 1.3 in [39] or Theorem 3 in [73].

We now motivate our estimator of the the node effects. The idea is that ARD is a Binomial random variable and thus we can equate the probability of an edge between node i and nodes in group k (which depends on the node effect and the node location, which we have already estimated above) with the observed number of edges. We then solve for the node effect. To state this estimator more formally, define the function

$$G_2(\nu_i, z_i) = E\{\exp(\nu)\} \exp(\nu_i) E[\exp\{-d(z_i, z)\}] \ ,$$

where here $z \sim F(\mu_k, \sigma_k^2)$. Since y_{ik}/n_k converges in probability to $G_2(z_i^\star, \nu_i^\star)$, this motivates the following estimator

$$\hat{\nu}_i = \log\left(\frac{y_{ik}}{n_k}\right) - \log(E[\exp\{-d(\hat{z}_i, z)\}]) - \log[\hat{E}\{\exp(\nu)\}] \ . \tag{B.3}$$

where $z \sim F(\hat{\mu}_k, \hat{\sigma}_k)$ and the term $\log[\hat{E}\{\exp(\nu)\}]$ is the estimate of $\log[E\{\exp(\nu)\}]$ computed using $\hat{\eta}$. Again, as in the case of the node locations, the user can select the group index k used in computing $\hat{\nu}_i$. As in the case of the node location, we can compute $\hat{\nu}_i$ for all group indices k and their average will be an improved estimate of $\nu_i^\star$.

In the next section, we prove Theorem 3.4.1 in the main chapter, which deals with showing that estimates of the node locations and node effects are consistent and satisfy a convergence rate of $\sqrt{3\log(\tilde{n})/2\tilde{n}}$ with probability at least $1 - O(m/\tilde{n}^3)$, where $\tilde{n} = n/K$ and K is assumed to be fixed. Our proof of Theorem 3.4.1 is based on two separate lemmas: Lemma B.2.2 proves the claimed convergence result for the node locations, and Lemma B.2.3 proves the claimed convergence result for the node effects.

To begin with some notation, the estimates of the node locations and the node effects depend on the group parameters, which we denote by η. We let $\hat{z}_i(\eta)$ denote the estimate of $z_i^\star$ that is computed using the known and true η, and we let $\hat{z}_i(\hat{\eta})$ denote the estimate based upon the plug-in estimate $\hat{\eta}$, which we define formally in Section B.2.2.

B.2.1 Proof of Theorem 3.4.1

We now provide a proof of Theorem 3.4.1 in the main text. For clarity, we repeat the statement of the proof here along with the necessary assumptions. The proof relies on consistent estimates of the global parameters. For ease of exposition, we have moved the derivation of these estimates to the subsequent section. We prove the result by constructing a series of Lemmas that, when combined, yield the desired result. We begin by restating the necessary assumptions. Additional discussion of the assumptions, including verification that they hold with distributional assumptions commonly used in practice is in Section B.2.3. Note that in the main part of the chapter, the following four assumptions are labeled as Assumptions 2-5.

Assumption B.2.1. *For each k, μ_k is in a compact subset of $\mathcal{M}^p(\kappa)$ and σ_k is in a compact subset of $(0, \infty)$.*

Assumption B.2.2. *The node effects $\nu_i^\star \overset{iid}{\sim} H$ satisfy $E\{\exp(\nu_i^\star)\} < \infty$.*

Assumption B.2.3. *The distribution F is a symmetric distribution on $\mathcal{M}^p(\kappa)$ that is completely characterized by its mean and variance and satisfies the following two conditions. The function $z_i \mapsto E_k[\exp\{-d(z_i, z)\}]$ is Lipschitz for every $k \in \{1, \ldots, K\}$ and $z_i \mapsto E_k[\exp\{-d(z_i, z)\}]/E_{k'}[\exp\{-d(z_i, z')\}]$ has a pseudo-inverse that is Lipschitz.*

Assumption B.2.4. *Define $F_1 : (z_i, \sigma_k, \sigma_{k'}) \mapsto E_k[\exp\{-d(z_i, z)\}]/E_{k'}[\exp\{-d(z_i, z')\}]$. The inverse function F_1^{-1} is continuous in σ and for every k, k', ℓ, and ℓ', the following two functions are Lipschitz:*

$$\eta \mapsto \frac{E_{kk'}[\exp\{-d(z, z')\}]}{E_{\ell\ell'}[\exp\{-d(z, z')\}]}, \quad \eta \mapsto \frac{E_{kk'}[\{\exp(-d(z, z')\}]^2}{E_{\ell\ell'}[\{\exp(-d(z, z')\}]^2} .$$

Under the four assumptions above, we now restate Theorem 3.4.1 in the main chapter.

Theorem B.2.1. *Suppose Assumptions B.2.1, B.2.2, B.2.3, and B.2.4 hold. The estimators $\hat{z}_i$ and $\hat{\nu}_i$ and $\hat{\eta}$ are consistent for $z_i^\star, \nu_i^\star$, and $\eta^\star$ as $m, n \to \infty$, up to isometry on $\mathcal{M}^p(\kappa)$ and satisfy*

$$\max_{1 \le i \le m(n)} d_{\mathcal{M}^p(\kappa)}(\hat{z}_i, z_i^\star) \le \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}},$$
$$\max_{1 \le i \le m(n)} |\hat{\nu}_i - \nu_i^\star| \le \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}},$$

with probability $1 - O(m/\tilde{n}^3)$.

Proof of Theorem 3.4.1 in the main chapter. For readability, we split up the proof of Theorem 3.4.1 in the main chapter into several lemmas. Theorem 3.4.1 claims a concentration inequality for the estimates of the node locations and node effects using the plug-in estimate $\hat{\eta}$ of the global parameters. We prove this result for the node locations (Lemma B.2.2) and for the node effects (Lemma B.2.3) separately. These two lemmas require us to first prove the consistency (without a rate) on the estimates of node locations and effects, which we do in Lemma B.2.1. The proofs of Lemmas B.2.2 and B.2.3 are based on Lemmas B.2.4 and B.2.5, which prove the concentration inequalities using the true and unknown group parameter η. Combining the arguments in these lemmas proves the desired result.

□

Our proof of Theorem 3.4.1 starts with the following lemma, which states the estimates that maximize the pseudo-likelihood of the ARD are consistent as $m, n \to \infty$. We use this result later on to prove Theorem 3.4.1. We would like to emphasize that maximizing the pseudo likelihood, which we do in Section B.10, is equivalent to a method-of-moments estimator in this case.

Lemma B.2.1. *Let the assumptions from Theorem 3.4.1 of the main chapter hold. Suppose that we have consistent estimates of the group parameters η, denoted by $\hat{\eta}$. Now suppose that $(\hat{\nu}_{1:m}, \hat{z}_{1:m})$ are the Z-estimators of the node effects and locations*

described in Section B.2. Then, $(\hat{\nu}_{1:m}, \hat{z}_{1:m})$ *are consistent for* $\nu^{\star}_{[1:m]}$ *and* $z^{\star}_{[1:m]}$ *as* $m, n \to \infty$*, up to an isometry on* $\mathcal{M}^p(\kappa)$*.*

For readability, we have moved the proof of Lemma B.2.1 to Section B.10. The main idea of the proof follows the standard M-estimator consistency steps: showing a well-separated extremum and a uniform law of large numbers [164].

Lemma B.2.2. *With probability at least* $1 - O(m/\tilde{n}^3)$*, the following inequality holds up to isometry on* $\mathcal{M}^p(\kappa)$*.*

$$\max_{1 \le i \le m(n)} d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), z_i^{\star}) \le \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}} \, .$$

Proof. By the triangle inequality,

$$d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), z_i^{\star}) \le d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), \hat{z}_i(\eta)) + d_{\mathcal{M}}(\hat{z}_i(\eta), z_i^{\star}). \tag{B.4}$$

We have two terms in the triangle inequality. We will only have to focus on the second one, because that will dominate the rate as we will soon show. We calculate that one below. The first one has an extremely fast rate as it tends to zero. This can be seen in a straightforward manner from using a Taylor expansion of the estimating equation in the usual way, because the estimating equation consists of an average taken over all pairs of groups and all pairs of potential links across every pair of group which gives order $O_P(1/\sqrt{K^2 mn})$, where again m is the size of the ARD sample. We will show later that this rate is much faster than the rate for the second term in the inequality, which means this term can be ignored when proving the rate of convergence on the term $d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), z_i^{\star})$.

We now study the second term in the triangle inequality above. Now, using the definition of $\hat{z}_i(\eta)$ as $\hat{z}_i = G_1^{-1}(a; \hat{\eta})$, we write

$$d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), \hat{z}_i(\eta)) = d_{\mathcal{M}}(G_1^{-1}(a; \hat{\eta}), G_1^{-1}(a; \hat{\eta}))$$

where $a = \log(y_{ik}/n_k) - \log(y_{ik'}/n_{k'})$.

Supposing that $G_1^{-1}(a;\sigma)$ is continuous in σ, which we assume in Theorem 3.4.1 in the main chapter, we combine Lemma B.2.6 with the continuous mapping theorem to show that $d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), \hat{z}_i(\eta))$ converges to zero in probability. All we need to do now is show that the second term in (B.4) satisfies the claimed concentration inequality. By Lemma B.2.4, which we state below, with probability at least $1 - O(1/n_k^3)$,

$$d_{\mathcal{M}}(\hat{z}_i(\eta), z_i^\star) \leq \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}} \,,$$

up to isometry on $\mathcal{M}$. By a union bound, and by recalling (B.4), we conclude that with probability at least $1 - O(m/\tilde{n}^3)$:

$$\max_{1\leq i\leq m(n)} d_{\mathcal{M}}(\hat{z}_i(\hat{\eta}), z_i^\star) \leq \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}}$$

up to isometry on $\mathcal{M}$. □

The next lemma shows that the estimate of ν_i, based on the plug-in estimate $\hat{\eta}$, satisfies a similar concentration inequality.

Lemma B.2.3. *The estimator $\hat{\nu}_i$ from (B.3) satisfies the following: With probability $1 - O(m/\tilde{n}^3)$,*

$$\max_{1\leq i\leq m(n)} |\hat{\nu}_i(\hat{\eta}) - \nu_i^\star| \leq \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}} \,.$$

Proof. The proof follows the same argument that we used in the proof of Lemma B.2.2. Since $\hat{\eta}$ is consistent for η, the second term in the definition of $\hat{\nu}_i$ can be ignored when proving the desired concentration inequality (again, this argument was used in the proof of Theorem 3 in [73]). It therefore suffices to just argue that the term $\log(y_{ik}/n_k)$ satisfies the claimed concentration inequality. We can prove this inequality by Hoeffding's inequality. See Lemma B.2.5, which proves this formally. Taking a union bound over all $i = 1, \ldots, m(n)$ to proves the desired result. □

In the case where $d(z_i, z_j) = 0$ (only node effects determine connection propensity) and $m = n$ (meaning that we observe the entire graph and not just the ARD), then Theorem 3.4.1 of the main chapter simplifies to Theorem 3.3 of [39].

Lemma B.2.4. *With probability at least* $1 - O(m/\tilde{n}^3)$*, the following inequality holds:*

$$\max_{1 \leq i \leq m(n)} d_{\mathcal{M}}(\hat{z}_i(\eta), z_i^\star) \leq \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}} \, .$$

The proof is based on similar ideas found in [39, 73]. The intuition behind the proof is as follows. The estimator $\hat{z}_i(\eta)$ is based on the ARD $y_{ik}/n_k = 1/n_k \sum_{j \in G_k} g_{ij}$, which converges exponentially fast to p_{ik} by Hoeffding's inequality. This insight allows us to conclude the uniform control over the error in $\hat{z}_i(\eta)$.

Proof. To begin, we recall that the estimator is $\hat{z}_i = G_1^{-1}(y_{ik}/n_k; \eta)$. This function will not be invertible, but we can choose a representative from the set of $\{x : G_1(x; \eta) = y_{ik}/n_k\}$. Any choice will lead to the right answer, up to isometry. Note also that because of properties of $\mathcal{M}^p(\kappa)$, it is locally Euclidean. See [115] and its references for a more complete description of this point. Since $\hat{z}_i(\hat{\eta})$ converges to $z_i(\eta)$, up to isometry, we therefore only need to prove the argument for the Euclidean case (this follows from Lemma B.2.1). The extension to the spherical and hyperbolic geometries follows since there is a neighborhood around z_i in which the distances are approximately Euclidean distances, and thus the Euclidean arguments apply here too.

Since

$$a = \log(y_{ik}/n_k) - \log(y_{ik'}/n_{k'})$$

converges in probability, as $n \to \infty$, to $G_1(z_i)$, this motivates our estimate of z_i. We set $\hat{z}_i = G_1^{-1}(a)$. See Section B.2.3 for a discussion on this inverse function. Since $G_1^{-1}\{\log(p_{ik}) - \log(p_{ik'})\} = z_i^\star$,

$$\begin{aligned}
\|\hat{z}_i(\eta) - z_i^\star\| &= \left\| G_1^{-1}(a) - G_1^{-1}\{\log(p_{ik}) - \log(p_{ik'})\} \right\| \\
&\leq C |\log(y_{ik}/n_k) - \log(y_{ik'}/n_{k'}) - \log(p_{ik}) - \log(p_{ik'})| \\
&\leq \tilde{C}_n \{|y_{ik}/n_k - p_{ik}| + |y_{ik'}/n_{k'} - p_{ik'}|\} \, .
\end{aligned}$$

for some sequence of constants $\tilde{C}_n$. We know that $\tilde{C}_n$ is on the order $n_k = O(n)$ when K is fixed (which we assume), since $x \mapsto \log(x)$ is Lipschitz on any interval $[a', b']$

with Lipschitz constant $1/a'$. In our case, with probability going to 1, $y_{ik} \geq 1$ and so $y_{ik}/n_k \geq 1/n_k$ and thus we can take $1/(1/n_k) = n_k$ to be the Lipschitz constant. We thus conclude that

$$\mathbb{P}(\|\hat{z}_i(\eta) - z_i^\star\| > \epsilon) \leq \mathbb{P}\left(\left|\frac{y_{ik}}{n_k} - p_{ik}\right| > \epsilon/\tilde{C}_n\right) + \mathbb{P}\left(\left|\frac{y_{ik'}}{n_{k'}} - p_{ik'}\right| > \epsilon/\tilde{C}_n\right) . \quad \text{(B.5)}$$

We now show that both terms on the right hand side converge to zero exponentially fast. Since y_{ik} is a sum of independent Bernoulli random variables, each with expectation p_{ik}, by Hoeffding's inequality,

$$\mathbb{P}\left(\left|\frac{y_{ik}}{n_k} - p_{ik}\right| > \epsilon/\tilde{C}_n\right) \leq 2\exp\left(-2\frac{\epsilon^2 n_k}{\tilde{C}_n^2}\right) .$$

Set $\epsilon^2 = \frac{3}{2}n_k^{-1}\tilde{C}_n^2\log(n_k) = O(\frac{3}{2}n_k^{-1}n_k^2\log(n_k))$. Then,

$$\mathbb{P}\left(\left|\frac{y_{ik}}{n_k} - p_{ik}\right| > \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}}\right) \leq 2\exp\{-3\log(n_k)\} = 2/n_k^3 .$$

Similarly, $\mathbb{P}\left(\left|\frac{y_{ik'}}{n_{k'}} - p_{ik'}\right| > \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}}\right) \leq 2/n_k^3$. Putting this together, and recalling (B.5), we see that

$$\mathbb{P}\left(\|\hat{z}_i(\eta) - z_i^\star\| > \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}}\right) \leq 4/n_k^3 .$$

By a union bound, with probability at least $1 - 4m/n_k^3$,

$$\max_{1\leq i\leq m}\|\hat{z}_i(\eta) - z_i^\star\| < \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}} .$$

□

In the following lemma, we prove that the estimate $\hat{\nu}_i$ satisfies a similar type of concentration inequality. The proof is identical to the one given above, so we omit the details.

Lemma B.2.5. *If each z_i is known, and the global parameter η is known, the estimator $\hat{\nu}_i$ defined in (B.3) satisfies the following: With probability at least $1 - O(m/\tilde{n}^3)$,*

$$\max_{1\leq i\leq m(n)}|\hat{\nu}_i(\eta) - \nu_i| \leq \sqrt{\frac{3\log(\tilde{n})}{2\tilde{n}}} .$$

B.2.2 Estimating Global Parameters in Latent Space Model

In this section, we provide estimates of the model parameters η. Our discussion comes in three parts. We first show how to estimate the within-group variance terms. To estimate the within-group variances, we equate the ARD responses of people in a group k to other nodes in the same group k with the probability that an arbitrary edge exists between nodes in group k. Since this probability depends on only the within-group variance, as all nodes from a given group are distributed about the same group center, we can therefore estimate the group variance in this way.

To formally define our estimator, fix two groups G_k and $G_{k'}$. The probability that an arbitrary node in group k connects to other nodes in group k is equal to, after integrating out all the parameters, $E\{\exp(\nu)\}^2 E_{kk}[\exp\{-d(z,z')\}]$, where z, z' are independent and $z, z' \sim F(\mu_k^\star, \sigma_k^\star)$. Note critically that this does not upon the mean parameter $\mu_k^\star$.

We let $m_k(n)$ be the number of nodes we sample that belong to group k. We define the quantity

$$t_{kk'} = \frac{1}{m_k(n)} \sum_{i \in G_k} \frac{y_{ik'}}{n_{k'}} \, . \tag{B.6}$$

Then, for large n (which implies that $|G_k| = n_k$ and $m_k(n)$ is large too), the ratio $t_k/t_{k'}$ converges in probability to

$$\frac{E\{\exp(\nu)\}^2 E_{kk}[\exp\{-d(z,z')\}]}{E\{\exp(\nu)\}^2 E_{k'k'}[\exp\{-d(z,z')\}]} = \frac{E_{kk}[\exp\{-d(z,z')\}]}{E_{k'k'}[\exp\{-d(z,z')\}]} \, . \tag{B.7}$$

which depends again on just the unknown variance terms $\sigma_k^\star$ and $\sigma_{k'}^\star$. In other words, by looking at the ratio $t_k/t_{k'}$, the term $E(\exp(\nu))^2$, which we have not yet estimated and do not know in practice, cancels. So this ratio depends only on the unknown variance vector $(\sigma_1^\star, \ldots, \sigma_\star^2)$. Motivated by this description, we define an estimator $\hat{\sigma}^2(n) = \{\hat{\sigma}_1^2(n), \ldots, \hat{\sigma}_K^2(n)\}$ as the root of the following system of equations

$$\frac{t_{kk}}{t_{k'k'}} = \frac{E_{kk}[\exp\{-d(z,z')\}]}{E_{k'k'}[\exp\{-d(z,z')\}]} \, . \tag{B.8}$$

If K is large enough to ensure the above solution has a unique zero in the limit as $m, n \to \infty$, this estimator is consistent for the true $(\sigma_1^\star, \ldots, \sigma_K^\star)$.

Lemma B.2.6. *The estimator $\hat{\sigma}^2(n) = \{\hat{\sigma}_1^2(n), \ldots, \hat{\sigma}_K^2(n)\}$ that is the root of the system from (B.8) is consistent as $n \to \infty$.*

Proof. We first sketch an outline of our argument. We will define a sequence of random functions $\hat{H}_n$ such that $\lim_n E\{\hat{H}_n(\sigma^2)\} = 0$ only at the true $\sigma^\star$. This sequence of functions $\hat{H}_n$ is defined such that the estimator from the lemma minimizes this expression. Thus, to show consistency of the estimator, we can simply verify the two conditions from Theorem 5.7 of [164], which for completeness we give in Section B.10. At a high level, Condition 1 requires that H have a well-separated zero, and Condition 2 requires that $\hat{H}_n$ converge uniformly to H. Once we verify these two conditions, we can then conclude from Theorem 5.7 of [164] the desired consistency result.

By recalling the definition of t_k in (B.6), we define the sequence of random functions $\hat{H}_n : (0, \infty)^K \to [0, \infty)$ by

$$\hat{H}_n(\sigma^2) = \sum_{k=1}^{K} \sum_{k'=1}^{K} \left\{ \frac{t_{kk}}{t_{k'k'}} - \frac{E_{kk}[\exp\{-d(z, z')\}]}{E_{k'k'}[\exp\{-d(z, z')\}]} \right\}^2 .$$

We then define $H_n(\sigma^2) = E\{\hat{H}_n(\sigma^2)\}$ and $H(\sigma^2) = \lim_{n\to\infty} H_n(\sigma^2)$. By (B.7) and using the weak law of large numbers, combined with the continuous mapping theorem, it is clear that H evaluated at the true σ^2 is zero. For sufficiently large K, this zero is unique, by using the same argument that we give in Lemma B.10.3 or by using Theorem 3 of [28]. So Condition 1 is satisfied.

We now prove Condition 2. Recall that our goal is to show that

$$\sup_{\sigma^2 \in S} |\hat{H}_n(\sigma^2) - H(\sigma^2)| \xrightarrow{p} 0$$

It suffices to show that $\sup_{\sigma^2 \in S} |\hat{H}_n(\sigma^2) - H_n(\sigma^2)| = o_P(1)$, because H_n converges uniformly to H deterministically and hence also in probability. To show *this* uniform law of large numbers, we will use Corollary 2.1 of [128]. For completeness, we provide

this corollary in Section B.10. The pointwise convergence is automatically satisfied, by recalling (B.7). We now fix a k, k' and expand inside the double sum in the expression for $\hat{H}_n$ as

$$\frac{t_{kk}}{t_{k'k'}} - 2\frac{t_{kk}}{t_{k'k'}}\frac{E_{kk}[\exp\{-d(z,z')\}]}{E_{k'k'}[\exp\{-d(z,z')\}]} + \frac{E_{kk}[\exp\{-d(z,z')\}]^2}{E_{kk}[\exp\{-d(z,z')\}]^2} .$$

By comparing the terms inside the expression $|\hat{H}_n(\sigma^2) - \hat{H}_n(\tilde{\sigma}^2)|$, we see that there are just two terms to consider. To show the Lipschitz condition required to use Corollary 2.1 of [128], let $\sigma, \tilde{\sigma} \in S \subseteq (0,\infty)^K$. To simplify the notation, we let $E_{kk}[\exp\{-d(z,z')\}]$ denote the expectation using the variance vector σ and $\tilde{E}_{kk}[\exp\{-d(z,z')\}]$ to denote the expectation using the variance $\tilde{\sigma}$.

By assumption, the first term satisfies

$$2\frac{t_{kk}}{t_{k'k'}}\left|\frac{E_{kk}[\exp\{-d(z,z')\}]}{E_{k'k'}[\exp\{-d(z,z')\}]} - \frac{\tilde{E}_{kk}[\exp\{-d(z,z')\}]}{\tilde{E}_{k'k'}[\exp\{-d(z,z')\}]}\right| \leq C\frac{t_{kk}}{t_{k'k'}}||\sigma^2 - \tilde{\sigma}^2|| .$$

By assumption, the second term satisfies a similar Lipschitz condition:

$$\left|\frac{E_{kk}[\exp\{-d(z,z')\}]^2}{E_{k'k'}[\exp\{-d(z,z')\}]^2} - \frac{\tilde{E}_{kk}[\exp\{-d(z,z')\}]^2}{\tilde{E}_{k'k'}[\exp\{-d(z,z')\}]^2}\right| \leq C'||\sigma^2 - \tilde{\sigma}^2||$$

Putting this all together, we see that

$$|\hat{H}_n(\sigma^2) - \hat{H}_n(\tilde{\sigma}^2)| \leq \sum_{k,k'}(C\frac{t_{kk}}{t_{k'k'}} + C')||\sigma^2 - \tilde{\sigma}^2|| .$$

Since $\sum_{k,k'} E(Ct_{kk}/t_{k'k'} + C') = O(1)$, we conclude by Corollary 2.1 of [128] that Condition 2 holds. By Theorem 5.7 of [164], we conclude the consistency claim in the theorem.

Estimating Group Means

In this section, we show how to use the consistent estimates of the within-group variances $\sigma_1^\star, \ldots, \sigma_K^\star$ to estimate the group mean parameters. Motivated by the same approach we used to prove consistency of $\sigma_1^\star, \ldots, \sigma_K^\star$, consider now four group centers.

The probability that nodes in the first two groups, say k and k' connect, divided by the probability that nodes in the last two groups, say ℓ and ℓ', connect is

$$\frac{E\{\exp(\nu)\}^2 E_{kk'}[\exp\{-d(z,z')\}]}{E\{\exp(\nu)\}^2 E_{\ell\ell'}[\exp\{-d(z,z')\}]} = \frac{E_{kk'}[\exp\{-d(z,z')\}]}{E_{\ell\ell'}[\exp\{-d(z,z')\}]} \ .$$

Having estimated the within-group variances terms, and noting that $t_{kk'}/t_{\ell\ell'}$ estimates the probability above, we can estimate the terms $\mu_1^\star, \ldots, \mu_K^\star$ by solving the following system of equations: for every 4-tuple (k, k', ℓ, ℓ') with distinct entries,

$$\frac{t_{kk'}}{t_{\ell\ell'}} = \frac{E_{kk'}[\exp\{-d(z,z')\}]}{E_{\ell\ell'}[\exp\{-d(z,z')\}]} \ . \tag{B.9}$$

The following lemma shows that this estimator is consistent as $n \to \infty$.

Lemma B.2.7. *Let $\hat{\mu}_1(n), \ldots, \hat{\mu}_K(n)$ be a root of the system in (B.9). This estimator is consistent as $n \to \infty$, up to an isometry on $\mathcal{M}$.*

Proof. The proof is nearly identical to the one given for Lemma B.2.6, so we only sketch the argument. We define the sequence of random functions

$$\hat{H}_n(\mu) = \sum_{k,k,\ell,\ell'} \left\{ \frac{t_{kk'}}{t_{\ell\ell'}} - \frac{E_{kk'}[\exp\{-d(z,z')\}]}{E_{\ell\ell'}[\exp\{-d(z,z')\}]} \right\}^2$$

We also define $H_n(\mu) = E\{\hat{H}_n(\mu)\}$ and $H(\mu) = \lim_{n\to} H_n$. At the true $\mu^\star$ parameter, $H(\mu^\star) = 0$ for sufficiently large K. For sufficiently large K, this is the only zero, up to an isometry on $\mathcal{M}$. (Again, by using the same argument that we give in Lemma B.10.3 or by using Theorem 3 of [28].) Thus, Condition 1 is satisfied. To show Condition 2, we use the same argument as we give in the proof of Lemma B.2.6. By assumption, we know that Condition 2 holds. Thus, by Theorem 5.7 of [164], we can conclude the desired consistency result. □

Estimating Node Effect Expectation

In the previous two sections, we have shown how to obtain consistent estimates of the within-group variances and the group means. In this section, we show how to estimate

the term $\tau = E[\{\exp(\nu)\}^2]$. The probability that any node in group k connects with any node in group k' is, after integrating out all parameters,

$$E[\{\exp(\nu)\}^2]E_{kk'}[\exp\{-d(z,z')\}] \,, \tag{B.10}$$

where $z \sim F(\mu_k^\star, \sigma_k^\star)$ and $z' \sim F(\mu_{k'}^\star, \sigma_{k'}^\star)$. By drawing $\hat{z} \sim F(\hat{\mu}_k, \hat{\sigma}_k)$ independently of $\hat{z}' \sim F(\hat{\mu}_{k'}, \hat{\sigma}_k)$, we can use $E_{kk'}[\exp\{-d(\hat{z},\hat{z}')\}]$ to estimate the quantity $E_{kk'}[\exp\{-d(z,z')\}]$. Since

$$t_{kk'} = \frac{1}{n_k}\sum_{i\in G_k}\frac{y_{ik'}}{n_{k'}}$$

converges in probability to the expression in (B.10), we can estimate $E[\{\exp(\nu)\}^2]$ by

$$\hat{\tau} = \frac{t_{kk'}}{E_{kk'}[\exp\{-d(\hat{z},\hat{z}')\}]}.$$

where $\hat{z} \sim F(\hat{\mu}_k, \hat{\sigma}_k)$ independently of $\hat{z}' \sim F(\hat{\mu}_{k'}, \hat{\sigma}_k)$. By the continuous mapping theorem and by recalling (B.10), we can consistently estimate τ. □

B.2.3 Discussion of Assumptions for Theorem 3.4.1

In this section we discuss two of the assumptions made in the main chapter and discuss when these hold.

The p-dimensional normal distribution in $\mathbb{R}^p$ and the von-Mises Fisher distribution on the p-sphere are two models commonly used in the literature. We now argue that these two model satisfy this assumption. Recall that the term in question, in the case of a p-dimensional Gaussian distribution, is

$$z_i \mapsto \int_{\mathbb{R}^p} \exp(-||z_i - z||) f(z) dz \,,$$

where f here is the pdf of the p-dimensional Gaussian distribution. Note that $z \mapsto d(z_i, z)$ is Lipschitz, and $x \mapsto \exp(-x)$ is Lipschitz over $[0,\infty)$, and thus since $\exp(-x)$ is bounded by 1 on $(0,\infty)$, we conclude that $z_i \mapsto \exp\{-d(z_i,z)\}$ is Lipschitz. Because the integral of a Lipschitz function is again Lipschitz, we conclude that the assumption holds.

We now look at the assumption that the inverse of the function $z_i \mapsto G_1(z)$ is invertible, where G_1 is defined in (B.2). To begin the discussion, recall the simulation exercise in Figure B.9.2. There are two group centers at $(2, 2)$ and $(-2, -2)$ in $\mathbb{R}^2$. The point we wish to estimate is at $(0, 0)$, so the distance between each group center and this point is $2\sqrt{2}$. There is a unique point in $\mathbb{R}^2$ that satisfies this constraint. However, consider the following two examples.

EXAMPLE **9.** *Consider two group centers at* $(2, 2)$ *and* $(-2, -2)$ *in* $\mathbb{R}^2$. *Suppose the point of interest* z_i *is 2 unit away from the first point and 2 away from the second point. Then, the points* $(2, -2)$ *and* $(-2, 2)$ *will both solve the expression* $F(z) = \log(p_{ik}) - \log(p_{ik'})$, *where* p_{ik} *depends on the distance between* z_i *and the group centers.*

EXAMPLE **10.** *Now let* $\mathcal{M}^p(\kappa) = S^1(1)$, *the circle with radius 1. Set two group centers at* $(0, 1)$ *and* $(-1, 0)$ *and suppose that the point of interest is* $\pi/2$ *away from the first group center and* $3\pi/2$ *away from the second group center. Then there are two points at* $(0, 1)$ *and* $(0, -1)$ *that solve the expression* $F(z) = \log(p_{ik}) - \log(p_{ik'})$, *where* p_{ik} *depends on the distance between* z_i *and the group centers.*

The discussion above highlights the fact that the mapping $z \mapsto G_1(z)$ might not be invertible. We therefore suggest that the user select a representative element of the pseudo-inverse (hence our language in the main part of the chapter).

We now turn to discussing Assumption B.2.4. We show that under mild distributional assumptions, the function $\sigma \mapsto \frac{E_{kk'}[\exp\{-d(z,z')\}]}{E_{\ell\ell'}[\exp\{-d(z,z')\}]}$ is Lipschitz. The discussion of the function $\mu \mapsto \frac{E_{kk'}[\exp\{-d(z,z')\}]}{E_{\ell\ell'}[\exp\{-d(z,z')\}]}$ is very similar. Suppose first that the function $\sigma_k \mapsto E[\exp\{-d(z_i, z)\}]$ is Lipschitz. Then, suppose that $g : (\sigma_k, \sigma_{k'}) \mapsto E(\exp\{-d(z_i, z)\})/E(\exp\{-d(z_i, z')\})$ is differentiable. It then has a gradient $\nabla g = (\nabla_k g, \nabla_{k'} g)$, where

$$\nabla_k g = \frac{\partial g}{\partial \sigma_k} = \frac{d}{d\sigma_k} E[\exp\{-d(z_i, z)\}]]/E[\exp\{-d(z_i, z')\}]]$$

Supposing that $E[\exp\{-d(z_i, z)\}]]$ is bounded away from zero, then this partial derivative is bounded because we assumed that the function $\sigma_k \mapsto E[\exp\{-d(z_i, z)\}]]$ is

Lipschitz. The other partial derivative is given by

$$\frac{\partial g}{\partial \sigma_{k'}} = E[\exp\{-d(z_i, z)\}]] / \frac{d}{d\sigma_{k'}} E[\exp\{-d(z_i, z')\}]]$$

Supposing that the function $\sigma_{k'} \mapsto E[\exp\{-d(z_i, z')\}]]$ has a derivative that is bounded away from zero, we can thus conclude that g is Lipschitz since each of its partial derivatives is bounded.

We now verify when the function $\sigma_k \mapsto E[\exp\{-d(z_i, z)\}]$ is Lipschitz. This function is given by

$$\sigma_k \mapsto \int_{\mathcal{M}} \exp\{-d(z_i, z)\} f_k(\mu_k, \sigma_k) dz \ .$$

Supposing that $\sigma_k \mapsto f_k(\mu_k, \sigma_k)$ is Lipschitz, then we can use the Leibnitz rule (which allows us to pass the derivative inside the integral) to conclude that the function $\sigma_k \mapsto E[\exp\{-d(z_i, z)\}]$ is Lipschitz. By explicitly calculating the derivative of this expression in the cause of a Gaussian distribution, we see that $\sigma_k \mapsto f_k(\mu_k, \sigma_k)$ is Lipschitz. Since by assumption, each σ_k is in a compact (and hence bounded subset of $(0, \infty)$), we can conclude that for each z_i, $\frac{\partial g}{\partial \sigma_k}$ is bounded. To show this, we need to show that $\frac{d}{d\sigma_{k'}} E[\exp\{-d(z_i, z')\}]]$ is bounded away from zero, which for a fixed z_i is true because the σ_k are by assumption in a compact subset of $(0, \infty)$. A similar argument applies to the function $\eta \mapsto \frac{E_{kk'}[\exp\{-d(z,z')\}]^2}{E_{\ell\ell'}[\exp\{-d(z,z')\}]^2}$.

B.3 Consistency of plug-in estimator $E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y}\}$ for $S_i(g_n^\star)$ (Theorem 3.5.1)

Proof of Theorem 3.5.1. By the triangle inequality,

$$|E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\} - S_i(g_n^\star)| \leq |E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\} - E\{S_i(g_n) \mid \theta_n\}| + |E\{S_i(g_n) \mid \theta_n\} - S_i(g_n^\star)| \ .$$

By Condition 2 of Theorem 3.5.1, $|E\{S_i(g_n) \mid \theta_n\} - S_i(g_n^\star)| = o_P(1)$. We now analyze the other term. Under Condition 3, the function $\theta_n \mapsto E\{S_i(g_n) \mid \theta_n\}$ is differentiable,

so by the mean value theorem, there exists a sequence of intermediate values $\bar{\theta}_n$ such that

$$E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\} = E\{S_i(g_n) \mid \theta_n\} + \nabla E_{\bar{\theta}_n} \cdot (E_{\hat{\theta}_n} - E_{\theta_n}) \, .$$

By re-arranging, we see that

$$\begin{aligned} |E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\} - E\{S_i(g_n) \mid \theta_n\}| &= |\sum_{i=1}^{n} \partial_i E_{\bar{\theta}_n}(\hat{\theta}_{n,i} - \theta_{n,i})| \\ &\leq \sum_{i=1}^{n} |\partial_i E_{\bar{\theta}_n}(\hat{\theta}_{n,i} - \theta_{n,i})| \\ &\leq \sup_{\tilde{\theta}_n} \sum_{i=1}^{n} |\partial_i E_{\tilde{\theta}_n}| \cdot |(\hat{\theta}_{n,i} - \theta_{n,i})| \end{aligned}$$

Under Condition 3, we have that $\sup_{\theta_n} \partial_i E_{\theta_n} \leq C/n$ for some constant C, so we can then upper bound

$$|E\{S_i(g_n) \mid \hat{\theta}_n(\mathbf{y})\} - E\{S_i(g_n) \mid \theta_n\}| \leq \frac{C}{n} \sum_j |\hat{\theta}_i(n) - \theta^\star(n)_i| \, ,$$

and this last term is $o_P(1)$ by Condition 1 of the theorem. This completes the proof. □

B.4 Proofs of taxonomy results (Corollaries 3.5.1 and 3.5.2)

Proof of Corollary 3.5.1. This is straightforward to calculate:

$$E\left[\left\{E(g_{ij}) - g^*_{ij}\right\}^2\right] = E\{E(g_{ij})^2 - 2E(g_{ij})g^*_{ij} + g^2_{ij^*}\} = p^2_{ij}(\theta) - 2p_{ij}(\theta)g^*_{ij} + \left(g^*_{ij}\right)^2$$

which completes the proof. □

Proof of Corollary 3.5.2. We begin the proof by verifying that $|E\{S_i(g_n) \mid \theta^\star_n\} - S_i(g^\star_n)| \xrightarrow{p} 0$ as $n \to \infty$.

For part 1, density, we have

$$\sum_{j\in\{1,\ldots,n\},j\neq i}\frac{\text{var}(g_{ij})}{(n-1)^2}=\sum_{j\in\{1,\ldots,n\},j\neq i}\frac{p_{ij}(\theta)\left(1-p_{ij}(\theta)\right)}{(n-1)^2}$$
$$\leq\sum_{j\in\{1,\ldots,n\},j\neq i}\frac{1}{(n-1)^2}=\frac{1}{n-1}\to 0$$

so the Kolmogorov condition is satisfied and

$$P\left\{\lim_{n\to\infty}\frac{d_i}{n}=\frac{E\left(d_i\right)}{n}\right\}=1$$

which satisfies the conditions of Proposition 1.

In part 2 we turn to diffusion centrality. Recall that.

$$DC_i\left(g;q_n,K\right)=\sum_j\left\{\sum_{t=1}^{K}\left(q_ng\right)^t\right\}_{ij}=\sum_j\sum_{t=1}^{K}\frac{C^t}{n^t}\sum_{j_1,\ldots,j_{t-1}}g_{ij_1}\cdots g_{j_{t-1}j}.$$

For any t, we have

$$\text{var}\left(\frac{1}{n^t}\sum_j\sum_{j_1,\ldots,j_{t-1}}g_{ij_1}\cdots g_{j_{t-1}j}\right)=\frac{1}{n^{2t}}\sum_j\sum_{j_1,\ldots,j_{t-1}}\text{var}(g_{ij_1}\cdots g_{j_{t-1}j})$$
$$+\frac{1}{n^{2t}}\sum_j\sum_{j_1,\ldots,j_{t-1}}\sum_k\sum_{k_1,\ldots,k_{t-1}}\text{cov}(g_{ij_1}\cdots g_{j_{t-1}j},g_{ik_1}\cdots g_{k_{t-1}k})$$

where $j_0=k_0=i$ and $j_s=j,k_s=k$. $\text{var}(g_{ij_1}\cdots g_{j_{t-1}j})$ has variance $\prod_{s=1}^{t}p_{j_{s-1}j_s}\left(1-\prod_{s=1}^{t}p_{j_{s-1}j_s}\right)\leq 1$ and $\text{cov}(g_{ij_1}\cdots g_{j_{t-1}j},g_{ik_1}\cdots g_{k_{t-1}k})\leq 1$. In order for $\text{cov}(g_{ij_1}\cdots g_{j_{t-1}j},g_{ik_1}\cdots g_{k_{t-1}k})\neq 0$, $g_{ij_1}\cdots g_{j_{t-1}j}$ and $g_{ik_1}\cdots g_{k_{t-1}k}$ need to have at least one edge in common. Notice that $g_{ij_1}\cdots g_{j_{t-1}j}$ has n^t combinations since i is given. Therefore, given a fixed common edge that $g_{ij_1}\cdots g_{j_{t-1}j}$ and $g_{ik_1}\cdots g_{k_{t-1}k}$ share, $g_{ij_1}\cdots g_{j_{t-1}j}$ has n^{t-2} free choices of actors in the path, and $g_{ik_1}\cdots g_{k_{t-1}k}$ also has n^{t-2} free choices of actors in the path. Therefore, for a given fixed common edge, there are $n^{2(t-2)}$ non-zero covariance terms. Since there are n^2 choices of a common edge, there are a total of n^{2t-2} non-zero covariance terms. Therefore,

$$\text{var}\left(\frac{1}{n^t}\sum_j\sum_{j_1,\ldots,j_{t-1}}g_{ij_1}\cdots g_{j_{t-1}j}\right)\leq\frac{n^t+n^{2t-2}}{n^{2t}}.$$

Let $DC_{i,t} = \frac{1}{n^t} \sum_j \sum_{j_1,\ldots,j_{t-1}} g_{ij_1} \cdots g_{j_{t-1}j}$, we have

$$\mathbb{P}\left\{ DC_{i,t} - E(DC_{i,t}) \geq \epsilon \right\} \leq \frac{n^t + n^{2t-2}}{n^{2t}\epsilon^2} \text{ by Chebyshev's inequality}$$

$$\mathbb{P}\left\{ DC_{i,t} - E(DC_{i,t}) \mid < \epsilon \right\} \geq 1 - \frac{n^t + n^{2t-2}}{n^{2t}\epsilon^2} \to 1 \text{ as } n \to \infty$$

Therefore, $DC_{i,t}$ goes in probability to $\mathrm{E}(DC_{i,t})$ as $n \to \infty$ and, by continuous mapping theorem,

$$DC_i\left(g; q_n, K\right) = \sum_{t=1}^{K} C^t \times DC_{i,t}$$

tends to $E(DC_i\left(g; q_n, K\right))$ in probability.

For part 3, clustering, the argument is identical to the convergence of clustering in Erdos-Renyi graphs because every link is conditionally edge independent. Let $N(i)$ denote the set of neighbors of actor i and $N(i)$ denote the size of neighbors, then

$$clustering_i(g) = \frac{\sum_{j,k \in N(i)} g_{jk}}{N(i) \cdot \{N(i) \mid -1\}}$$

Similar to the proof for density, we have

$$\sum_{j,k \in N(i)} \frac{\mathrm{var}(g_{jk})}{[\mid N(i) \mid \times \{\mid N(i) \mid -1\}]^2} = \sum_{j,k \in N(i)} \frac{p_{jk}(\theta)\left(1 - p_{jk}(\theta)\right)}{[\mid N(i) \mid \times \{\mid N(i) \mid -1\}]^2}$$
$$\leq \sum_{j,k \in N(i)} \frac{1}{[\mid N(i) \mid \times \{\mid N(i) \mid -1\}]^2} = \frac{1}{\mid N(i) \mid \times \{\mid N(i) \mid -1\}} \to 0$$

so the Kolmogorov condition is satisfied and $clustering_i(g)$ goes in probability to

$$E_{z_j,\nu_j,z_k,\nu_k \mid j,k \in N(i)} \{\mathbb{P}(g_{jk} = 1 \mid \nu_j, \nu_k, z_j, z_k)\}$$

as n tends to infinity.

Finally, we now verify the conditions of Theorem 3.5.1 of the main chapter. Specifically, we need to show that the derivative $\partial_i E_{\theta_n}$ is uniformly bounded over Θ.

The degree of a node i is $S_i(g_n) = 1/(n-1)\sum_{j\neq i} p_{ij}(\theta)$. In this case, for any k,

$$\partial_k E\{S_i(g_n) \mid \theta_n\} = \frac{1}{n-1}\frac{d}{d\theta_k}p_{ik}(\theta)$$

So, supposing that $\frac{d}{d\theta_k}p_{ik}(\theta)$ is uniformly bounded, we can conclude that $\partial_k E\{S_i(g_n) \mid \theta_n\} \leq C/(n-1)$ for some constant C, so Condition 2 holds assuming that $1/n\sum_j |\hat{\theta}_i(n) - \theta_i(n)| = o_P(1)$. A similar argument applies to the clustering coefficient of a node, defined as

$$S_i(g_n) = \frac{1}{\binom{N_i}{2}}\sum_{j,k\in N_i} g_{ij}g_{jk}$$

where N_i is the set of neighbors of node i: $N_i = \{j : g_{ij} = 1\}$.

We finally look at the centrality parameter of a node. We only look at the case of $T = 2$, since the argument for $T > 2$ is similar. We begin by computing $E\{S_i(g_n) \mid \theta_n\}$, which is equal to

$$E\{S_i(g_n) \mid \theta_n\} = \sum_j \frac{C}{n}E[A_{ij}] + \sum_j \frac{C^2}{n^2}E\{[A^2]_{ij}\} .$$

where A^2 is the matrix square of the matrix A and A is the adjacency matrix of the graph g. We are interested in the derivative of $E\{S_i(g_n) \mid \theta_n\}$. Supposing that $\frac{d}{d\theta_k}p_{ik}(\theta)$ is uniformly bounded, the derivative of the first term satisfies Condition 3. So we now turn to the second sum and expand

$$E\{A_{ij}\}^2 = E\{\sum_k A_{ik}A_{kj}\} = \sum_k E\{A_{ik}A_{kj}\} = \sum_k E\{A_{ik}\}E\{A_{kj}\} = \sum_k p_{ik}(\theta)p_{kj}(\theta) .$$

Under the same assumption that the derivative $\frac{d}{d\theta_k}p_{ik}(\theta)$ is uniformly bounded, we can conclude that the second sum is also satisfies Condition 2. It follows then that $E\{S_i(g_n) \mid \theta_n\}$ satisfies Condition 3. □

B.5 *Proof of consistency of OLS estimators in many networks setting (Theorem 3.5.2)*

Proof of Theorem 3.5.2. We consider the case where there is no intercept ($\alpha = 0$) to simplify the calculations, but the same argument applies to the case where $\alpha \neq 0$.

We begin by expanding

$$O_r = \beta E\{S_r(g_n) \mid \hat{\theta}_r(n)\} + \epsilon_r = \beta S_r^\star + \big(\epsilon_r + \beta E\{S_r \mid \hat{\theta}_r(n)\} - \beta E\{S_r \mid \theta_r\} + \beta E\{S_r \mid \theta_r\} - \beta S_r^\star\big)$$

Let $\tilde{\epsilon}_{n,r} = \big(\epsilon_r + \beta E\{S_r \mid \hat{\theta}_r(n)\} - \beta E\{S_r \mid \theta_r\} + \beta E\{S_r \mid \theta_r\} - \beta S_r^\star\big)$. Now, by using the analytic expression for the OLS estimator, we have that

$$\begin{aligned}
|\hat{\beta} - \beta| &= \frac{1}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R |E\{S_r \mid \hat{\theta}_r(n)\} \tilde{\epsilon}_{n,r}| \\
&= \frac{1}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\} \big(|\epsilon_r + \beta E\{S_r \mid \hat{\theta}_r(n)\} - \beta E\{S_r \mid \theta_r\} \\
&+ \beta E\{S_r \mid \theta_r\} - \beta S_r^\star|\big) \\
&\leq \frac{1}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)^2} \sum_{r=1}^R |E\{S_r \mid \hat{\theta}_r(n)\} \epsilon_r| + \\
&\frac{\beta}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R |E\{S_r \mid \hat{\theta}_r(n)\} \Big(E\{S_r \mid \hat{\theta}_r(n)\} - E\{S_r \mid \theta_r\}\Big)| + \\
&\frac{\beta}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R |E\{S_r \mid \hat{\theta}_r(n)\} \left(E\{S_r \mid \theta_r\} - S_r^\star\right)| \\
&= I + II + III \, .
\end{aligned}$$

Now, I is $o_P(1)$ assuming that $E(\epsilon_r | E\{S_r \mid \hat{\theta}_r(n)\}) = 0$. Now, let us look at the second term,

$$II = \frac{1}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\} \times |E\{S_r \mid \hat{\theta}_r(n)\} - E\{S_r \mid \theta_r\}| \, ,$$

and the third term is

$$III = \frac{1}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\} \times |E\{S_r \mid \theta_r\} - S_r^\star|$$

For the third term, supposing that $E\{S_r \mid \hat{\theta}_r(n)\} \leq C$, I can upper bound

$$III \leq \frac{C}{R^{-1} \sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \frac{1}{R} \sum_{r=1}^R |E\{S_r \mid \theta_r\} - S_r^\star|$$

Now suppose that that $E\{S_r^\star \mid \theta\}$ has finite mean. We then can then conclude that

$$III \leq \frac{C}{R^{-1}\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \frac{1}{R} \sum_{r=1}^R |E\{S_r \mid \theta_r\} - S_r^\star| \, .$$

By Hoeffding's inequality, we can conclude that the average $\frac{1}{R}\sum_{r=1}^R |E\{S_r \mid \theta_r\} - S_r^\star| = o_P(1)$, and so by Slutksy's lemma, we can conclude that $III = o_P(1)$ as $n, R \to \infty$.

We now move to the second term II. Using a Taylor series expansion, we can write

$$\begin{aligned} E\{S_r \mid \hat{\theta}_r(n)\} - E\{S_r \mid \theta_r(n)\} &= D^T(\bar{\theta}_n)||\hat{\theta}_r(n) - \theta_r(n)|| \\ &= \sum_{i=1}^n \partial_i E\{S_r \mid \bar{\theta}_r(n)\}|\hat{\theta}_r(n) - \theta_r(n)|_i \end{aligned}$$

for some sequence of intermediate values $\bar{\theta}_n$. So,

$$\begin{aligned} II &\leq \frac{1}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\} \sum_{i=1}^n \partial_i E\{S_r \mid \bar{\theta}_r(n)\} \times |\hat{\theta}_r(n) - \theta_r(n)|_i \\ &\leq \frac{C}{\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \sum_{r=1}^R \frac{1}{n} \sum_{i=1}^n |\hat{\theta}_r(n) - \theta_r(n)|_i \\ &= \frac{C}{R^{-1}\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \frac{1}{R} \sum_{r=1}^R \sum_{i=1}^n |\hat{\theta}_r(n) - \theta_r(n)|_i \end{aligned}$$

where the first inequality follows from the Taylor series expansion and the second inequality follows from the assumptions of this theorem. Supposing that that $E(E\{S_r \mid \hat{\theta}_r(n)^2) < \infty$, we bound

$$II \leq \frac{C}{R^{-1}\sum_{r=1}^R E\{S_r \mid \hat{\theta}_r(n)\}^2} \max_{1 \leq r \leq R} \sum_{i=1}^n |\hat{\theta}_r(n) - \theta_r(n)|_i$$

Under the assumptions of the theorem, we have that $\max_{1\leq r\leq R} \sum_{i=1}^n |\hat{\theta}_r(n) - \theta_r(n)|_i = o_P(1)$, so we conclude that $|\hat{\beta}_{n,R} - \beta| = o_P(1)$, as claimed.

To prove that the estimator $\hat{\gamma}_{n,R}$ is consistent, the argument is nearly identical. To see why, we simple re-arrange the supposed data generating model:

$$S_r^\star = \alpha + \gamma T_r + \epsilon_r - E\{S_i(g_n) \mid \hat{\theta}_r(n)\} + S_r^\star \, . \tag{B.11}$$

The same argument applies to show that the OLS estimates of γ are also consistent under the conditions of the theorem. □

B.6 Checking conditions of Theorem 3.5.2 for common network statistics (Theorem 3.5.3)

Proof of Theorem 3.5.3. We only prove the case for the density. The arguments for the other two statistics are similar.

From the proof of Theorems 3.2.1, 3.3.1, 3.4.1, we showed that for any network, each estimator $\hat{\theta}_{i,r}(n)$ satisfies an exponential concentration inequality, and by taking a union bound over all nodes in a network, we see that

$$\mathbb{P}(\frac{1}{n}\sum_{i=1}^{n}|\hat{\theta}_{i,r}(n)-\theta^{\star}_{i,r}|>\epsilon)\leq\mathbb{P}(\max_{1\leq i\leq n}|\hat{\theta}_{i,r}(n)-\theta^{\star}_{i,r}|>\epsilon)\leq nC\exp(-C'\epsilon^2 n)\ .$$

for some constants C and C'. By taking a union bound over all R villages, we conclude that

$$\mathbb{P}(\max_{1\leq r\leq R}\frac{1}{n}\sum_{i=1}^{n}|\hat{\theta}_{i,r}(n)-\theta^{\star}_{i,r}|>\epsilon)\leq Rn\exp(-\epsilon^2 n)\ .$$

Under the assumptions of the theorem, we have that $Rn\exp(-n)\to 0$, so Condition 2 holds. We now discuss Condition 3 of Theorem 3.5.2. One way to satisfy this is to require that the network statistic for each network is the same (i.e., we are considering just the centrality of a set of nodes). In this case, since the network statistic $S_{i,r}$ satisfies the required derivative condition, per Theorem 3.5.1, we can then conclude that the maximum also satisfies such a derivative condition. This completes the proof. □

B.7 Simulations using fully-elicited graphs

In this section we present additional results using fully-elicited, observed graphs. We use data from [13], which consists of completely observed graphs from 75 villages in rural India. The goal of these results is two-fold. First, we aim to demonstrate that our results hold in networks that have the level of sparsity and complexity that a user

could reasonably find in practice. Second, we aim to show that the perfomance of our method improves as the graph size increases, as indicated by our results.

In each village, about one-third of respondents were asked ARD questions. [28] compare statistics estimated with ARD from these graphs with the same statistics calculated using the complete graph. We leverage these results and present a different aspect, how the MSE changes as the size of the graph grows. We present results for individual-level statistics from these graphs and compute MSE across individuals. Figure B.7.1 presents these results.

B.8 Additional simulation results with estimated formation model parameters

In this section we present additional simulation results to complement the simulations we present in the main text. We present results when the parameters are estimated using the procedure in [28], rather than assumed to be consistently estimated. These simulations are presented in Figures B.8.1, B.8.2, and B.8.3. The results we present here use the same simulation setup as Figure 3.6.1 in the main text.

B.9 Simulations to demonstrate consistency of latent space model parameter estimators

In this section, we study simulation experiments to we show that the estimates of $z_i^\star$ and $\nu_i^\star$ are consistent as $n \to \infty$.

We start with the estimates of the node locations. To do this, we create two group centers $\mu_1 = (2, 2)$ and $\mu_2 = (-2, -2)$ and set $z_0 = (0, 0)$. Our goal is to estimate the location of z_0. In Figure B.9.1, we plot a sample realization of the z_i and z_0 for $n = 500$.

We assign n nodes to be in group 1, and n nodes to be in group 2. Given these group memberships c_i, we draw

$$z_i \mid \{c_i = j\} \sim N\left(\mu_j, \frac{1}{3} I_2\right) \quad j = 1, 2 \, .$$

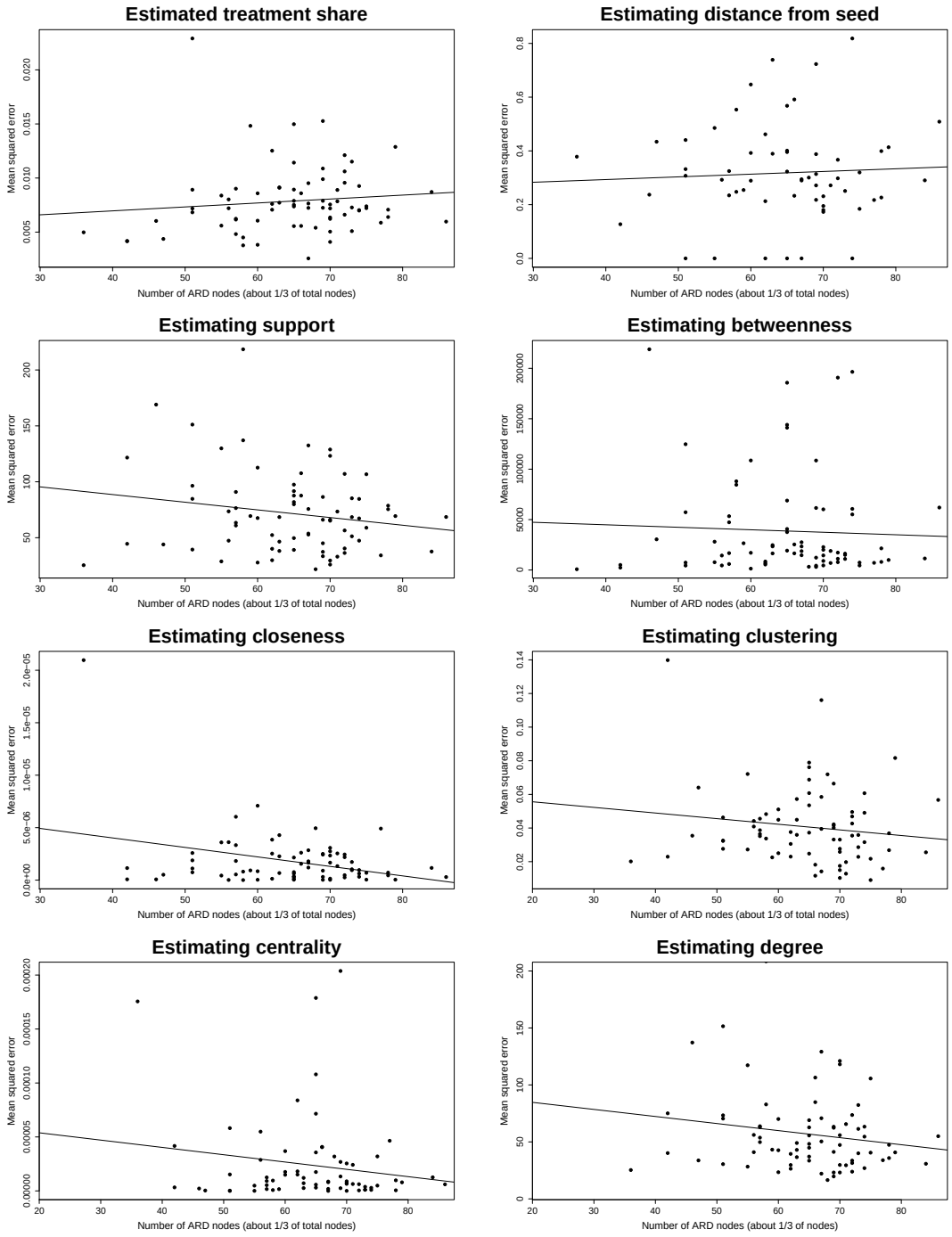


Figure B.7.1: MSE and graph size. Each plot shows the MSE (computed across nodes) plotted as a function of the number of respondents who received ARD using data from [13].

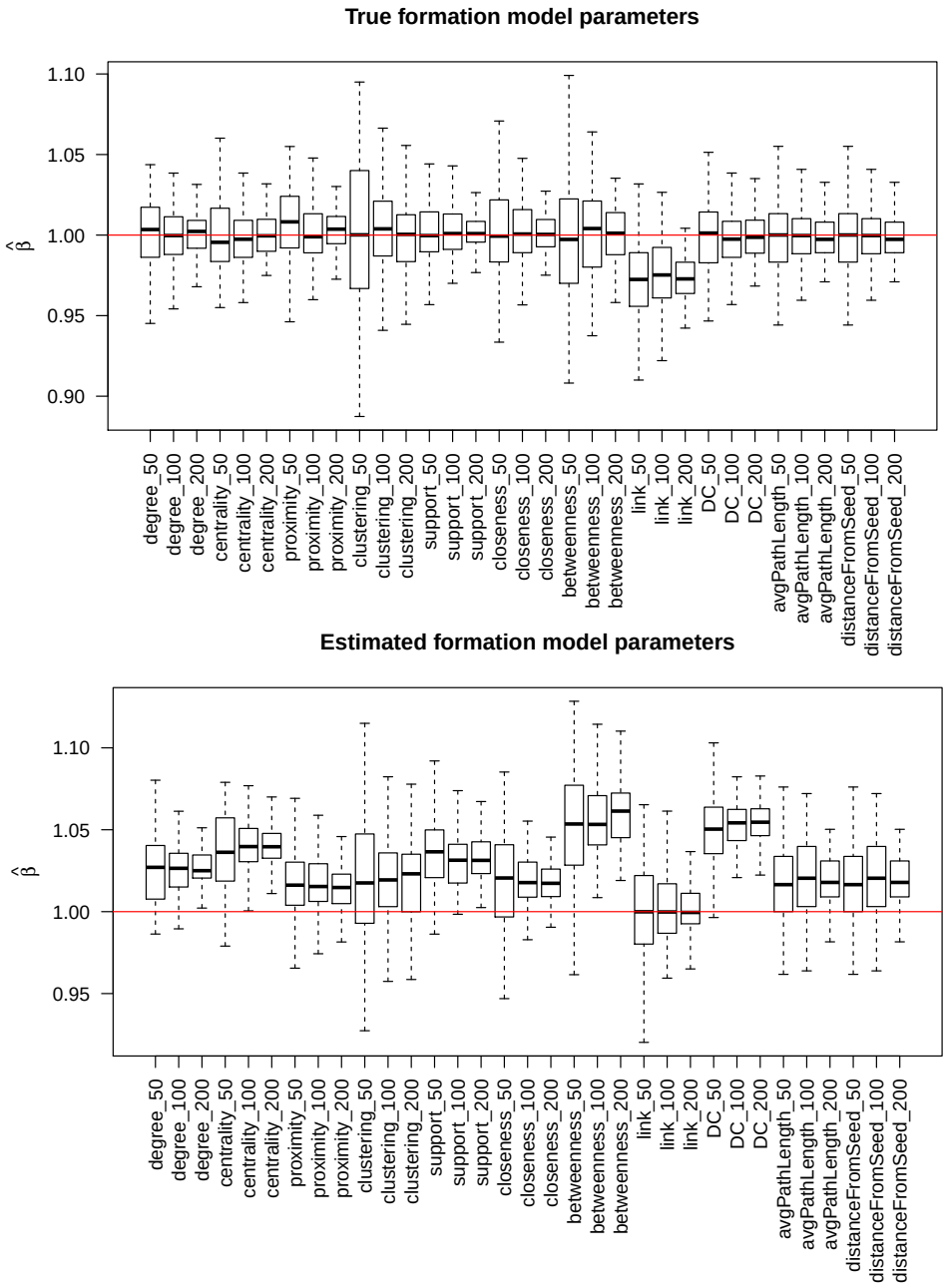


Figure B.8.1: Boxplot of $\hat{\beta}$ for β in regression $y_{ij,r} = \alpha + \beta \bar{S}_{ij,r} + \epsilon_r$, where $S_{ij,r}$ and $\bar{S}_{ij,r}$ represent a true and mean individual-level measure, respectively. Each box represents the distribution of $\hat{\beta}$ for one measure and use of R=50, 100 or 200 networks in regression. 50 actors and 1000 pairs (for link) are randomly selected for each network. The middle line of the boxplot denotes median, and borders of the boxes denote first and third quartile. The red line denotes the true $\beta = 1$ used to generate $y_{ij,r} = \alpha + \beta S^*_{ij,r} + \epsilon_r$ in the simulation. These results corroborate the theoretical intuition developed in Theorems 3.5.1 and 3.5.2.

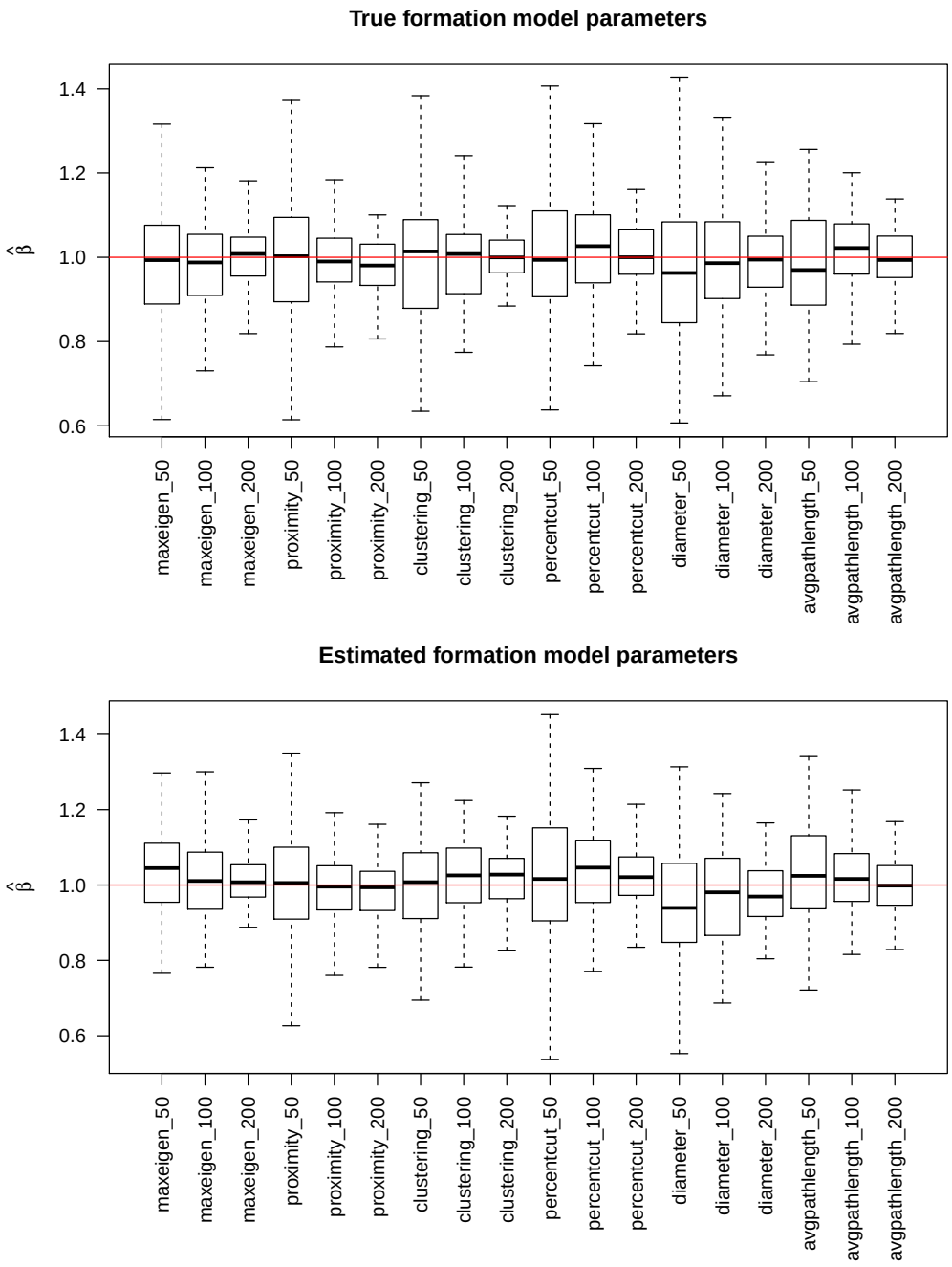


Figure B.8.2: Boxplot of $\hat{\beta}$ for β in regression $y_r = \alpha + \beta \bar{S}_r + \epsilon_r$, where S_r and $\bar{S}_r$ represent a true and mean network-level measure, respectively. Each box represents the distribution of $\hat{\beta}$ for one measure and use of R=50, 100 or 200 networks in regression. The middle line of the boxplot denotes median, and borders of the boxes denote first and third quartile. The red line denotes the true $\beta = 1$ used to generate $y_r = \alpha + \beta S_r^* + \epsilon_r$ in the simulation. These results corroborate the theoretical intuition developed in Theorems 3.5.1 and 3.5.2.

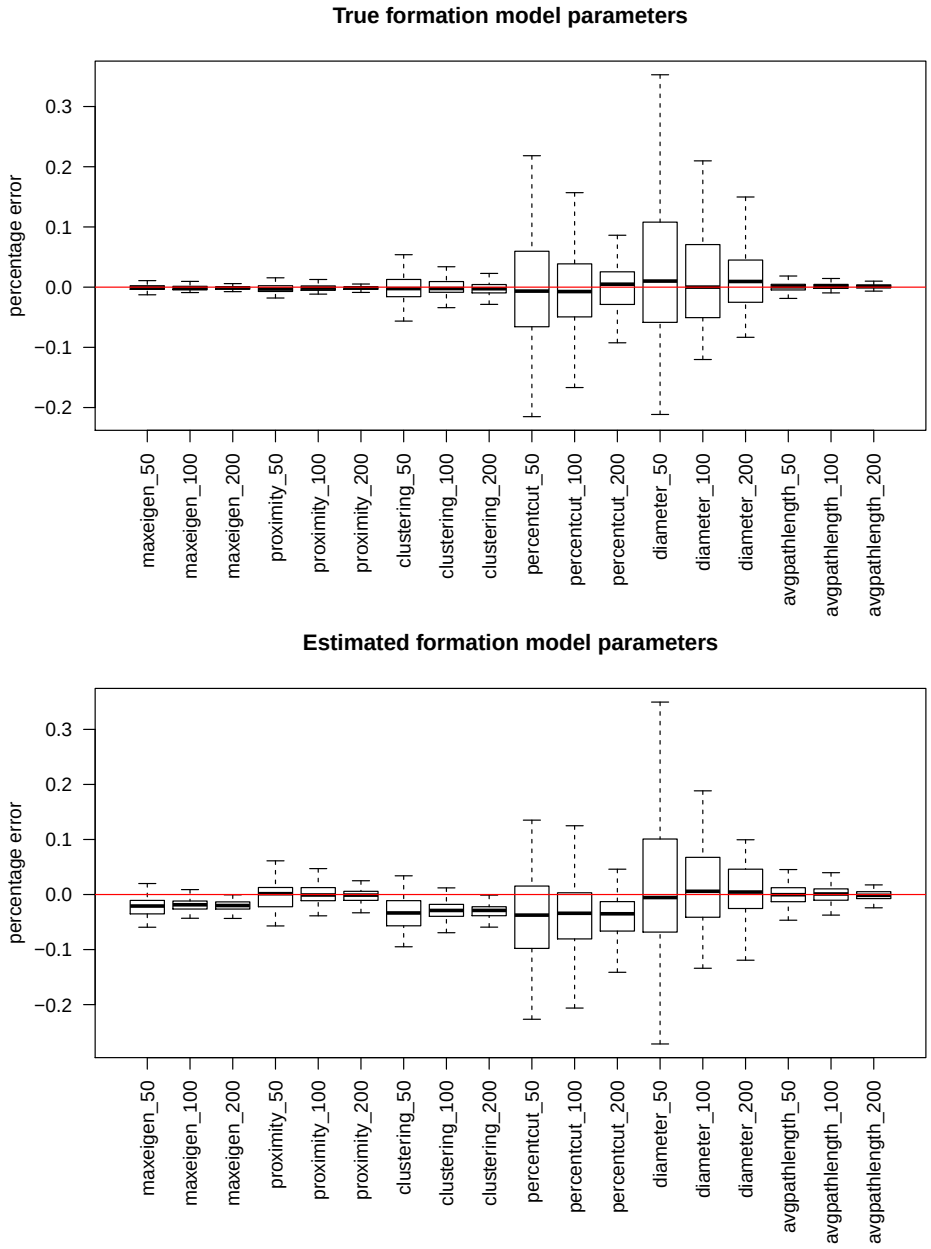


Figure B.8.3: Boxplot of percentage errors of $\hat{\gamma}$ for γ in regression $\bar{S}_r = \alpha + \gamma T_r + \epsilon_r$, where S_r and $\bar{S}_r$ represent a true and mean network-level measure, respectively. Each box represents the distribution of percentage errors for one measure and use of R=50, 100 or 200 networks in regression. The middle line of the boxplot denotes median, and borders of the boxes denote first and third quartile. These results corroborate the theoretical intuition developed in Theorems 3.5.1 and 3.5.2.

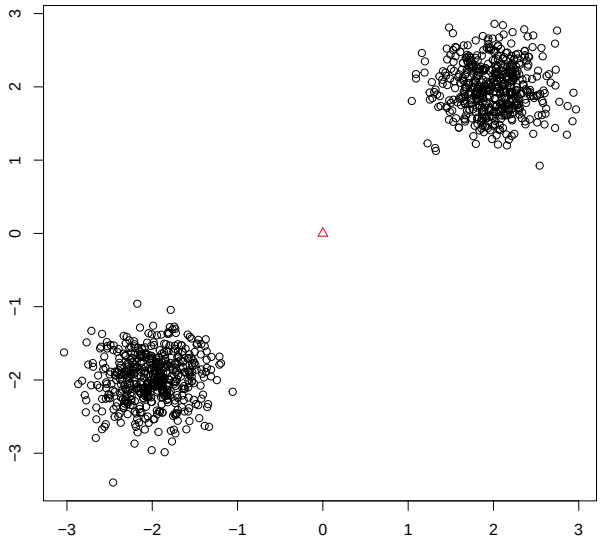


Figure B.9.1: Plot of $n = 500$ locations (black circle) centered at $(2, 2)$ and $(-2, 2)$. The point at $(0, 0)$ (the red triangle) is the location we want to estimate with the ARD.

where I_2 is the 2×2 identity matrix. We then create generate edges between the node at location z_i and z_0 by defining

$$P_i = \exp(-||z_i - z_0||) = \exp(-||z_i||) \; .$$

where the second equality follows since $z_0 = (0, 0)$. We then generate the edges between nodes in groups 1 and 2 and the node at z_0 in this way:

$$G_{i1} = \text{Bernoulli}(P_i), \quad c_i = 1$$
$$G_{i2} = \text{Bernoulli}(P_i), \quad c_i = 2 \; .$$

The ARD responses are then $y_{i1} = \sum_{i=1}^{n} G_{i1}$ and $y_{i2} = \sum_{i=n+1}^{2n} G_{i2}$. We then estimate the node location z_0 by the estimation procedure described above. In particular, the

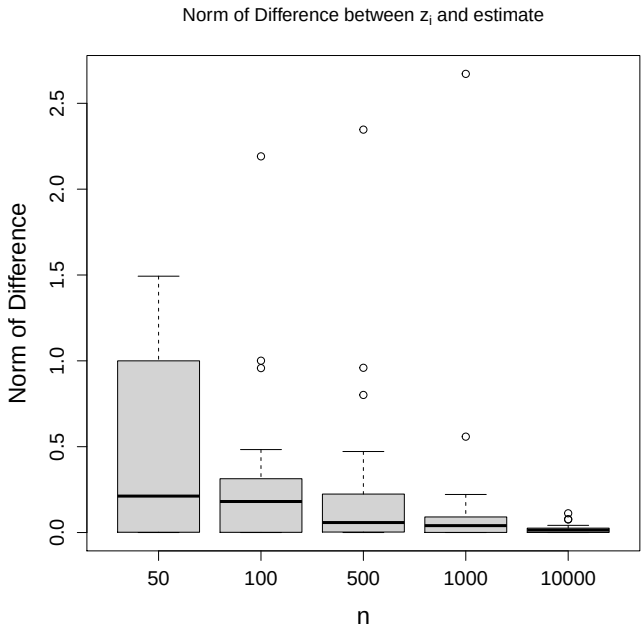


Figure B.9.2: Norm of difference $\hat{z}_i - z_0$ for various values of n on the x-axis.

estimate $\hat{z}_i$ solves $\hat{z}_i = G_1(a)$, where $a = \log(Y_{i1}/n) - \log(Y_{i2}/n)$. We repeat the above process 25 times for each value of $n = 50, 100, 500, 1000, 10^4$. In Figure B.9.2, we plot $||\hat{z}_i - z_i|| = ||\hat{z}_i||$. We see that the norm is decreasing as n increases.

To demonstrate the consistency claim for the node effect estimate $\hat{\nu}_i$, we simulate n locations $z_i \sim N\left((2,2), \frac{1}{3}I_2\right)$ and $\nu_i^\star \overset{\text{i.i.d.}}{\sim} \text{Unif}(-2, 0)$. We then let $\nu_i^\star = -1$. Our estimate of the node effects is, recalling (B.3), the $\hat{\nu}_i$ that solves

$$\frac{y_{ik}}{n_k} = E\{\exp(\nu^\star)\} \exp(\hat{\nu}_i) E[\exp\{-d(z_i, z)\}] \ ,$$

where $z \sim F(\mu_k^\star, \sigma_k^\star)$. We suppose that the terms z_i, $E\{\exp(\nu^\star)\}$ and $\mu_k^\star, \sigma_k^\star$ are known, which allows us to solve for the estimate $\hat{\nu}_i$. We repeat this process 100 times for $n = 250, 500, 1000, 10^4$. In Figure B.9.3, we plot the estimation error and see that as n increases, the error decreases.

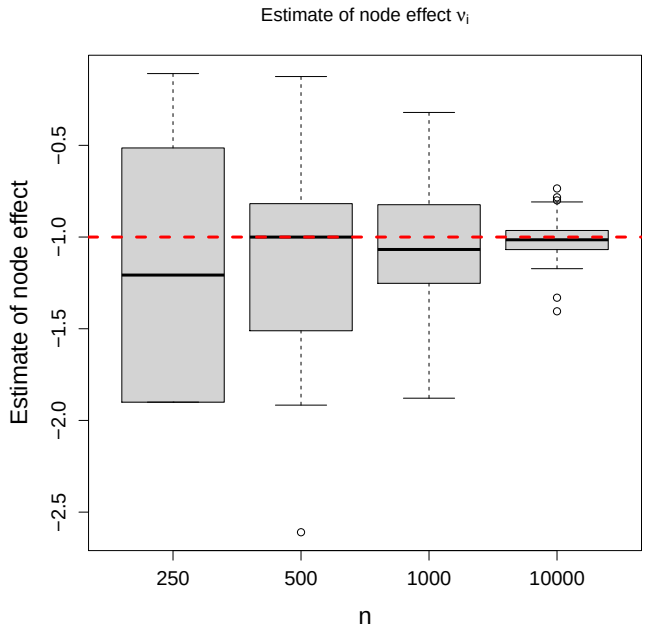


Figure B.9.3: Estimate of the node effect $\nu_i^\star$ using the esimate defined in (B.3). We set $\nu_i^\star = -1$ and generate estimates of this parameter using various values of n on the other x-axis. As n increases, we see convergence of the estimate to $\nu_i^\star$.

B.10 Supplemental results used to prove Theorem 3.4.1

In this section, we prove Lemma B.2.1 which is used to prove Theorem 3.4.1. To do that, we introduce the pseudo-log likelihood of the ARD. We note here that maximizing the pseudo-likelihood is equivalent to the method-of-moments (or equivalently, Z-estimator) approach taken in Section B.2 but by maximizing the pseudo log likelihood, we are able to use the classical M-estimator results to conclude consistency [164].

We now discuss the pseudo-likelihood of the ARD. As described above, the data we observe, when conditioned on the ego's parameters and marginalizing over the alters' parameters, are simply Binomial draws. We can write the log-likelihood for the number of links that i has to a random set of n_k members of group k as

$$\log f(y_{ik} \mid \nu_i, z_i, \eta) = \log \left\{ \binom{n_k}{y_{ik}} \right\} + y_{ik} \log(p_{ik}) \tag{B.12}$$

$$+ (n_k - y_{ik}) \log(1 - p_{ik}). \tag{B.13}$$

for an arbitrary $\nu_i, z_i,\ \eta$.

We can build our target objective function by summing up over all k traits for each node and then all nodes

$$\sum_{i=1}^{m} \sum_{k=1}^{K} \log f(y_{ik} \mid \nu_i, z_i, \eta).$$

For each i, the counts of links across groups are independent conditional on the latent positions. We describe this as the pseudo-likelihood because the full likelihood also accounts for correlation between $Y_{ik(j)}$ and $Y_{jk(i)}$, where $k(i)$ is person i's group. Nonetheless, this pseudo-likelihood delivers consistent estimates, similar to other recent work in consistent estimators for graph models. See [**?**] and its references for a discussion on this point. In practice, we do not know the parameter $\eta^\star$, which contains the means and variances of the distribution of node locations as well as the expxected value of $\exp(\nu_i)$. Suppose that we have a consistent estimator $\hat{\eta} \xrightarrow{p} \eta^\star$.

We can then use this plug-in estimator in place of $\eta^\star$, which leads to the final ARD pseudo-likelihood

$$\hat{\ell}_n(\mathbf{y} \mid \theta) = \sum_{i=1}^{m} \sum_{k=1}^{K} \log f(y_{ik} \mid \nu_i, z_i, \hat{\eta}). \tag{B.14}$$

We then define the estimates of the node locations and effects as the maximisers of the following pseudo-likelihood:

$$(\hat{\nu}_1, \ldots, \hat{\nu}_m, \hat{z}_1, \ldots, \hat{z}_m) = \underset{\nu_{[1:m]}, z_{[1:m]}}{\arg\max}\, \hat{\ell}_n \left(\mathbf{y} \mid \nu_{[1:m]}, z_{[1:m]}, \hat{\eta}\right). \tag{B.15}$$

We begin by including the following result, Theorem 5.7 of [164], that allows us to conclude consistency of an M-estimator. This result requires two conditions, which we now state below.

Condition 1. *For all* $\epsilon > 0$,

$$\sup_{\theta : d(\theta^\star, \theta) \geq \epsilon} Q(\theta) < Q(\theta^\star) \, .$$

When Θ is compact, which we assume is true in Condition 3 below, a sufficient condition for Condition 1 to hold is that Q has a unique maximum at $\theta^\star$.

Condition 2 (Uniform law of Large Numbers)**.** *We require that*

$$\sup_{\theta \in \Theta} |\hat{Q}_n(\theta) - Q(\theta)| \xrightarrow{p} 0 \, .$$

Under these two conditions, we can conclude that any M-estimator of the form $\hat{\theta}_n = \arg\max Q_n(\theta)$ is consistent, in the sense specified below.

Lemma B.10.1 (Theorem 5.7 of [164])**.** *Let* $\hat{Q}_n$ *be a sequence of random functions indexed by* $\theta \in \Theta$*, where* (Θ, d) *is a metric space. Suppose that Conditions 1 and 2 hold. Then,* $d(\hat{\theta}, \theta^\star) \xrightarrow{p} 0$ *as* $n \to \infty$.

There are many ways to verify the uniform law of large numbers result in Condition 2. See, among others, [128, 8, 141]. In this work, we follow the approach outlined

by [128], which requires a compact parameter space, that the functions $\hat{Q}_n$ converge pointwise to $E(\hat{Q}_n)$, and that the functions $\hat{Q}_n$ satisfy a Lipschitz-type condition.

The following two conditions are used in the uniform law of large numbers results from [128].

Condition 3 (Compact Parameter Space)**.** *We suppose that* (Θ, d) *is a compact metric space.*

Condition 4 (Pointwise Convergence)**.** *For each* $\theta \in \Theta$, $\hat{Q}_n(\theta) = \bar{Q}(\theta) + o_P(1)$

Lemma B.10.2 (Corollary 2.1 of [128])**.** *Suppose Conditions 3 and 4 hold and that* $\bar{Q}_n$ *is equicontinuous. Also suppose that* Θ *is a metric space with metric* $d(\theta, \theta')$ *and there exists* B_n *such that for all* $\theta, \theta' \in \Theta$, $|\hat{Q}_n(\theta) - \hat{Q}_n(\theta')| \leq B_n d(\theta, \theta')$ *and* $B_n = O_P(1)$*. Then* $\sup_{\theta \in \Theta} |\hat{Q}_n(\theta) - \bar{Q}_n(\theta)| = o_P(1)$.

As [128] points out immediately after Corollary 2.1, if $\bar{Q}_n = E(\hat{Q}_n)$ and $E(B_n)$ is bounded, then we can drop the assumption that $\hat{Q}_n$ is equicontinuous and instead include it as a conclusion to the lemma. In other words, we do not need to check the condition of equicontinuity to use the lemma above.

Lemma B.10.3. *The likelihood function of the data* y_{ik}*, conditioned on node* i*'s parameters, which we denote by* $f(\nu_i, z_i)$*, from the proof of Lemma B.2.1 has a unique maximum at* $(\nu_i^\star, z_i^\star, \eta^\star)$ *for sufficiently large* K*.*

Proof. By the information decomposition, and again using f to denote the likelihood of y_{ik} given node i's parameters, we have that

$$E[\log\{f(y_{ik} \mid \nu_i, z_i)\}] = H_{ik}(\theta^\star) - KL_{ik}(\theta \mid \theta^\star) \, .$$

where H is the entropy of $y_{ik} \mid \nu_i^\star, z_i^\star$ and KL is the KL-divergence between $y_{ik} \mid \nu_i^\star, z_i^\star$ and $y_{ik} \mid \nu_i, z_i$. See [44] for more information on this decomposition.

So to maximize the $E[\log\{f(y_{ik} \mid \nu_i, z_i)\}]$, we need to minimize the KL divergence. Hence, by summing over $k = 1, \ldots, K$,

$$\sum_{k=1}^{K} KL_k(\theta \mid \theta^\star) = \sum_{k=1}^{K} \log\left\{\frac{p_{ik}(\nu_i, z_i)}{p_{ik}(\nu_i^\star, z_i^\star)}\right\} n_k p_{ik}(\nu_i, z_i) + \log\left\{\frac{1 - p_{ik}(\nu_i, z_i)}{1 - p_{ik}(\nu_i^\star, z_i^\star)}\right\} n_k\{1 - p_{ik}(\nu_i, z_i)\} .$$

Now, note first that the KL divergence is always greater than or equal to zero. Second, the KL divergence is zero if and only if $\theta = \theta^\star$. Note that there are just two parameters ν_i and z_i. For any $k = 1, \ldots, K$, we define the set A_k to be

$$A_k = \{(\nu_i, z_i) : \exp(\nu_i) H_k(z_i) E\{\exp(\nu)\} = p_{ik}(\nu_i^\star, z_i^\star)\} .$$

In words, A_k is the set of parameters (ν_i, z_i) that lead to the same probability $p_{ik}(\nu_i^\star, z_i^\star)$. Since the KL divergence is always greater than or equal to zero, with equality if and only if the parameters are equal, we see that $\bigcap_{k=1}^{K} A_k$ is the set of maximizers of the function f.

Clearly, $(\nu_i, z_i) \in A_k$ for each k and thus $(\nu_i^\star, z_i^\star) \in \bigcap_{k=1}^{K} A_k$. To argue that f has a unique maximum at $(\nu_i^\star, z_i^\star)$, we now need to argue that $\{(\nu_i^\star, z_i^\star)\} = \bigcap_{k=1}^{K} A_k$. Supposing that $p_{ik}(\nu_i^\star, z_i^\star) \neq p_{ik'}(\nu_i^\star, z_i^\star)$ for some $k \neq k'$, meaning we have at least two distinct probabilities, then f has a unique maximum. For K sufficiently large, we will have that $\{(\nu_i^\star, z_i^\star)\} = \bigcap_{k=1}^{K} A_k$. Thus, f has a unique maximum. □

Proof of Lemma B.2.1. To show consistency of the estimates based on maximizing the pseudo-likelihood, we first note that each pair ν_i, z_i appears in exactly K of the terms in the expression from (B.14). That is,

$$(\hat{\nu}_i, \hat{z}_i) = \arg\max_{\nu, z} \sum_{k=1}^{K} n_k^{-1} \log f(y_{ik} \mid \nu_i, z_i, \hat{\eta})$$

Thus, we will show that each pair $(\hat{z}_i, \hat{\nu}_i)$ converges to the true value. By recalling

that $y_{ik} \mid \nu_i^\star, z_i^\star$ is Binomial, we see that

$$\sum_{k=1}^{K} n_k^{-1} \log f(y_{ik} \mid \nu_i, z_i) = \sum_{k=1}^{K} \left\{ n_k^{-1} \log \binom{n_k}{y_{ik}} + \frac{y_{ik}}{n_k} p_{ik} + \left(1 - \frac{y_{ik}}{n_k}\right) \log(1 - p_{ik}) \right\} .$$

To argue consistency, we will use Theorem 5.7 of [164]. To simplify the analysis, first note that the term $n_k^{-1} \log \binom{n_k}{y_{ik}}$ does not depend on the parameters, and also $y_{ik} \mid \nu_i, z_i = \sum_{j \in G_k} g_{ij}$, so the maximum pseudo likelihood estimates $(\hat{\nu}_i, \hat{z}_i)$ also satisfy

$$\begin{aligned} (\hat{\nu}_i, \hat{z}_i) &= \arg\max_{\nu, z} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j \in G_k} \left\{ g_{ij} \log(\hat{p}_{ik}) + (1 - g_{ij}) \log(1 - \hat{p}_{ik}) \right\} \\ &= \arg\max_{\nu, z} \hat{f}_n(\mathbf{y}, \nu_i, z_i, \hat{\eta}) . \end{aligned}$$

We now define the term $\hat{p}$ in the expression above. Given estimates of the structural parameters $E\{\exp(\nu)\}, \mu_k, \sigma_k^2$, we define

$$\hat{p}_{ik} := \exp(\nu_i) \hat{E}\{\exp(\nu)\} \hat{H}_k(z_i)$$

where $\hat{H}_k(z_i) = E[\exp\{-d(z_i, z)\}]$ is computed using z_j drawn iid from $F(\hat{\mu}_k, \hat{\sigma}_k^2)$ and $\hat{E}\{\exp(\nu)\}$ is the estimate of $E\{\exp(\nu)\}$ defined in the previous section.

Define $f_n(\nu_i, z_i) = E\{\hat{f}_n(\mathbf{y}, \nu_i, z_i, \hat{\eta})\}$ and $f(\nu_i, z_i) = \lim_{n \to \infty} f_n(\nu_i, z_i)$. In the defintion of f_n, the expectation is over the distribution of $\mathbf{y}$ (and note that the distribution of $\hat{\eta}$ is also determined by the distribution of $\mathbf{y}$). To see why, see our discussion where we define particular estimates of $\hat{\eta}$ and note that these estimates depend on $\mathbf{y}$. By Lemma B.10.3, f has a unique maximum at $(\nu_i^\star, z_i^\star, \eta^\star)$. Thus, since $V \times M \times E$ is compact, it follows that Condition 3 is satisfied. To verify Condition 2, we first use the triangle inequality to see that $\sup_{\nu_i, z_i} |\hat{f}_n(\mathbf{y}, \nu_i, z_i, \hat{\eta}) - f(\mathbf{y}, \nu_i, z_i, \hat{\eta})|$ is upper bounded by

$$\sup_{\nu_i, z_i} |\hat{f}_n(\mathbf{y}, \nu_i, z_i, \hat{\eta}) - f_n(\mathbf{y}, \nu_i, z_i, \hat{\eta})| + \sup_{\nu_i, z_i} |f_n(\mathbf{y}, \nu_i, z_i, \hat{\eta}) - f(\mathbf{y}, \nu_i, z_i, \hat{\eta})| .$$

The second term, which is deterministic, converges to zero uniformly over all (ν_i, z_i) by the Weierstrass M-test, which we provide for completeness as Lemma B.10.4 and state below:

Lemma B.10.4 (Weierstrass M-test)**.** *Let* $f_n(x) = \sum_{i=1}^{n} f_i(x)$ *and* $f = \lim_n f_n(x)$. *Suppose that there exists* M_n *such that for each* n, $|f_n(x)| \leq M_n$ *for all* x *and* $\sum_{i=1}^{\infty} M_i < \infty$. *Then* f_n *converges uniformly to* f.

Hence this second term converges uniformly in probability over all (ν_i, z_i). We now look at the first term. To show that this converges uniformly in probability to zero, we will use Corollary 2.1 from [128] which for completeness we provide in Section B.10. In particular, if we can show (1) that $\hat{f}_n$ converges pointwise to $E(\hat{f}_n)$ and (2) that $\hat{f}_n$ satisfies the Lipschitz inequality

$$|\hat{f}_n(\mathbf{y}, \nu_i, z_i, \hat{\eta}) - \hat{f}_n(\mathbf{y}, \nu_i', z_i', \hat{\eta})| \leq B_n d\{(\nu_i, z_i), (\nu_i', z_i')\} \,, \tag{B.16}$$

where $B_n = O_P(1)$, then Condition 2 holds by Corollary 2.1 of [128].

We first show the pointwise convergence. By assumption, $\hat{p}_{ik} = \exp(\nu_i)\hat{\tau}\hat{H}(z_i)$ is a continuous function of its arguments, and since $\hat{\eta} \xrightarrow{p} \eta^\star$, $\hat{p}_{ik} \xrightarrow{p} p_{ik}$ as $n \to \infty$ by the continuous mapping theorem. Also, conditioned on the ego's parameters, $y_{ik}/n_k \xrightarrow{p} p_{ik}$ (by Chebyshev's inequality, since g_{ij} are independent and bounded), so we conclude the pointwise convergence.

To show (B.16), we upper bound the left hand side by $t_1 + t_2$, where

$$\begin{aligned} t_{1k} &= g_{ij} |\log(\hat{p}_{ik}) - \log(\hat{p}_{ik})| \leq g_{ij} |\nu_i - \nu_i' + \log \hat{H}(z_i) - \log \hat{H}(z_i')| \\ t_{2k} &= (1 - g_{ij}) |\log(\hat{p}_{ik}) - \log(\hat{p}_{ik})| \leq g_{ij} |\nu_i - \nu_i' + \log \hat{H}(z_i) - \log \hat{H}(z_i')| \,. \end{aligned}$$

By assumption, $\hat{H}$ is Lipschitz in z and so $|\log\{\hat{H}(z_i)\} - \log\{\hat{H}(z_i')\}| \leq C d(z_i, z_i')$ for some constant C, so

$$t_{1k} \leq g_{ij} \left\{ |\nu_i - \nu_i'| + C d(z_i, z_i') \right\} \leq g_{ij} C' d((\nu_i, z_i), (\nu_i', z_i')) \,,$$

and a similar argument holds for t_{2k}. Since the left hand side of (B.16) is upper bounded by $\sum_{k=1}^{K} t_{1k} + t_{2k}$, and since $\sum_{j \in G_k} n_k^{-1} g_{ij}$ is $O_P(1)$, we conclude that (B.16) holds and so we conclude by Corollary 2.1 of [128] that Condition 2 holds. It follows from Theorem 5.7 of [164] that the maximum pseudo likelihood estimator $(\hat{\nu}_i, \hat{z}_i)$ is consistent. □

Appendix 3

APPENDIX FOR CHAPTER 4

This appendix contains the algorithms and supplemental results for Chapter 4.

C.1 Bootstrap correction for directed data

We provide in Algorithm 9 a bootstrap correction algorithm for directed network data.

C.2 Using BIC to select dimension of latent space

Suppose G is a network with $n = 100$ nodes drawn from a latent space model, defined in [85], where its latent space dimension is $d_{\text{true}} = 2$. We fit the observed network G to the latent space models with dimensions $d_{\text{fit}} = 1, 2, 3, 4$ and calculate the corresponding BIC with the `latentnet::ergmm.bic` command. We summarize the results in Table C.2.1. The BIC method provides a false prediction, suggesting the dimension to be $d_{\text{fit}} = 4$ instead of $d_{\text{true}} = 2$. This indicates the BIC method might not be an optimal approach for latent space dimension detection as stated in the `latentnet` manual: "*It is not clear whether it is appropriate to use this BIC to select the dimension of latent space ...*" This motivates us to develop Algorithm 5 in Section 4.4.2 to robustly address the problem. Our method correctly predicts the latent space dimension 80% of the time or better for a variety of true dimensions and for values of n (the number of nodes) as small as 100. Crucially, this sample size covers many empirically-relevant networks, such as the Indian villages network studied in [28] and others.

Algorithm 9: Bootstrap correction of Directed Tracy Widom statistic

Input: Observed sociomatrix: G; Estimated probability matrix $\hat{P}$; Bootstrap iterates: B; TW_1 mean: μ_{TW}; TW_1 standard deviation: s_{TW}; Significance Level: α.

Compute

$$\hat{A}_{ij} = (G_{ij} - \hat{P}_{ij})/\sqrt{\hat{P}_{ij}(1 - \hat{P}_{ij})} \ ;$$

for $b = 1$ **to** B **do**

  Sample $G_b^\star \sim F_{\hat{\theta}}$;

  Compute

$$[A_b^\star]_{ij} = ([G_b^\star]_{ij} - \hat{P}_{ij})/\sqrt{\hat{P}_{ij}(1 - \hat{P}_{ij})} \ ;$$

  Set $\lambda_b^\star = s_{\max}(A_b^\star)$;

end

Define μ to be the sample mean of the $\{\lambda_b^\star\}_{b=1}^B$ and s to be the sample standard deviation of $\{\lambda_b^\star\}_{b=1}^B$;

Compute the test statistic t

$$t := \mu_{TW} + s_{TW}\left(\frac{s_{\max}(\hat{A}) - \mu}{s}\right)$$

if $TW_1(\alpha/2) < t < TW_1(1 - \alpha/2)$ **then**

  Do not reject t and set Rej = FALSE;

else

  Reject t and set Rej = TRUE.

end

Output: Rejection of bootstrap statistic: Rej.

	$d_{\text{fit}} = 1$	$d_{\text{fit}} = 2$	$d_{\text{fit}} = 3$	$d_{\text{fit}} = 4$
BIC	6047.10	5774.94	5750.63	**5721.85**

Table C.2.1: Fitted BIC of the observed network G with $d_{\text{true}} = 2$, which suggest $d_{\text{fit}} = 4$ be the underlying latent dimension.

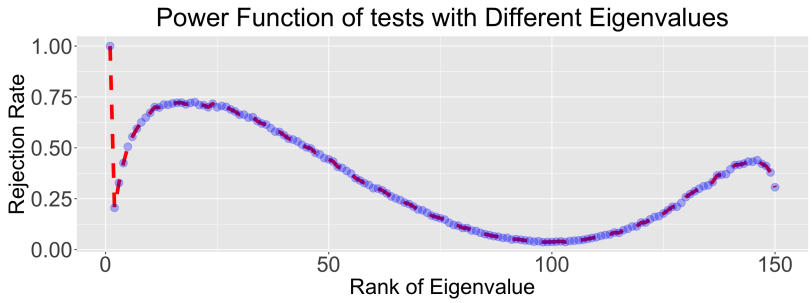
C.3 Power of tests with different eigenvalues

In Theorem 4.2.1, only the extreme eigenvalues of the adjusted adjacency matrix are used to construct the proposed goodness-of-fit test. We further attempted to construct tests using non-extreme eigenvalues and assess the power of tests via the following simulation. First, as we lack asymptotic results of the non-extreme eigenvalues, we use their empirical distributions as reference distributions of the tests. Specifically, for a given network size n, we repeatedly sample normalized random matrix as described in Theorem 4.2.1, that is, $n \times n$ matrix A with $A_{ij} \sim_{i.i.d.} N(0, 1/(n-1))$ and $A_{ii} = 0$, and compute its eigenvalues to access the empirical distribution. We then consider a simple test, where we assume that the networks are sampled from a two-communities SBM model and test the null hypothesis that the networks are drawn from an Erdos-Reyni. Specifically, for $n = 150$, we sample 10,000 networks from two-communities SBM models with some community probability matrix B, where B_{ij} denotes the probability to form a link between nodes from community i and j. For each sampled network adjacency matrix, we normalize it as described in Theorem 4.2.1 under the null hypothesis and compute the eigenvalues. We use each eigenvalue as test statistics and reject H_0 if the observed eigenvalue falls out of $\alpha/2$ or $1 - \alpha/2$ quantiles of its simulated empirical distribution. We compute the rejection rate of tests using different eigenvalues and plot the power functions in Figure C.3.1. We observe that the tests using extreme eigenvalues achieve the best rejection rates and are substantially better

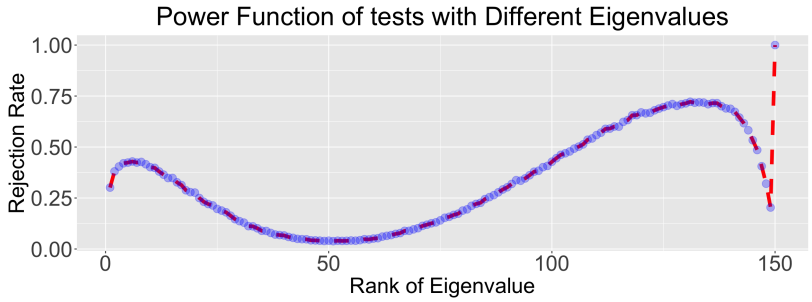
	Political Blog		Simmons College		Caltech	
d_{fit}	t_{TW}	R_{mis}	t_{TW}	R_{mis}	t_{TW}	R_{mis}
1	2.23	5.16	48.03	16.00	7.94	56.17
2	2.70	4.58	32.12	15.39	8.31	33.41
3	12.20	4.34	17.95	18.21	13.82	37.37
4	15.11	4.42	19.99	14.89	7.15	36.57
5	10.80	4.91	16.18	11.13	5.79	31.48
6	8.67	4.58	8.29	10.30	2.90	21.11
7	-1.75^*	**4.26**	2.29	10.28	2.47	27.57
8	-2.78^*	4.42	1.14^*	10.37	-0.83^*	**18.35**
9	-1.33^*	5.07	0.84^*	9.87	0.06^*	19.27
10	-1.76^*	5.32	2.37	9.67	-0.73^*	19.40
11	-1.26^*	5.33	0.30^*	**9.62**	0.46^*	19.07
12	-	-	1.92	10.21	-	-

Table C.2.2: Tracy Widom statistics and mis-classification rates of Political Blog data, Simmons College data, and Caltech data. Tracy Widom statistics that are not rejected are labeled with stars. Optimal mis-classification rates are highlighted in bold text.

than those of tests using non-extreme eigenvalues. This suggests that the non-extreme eigenvalues can have weak power in testing against improper model fit, and using more eigenvalues of the adjacency matrix may not increase the power of the test.



(a)

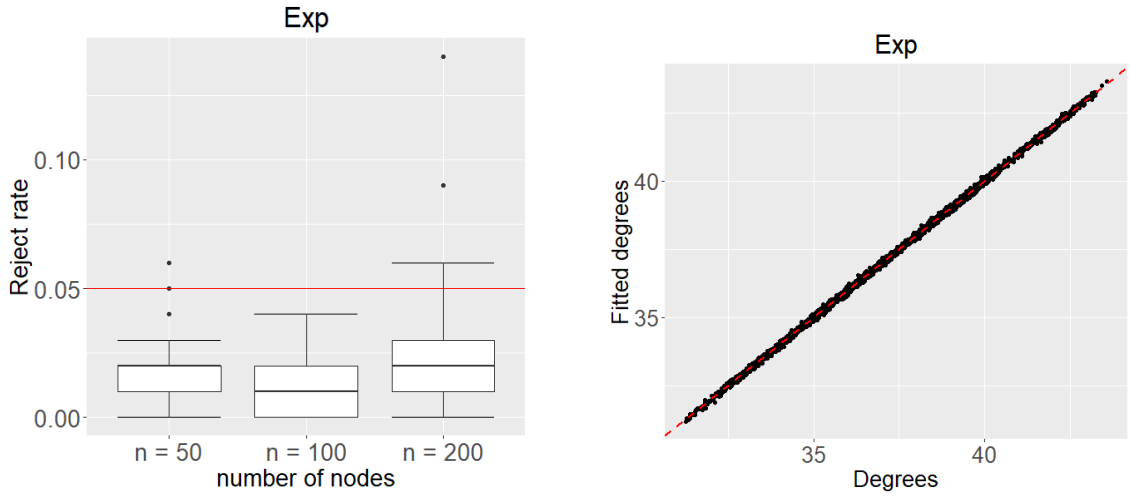


(b)

Figure C.3.1: (a): Power function of tests using different eigenvalues with $B_{11} = B_{22} = 0.5$ and $B_{12} = B_{21} = 0.25$. (b): Power function of tests using different eigenvalues with $B_{11} = B_{22} = 0.25$ and $B_{12} = B_{21} = 0.5$. The two plots correspond to the situation where (a) within-cluster links and (b) between-cluster links are more likely to form. Best rejection rates are achieved with tests using the largest and the smallest eigenvalues. The rejection rates of tests with non-extreme eigenvalues are not comparable with those of tests with extreme eigenvalues and thus has less power.

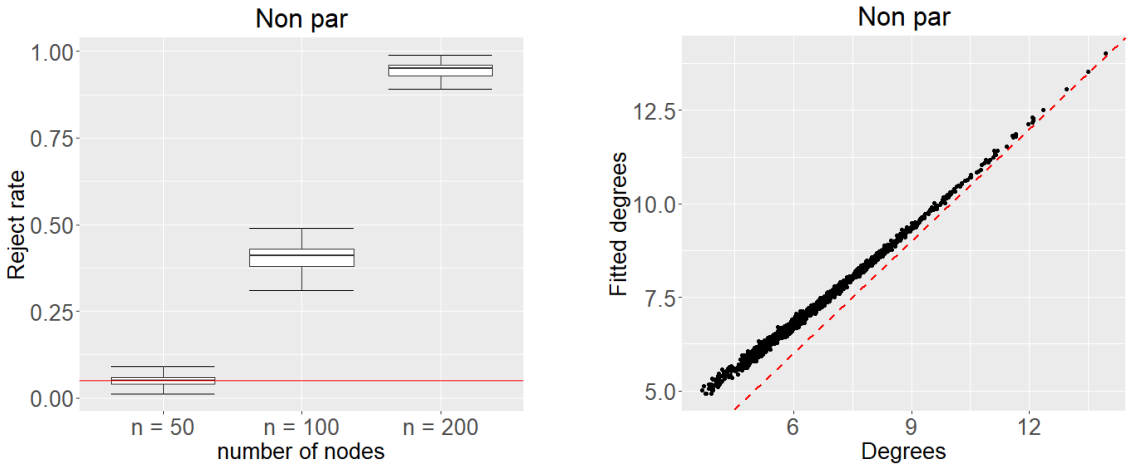
C.4 Additional figures

This appendix contains additional figures and simulations.



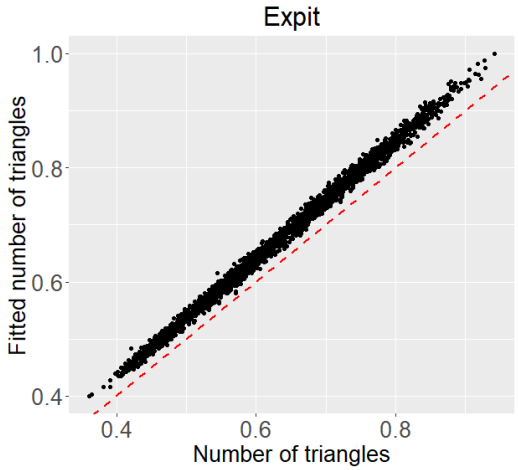
(a) (b)

Figure C.4.1: Left: Power for the null hypothesis in (4.10) against Beta model with exp link function for $n = 50, 100, 200$. The powers centered below 0.05 and is smaller than the corresponding Type I error. Right: Identical settings as in Figure 4.4.1, with true model altered to exp link function. We observed that most of the points align upon the diagonal, which potentially indicates that the exp model can also be a good fit. Such a phenomenon is observed with other network statistics, i.e. average path length, number of 3/4-cliques, etc, which suggests there might exist an equivalent relationship between the expit and exp link function.

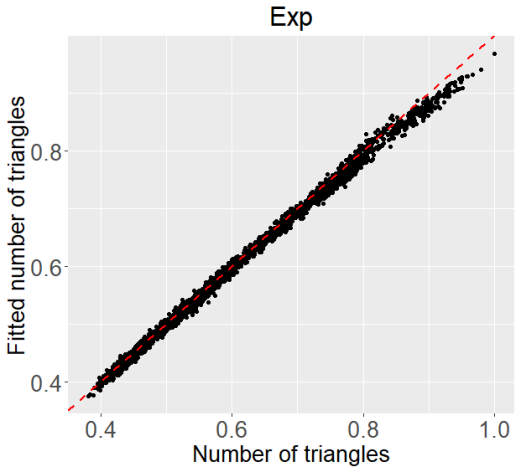


(a) (b)

Figure C.4.2: Left: Power for the null hypothesis in (4.10) against non-parametric network structures for $n = 50, 100, 200$. The powers increases sharply as network sizes grows. Right: Identical settings as in Figure 4.4.1, with true model altered to non-parametric structures. We observed that the trend of the points tilts up at the left end, with more mass concentrates around smaller degree distributions. Such a behavior differs significantly with that of the β-model with expit link function, which is consistent with our observation on the left that our method reject almost 100% of the time for $n = 200$.



(a) Beta model with expit link function



(b) Beta model with exp link function

(c) Non-parametric structure

Figure C.4.3: We plot the number of triangles in observed networks against the number of triangles simulated via fitted MLE estimates w.r.t. expit link function. The red dash line corresponds to $y = x$. If the fit is good, we will observe the data points align upon $y = x$. Compared to the poorly behaved non-parametric structure, we observed a good correspondence between the observed and fitted on Beta model with expit and exp function, indicating the goodness-of-fit of the two models is probably good. This further tell us there is potentially an equivalent relationship between the two link functions and can be achieved with the fixed point method in [38]. The difference between the simulated values in black and the diagonal line in (A) decreases as the sample size n increases.

Appendix 4

APPENDIX FOR CHAPTER 5

D.1 Reasoning of the prior choice

Note that, since each $\boldsymbol{x}_i$ is defined on $\mathbb{H}^p(\kappa)$, we cannot apply a usual multivariate normal prior on $\boldsymbol{X}$ as in [132], which is originally defined on the Euclidean geometry. It is tempting to consider distributions defined on the hyperbolic space as substitutes of the multivariate normal distribution, such as the distributions described in [62] and [127], and directly specify the prior on $\boldsymbol{X}$. However, since the hyperbolic space is centered around $\boldsymbol{\mu}_0^p = (1, 0, \cdots, 0) \in \mathbb{H}^p \subset \mathbb{R}^{p+1}$ instead of the Euclidean origin $(0, \cdots, 0) \in \mathbb{R}^{p+1}$, directly impose a hyperbolic prior on $\boldsymbol{X}$ will generally lead to an asymmetric proposal function, and restrict us from applying the random walk Metropolis-Hastings sampling algorithm described in [132], which uses a symmetric proposal function, and is more tractable and computationally efficient.

D.2 Curvature estimation through stress minimization

In this section, we define an estimate of the curvature of the $\mathbb{H}^p(\kappa)$ given (potentially) noisy dissimilarities between points on $\mathbb{H}^p(\kappa)$. An advantage of the estimator we now propose is that it does not depend on knowing the dimension p of the space. For any curvature value κ, we let $\boldsymbol{x}_i(\kappa)$ for $i = 1, \ldots, n$ denote the set of embedding coordinates obtained from hyperbolic MDS ([99]) computed using the curvature value κ, and we let $d_{ij}(\kappa)$ be the distance between $\boldsymbol{x}_i(\kappa)$ and $\boldsymbol{x}_j(\kappa)$. Our estimate $\hat{\kappa}$ is then the estimate that minimizes the stress between the observed dissimilarities $\{\widehat{\delta}_{ij}\}$ and

the distances $\{d_{ij}(\kappa)\}$. That is, we set

$$\hat{\kappa} = \arg\min_{\kappa} \text{stress}\left(\{d_{ij}(\kappa)\}, \{\widehat{\delta}_{ij}\}\right) =: \arg\min_{\kappa} \sqrt{\frac{\sum_{i<j}\left(d_{ij}(\kappa) - \widehat{\delta}_{ij}\right)^2}{\sum_{i<j} d_{ij}(\kappa)^2}}\,.$$

This is not the only way to estimate curvature of $\mathbb{H}^p(\kappa)$ given (potentially) noisy dissimilarity data. [116], for example, proposed a different estimate of curvature and proved it is consistent as the error in $\{\widehat{\delta}_{ij}\}$ goes to zero. But we found in simulations that the above estimator outperforms the estimate in [116] for the noise levels we consider in this work.

D.3 MCMC convergence

In this section, we provide typical trace plots from our simulations and data sets. We plot the MCMC proposed δ_{ij} and σ values over iterations t for the MCMC simulation in Section B.9. Specifically, to comprehensively investigate the convergence of δ_{ij}, we ran a BHMDS simulation for sample size $n = 200$, hyperbolic dimension $p = 2$, error size $\sigma = 1$, with 20,000 MCMC iterations and burn-in of 3,000 iterations, randomly pick ten δ_{ij} entries from the dissimilarity matrix, and record their MCMC sampled values after the MCMC burn-in. We provide the trace plots below. We also assessed convergence using the diagnostic of [144]. We analyze the traces using the default `raftery.diag()` function from the `coda` R package [139], and summarize the results in Table D.3.1. Our choice of total number of MCMC iterations is close to the average of the suggested total number of MCMC iterations, $\bar{N} = 21596$ iterations $Nmin = 3746$. This suggests that around 20,000 MCMC iterations is enough to estimate the parameters of interest in BHMDS.

Table D.3.1: Summary of the `raftery.diag()` MCMC diagnosis for traces of the ten δ_{ij} random samples and error size σ. M is small for all traces, indicating we burn-in enough of the MCMC samples. Our choice of total number of MCMC iterations is close to the suggested total number of MCMC iterations N's, and greatly exceeds the suggested minimal number of iterations $Nmin$, indicating the proposed MCMC sampler mixes well with ~ 20000 MCMC iterations.

	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
Sample 1	21	24222	3746	6.47
Sample 2	18	19083	3746	5.09
Sample 3	18	20487	3746	5.47
Sample 4	24	24474	3746	6.53
Sample 5	20	22194	3746	5.92
Sample 6	18	20061	3746	5.36
Sample 7	18	21762	3746	5.81
Sample 8	20	22152	3746	5.91
Sample 9	20	23348	3746	6.23
Sample 10	18	21612	3746	5.77
σ	18	18158	3746	4.85

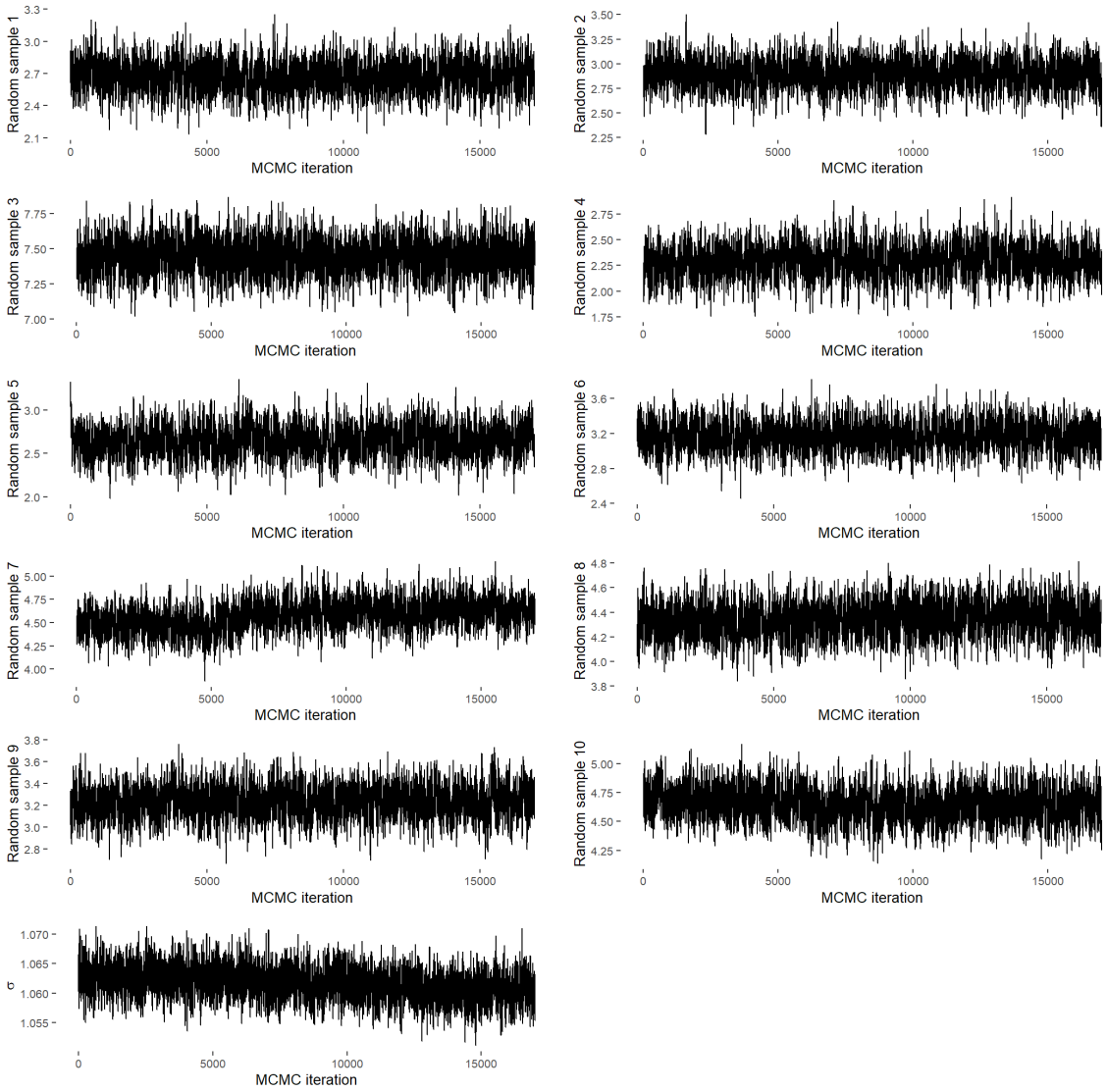


Figure D.3.1: Trace plots of ten randomly sampled MCMC proposed δ_{ij} and σ values over iteration t for MCMC simulation in Section B.9 with $n = 200$, $p = 2$, $\sigma = 1$.

BIBLIOGRAPHY

[1] Farras Abdelnour, Michael Dayan, Orrin Devinsky, Thomas Thesen, and Ashish Raj. Functional brain connectivity is predictable from anatomic network's laplacian eigen-structure. *NeuroImage*, 172:728–739, 2018.

[2] Daron Acemoglu, Asuman Ozdaglar, and Alireza Tahbaz-Salehi. Systemic risk and stability in financial networks. *American Economic Review*, 105(2):564–608, 2015.

[3] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, Eric P Xing, and Tommi Jaakkola. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, volume 15, 2006.

[4] David J Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

[5] Hossein Alidaee, Eric Auerbach, and Michael P. Leung. Recovering network structure from aggregated relational data using penalized regression. *arXiv preprint arXiv:2001.06052*, 2020.

[6] Attila Ambrus, Markus Mobius, and Adam Szeidl. Consumption risk-sharing in social networks. *American Economic Review*, 104(1):149–82, 2014.

[7] Donald Andrews. Inconsistency of the bootstrap when a parameter is on the boundary of the parameter space. *Econometrica*, 68(2):pp. 399–405, 2000.

[8] Donald WK Andrews et al. Generic uniform convergence. *Econometric Theory*, 8(2):241–257, 1992.

[9] Dena Asta and Cosma Shalizi. Geometric network comparison. *Uncertainty in Artificial Intelligence - Proceedings of the 31st Conference, UAI 2015*, 11 2014.

[10] Dena Asta and Cosma Rohilla Shalizi. Geometric network comparison. *arXiv preprint arXiv:1411.1350*, 11 2014.

[11] Dena Marie Asta and Cosma Rohilla Shalizi. Geometric network comparisons. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 102–110. AUAI Press, 2015.

[12] A. Banerjee, E. Breza, A. Chandrasekhar, E. Duflo, C. Kinnan, and M.O. Jackson. Changes in social network structure in response to exposure to formal credit markets. *Working Paper*, 2020.

[13] A. Banerjee, A. Chandrasekhar, E. Duflo, and M.O. Jackson. Diffusion of microfinance. *Science*, 341(6144):1–7, 2013.

[14] Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. Using gossips to spread information: Theory and evidence from two randomized controlled trials. *The Review of Economic Studies*, 2019.

[15] Shweta Bansal, Jonathan Read, Babak Pourbohloul, and Lauren Ancel Meyers. The dynamic nature of contact networks in infectious disease epidemiology. *Journal of biological dynamics*, 4(5):478–489, 2010.

[16] L.A. Beaman. Social networks and the dynamics of labour market outcomes: Evidence from refugees resettled in the u.s. *Review of Economic Studies*, 79 (1):128–161, 2012.

[17] Lori Beaman, Ariel BenYishay, Jeremy Magruder, and Ahmed Mushfiq Mobarak. Can network theory based targeting increase technology adoption? *Working Paper*, 2016.

[18] Evgeni Begelfor and Michael Werman. The world is not always flot or learning curved manifolds. *School of Engineering and Computer Science, Hebrew University of Jerusalem., Tech. Rep*, 3(7):8, 2005.

[19] Alexander Belton, Dominique Guillot, Apoorva Khare, and Mihai Putinar. A panorama of positivity. i: Dimension free. *Analysis of Operators on Function Spaces*, page 117–165, 2019.

[20] H Russell Bernard, Tim Hallett, Alexandrina Iovita, Eugene C Johnsen, Rob Lyerla, Christopher McCarty, Mary Mahy, Matthew J Salganik, Tetiana Saliuk, Otilia Scutelniciuc, et al. Counting hard-to-count populations: the network scale-up method for public health. *Sexually Transmitted Infections*, 86(Suppl 2):ii11–ii15, 2010.

[21] Peter J. Bickel and Purnamrita Sarkar. Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society B*, 78:253–273, 2015.

[22] P.J. Bickel, A. Chen, and E. Levina. The method of moments and degree distributions for network models. *Annals of Statistics*, 39(5):2280–2301, 2011.

[23] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. New York: Springer-Verlag, 1997.

[24] V Boucer and A Houndetoungan. Estimating peer effects using partial network data. *(Centre de recherche sur les risques les enjeux economiques et les politiques . . .)*, 2020.

[25] Emily Breza. Field experiments, social networks, and development. *The Oxford Handbook on the Economics of Networks, Oxford: Oxford University Press*, 2016.

[26] Emily Breza and Arun G Chandrasekhar. Social networks, reputation, and commitment: evidence from a savings monitors experiment. *Econometrica*, 87(1):175–216, 2019.

[27] Emily Breza, Arun G. Chandrasekhar, Tyler McCormick, and Mengjie Pan. Consistently estimating graph statistics using aggregated relational data. *arXiv preprint arXiv:1908.09881*, 08 2019.

[28] Emily Breza, Arun G Chandrasekhar, Tyler H McCormick, and Mengjie Pan. Using aggregated relational data to feasibly identify network structure without network data. *American Economic Review*, 2020.

[29] J. Cai, A. deJanvry, and E. Sadoulet. Social networks and the decision to insure. *University of Michigan Working Paper*, 2013.

[30] Jing Cai and Adam Szeidl. Interfirm relationships and business performance. *Quarterly Journal of Economics*, 133(3):1229–1282, 2017.

[31] A. Calvo-Armengol. Job contact networks. *Journal of Economic Theory*, 115(1):191–206, 2004.

[32] Antoni Calvó-Armengol, Eleonora Patacchini, and Yves Zenou. Peer effects and social networks in education. *The Review of Economic Studies*, 76(4):1239–1267, 2009.

[33] Djalil Chafaï. Singular value of random matrices, 2009. https://djalil.chafai.net/docs/sing.pdf.

[34] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. *arXiv preprint arXiv: 2005.00545*, 01 2020.

[35] A. Chandrasekhar and R. Lewis. Econometrics of sampled networks. Stanford Working Paper, 2016.

[36] Thomas Chaney. The network structure of international trade. *American Economic Review*, 104(11):3600–3634, 2014.

[37] Sourav Chatterjee. Matrix estimation by universal singular thresholding. *Annals of Statistics*, 43(1), 2015.

[38] Sourav Chatterjee, Persi Diaconis, and Allan Sly. Random graphs with a given degree sequence. *The Annals of Applied Probability*, 21(4):1400 – 1435, 2011.

[39] Sourav Chatterjee, Persi Diaconis, Allan Sly, et al. Random graphs with a given degree sequence. *The Annals of Applied Probability*, 21(4):1400–1435, 2011.

[40] Li Chen, Lizehn Lin, and Jie Zhou. A hypothesis testing for large weighted networks with applications to functional neuroimaging data. *IEEE Access*, 8, 2020.

[41] Sixing Chen and Jukka-Pekka Onnela. A bootstrap method for goodness of fit and model selection with a single observed network. *Scientific Reports*, 2019.

[42] Yoon-Sik Cho, Greg Ver Steeg, Emilio Ferrara, and Aram Galstyan. Latent space model for multi-modal social data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 447–458, 2016.

[43] J.S. Coleman. Social Capital in the Creation of Human Capital. *American Journal of Sociology*, 94(1):S95–S120, 1988.

[44] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[45] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. London: Chapman Hall, 2001.

[46] S. Currarini, M.O. Jackson, and P. Pin. An economic model of friendship: Homophily, minorities, and segregation. *Econometrica*, 77(4):1003–1045, 2009.

[47] L. David and J. Seinfeld. The pool guy. *Seinfeld*, 1995.

[48] M. L. Davison. *Multidimensional Scaling*. New York: Wiley, 1983.

[49] Siemon de Lange, Marcel de Reus, and Martijn Van Den Heuvel. The laplacian spectrum of neural networks. *Frontiers in computational neuroscience*, 7:189, 2014.

[50] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. *Proceedings of Machine Learning Research*, 80, 04 2018.

[51] Thomas A DiPrete, Andrew Gelman, Tyler McCormick, Julien Teitler, and Tian Zheng. Segregation in social networks based on acquaintanceship and trust. *American Journal of Sociology*, 116(4):1234–83, 2011.

[52] Thomas A DiPrete, Andrew Gelman, Tyler McCormick, Julien Teitler, and Tian Zheng. Segregation in Social Networks Based on Acquaintanceship and Trust. *The American Journal of Sociology*, 116:1234–1283, March 2011.

[53] Patrick Doreian. Exploratory social network analysis with Pajek, W. de nooy, A. Mrvar, V. Batagelj. Cambridge University Press, New York (2005). *Social Networks*, 28:269–274, 07 2006.

[54] Morris L. Eaton and David E. Taylor. On weilandt's inequality and its application to the asymptotic distribution of eigenvalues of a random symmetric matrix. *Annals of Statistics*, 19(1):260–271, 1991.

[55] Matthew Elliott, Benjamin Golub, and Matthew O Jackson. Financial networks and contagion. *American Economic Review*, 104(10):3115–53, 2014.

[56] Michael Elowitz, Arnold Levine, Eric Siggia, and Peter Swain. Stochastic gene expression in a single cell. *Science (New York, N.Y.)*, 297:1183–6, 09 2002.

[57] P. Erdős and A. Rényi. On random graphs i. *Publicationes Mathematicae*, 1959.

[58] László Erdős, Horng-Tzer Yau, and Jun Yin. Rigidity of eigenvalues of generalized wigner matrices. *Advances in Mathematics*, 2012.

[59] Satoshi Ezoe, Takeo Morooka, Tatsuya Noda, Miriam Lewis Sabin, and Soichi Koike. Population size estimation of men who have sex with men through the network scale-up method in japan. *PLOS ONE*, 7(1):1–7, 01 2012.

[60] Dennis M Feehan, Mary Mahy, and Matthew J Salganik. The network survival method for estimating adult mortality: Evidence from a survey experiment in rwanda. *Demography*, 54(4):1503–1528, 2017.

[61] Dennis M Feehan, Aline Umubyeyi, Mary Mahy, Wolfgang Hladik, and Matthew J Salganik. Quantity versus quality: A survey experiment to improve the network scale-up method. *American Journal of Epidemiology*, 183(8):747–757, 2016.

[62] Thais Fonseca, Helio Migon, and Marco Ferreira. Bayesian analysis based on the jeffreys prior for the hyperbolic distribution. *Brazilian Journal of Probability and Statistics*, 26, 11 2012.

[63] O. Frank and D. Strauss. Markov graphs. *Journal of the American Statistical Association*, pages 832–842, 1986.

[64] Dinorah Friedmann-Morvinski and Inder Verma. Dedifferentiation and reprogramming: Origins of cancer stem cells. *EMBO reports*, 15, 03 2014.

[65] Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1981.

[66] Prasanna Gai and Sujit Kapadia. Contagion in financial networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2120):2401–2423, 2010.

[67] Chao Gao and John Lafferty. Testing for global network structure using small subgraph statistics. *arXiv preprint arXiv:1710.00862*, 2017.

[68] Lucy L. Gao, Daniela Witten, and Jacob Bien. Testing for association in multiview network data. *arXiv preprint arXiv: 1909.11640*, 2019.

[69] Cedric E. Ginestet, Jun Li, Prakash Balachandran, Steven Rosenberg, and Eric D. Kolaczyk. Hypothesis testing for network data in functional neuroimaging, 2017.

[70] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69:243–268, 04 2007.

[71] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *arXiv preprint arXiv:0912.541*, 2009.

[72] Isobel Claire Gormley and Thomas Brendan Murphy. A mixture of experts

latent position cluster model for social network data. *Statistical methodology*, 7(3):385–405, 2010.

[73] Bryan S Graham. An econometric model of network formation with degree heterogeneity. *Econometrica*, 85(4):1033–1063, 2017.

[74] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[75] G. R. Grimmett and C. J. H. McDiarmid. On colouring random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 77(2):313–324, 1975.

[76] P. J. F. Groenen. *The Majorization Approach to Multidimensional Scaling: Some Problems and Extensions*. Liden, The Netherlands: DSWO, 1993.

[77] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in products of model spaces. 2019.

[78] Mark S Handcock and James Holland Jones. Likelihood-based inference for stochastic models of sexual network formation. *Theoretical population biology*, 65(4):413–422, 2004.

[79] Mark S. Handcock, Adrian E. Raftery, and Jeremy M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

[80] Rachel Heath. Why do firms hire using referrals? evidence from bangladeshi garment factories. *Journal of Political Economy*, 126(4):1691–1746, 2018.

[81] Wolfgang Hofbauer, Laura Forrest, Peter Hollingsworth, and Michelle Hart. Preliminary insights from DNA barcoding into the diversity of mosses colonising modern building surfaces. *Bryophyte Diversity and Evolution*, 38:1, 04 2016.

[82] P.D. Hoff, A.E. Raftery, and M.S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:460:1090–1098, 2002.

[83] Peter Hoff. Random effects models for network data. *Working Paper no. 28, Center for Statistics and the Social Sciences, University of Washington*, January 2003.

[84] Peter Hoff. Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association*, 100:286–295, 2005.

[85] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

[86] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 2002.

[87] Andrew J. Holbrook, Philippe Lemey, Guy Baele, Simon Dellicour, Dirk Brockmann, Andrew Rambaut, and Marc A. Suchard. Massive parallelization boosts big Bayesian multidimensional scaling. *Journal of Computational and Graphical Statistics*, 30:11–24, 2021.

[88] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 1983.

[89] Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981.

[90] David R Hunter, Steven M Goodreau, and Mark S Handcock. Goodness of fit of social network models. *Journal of the American Statistical Association*, 2012.

[91] David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3), feb 2008. Copyright: Copyright 2018 Elsevier B.V., All rights reserved.

[92] Matthew O. Jackson, Tomas R. Rodriguez-Barraquer, and Xu Tan. Social capital and social quilts: Network patterns of favor exchange. *American Economic Review*, 102(5):1857–1897, 2012.

[93] M.O. Jackson. Unraveling peers and peer effects: Comments on goldsmith-pinkham and imbens' "social networks and the identification of peer effects". *Journal of Business and Economic Statistics*, 31:3:270–273, DOI: 10.1080/07350015.2013.794095, 2013.

[94] M.O. Jackson and D. Lopez-Pintado. Diffusion and contagion in networks with heterogeneous agents and homophily. *Network Science*, 1:1:49–67, 2013.

[95] Liwei Jing, Chengyi Qu, Hongmei Yu, Tong Wang, and Yuehua Cui. Estimating the sizes of populations at high risk for HIV: a comparison study. *PloS ONE*, 9(4):e95601, 2014.

[96] Iain M. Johnstone. On the distribution of the largest eigenvalue in principal-components analysis. *The Annals of Statistics*, 29(2):295 – 327, 2001.

[97] Marcus Kaiser and Claus C Hilgetag. Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems. *PLoS computational biology*, 2(7), 2006.

[98] Waldemar Karwowski, Farzad Vasheghani Farahani, and Nichole Lighthall. Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review. *Frontiers in Neuroscience*, 13:585, 2019.

[99] Martin Keller-Ressel and Stephanie Nargang. Hydra: a method for strain-minimizing hyperbolic embedding of network- and distance-based data. *Journal of Complex Networks*, 8, 02 2020.

[100] Wilheml Killing. Ueber die clifford-klein'schen raumformen. *Mathematische Annalen*, 39(2):257–278, 1891.

[101] Peter D Killworth, Eugene C Johnsen, H Russell Bernard, Gene Ann Shelley, and Christopher McCarty. Estimating the size of personal networks. *Social Networks*, 12(4):289–312, 1990.

[102] Peter D. Killworth, Christopher McCarty, H. Russell Bernard, Eugene C. Johnsen, John Domini, and Gene A. Shelley. Two interpretations of reports of knowledge of subpopulation sizes. *Social Networks*, 25:141–160, 2003.

[103] Peter D Killworth, Christopher McCarty, H Russell Bernard, Gene Ann Shelley, and Eugene C Johnsen. Estimation of seroprevalence, rape, and homelessness in the united states using a social network approach. *Evaluation Review*, 22(2):289–308, 1998.

[104] Cynthia Kinnan and Robert Townsend. Kinship and financial networks, formal financial access, and risk reduction. *The American Economic Review*, 102(3):289–293, 2012.

[105] Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature Communications*, 11, 06 2020.

[106] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[107] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[108] M. A. Langston, Y. Zhang, E. J. Chesler, N. F. Samatova, N. E. Baldwin, and F. N. Abu-Khzam. Genome-scale computational approaches to memory-intensive applications in systems biology. In *SC Conference*, page 12, Los Alamitos, CA, USA, nov 2005. IEEE Computer Society.

[109] J. O. Lee and J. Yin. A necessary and sufficient condition for edge universality of wigner matrices. *Duke Mat. J*, pages 117–173, 2014.

[110] Jing Lei. A goodness-of-fit test for stochastic block models. *The Annals of Statistics*, 44(1):401 – 424, 2016.

[111] Peter Lenk. Simulation pseudo-bias correction to the harmonic mean estimator of integrated likelihoods. *Journal of Computational and Graphical Statistics*, 18(4):941–960, 2009.

[112] M. Leung. Two-step estimation of network-formation models with incomplete information. *mimeo*, 2013.

[113] Maxwell CK Leung, Phillip L Williams, Alexandre Benedetto, Catherine Au, Kirsten J Helmcke, Michael Aschner, and Joel N Meyer. Caenorhabditis elegans: an emerging model in biomedical and environmental toxicology. *Toxicological sciences*, 106(1):5–28, 2008.

[114] Tianxi Li, Elizaveta Levina, and Ji Zhu. Network cross-validation by edge sampling. *arXiv preprint arXiv:1612.04717*, 2020.

[115] Shane Lubold, Arun Chandrasekhar, and Tyler McCormick. Identifying latent space geometry of network formation models via analysis of curvature. *arXiv preprint arXiv:2012.10559*, 2020.

[116] Shane Lubold, Arun Chandrasekhar, and Tyler McCormick. Identifying the latent space geometry of network models through analysis of curvature. *arXiv preprint arXiv:2012.10559*, 2020.

[117] Shane Lubold, Bolun Liu, and Tyler H. McCormick. Spectral goodness-of-fit tests for complete and partial network data, 2021.

[118] Shane Lubold, Bolun Liu, and Tyler H. McCormick. Spectral goodness-of-fit tests for complete and partial network data, 2021.

[119] Margus Lukk, Misha Kapushesky, Janne Nikkilä, Helen Parkinson, Angela Goncalves, Wolfgang Huber, Esko Ukkonen, and Alvis Brazma. A global map of human gene expression. *Nature Biotechnology*, 28:322–4, 04 2010.

[120] Wei Luo and Bing Li. Combining eigenvalues and variation of eigenvectors for order determination. *Biometrika*, 103:875–887, 2016.

[121] Zhuang Ma, Zongming Ma, and Hongsong Yuan. Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4):1–67, 2020.

[122] D. MacKay. Probabilistic multidimensional scaling: An anisotropic model for distance judgements. *Marketing Science*, 5:325–334, 1989.

[123] Christopher McCarty, Peter D Killworth, H Russell Bernard, Eugene C Johnsen, and Gene A Shelley. Comparing two methods for estimating network size. *Human organization*, 60(1):28–39, 2001.

[124] Tyler H McCormick, Matthew J Salganik, and Tian Zheng. How many people do you know?: Efficiently estimating personal network size. *Journal of the American Statistical Association*, 105(489):59–70, 2010.

[125] Tyler H McCormick and Tian Zheng. Latent surface models for networks using aggregated relational data. *Journal of the American Statistical Association*, 110(512):1684–1695, 2015.

[126] Seth A Myers and Jure Leskovec. The bursty dynamics of the twitter information network. In *Proceedings of the 23rd international conference on World wide web*, pages 913–924, 2014.

[127] Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. *arXiv preprint arXiv: 1902.02992*, 2019.

[128] Whitney K. Newey. Uniform convergence in probability and stochastic equicontinuity. *Econometric Research Program Research Memorandum No. 342*, 1989.

[129] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of Econometrics*, 4:2111–2245, 1994.

[130] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *arXiv preprint arXiv: 1705.08039*, 05 2017.

[131] Man-Suk Oh and Adrian E. Raftery. Bayesian multidimensional scaling and choice of dimension. *Journal of the American Statistical Association*, 96:1031–1044, 2001.

[132] Man-Suk Oh and Adrian E Raftery. Bayesian multidimensional scaling and choice of dimension. *Journal of the American Statistical Association*, 96(455):1031–1044, 2001.

[133] Marjorie Oleksiak, Gary Churchill, and Douglas Crawford. Variation in gene expression within and among natural populations. *Nature Genetics*, 32:261–6, 11 2002.

[134] Barrett O'Neill. *Semi-Riemannian Geometry with Applications to Relativity*, volume 103. Academic press, 1983.

[135] Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):437–461, 2015.

[136] Sarah Ouadah, Stéphane Robin, and Pierre Latouche. Degree-based goodness-of-fit tests for heterogeneous random graph models : independent and exchangeable cases. *arXiv preprint arXiv: 1507.08140*, 2019.

[137] Tiago P. Peixoto. Disentangling homophily, community structure, and triadic closure in networks. *Physical Review X*, 12(1), jan 2022.

[138] Steven E Petersen and Olaf Sporns. Brain networks and cognitive architectures. *Neuron*, 88(1):207–219, 2015.

[139] Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

[140] Dimitris N. Politis and Joseph P. Romano. Large sample confidence regions based on subsamples under minimal assumptions. *Annals of Statistics*, 22(4):2031–2050, 1994.

[141] Benedikt M. Potscher and Ingmar Prucha. A uniform law of large numbers for dependent and heterogeneous data processes. *Econometrica*, May 1989.

[142] Anoop Praturu and Tatyana Sharpee. A Bayesian approach to hyperbolic multidimensional scaling. *bioRxiv*, 2022.

[143] Adrian Raftery, Xiaoyue Niu, Peter Hoff, and Ka Yee Yeung. Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, 21, 10 2012.

[144] Adrian E Raftery and Steven M Lewis. Comment: one long run with diagnostics: implementation strategies for Markov chain Monte Carlo. *Statistical science*, 7(4):493–497, 1992.

[145] Adrian E. Raftery, Michael A. Newton, Jaya M. Satagopan, and Pavel N. Krivitsky. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Bayesian Analysis*, pages 1–45, 2007.

[146] Arjun Raj and Alexander Oudenaarden. Nature, nurture, or chance: Stochastic gene expression and its consequences. *Cell*, 135:216–26, 11 2008.

[147] Stefano Recanatesi, Matthew Farrell, Guillaume Lajoie, Sophie Deneve, Mattia Rigotti, and Eric Shea-Brown. Predictive learning extracts latent space representations from sensory observations. *bioRxiv*, page 471987, 2019.

[148] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39(4):1878–1915, 2011.

[149] Daniel M Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 695–704, 2011.

[150] Evan Rosenman. Some new results for poisson binomial models, 2019.

[151] Evan Rosenman and Nitin Viswanathan. Using poisson binomial glms to reveal voter preferences, 2018.

[152] Evan Sadler. Seeding a simple contagion. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 247–248, New York, NY, USA, 2022. Association for Computing Machinery.

[153] Michael Salter-Townshend and Tyler H McCormick. Latent space models for multiview network data. *The Annals of Applied Statistics*, 11(3):1217, 2017.

[154] Michael Salter-Townshend and Tyler H. McCormick. Latent space models for multiview network data. *Annals of Applied Statistics*, 2017.

[155] I. J. Schoenberg. Remarks to maurice frechet's article "sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de hilbert. *Annals of Mathematics*, 36(3):724–732, 1935.

[156] O Scutelniciuc. Network scale-up method experiences: Republic of Kazakhstan. *Consultation on estimating population sizes through household surveys: Successes and challenges (New York, NY)*, 2012.

[157] Daniel K Sewell and Yuguo Chen. Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657, 2015.

[158] Cosma Rohilla Shalizi and Dena Asta. Consistency of maximum likelihood for continuous-space network models. *arXiv preprint arXiv:1711.02123*, 2017.

[159] Jesse Shore and Benjamin Lubin. Spectral goodness of fit for network models. *Social Networks*, 2015.

[160] Anna L Smith, Dena M Asta, Catherine A Calder, et al. The geometry of continuous latent space models for network data. *Statistical Science*, 34(3):428–453, 2019.

[161] T.A.B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):240, 2002.

[162] Y. Takane and J. D. Caroll. Nonmetric maximum likelihood multidimensional scaling from directional rankings of similarities. *Psychometrika*, 46:389–405, 1981.

[163] Terrence Tao. Topics in random matrix theory (graduate studies in mathematics). *American Mathematical Society*, 2012.

[164] A.W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

[165] Rufin VanRullen and Leila Reddy. Reconstructing faces from fmri patterns using deep generative neural networks. *Communications biology*, 2(1):1–10, 2019.

[166] Y. H. Wang. On the number of successes in independent trials. *Statistica Sinica*, pages 295–312, 1993.

[167] S. Wasserman and P. Pattison. Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp. *Psychometrika*, 61(3):401–425, 1996.

[168] Melanie Weber and Maximilian Nickel. Curvature and representation learning: Identifying embedding spaces for relational data. In *NeurIPS Relational Representation Learning*, 2018.

[169] E. P. Wigner. On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 1958.

[170] Steven Wilkins-Reeves and Tyler McCormick. Asymptotically normal estimation of local latent network curvature. *arXiv preprint 2211.11673*, 2022.

[171] Richard C. Wilson, Edwin R. Handcock, Elżbieta Pekalska, and Robert P.W. Duin. Spherical and hyperbolic embeddings of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2255–2269, 2014.

[172] Wenkai Xu and Gesine Reinert. A stein goodness-of-fit test for exponential random graph models. *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021.

[173] Xiaoran Yan, Cosma Shalizi, Jacob E Jensen, Florent Krzakala, Cristopher Moore, Lenka Zdeborová, Pan Zhang, and Yaojia Zhu. Model selection for degree-corrected block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(5):P05007, 2014.

[174] Anna K. Yanchenko and Peter D. Hoff. Hierarchical multidimensional scaling for the comparison of musical performance styles. *The Annals of Applied Statistics*, 14(4), Dec 2020.

[175] Xingchen Yu and Abel Rodriguez. Spatial voting models in circular spaces: A case study of the u.s. house of representatives. *Annals of Applied Statistics*, 15(4):1897–1922, 2019.

[176] Wayne Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33, 11 1976.

[177] Tian Zheng, Matthew J Salganik, and Andrew Gelman. How many people do you know in prison? using overdispersion in count data to estimate social structure in networks. *Journal of the American Statistical Association*, 101(474):409–423, 2006.

[178] Yuansheng Zhou and Tatyana Sharpee. Hyperbolic geometry of gene expression. *iScience*, 24:102225, 02 2021.

[179] Pavlos Zoubouloglou, Eduardo García-Portugués, and J. S. Marron. Scaled torus principal component analysis. *arXiv preprint arXiv:2110.04758*, 2021.