Statistical Learning and Modeling with Graphs and Networks

Zeyu Wei

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

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Yen-Chi Chen, Co-chair
Tyler H. McCormick, Co-chair
Marina Meila

Program Authorized to Offer Degree:
Statistics

University of Washington

Abstract

Statistical Learning and Modeling with Graphs and Networks

Zeyu Wei

Co-Chairs of the Supervisory Committee:

Yen-Chi Chen
Tyler H. McCormick

Department of Statistics

Graph, consisting of a set of vertices and a set of edges, is a natural tool to study relations. From a geometric perspective, relations between data points reveal information about the underlying structure, and a graph as a geometric object can not only visualize but also mathematical characterize such geometric structures in the data. From a network perspective, graphs can also model connections between different units and have applications in various fields such as epidemiology, econometrics, sociology, biology, and astronomy. We first take advantage of graphs from a geometric perspective and propose a data analysis framework that constructs weighted graphs, called skeletons, to encode the geometric structures in the data and utilize the learned graphs to assist downstream analysis tasks such as clustering and regression.

For clustering, we introduce a density-aided method that can detect clusters in multivariate and even high-dimensional data with irregular shapes. To bypass the curse of dimensionality, we propose surrogate density measures that are less dependent on the dimension and have intuitive geometric interpretations. The clustering framework constructs a concise graph representation of the given data as an intermediate step and can be thought of as a combination of prototype methods, density-based clustering, and hierarchical clustering. We show by theoretical analysis and empirical studies that skeleton clustering leads to reliable clusters in multivariate and high-dimensional scenarios.

For regression tasks, we propose a novel framework specialized for covariates concentrated around some low-dimension geometric structures. The proposed framework first learns a graph representation of the covariates which encodes the geometric structures. Then we apply nonparametric regression techniques to estimate the regression function on the skeleton graph, which, notably, bypasses the curse of dimensionality. We derive statistical and computational properties of the proposed regression framework and use simulations and real data examples to illustrate its effectiveness. Our framework has the advantage that predictors for distinct geometric structures can be accounted for and is robust to additive noise and noisy observations.

Graphs are widely used to represent networks of connections and serve as a helpful tool in modeling real-world diffusion processes. Network diffusion models are used to study things like disease transmission, information spread, and technology adoption. However, small amounts of mismeasurement are extremely likely in the networks constructed to operationalize these models. We show that estimates of diffusions are highly non-robust to this measurement error. First, we show that even when measurement error is vanishingly small, such that the share of missed links is close to zero, forecasts about the extent of diffusion will greatly underestimate the truth. Second, a small mismeasurement in the identity of the initial seed generates a large shift in the locations of the expected diffusion path. We show that both of these results still hold when the vanishing measurement error is only local in nature. Such non-robustness in forecasting exists even under conditions where the basic reproductive number is consistently estimable. Possible solutions, such as estimating the measurement error or implementing widespread detection efforts, still face difficulties because the number of missed links is so small. Finally, we conduct Monte Carlo simulations on simulated networks, and real networks from three settings: travel data from the COVID-19 pandemic in the western US, a mobile phone marketing campaign in rural India, and an insurance experiment in China.

ACKNOWLEDGEMENT

I want to express my deepest gratitude to my supervisors, Prof. Yen-Chi Chen and Prof. Tyler H. McCormick, for providing me with invaluable guidance and support throughout my journey as a Ph.D. student. Their knowledge, expertise, insightful ideas, and, foremost, the mindset of curiosity to explore the unknown and uncertainty have fundamentally shaped my learning experience and research endeavors. Their personal advice is also invaluable and helps guide my path forward in my career development. I am fortunate to have had such supportive and intelligent mentors.

I would like to thank my committee members, Prof. Marina Meila and Prof. John M. Lee, for their invaluable feedback, suggestions, and critiques that helped me refine my research and provided me with a broader perspective on my work.

I am grateful to the Department of Statistics at the University of Washington for offering such a great Ph.D. program for me to develop into a statistician. In particular, I want to thank Prof. Yen-Chi Chen for his great lectures on Statistical Machine Learning. The insightful and intuitive explanations he gave on various methods along with his enthusiasm for such topics fundamentally shaped my interest in statistical machine learning and well-prepared me with the background to start working on research topics in this realm with him. I am also thankful to Prof. Marina Meila for her class on Machine Learning for Big Data which introduces many interesting methods along with theoretical results which significantly deepened my knowledge in the machine learning realm. I would also love to thank Prof. Tyler H. McCormick for his instructions on Network Analysis which not only introduces

technical models but also a broad perspective on social and economic topics. The required Mathematical Statistics sequence taught by Prof. Alex Luedtke and Prof. Marco Carnone helped me build a solid background in mathematical statistics, and the Advance Probability sequence taught by Prof. Jon A Wellner honed my mathematical skills. The Prelim Exam preparation course and the consulting class are also invaluable experiences for my academic and professional development, and I genuinely appreciate the Stat department for putting all of these together into such a meaningful Ph.D. program.

I am fortunate and incredibly grateful for the opportunity to be part of the UW Stat department. In particular, I truly appreciate the effort made by our department chair, Prof. Abel Rodriguez, in gathering feedback, improving guidelines, and dedicated supervision of the department administration to ensure we all can enjoy the time with UW Stats. I also want to thank the staff, including Ellen Reynolds, Kristine Chan, Vickie J Graybeal Tracy Pham, and Veronica Bae, for their assistance and support during my studies.

I would like to express my appreciation to my collaborators Prof. Arun G. Chandrasekhar, Prof. Paul Goldsmith-Pinkham, and Samuel Thau. Working with them has been an exceptional experience where I have gained valuable insights from socio-economic perspectives.

I owe a debt of gratitude to my parents for always supporting me and my friends for their understanding and help. I cannot state how grateful I am to my love, Yilun Xing, for her unwavering support and love and for always being there with me along this memorable journey.

DEDICATION

To My Parents And To My Love Yilun

Contents

List of Figures

List of Tables

Chapter 1

Overview

Graphs, consisting of a set of vertices and a set of edges, have many applications in various research fields such as machine learning, network analysis, and causal inference. Our research focuses on two perspectives of graphs. On one hand, graphs, as geometric objects, can help not only visualize but also mathematical characterize the geometric structures in data. On the other hand, from a network perspective, graphs model relations or connections between different units and have applications in various fields such as epidemiology, sociology, economy, biology, and chemistry.

We first take advantage of graphs from a geometric perspective. Finding meaningful geometric or topological descriptions of datasets is of great interest in virtue of uncovering hidden structural information, particularly when data in a high-dimensional Euclidean space is assumed to lie on a low-dimensional manifold. This is a major focus of Topological Data Analysis (Wasserman, 2016) and Manifold Learning, in which graphs play an important role. For nonlinear dimension reduction techniques such as Laplacian Eigenmaps (Belkin

and Niyogi, 2003) and Diffusion Maps (Coifman and Lafon, 2006), a weighted graph is first constructed based on local neighborhoods, some versions of graph Laplacian is constructed, and spectral analysis of the graph Laplacian leads to the desired results. Latter works have shown the convergence of such discrete graph Laplacian to the Laplace-Beltrami operator (Belkin et al., 2006a; Belkin and Niyogi, 2008; Berry and Harlim, 2014; Berry and Sauer, 2019), which adds topological interpretations to such approaches.

Geometric data is also attracting attention from the deep learning field. Under the similar principle as Felix Klein's "Erlangen Programme" (Klein, 1893) that characterizes geometries through symmetry groups, Geometric Deep Learning (Bronstein et al., 2017; Battaglia et al., 2018; Bronstein et al., 2021) derives neural network architectures through group invariance and equivariance. Under this general blueprint, one approach encodes geometric information through graphs and performs learning tasks with the Graph Neural Networks (GNNs) (Veličković et al., 2018; Xu et al., 2019; Chamberlain et al., 2021a,b; Bouritsas et al., 2022). In particular, Wang et al. (2019) dynamically builds neighborhood graphs from point clouds and aggregates edge features through layers for classification and segmentation tasks. Kazi et al. (2022) learns the probabilistic latent graphs in the deep learning architecture for optimal classification.

One direction of our research is to use graphs to extract the underlying geometric information in a dataset. Unlike the approaches that set graph vertices as individual points in the data point cloud, we propose a data analysis framework that constructs representational weighted graphs, called skeletons, to encode the geometric structures in data with a small

number of vertices, and utilizes the learned graph to assist the downstream analysis tasks such as clustering and regression.

In addition to representing geometric information, a graph is a structure of connections, which makes it natural to represent various networks, with the social network or the contact network being examples. Particularly, due to the advancement in mobile communication technology, the collection of contact network data, or at least some proxies for it, becomes feasible, and studies have directly incorporated such data to model epidemic behaviors. Some early works collect mobility data based on phone calls and text records to model disease transmission behaviors (Wesolowski et al., 2012; Bengtsson et al., 2015; Engebretsen et al., 2020; Milusheva, 2020). Mobility networks derived from commute flow data are also used as proxies to contact networks for epidemic modeling (Fajgelbaum et al., 2021a; Alsing et al., 2020). Facing the challenge of the global pandemic, the Google COVID-19 Aggregated Mobility Research Dataset has become a major source to drive research in epidemic modeling (Kapoor et al., 2020; Ruktanonchai et al., 2020; Venkatramanan et al., 2021).

Despite the importance of contact data in modeling epidemic behavior, collecting contact networks is still difficult, and, as described above, research teams use proxies for contact networks, with mismeasurements inevitable. Chandrasekhar et al. (2021) demonstrates that small misalignment of the model with the underlying network of interactions necessitates non-trivial failure of local targeting policy guided by epidemiological models. Changes in contact networks have substantial implications for disease transmissions, which raises concern over the robustness of epidemic models in this regard. To address one aspect of this concern,

we assess the sensitivity of the diffusion models, in terms of policy decisions, to missingness about the underlying contact graph.

In Chapter 2, we use graphs to represent the data structures and perform clustering. We introduce a density-aided method called Skeleton Clustering that can detect clusters in multivariate and even high-dimensional data with irregular shapes. To bypass the curse of dimensionality, we propose surrogate density measures that are less dependent on the dimension but have intuitive geometric interpretations. The clustering framework constructs a concise representation of the given data as an intermediate step and can be thought of as a combination of prototype methods, density-based clustering, and hierarchical clustering. We show by theoretical analysis and empirical studies that skeleton clustering leads to reliable clusters in multivariate and high-dimensional scenarios.

In Chapter 3, we use graphs to encode the geometric information in the covariate space and to fit regression functions. We propose a novel framework specialized for covariates concentrated around some low-dimension geometric structures. The proposed framework first learns a graph representation of the covariates, which we call the skeleton, to summarize the geometric structures. Then we apply nonparametric regression techniques to estimate the regression function on the skeleton, which, notably, bypasses the curse of dimensionality. We derive statistical and computational properties of the proposed regression framework and use simulations and real data examples to illustrate its effectiveness. Our framework has the advantage that predictors from distinct geometric structures can be accounted for and is robust to additive noise and noisy observations.

In Chapter 4, we focus on the non-robustness of diffusion estimates on networks with measurement error. We show that estimates of diffusions are highly non-robust to this measurement error. First, we show that even when measurement error is vanishingly small, such that the share of missed links is close to zero, forecasts about the extent of diffusion will greatly underestimate the truth. Second, a small mismeasurement in the identity of the initial seed generates a large shift in the locations of the expected diffusion path. We show that both of these results still hold when the vanishing measurement error is only local in nature. Such non-robustness in forecasting exists even under conditions where the basic reproductive number is consistently estimable. Possible solutions, such as estimating the measurement error or implementing widespread detection efforts, still face difficulties because the number of missed links is so small. Finally, we conduct Monte Carlo simulations on simulated networks, and real networks from three settings: travel data from the COVID-19 pandemic in the western US, a mobile phone marketing campaign in rural India, and an insurance experiment in China.

Chapter 2

Skeleton Clustering: Dimension-Free Density-Aided Clustering

2.1 Introduction

Density-based clustering (Azzalini and Torelli, 2007; Menardi and Azzalini, 2014; Chacón, 2015) is a popular framework for grouping observations into clusters defined based on the underlying probability density function (PDF). In practice, when the PDF is usually unknown, it is estimated via the random sample, and the estimated PDF is then used to obtain the resulting clusters. Many clustering methods have been proposed within the framework of density-based clustering. The mode clustering (Li et al., 2007; Chacón and Duong, 2013; Chen et al., 2016) finds clusters via the local modes of the underlying PDF. When the kernel density estimator (KDE) is used for density estimation, the mode clustering can be done easily via the mean-shift algorithm (Fukunaga and Hostetler, 1975; Cheng, 1995;

Carreira-Perpinán, 2015). Another famous density-based clustering approach is the level-set clustering (Cuevas et al., 2000, 2001; Mason et al., 2009; Rinaldo et al., 2012), which creates clusters as the connected components of high-density regions. The well-known DBSCAN (Density-Based Spatial Clustering of Applications with Noise) method (Ester et al., 1996) is also a special case of level-set clustering. Moreover, the cluster tree (Stuetzle and Nugent, 2010; Chaudhuri and Dasgupta, 2010; Chaudhuri et al., 2014; Eldridge et al., 2015; Kim et al., 2016) is a density-based clustering approach combining information from both modes and level sets. This method creates a tree structure with each leaf representing a mode and the tree describes the evolution of level-set clusters at different density levels.

Compared to the classical k-means clustering (Lloyd, 1982; Hartigan and Wong, 1979; Pollard, 1982) and the model-based clustering methods (Fraley and Raftery, 2002), a density-based clustering approach is capable of finding clusters with irregular shapes and gives an intuitive interpretation based on the underlying PDF. Furthermore, defining clusters based on the density function makes it possible to view the clustering problem as an estimation problem: the clusters from the true PDF are the parameters of interest and the estimated clusters are sample quantities utilized for approximation.

Although density-based clustering enjoys many advantages, it has a fundamental limitation: the curse of dimensionality. Because a density-based clustering method often involves a density estimation step, it does not scale well with the dimension. Specifically, the convergence rate of a density estimator is $O_P\left(n^{-\frac{2}{4+d}}\right)$ under usual smoothness conditions (Scott, 2015; Wasserman, 2006a), which is slow when d is large. To overcome the curse of dimensionality

and to apply density-based clustering to high-dimensional data, we borrow the idea of merging a large number of k-means clusters from (Peterson et al., 2018; Fred and Jain, 2005; Maitra, 2009; Baudry et al., 2010; Shin et al., 2019) and propose density-aided similarity measures suitable for high-dimensional settings.

The idea of merging prototypes has also attracted great attention from model-based clustering to overcome the limitations of parametric assumptions. In particular, there are several methods based on merging Gaussian-mixture models (Hennig, 2010), such as Dip test approach (Hartigan and Hartigan, 1985), ridgeline elevation (Ray and Lindsay, 2005), misclassification method (Tibshirani and Walther, 2005), multi-layer approach (Li, 2005), entropy-based method (Baudry et al., 2010), level set-based method (Scrucca, 2016), and modal clustering (Chacón, 2019). The work by Aragam et al. (2020) reconstructs a nonparametric mixture model by fitting the data with a large number of general nonparametric mixture components and then partitions them into a small number of final clusters.

Our idea can be summarized as follows. We first find a large set of protoclusters (called *knots*) by running k-means clustering. Nearby knots are then connected by edges to form a graph that we call the *skeleton*. The similarities between connected knots are measured by density-aided criteria that are estimable even in high dimensions. Finally, we merge knots according to a linkage criterion to create the final clusters. Because the construction involves creating a *skeleton* representation of the data, we call this method *Skeleton Clustering*.

To illustrate the limitations of the classical approaches and to highlight the effectiveness of skeleton clustering, we conduct a simple simulation in Figure 2.1. It is a $d = 200$ dimensional

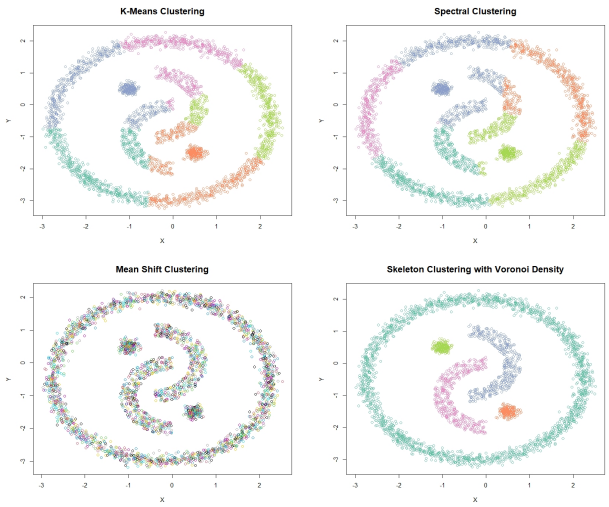


Figure 2.1: Yinyang Data with dimension 200. On the bottom right is the clustering result of the skeleton clustering with the proposed Voronoi density similarity measure.

data consisting of five components with non-spherical shapes. The actual structure is in 2-dimensional space as illustrated in Figure 2.1. We add Gaussian noises in other dimensions to make it a $d = 200$ dimensional data (see Section 2.5 for more details). Traditional k-means and spectral clustering fail to find the five components and the mean shift algorithm cannot form clusters due to the high dimensionality of the data. However, our proposed method (bottom-right panel) can successfully recover the underlying five components.

Outline. In section 2.2, we describe the skeleton clustering framework. In section 2.3, we introduce similarity measures that can be utilized in the skeleton clustering framework. In section 2.4, we provide some consistency results of the sample similarity measures and the clustering performance guarantee. In section 2.5, we present simulation results to demonstrate the effectiveness of skeleton clustering in dealing with different data scenarios and to guide some choices in the framework for applications. In section 2.6, we test the performance of skeleton clustering on real datasets. In section 2.7, we conclude the paper

and point out some directions for future research.

2.2 Skeleton Clustering Framework

Algorithm 1 Skeleton clustering

Input: Observations $X_1, \cdots, X_n$, final number of clusters S.

1. **Knot construction.** Perform k-means clustering with a large number of k; the centers are the knots (Section 2.2.1).
2. **Edge construction.** Apply approximate Delaunay triangulation to the knots (Section 2.2.2).
3. **Edge weights construction.** Add weights to each edge using either Voronoi density, Face density, or Tube density similarity measure (Section 2.3).
4. **Knots segmentation.** Use linkage criterion to segment knots into S groups based on the edge weights (Section 2.2.4).
5. **Assignment of labels.** Assign a cluster label to each observation based on which knot group the nearest knot belongs (Section 2.2.5).

In this section we formally introduce the skeleton clustering framework. Let $\mathbb{X} = \{X_1, \ldots, X_n\}$ be a random sample from an unknown distribution with density p supported on a compact set $\mathcal{X} \in \mathbb{R}^d$. The goal of clustering is to partition $\mathbb{X}$ into clusters $\mathbb{X}_1, \ldots \mathbb{X}_S$, where S is the final number of clusters.

A summary of the skeleton clustering framework is provided in Algorithm 1. [1] Figure 2.2 illustrates the overall procedure of the skeleton clustering method. Starting with a collection of observations (panel (a)), we first find knots, the representative points of the entire data (panel (b)). Then we compute the corresponding Voronoi cells induced by the knots (panel

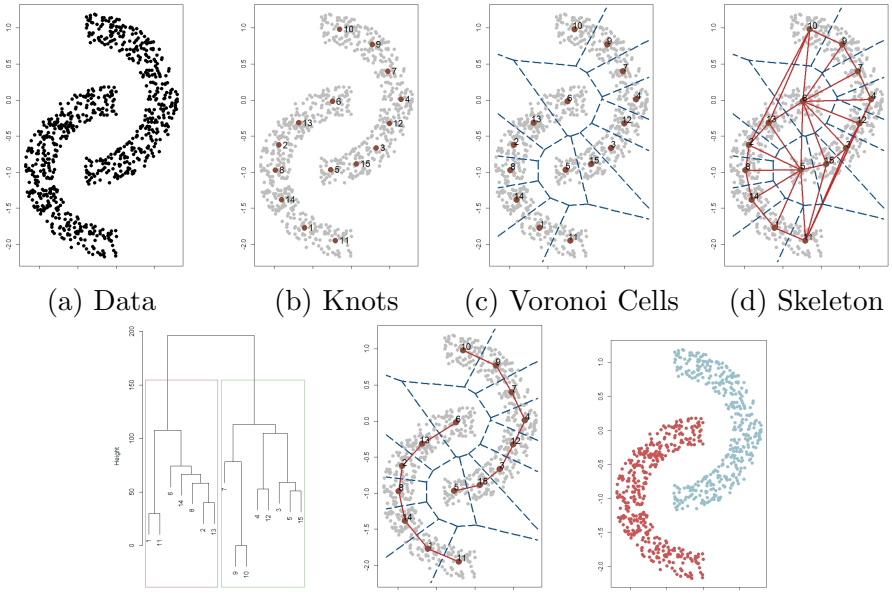
[1] See https://cse512-22sp.pages.cs.washington.edu/SkeletonVis/ for interactive visualizations of the framework.

(c)) and the edges associating the nearby Voronoi cells (panel (d)). For each edge in the graph, we compute a density-aided similarity measure that quantifies the closeness of each pair of knots. For the next step, we segment knots into groups based on a linkage criterion (single linkage in this example), leading to the dendrogram in panel (e). Finally, we choose a threshold that cuts the dendrogram into $S = 2$ clusters (panel (f)) and assign a cluster label to each observation according to the knot-cluster that it belongs to (panel (g)).

In summary, the skeleton clustering consists of the following five steps: (1) Knots construction, (2) Edges construction, (3) Edge weights construction, (4) Knots segmentation, and (5) Assignment of labels. In what follows in this section, we provide a detailed description of each step except Step 3. Step 3 is the key step in our clustering framework where we incorporate the information from the underlying density for clustering in a less dimension-dependent way and we defer the detailed discussion of Step 3 to Section 2.3 and Section 2.4. We include a short analysis of the computational complexity of our skeleton clustering framework in Appendix A.

2.2.1 Knots Construction

The construction of knots is a step aiming at finding representative points in the data that can help measure similarities between regions in the later stage. The knots can be viewed as landmarks inside the data where we can shift our focus from the entire data to these local locations. A simple but reliable approach for constructing knots is the k-means algorithm. We apply the k-means algorithm with a large number $k \gg S$ the desired number of final



(a) Data (b) Knots (c) Voronoi Cells (d) Skeleton

(e) Dendrogram (f) Segmentation (g) Clustering

Figure 2.2: Skeleton Clustering illustrated by Two Moon Data (d=2).

clusters, and this procedure behaves like overfitting the k-means. Notably, we do not use the k-means procedure to obtain final clustering, but instead, we use it as an intermediate step to find concise representations of the original data.

The number of knots k is a key parameter in the knots construction step. It controls the trade-off between the quality of the data representation and the reliability of each knot. More knots can give a better representation of the data, but, if we have too many knots, the number of observations per knot will be small, so the uncertainty in estimation in the later stage will be large. We find that a simple reference rule for k to be around $\sqrt{n}$ works well in our empirical studies (Section F). In practice, it is also advisable to prune knots with a small number of corresponding observations because the density-aided weights (in Step 3, Section 2.3) are estimated locally by the data belonging to each pair of knots. Knots with a few data points can lead to unstable similarity measurements and unreliable

final clustering. Moreover, to take care of observations in the low-density areas that could cause problems for the k-means clustering, one may first pre-process or denoise the data by removing observations in the low-density area and then apply the k-means clustering to find out the knots.

In this work, we use overfitting k-means as the default way for knots construction, but there are alternative approaches to find knots such as subsampling, the coreset construction methods (Bachem et al., 2017; Turner et al., 2020), and the Self-Organizing Maps (SOM) (Heskes, 2001). We show in Appendix F that the SOM can also be used to find knots but requires more careful treatments such as removing knots with few or even no observations and the performance is slightly worse than that of the overfitting k-means. The k-medians algorithm can be another alternative method but it gave an unstable result when the dimension is large. Therefore, we choose to use the overfitting k-means algorithm in this work and recommend using it in practice.

Remark 1. Since the k-means algorithm does not always find the global optimum, we repeat it many times with random initial points (generally $1,000$ times or more) and choose the one with the optimal objective function. This works well for all of our numerical analyses. Moreover, since we are only using k-means as a tool to find a useful representation, we do not need to find the actual global optimum. All we need is a set of knots forming a useful representation.

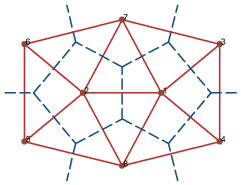


Figure 2.3: Voronoi Tessellation as blue dashed lines and Delaunay Triangulation by red solid lines.

2.2.2 Edges Construction

With the constructed knots, our next step is to find the edges connecting them. Let $c_1, \cdots, c_k$ be the given knots and we use $\mathcal{C} = \{c_1, \cdots, c_k\}$ to denote the collection of them. We add an edge between a pair of knots if they are neighbors, with the neighboring condition being that the corresponding Voronoi cells (Voronoi, 1908) share a common boundary. The Voronoi cell, or Voronoi region, $\mathbb{C}_j$, associated with a knot c_j is the set of all points in $\mathcal{X}$ whose distance to c_j is the smallest compared to other knots (See Figure 2.3). That is,

$$\mathbb{C}_j = \{x \in \mathcal{X} : d(x, c_j) \leq d(x, c_\ell) \quad \forall \ell \neq j\}, \tag{2.2.1}$$

where $d(x, y)$ is the usual Euclidean distance. Therefore, we add an edge between knots (c_i, c_j) if $\mathbb{C}_i \cap \mathbb{C}_j \neq \emptyset$. Such resulting graph is the Delaunay triangulation (Delaunay, 1934) of the set of knots $\mathcal{C}$ and we denote it as $DT(\mathcal{C})$. In a nutshell, the skeleton graph in our framework is given by the Delaunay triangulation of $\mathcal{C}$.

The Delaunay triangulation graph is conceptually intuitive and appealing and is utilized by some clustering methods to identify connected components (Azzalini and Torelli, 2007; Scrucca, 2016), but empirically the computational complexity of the exact Delaunay triangulation algorithm has an exponential dependence on the ambient dimension d (Amenta et al., 2007;

Chazelle, 1993). Given our multivariate and even high-dimensional data setting, exact Delaunay triangulation is empirically unfavorable. Therefore, in practice, we approximate the exact Delaunay Triangulation with $\hat{DT}(\mathcal{C})$ by examining the 2-nearest knots of the sample data points. The key observation is that, if the Voronoi cells of two knots c_i, c_j share a boundary, there is a non-empty region of points whose 2-nearest knots are c_i, c_j. Consequently, for approximation, we query the two nearest knots for each data point and have an edge between c_i, c_j if there is at least one data point whose two nearest neighbors are c_i, c_j. The complexity of the neighbor search depends linearly on the dimension d, which is desirable for high-dimensional setting (Weber et al., 1998), and this sample-based approximation to the Delaunay Triangulation has reliable empirical performance.

2.2.3 Edge Weight Construction

Given the constructed edges and knots, we assign each edge a weight that represents the similarity between the pair of knots. In this work, we propose some novel density-aided quantities as the edge weights. Since the description of the similarity measures is more involved, we defer the detailed discussion of the similarity measures to Section 2.3. It is worth noting here that the similarity measures proposed in this work are estimated based on surrogates of the underlying density function (hence density-aided) and the estimation procedure has minimal dependence on the ambient dimension. Therefore, the estimations of the newly proposed similarity measures are reliable even under high-dimensional settings.

2.2.4 Knots Segmentation

Given the weighted skeleton graph, the next step is to partition the knots into the desired number of final clusters, and we apply hierarchical clustering with the inverses of the similarity measures as the distance. The choice of linkage criterion for hierarchical clustering may depend on the underlying geometric structure of the data. We analyze several linkage criteria under various simulation scenarios in Appendix E. Generally, single linkage gives reliable clustering results when the components are well-separated, but average linkage works better when there are overlapping clusters of approximately spherical shapes. Therefore, in practice, such a choice of linkage should be made based on some exploratory understanding of the data structure, and experimenting with different linkage methods is computationally tractable as only the knots need to be segmented.

The number of final clusters S is an essential parameter for the hierarchical clustering procedure but can be unknown. The dendrograms given by hierarchical clustering can be a helpful tool in this situation, displaying the clustering structure at different resolutions. Consequently, analysts can experiment with different numbers of final clusters and choose a cut that preserves the meaningful structures based on the dendrograms, which takes little extra computation. However, it is worth pointing out that with the presence of noisy data points, the final number S being larger than the true number of meaningful components may be needed to achieve better clustering results (see Appendix E).

Remark 2. Although the dendrogram for knots given by our method is not exactly the cluster tree, the pruning graph cluster tree procedure proposed in Nugent and Stuetzle (2010) with

excess mass can be applied to help decide the final segmentation. Peterson et al. (2018) also presented similar ideas choosing the final number of clusters by looking at the lifetime of the clusters in the dendrogram. Additionally, the traditional "elbow" methods can be used to determine the number of clusters. An inferential choice can also be made using the gap statistics (Tibshirani et al., 2001).

2.2.5 Assignment of Labels

In the previous step, we created S groups of knots and each group has a cluster label. To pass the cluster membership to each observation, we assign a hard clustering label to each observation according to which group its nearest knot belongs. For instance, if an observation X_i is closest to knot c_j and c_j belongs to cluster ℓ, we assign cluster membership label ℓ to observation X_i.

Remark 3. There are other methods in clustering literature for assigning labels of observations based on identified structures. Azzalini and Torelli (2007) and Scrucca (2016) assign unlabelled data based on density ratios. DBSCAN and HDBSCAN (Campello et al., 2015; Ester et al., 1996) assign labels (and identify noisy points) based on k-nearest-neighbor considerations. One may use these alternatives to assign the cluster label to each observation.

2.3 Density-Based Edge Weights Construction

To incorporate the information of density into clustering, we calculate the edge weights based on the underlying density function. However, the conventional notion of PDF is not

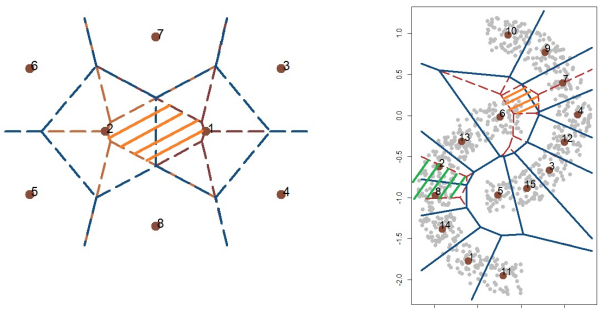


Figure 2.4: **Left:** Orange shaded area illustrates the 2-NN region of knots $1, 2$. **Right:** Shaded areas illustrate the 2-NN region of knots $6, 7$ and knots $2, 8$.

feasible in multivariate or even high-dimensional data due to the curse of dimensionality. To resolve this issue, we introduce three density-related quantities that are estimable even when the dimension is high.

2.3.1 Voronoi Density

The *Voronoi density (VD)* measures the similarity between a pair of knots (c_j, c_ℓ) based on the number of observations whose 2-nearest knots are c_j and c_ℓ. We start with defining the Voronoi density based on the underlying probability measure and then introduce its sample analog. Given a metric d on $\mathbb{R}^d$, the 2-Nearest-Neighbor (2-NN) region of a pair of knots (c_j, c_ℓ) is defined as

$$A_{j\ell} = \{x \in \mathcal{X} : d(x, c_i) > \max\{d(x, c_j), d(x, c_\ell)\}, \forall i \neq j, \ell\}. \tag{2.3.1}$$

In this work, we take $d(.,.)$ to be the usual Euclidean distance and use $||.||$ to denote the Euclidean norm. An example 2-NN region of a pair of knots is illustrated in Figure 2.4.

Following the idea of density-based clustering, two knots c_j, c_ℓ belong to the same clusters if they are in a connected high-density region, and we would expect the 2-NN region of c_j, c_ℓ

to have a high probability measure. Hence, the probability $\mathbb{P}(A_{j\ell}) = P(X_1 \in A_{j\ell})$ can measure the association between c_j and c_ℓ (see illustration in Figure 3.2 right). Based on this insight, the Voronoi density measures the edge weight of (c_j, c_ℓ) with

$$S_{j\ell}^{VD} = \frac{\mathbb{P}(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{2.3.2}$$

Namely, we divide the probability of in-between region by the mutual Euclidean distance. The division of the distance adjusts for the fact that 2-NN regions have different sizes and provides more weights to edges between knots close in distance. However, such division makes the Voronoi density be in the unit of $1/\|c_j - c_\ell\|$ and hence can be scale-dependent.

In practice, we estimate $S_{j\ell}^{VD}$ by a sample average. Specifically, the numerator $\mathbb{P}(A_{j\ell})$ is estimated by $\hat{P}_n(A_{j\ell}) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A_{j\ell})$ and the final estimator for the VD is

$$\hat{S}_{j\ell}^{VD} = \frac{\hat{P}_n(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{2.3.3}$$

Note that here we are assuming that $c_1, \cdots, c_k$ as given beforehand. In the sample version, we replace them with the sample analog $\hat{c}_1, \cdots, \hat{c}_k$ and replace the region $A_{j\ell}$ by $\hat{A}_{j\ell}$.

The Voronoi density can be computed in a fast way. The numerator, which only depends on 2-nearest-neighbors calculation, can be computed efficiently by the k-d tree algorithm (Bentley, 1975). For high-dimensional space, space partitioning search approaches like the k-d tree can be inefficient but a direct linear search still gives a short run-time (Weber et al., 1998), and with a large number of observations approximate nearest neighbor algorithms can be incorporated. The denominator requires distance calculation and can be burdensome in high-dimensional settings, but note that we only need to calculate the distance for edges present in $\hat{DT}(\mathcal{C})$, which is far less than $k(k-1)/2$, where k is the number of knots. Hence,

the calculation of VD can be carried out in a fast way even for high-dimensional data with a large sample size.

2.3.2 Face Density

Here we present another density-based quantity to measure the similarity between two knots. Since the Voronoi cell of a knot describes the associated region, a natural way to measure the similarity between two knots is to investigate the shared boundary of the corresponding Voronoi cells. If two knots are highly similar, we would expect the boundary to lie in a high-density region and to be surrounded by many observations. Based on this idea, we define the *Face Density (FD)* as the integrated PDF over the "face" (boundary) region. Note that, although the density is involved in FD, by integrating over the face region the problem reduces to a 1-dimensional density estimation task regardless of the dimension of the ambient space. Formally, let the face region between two knots c_j, c_ℓ be $F_{j\ell} = \mathbb{C}_j \cap \mathbb{C}_\ell$. At the population level, the FD is defined as

$$S_{j\ell}^{FD} = \int_{F_{j\ell}} p(x)\mu_{d-1}(dx) = \int_{F_{j\ell}} d\mathbb{P}(x), \tag{2.3.4}$$

where $\mu_m(dx)$ denotes the m-dimensional volume measure.

To estimate the FD, we utilize the idea of kernel smoothing in combination with data projection. By the construction of the Voronoi diagram, the boundary of two Voronoi cells is orthogonal to the line passing through the two corresponding knots (called the 'central line') and intersects the central line at the middle point regardless of the dimension of the data (see Figure 2.3 for reference). Therefore, we estimate the FD by first projecting the observations

onto the central line and then using the 1-dimensional kernel density estimator(KDE) to evaluate the density at the midpoint. Specifically, fix two knots c_j, c_ℓ, let $\mathbb{C}_j, \mathbb{C}_\ell$ be the corresponding Voronoi cells, and denote $\Pi_{j\ell}(x)$ as the projection of $x \in \mathcal{X}$ onto the central line passing through c_j and c_ℓ, we define the estimator $\hat{S}_{j\ell}^{FD}$ to be

$$\hat{S}_{j\ell}^{FD} = \frac{1}{nh} \sum_{X_i \in \mathbb{C}_j \cup \mathbb{C}_\ell} K\left(\frac{\Pi_{j\ell}(X_i) - (c_\ell + c_j)/2}{h}\right) \tag{2.3.5}$$

where K is a smooth, symmetric kernel function (e.g. Gaussian kernel) and $h > 0$ is the bandwidth that controls the amount of smoothing. It is noteworthy that, while conventional kernel smoothing suffers from the curse of dimensionality (Chen et al., 2017; Chacón et al., 2011; Wasserman, 2006a), the kernel estimator in equation (2.3.5) bypasses it.

2.3.3 Tube Density

While FD is conceptually appealing, the characterization of the face between two Voronoi cells could be challenging since the shapes of the boundaries can be irregular. Here we propose a measure similar to the Face density measure but has a predefined regular shape. For a point x, we define the *Disk Area* centered at x with radius R and normal direction ν (see Figure 2.5 for an illustration) as

$$\mathsf{Disk}(x, R, \nu) = \{y : ||x - y|| \leq R, (x - y)^T \nu = 0\} \tag{2.3.6}$$

To measure the similarity between knots c_j and c_ℓ, we examine the integrated density within the disk areas along the central line. In more detail, the central line can be expressed as $\{c_j + t(c_\ell - c_j) : t \in [0, 1]\}$, and any point on the central line can be written as $c_j + t(c_\ell - c_j)$ for some t. For a point $c_j + t(c_\ell - c_j)$, we define the integrated density in the disk region

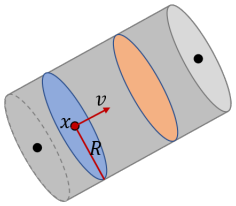


Figure 2.5: The disk area centered at x with a radius R and a direction ν.

(called *Disk Density*) as

$$\mathsf{pDisk}_{j\ell,R}(t) = \mathbb{P}\left(\mathsf{Disk}(c_j + t(c_\ell - c_j), R, c_\ell - c_j)\right) = \int_{\mathsf{Disk}(c_j+t(c_\ell-c_j),R,c_\ell-c_j)} p(x)dx. \quad (2.3.7)$$

The *Tube Density (TD)* measures the similarity between c_j and c_ℓ as the minimal disk density along the central line, i.e.,

$$S_{j\ell}^{TD} = \inf_{t\in[0,1]} \mathsf{pDisk}_{j\ell,R}(t) \quad (2.3.8)$$

In other words, with given c_j, c_ℓ, we survey all Disk Density along the central line and retrieve the infimum as the similarity measure between two knots.

In this work, we set R based on the root mean squared distances within each Voronoi cell. Specifically, for knot c_j and the corresponding Voronoi cell $\mathbb{C}_j$, we calculate

$$R_j = \sqrt{\frac{1}{|\mathbb{C}_j| - 1} \sum_{X_\ell \in \mathbb{C}_j} \|X_\ell - c_j\|^2} \quad (2.3.9)$$

where $|\mathbb{C}_j|$ denotes the size of set $\mathbb{C}_j$. With the uniform radius paradigm where the radius is the same for all pairs of knots, we set $R = \frac{1}{k}\sum_{j=1}^{k} R_j$. Our empirical studies show that this rule leads to good clustering performances and theoretical analysis also shows that this reference rule for R leads to the consistency of the sample analog of the TD.

Note that the radius may also be chosen adaptively for each pair: we set the disk radius at c_j to be R_j for all knots and set the disk radius along the edge to be the linear interpolation of the radii at the two connected knots. The comparison between the uniform and adaptive

R is presented in Appendix F, and similar clustering performance is observed for the two approaches. Hence we use uniform R by default for simplicity.

Similar to the FD, we estimate the TD by a projected KDE. Let $\Pi_{j\ell}(x)$ be the projection of a point x on the line through c_j, c_ℓ. We first estimate the pDisk via

$$\widehat{\mathsf{pDisk}}_{j\ell,R}(t) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - c_j - t(c_\ell - c_j)}{h}\right) I(||X_i - \Pi_{j\ell}(X_i)|| \leq R)$$

and then estimate the TD as

$$\hat{S}_{j\ell}^{TD} = \inf_{t\in[0,1]} \widehat{\mathsf{pDisk}}_{j\ell,R}(t). \tag{2.3.10}$$

where the infimum is approximated by grid search.

Remark 4. The estimations of the FD and the TD involve the use of the projected kernel density estimation, and we discuss the choices of the kernel and the bandwidth selections for kernel density estimations in Appendix F. By default, we use the Gaussian kernel with the normal scale bandwidth selector (NS) (Chacón et al., 2011) for the best empirical results.

2.4 Asymptotic Theory of Edge Weight Estimation

In this section, we focus on the theoretical properties of the similarity measures to theoretically explain the effectiveness of the newly proposed density-aided similarity measures. We assume the set of knots $\mathcal{C} = \{c_1, \ldots, c_k\}$ is given and non-random to simplify the analysis because (1) it is hard to quantify k-means uncertainty, and (2) with large k, it is extremely likely for k-means to stuck within the local minimum. Note that this implies the corresponding Voronoi cells $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_k\}$ and the 2-NN regions $\{A_{j\ell}\}_{j,\ell=1,\ldots,k,j\neq\ell}$

(Equation 2.3.1) of all pairs of knots are fixed as well. We allow $k = k_n$ to grow with respect to the sample size n. Theoretical results for Voronoi density are described in this section and theoretical properties for the Face density and Tube density are deferred to Appendix B and C respectively. In summary, the consistency of FD and TD are obtained based on the analysis of KDE with additional geometric considerations, resulting in rates similar to that of the 1-dimensional KDE under some regularity conditions. All proofs are included in Appendix D.

2.4.1 Voronoi Density Consistency

We start with the convergence rate of the VD and consider the following condition:

(B1) There exists a constant c_0 such that the minimal knot size $\min_{(j,\ell)\in E} \mathbb{P}(A_{j\ell}) \geq \frac{c_0}{k}$ and $\min_{(j,\ell)\in E} \|c_j - c_\ell\| \geq \frac{c_0}{k^{1/d}}$.

where $(j, \ell) \in E$ means that there is an edge between knots c_j, c_ℓ in the Delaunay Triangulation. Condition (B1) is a condition requiring that no Voronoi cell $A_{j\ell}$ has a particularly small size and all edges have sufficient length. This condition is mild because when the dimension of data d is fixed, the total number of edges in the Delaunay triangulation of k points scale at rate $O(k)$. Because the volume shrinks at rate $O(k^{-1})$, the distance is expected to shrink at rate $O(k^{-1/d})$.

Theorem 1 (Voronoi Density Convergence). Assume (B1). Then for any pair $j \neq \ell$ that shares an edge, the similarity measure based on the Voronoi density satisfies

$$\left| \frac{\hat{S}^{VD}_{j\ell}}{S^{VD}_{j\ell}} - 1 \right| = O_p\left(\sqrt{\frac{k}{n}}\right), \tag{2.4.1}$$

$$\max_{j,\ell} \left| \frac{\hat{S}_{j\ell}^{VD}}{S_{j\ell}^{VD}} - 1 \right| = O_p \left(\sqrt{\frac{k}{n}} \log k \right), \tag{2.4.2}$$

when $n \to \infty, k \to \infty, \frac{n}{k} \to \infty$.

Theorem 1 provides the convergence rates of the sample-based Voronoi density to the population version of Voronoi density. This result is reasonable because when the knots $\mathcal{C}$ are given, the randomness in the sample-based Voronoi density is just the empirical proportion in each cell, so it is a square-root-rate estimator based on the effective local sample size n/k. Consequentially, Theorem 1 suggests that estimating the Voronoi density is easy in multivariate cases when the knots are given–there is no dependency with respect to the ambient dimension. The extra $\log k$ factor in the uniform bound (Equation 2.4.2) comes from the Gaussian concentration bounds.

2.4.2 Performance Guarantee for Voronoi Density

We provide below a performance guarantee in terms of the adjusted Rand Index (Rand, 1971; Hubert and Arabie, 1985) for skeleton clustering with Voronoi density edge similarity. To simplify the problem, we define the true clusters as the connected components of the skeleton graph with edges having true Voronoi density similarities $S_{j\ell}^{VD}$ over a known threshold $\tau > 0$. We show below that cutting the skeleton graph based on estimated edge similarities at the same threshold τ recovers the true clustering with a high probability. Since the knots are fixed, the clustering error comes from partitioning knots into the wrong groups, so we will focus on the adjusted Rand Index of clustering the knots. Let the true partition of the knots be $\mathcal{L}^* = \{\mathcal{L}_\ell^*\}_{\ell=1,\ldots,L}$, where $\mathcal{L}_\ell^*$ contains all the knot indices belonging to the partition

ℓ. Let the partition based on estimated edge similarities be $\hat{\mathcal{L}}$. We assume that

(P1) The true partition $\mathcal{L}^*$ under the threshold τ remains the same when the thresholding level is within $(\tau(1-\varepsilon), \tau(1+\varepsilon))$ for some $\varepsilon > 0$.

This is a mild assumption because when we vary the threshold level τ, only a finite number of values will create a change in the partition. So (P1) holds under almost all values of τ except for a set of Lebesgue measure 0. Let $ARI(\mathcal{L}^*, \hat{\mathcal{L}})$ denote the adjusted Rand Index of the estimated partition.

Theorem 2 (Adjusted Rand Index Guarantee). Assume (B1) and (P1) and let $p_{min} = \min_{j,\ell} \mathbb{P}(A_{j\ell})$, then

$$\mathbb{P}\left\{ ARI(\mathcal{L}^*, \hat{\mathcal{L}}) < 1 \right\} \leq k(k-1) \exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1-p_{min}) + \frac{1}{3}\varepsilon} \right) \tag{2.4.3}$$

Theorem 2 shows that we have a good chance of recovering the "true" clusters defined by the actual Voronoi density. The above bound is derived from the uniform concentration bound of the Voronoi density.

2.5 Simulations

To study the effectiveness of skeleton clustering as a clustering method, we conduct several Monte Carlo experiments. In this section, we present some empirical results to illustrate the performance of skeleton clustering in multivariate and high-dimensional settings (with additional data examples in Appendix G). Generally, our framework with the Voronoi density similarity measure is superior among all the compared clustering methods. In Appendix

E, we use a systematic set of simulation studies to discuss the choice of linkage criteria within our clustering framework when dealing with different datasets and at the same time to demonstrate the robustness of the proposed framework to noisy data points and overlapping clusters. We include some additional simulations to support some choices within our framework in Appendix F.

2.5.1 High-dimensional Setting

In this section, we demonstrate the performance of skeleton clustering on simulated datasets: the Yinyang data and the Mickey data. We also include a simulated dataset consisting of manifold structures of different dimensions, called the Manifold Mixture data, in Appendix G and an additional simulation called the Ring data in Appendix G. For the simulations within Section 2.5.1 and Appendix G, when using the skeleton clustering methods, the number of knots is set to be $k = [\sqrt{n}]$ and the knots are chosen by k-means with 1000 random initialization. We select smoothing bandwidth by the normal scale bandwidth selector for the FD and TD, and the radius of TD is set to be the same for all edges with the value chosen as described in Section 2.3.3. We use single linkage hierarchical clustering when merging knots into final clusters with the true number of final clusters S being provided.

To highlight the importance of density-aided similarity measures, we include a similarity measure called the average distance (AD) for comparison. AD measures the similarity between c_j and c_ℓ using the inverse of the average Euclidean distances between all pairs of observations in the two corresponding Voronoi cells. All simulations are repeated 100

times to obtain the distribution of the empirical performances.

Yinyang Data

The Yinyang dataset is an intrinsically 2-dimensional data containing 5 components: a big outer circle with 2000 uniformly distributed data points, two inner semi-circles each with 200 data points generated as 2D Gaussian with standard deviation 0.1, and two clumps each with 200 data points (generated with the `shapes.two.moon` function with default parameters in the `clusterSim` library in R (Walesiak and Dudek, 2020)). The total sample size is $n = 3200$ and according to our reference rule, we choose $k = [\sqrt{3200}] = 57$ knots for the skeleton clustering procedure. To make the data high-dimensional, we include additional variables from a Gaussian distribution with mean 0 and standard deviation 0.1, and we increase the dimension of noise variables so that the total dimensions are $d = 10, 100, 500, 1000$. We present results with larger standard deviations for the noisy variable in Appendix F. We empirically compare the following clustering approaches: direct single-linkage hierarchical clustering (SL), direct k-means clustering (KM), spectral clustering (SC), skeleton clustering with average distance density (AD), skeleton clustering with Voronoi density (Voron), skeleton clustering with Face density (Face), and skeleton clustering with Tube density (Tube). Since this is simulated data, we know that there are exactly 5 clusters and we know which cluster an observation belongs to. The true number of clusters is provided to all the clustering algorithms. We use the adjusted Rand Index to measure the performance of each clustering method.

The results are given in Figure 2.7. We observe that when dimension increases, traditional

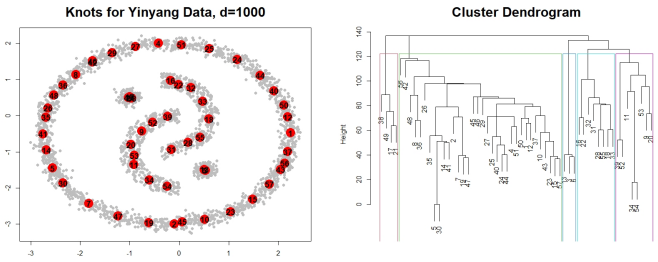


Figure 2.6: Knots chosen by k-means on Yinyang data and the Dendrogram for single linkage hierarchical clustering with similarity measured by Voronoi density.

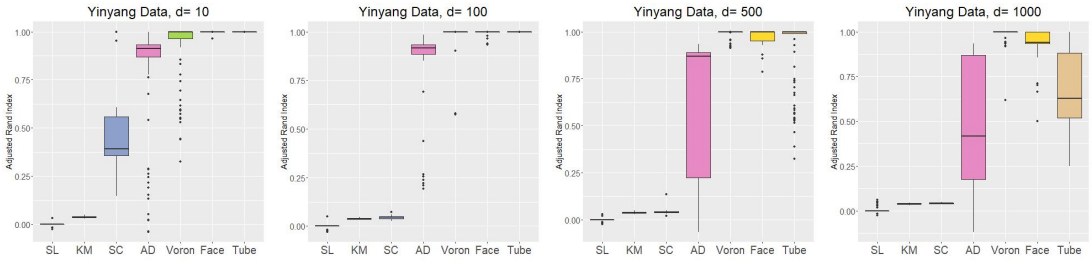


Figure 2.7: Comparison of the final clustering performance in terms of adjusted Rand Index with different clustering methods on Yinyang Data with dimensions 10, 100, 500, and 1000.

methods (SL, KM, SC) fail to give good clustering results while skeleton clustering can generate nearly perfect clustering. Across all the data dimensions, the Voronoi density, the simplest measure among the three proposed similarity measures, gives the best performance in the skeleton clustering framework. Average distance density becomes problematic in high-dimensional settings but still gives better performance compared to the classical methods. The fact that all skeleton clustering methods perform better than the traditional methods highlights the effectiveness of using the skeleton clustering framework. Moreover, all three density-aided similarity measures outperform the average distance, which illustrates the power of using density-aided weights in clustering.

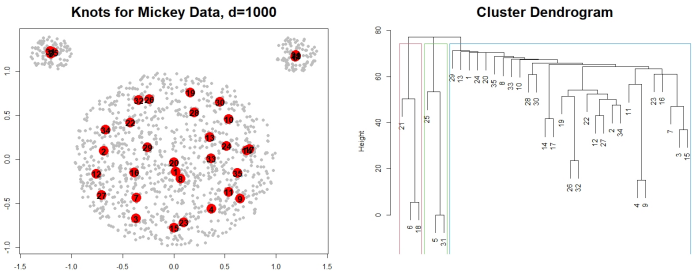


Figure 2.8: An illustration of the analysis of the Mickey data with dimension 100.

Mickey Data

The simulated Mickey data is an intrinsically 2-dimensional data consisting of one large circular region with 1000 data points and two small circular regions each with 100 data points. As a result, the structures have unbalanced sizes. The total sample size is $n = 1200$ and we choose the number of knots to be $k = [\sqrt{1200}] = 35$. We include additional variables with random Gaussian noises to make it a high dimensional data ($d = 10, 100, 500, 1000$) the same way as in Section 2.5.1. The left panel of Figure 2.8 shows the scatter plot of the first two dimensions.

We perform the same comparisons as done on the Yinyang data with the true number of components $S = 3$ provided to all the clustering algorithms, and the results are displayed in Figure 2.9. All methods perform well when d is small but starting at $d = 100$, traditional methods fail to recover the underlying clusters. On the other hand, all methods in the skeleton clustering framework work well even when $d = 1000$.

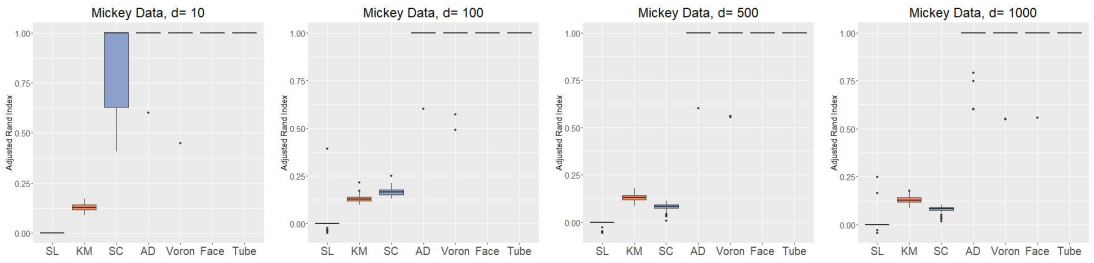


Figure 2.9: Comparison of adjusted Rand index using different similarity measures on Mickey data with dimensions 10, 100, 500, 1000.

2.6 Real Data

In this section, we apply skeleton clustering to one real data example: the graft-versus-host disease (GvHD) data (Brinkman et al., 2007). Additionally, we analyze the Zipcode data (Stuetzle and Nugent, 2010) in Appendix H and the Olive Oil data (Tsimidou et al., 1987) in Appendix H.

GvHD is a significant problem in the field of allogeneic blood and marrow transplantation which occurs when allogeneic hematopoietic stem cell transplant recipients when donor-immune cells in the graft attack the tissues of the recipient. The data include samples from a patient with GvHD containing $n_1 = 9083$ observations and samples from a control patient with $n_2 = 6809$ observations. Both samples include four biomarker variables, CD4, CD8β, CD3, and CD8. Previous studies (Lo et al., 2008; Baudry et al., 2010) have identified the presence of high values in CD3, CD4, CD8β cell sub-populations as a significant characteristic in the GvHD positive sample and a major objective of our analysis is to rediscovery this region with the proposed skeleton clustering methods. In addition, our skeleton clustering procedure shows more information and leads to a novel two-sample test.

The two samples are plotted in the left panel of Figure 2.10 focusing on the three key

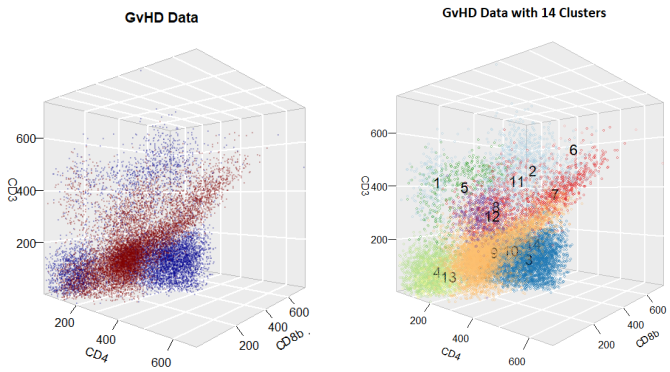


Figure 2.10: **Left:** 3D scatterplot of the positive sample (red) and the control sample (blue). **Right:** Final clustering result of combined GvHD data.

variables (CD3, CD4, CD8β) with blue points from the control sample and the red points from the GvHD positive sample. We observe that, in addition to the high CD3, CD4, CD8β region, the distribution of the positive sample is different from the control sample also in some regions with medium to low CD3, CD4, and CD8β. Later we will demonstrate that our clustering framework can identify all such differences in distributions.

To apply the skeleton clustering for a fair comparison of the two samples, we first construct knots from each sample separately. Specifically, we apply the k-means method to find $k_1 = [\sqrt{n_1}]$ knots for the positive sample and find $k_2 = [\sqrt{n_2}]$ knots for the control sample. This ensures that both samples are well-represented by knots. We then combine the two samples into one dataset and combine the two sets of knots into one set with $k_1 + k_2$ knots. We create edges among the combined knots and apply the Voronoi density (VD) to measure the edge weights. To segment the knots, we use the average linkage criterion because the clusters can be overlapping and the analysis in Appendix E suggests average linkage for this scenario. The skeleton clustering result is displayed in the right panel of Figure 2.10 with the number of final clusters chosen to be $S = 14$ (Baudry et al., 2010).

Cluster	1	2	3	4	5	6	7
Size	202	948	3881	1859	338	17	812
Prop	.458	.343	.008	.296	.341	.000	.934
p-value	.30	7×10^{-20}	0	3×10^{-63}	4×10^{-8}	1×10^{-4}	6×10^{-103}
Cluster	8	9	10	11	12	13	14
Size	468	6191	251	37	478	402	8
Prop	.690	.888	.673	.669	.794	.841	.310
p-value	2×10^{-13}	0	1×10^{-6}	.09	6×10^{-30}	3×10^{-33}	.52

Table 2.1: Table of the sizes of the clusters and the weighted proportion of positive observations within each cluster. A proportion of 0.5 indicates that the two samples have equal proportions in the region. The p-value is the simple proportional test to examine if the two samples have equal proportions in that cluster.

For further insights, we examined the weighted proportion of positive observations in each cluster. A proportionally smaller weight is assigned to each positive observation to accommodate the fact that there are more positive observations ($n_1 = 9083 > n_2 = 6809$). After such normalization, a weighted proportion of 0.5 means that the positive and control observations are balanced in one region. A summary of the weighted proportion of clusters is presented in Table 2.1. We note that clusters 7,9,12, and 13 are majorly composed of positive observations (proportion > 0.75), and clusters 3 and 6 are majorly composed of observations from the control sample (proportion < 0.25). We also include the p-value for testing if the proportions equal 0.5. Admittedly, because we use the data to find clusters and use the same data to do the test, the p-values in Table 2.1 may tend to be small.

Clusters with majorly positive observations and clusters with majorly control observations are depicted in the two panels in Figure 2.11. Cluster 7 corresponds to the high CD3, CD4, and CD8β region identified by previous works with nearly all data points belonging to the positive patient. Cluster 6 is also scattered in the high CD3, CD4, and CD8β regions but has all the observations coming from the control sample. However, the small size (only 17

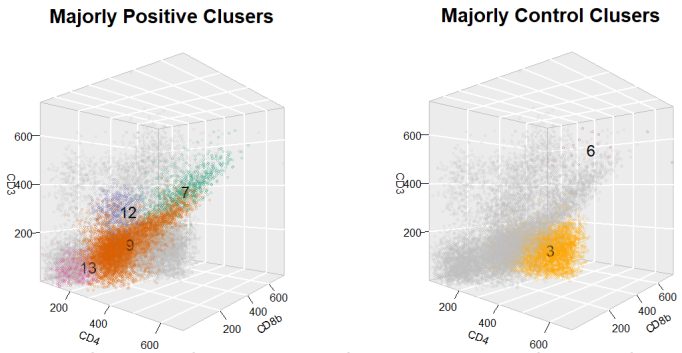


Figure 2.11: Clusters with majorly positive observations and majorly control observations

data points) of Cluster 6 makes it unclear if it is a real structure or due to pure randomness. Overall our method succeeds in identifying the CD3+ CD4+ CD8β+ area for the GvHD-positive patient like the previous model-based clustering approaches. Note that the data we are using are two individuals from the original 31 individuals in the GvHD study, which does not account for the inter-individual variability.

Our clustering approach has some additional findings. Cluster 9, 12, and 13 also have a high proportion of positive samples. These clusters are in the mid to low CD3, CD4, CD8β region. For the control case, in addition to the small Cluster 6, Cluster 3 is a large cluster with nearly all the observations from the control sample. It is located in the high CD8β but low CD3 and CD4 region.

Model-based clustering approaches Lo et al. (2008); Baudry et al. (2010) have an advantage for managing this cytometry data as they can parametrically describe the behaviors of data samples in different regions. The overlapping between different structures and the overall 4-dimensional feature space are also applicable with model-based clustering methods. However, the proposed skeleton clustering approach can result in a graphical representation of each

cluster that can be visualized for intuitive understanding. We include the skeleton graphs of the GvHD data clusters from the proposed clustering approach in Appendix F. Moreover, model-based approaches can still be limited to some regular shapes of the clusters in the ambient space, while applying the proposed clustering method helps identify clusters with complex structures. Cluster 9, for instance, shows a hammer-like structure based on the skeleton representation (see Figure A.27).

Our results suggest a potential procedure for diagnosing GvHD. Biomarkers from a new patient can be divided into clusters with respect to the learned segmentation, and doctors can mainly focus on the sample points that fall into regions 3, 7, 9, 12, and 13. If the patient has many points in Clusters 7, 9, 12, and 13, the patient likely has GvHD. Note that our current result is only based on two individuals and, with a descriptive purpose, is not accounting for the variability between different individuals and different cases. To use it for practical diagnosis, a more comprehensive analysis based on a larger and more representative sample is required.

2.7 Conclusion

In this work, we introduce the skeleton clustering framework that can handle multivariate and even high-dimensional clustering problems with complex, manifold-based cluster shapes. Our method adopts the density-based clustering idea to the high dimensional regime. The key to bypassing the curse of dimensionality is the use of density surrogates such as Voronoi density, Face density, and Tube density that are less sensitive to the dimension. We use both

theoretical and empirical analysis to illustrate the effectiveness of the skeleton clustering procedure. In what follows, we discuss some possible future directions:

- **Accounting for the randomness of knots.** For our current theoretical analysis, we assume that the knots are given and non-random to simplify the problem. But in practice, knots are computed from the sample data with inherent uncertainty. The randomness of knots can affect the clustering performance because the location of knots directly impacts the Voronoi cells, which changes the value of the similarity measures and consequently the cluster label assignments. In particular, observations on the boundary of clusters will be more sensitive to any perturbations in the location of knots. Currently, there are two technical challenges when dealing with random knots. First, the randomness of knots may be correlated with the randomness of estimated edge weight, so the calculation of rates is much more complicated. Second, while there are established theories for k-means algorithm (Graf and Luschgy, 2000, 2002; Hartigan and Wong, 1979), these results only apply to the global minimum of the objective function. In reality, we are unlikely to obtain the global minimum, but instead, our inference is based on a local minimum. It is unclear how to properly derive a theoretical statement based on local minima, so we leave this as future work.
- **Skeleton clustering with similarity matrix.** The idea of skeleton clustering may be generalized to data where we only observe the similarity/distance matrices such as network data. Knots can be restricted to indices in the data and we choose them by minimizing some network-based or diffusion-related criteria. While Face and Tube

density can be difficult to adopt, the Voronoi density is still applicable since we only need the information about pairs of observations. This might provide a new approach for community detection in network data (Zhao, 2017; Abbe, 2017).

- **Detecting boundary points between clusters.** Our skeleton clustering method can be applied to detect points on the boundary between two clusters. The idea is simple: in the final cluster assignment, instead of assigning only one label to an observation, we assign h labels to an observation based on the cluster labels of h-nearest knots. The homogeneity of the label assignments can be used as a quantity to detect if a point is on the boundary or in the interior of a cluster and may serve as an uncertainty quantification of clustering. We will pursue this in the future.
- **Anomaly and noise detection.** As illustrated in Appendix E, E, and E, the single linkage criterion in our Skeleton clustering framework may detect noisy observations in the data. This suggests the possibility of using our approach for noises or anomalies similar to the DBSCAN (Campello et al., 2015; Ester et al., 1996). We will explore this direction in the future.

Chapter 3

Skeleton Regression: A Graph-Based Approach to Estimation on Manifold

3.1 Introduction

Many data nowadays are geometrically structured that the covariates lie around a low-dimensional manifold embedded inside a large-dimensional vector space. Among many geometric data analysis tasks, the estimation of functions defined on manifolds has been extensively studied in the statistical literature. A classical approach to explicitly account for geometric structure takes two steps: map the data to the tangent plane or some embedding space and then run regression methods with the transformed data. This approach is pioneered by the Principle Component Regression (PCR) Massy (1965) and the Partial Least Squares (PLS) Wold (1975). Aswani et al. (2011) innovatively relates the regression coefficients to exterior derivatives. They propose to learn the manifold structure through local principal

components and then constrain the regression to lie close to the manifold by solving a weighted least-squares problem with Ridge regularization. Cheng and Wu (2013) present the Manifold Adaptive Local Linear Estimator for the Regression (MALLER) that performs the local linear regression (LLR) on a tangent plane estimate. However, because those methods directly exploit the local manifold structures in an exact sense, they are not robust to variations in the covariates that perturb them away from the true manifold structure.

Many other manifold estimation approaches exist in the statistical literature. Guhaniyogi and Dunson (2016) utilize random compression of the feature vector in combination with Gaussian process regression. Zhang et al. (2013) follows a divide-and-conquer approach that computes an independent kernel Ridge regression estimator for each randomly partitioned subset and then aggregates. Other nonparametric regression approaches such as kernel machine learning (Schölkopf and Smola, 2002), manifold regularization (Belkin et al., 2006b), and the spectral series approach (Lee and Izbicki, 2016) also account for the manifold structure of the data. More recently, Green et al. (2021) proposes the Principal Components Regression with Laplacian-Eigenmaps (PCR-LE) that projects data onto the eigenvectors output by Laplacian Eigenmaps and provides the rates of convergence of such nonparametric regression method over Sobolev spaces. However, those methods still suffer from the curse of dimensionality with large-dimensional covariates.

In addition to data with manifold-based covariates, manifold learning has been applied to other types of manifold-related data. Marzio et al. (2014) develop nonparametric smoothing for regression when both the predictor and the response variables are defined on a sphere.

Zhang et al. (2019) deal with the presence of grossly corrupted manifold-valued responses. Lin and Yao (2020) address data with functional predictors that reside on a finite-dimensional manifold with contamination. In this work, we focus on manifold-based covariates and may incorporate other types of manifold-related data in the future.

The main goal of this work is to estimate a scalar response with covariates lying around some manifold structures in a way that utilizes the geometric structure and bypasses the curse of dimensionality. This is achieved by proposing a new framework that combines graphs and nonparametric regression techniques. Our framework follows the two-step idea: first, we learn a graph representation, which we call the *skeleton*, of the manifold structure based on the methods from Wei and Chen (2023) and project the covariates onto the skeleton. Then we apply different nonparametric regression methods with the skeleton-projected covariates. We give brief descriptions of the relevant nonparametric regression methods below.

Kernel smoothing is a widely used technique that estimates the regression function as locally weighted averages with the kernel as the weighting function. Pioneered by the famous Nadaraya–Watson estimator from Nadaraya (1964) and Watson (1964), this technique has been widely used and extended by recent works (Fan and Fan, 1992; Hastie and Loader, 1993; Fan et al., 1996; Kpotufe and Verma, 2017). Splines (Hastie et al., 2009; Friedman, 1991) are popular nonparametric regression constructs that take the derivative-based measure of smoothness into account when fitting a regression function. Moreover, k-Nearest-Neighbors (kNN) regression (Altman, 1992; Hastie et al., 2009) has a simple form based on a distance metric but is powerful and widely used in many applications. These techniques are incorporated

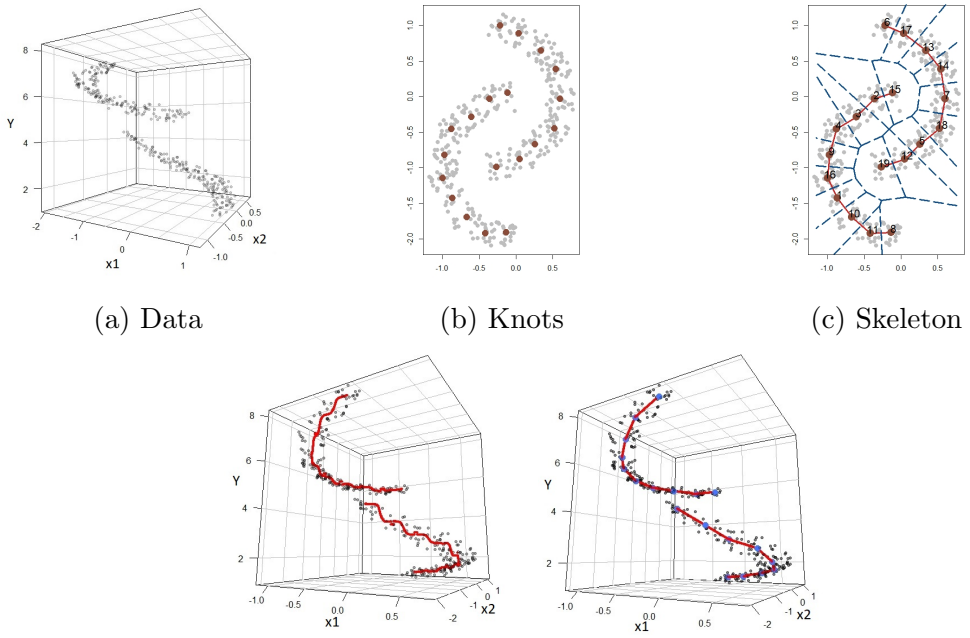
into our proposed regression framework.

In recent years, many nonparametric regression techniques have been shown to adapt to the manifold structure of the data, with convergence rates that depend only on the intrinsic dimension of the data space. Specifically, the classical kNN and kernel regressor have been shown to be manifold-adaptive with proper parameter tuning procedures (Kpotufe, 2009, 2011; Kpotufe and Garg, 2013; Kpotufe and Verma, 2017), while recent methods like the Spectral Series regression and PCR-LE also enjoy this property (Green et al., 2021). The proposed regression framework in this work also adapts to the manifold, as the nonparametric regression models fitted on a graph are dimension-independent. This framework has several additional advantages such as the ability to account for predictors from distinct manifolds and being robust to additive noise and noisy observations.

Outline. We start by presenting the procedures of the skeleton regression framework in section 3.2. In section 3.3, we apply nonparametric regression techniques to the constructed skeleton graph along with theoretical justifications. In section 3.4, we present some simulation results for skeleton regression and demonstrate the effectiveness of our method on real datasets in Section 3.5. In section 3.6, we conclude the paper and point out some directions for future research.

3.2 Skeleton Regression Framework

In this section, we introduce the skeleton regression framework. Given design vectors $\{\boldsymbol{x}_i\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ for each i and the corresponding responses $\{Y_i\}_{i=1}^n$ in $\mathbb{R}$, a



(a) Data (b) Knots (c) Skeleton

(d) S-Kernel Regression (e) Linear Spline

Figure 3.1: Skeleton Regression illustrated by data with covariates having the shape of two moons in a 2D space.

traditional regression approach is to estimate the regression function $m(\boldsymbol{x}) = \mathbb{E}(Y|X = \boldsymbol{x})$. However, the ambient dimension d can be large while the covariates are distributed around some low-dimensional manifold structures. In this case, $\mathcal{X}$ can be the union of several disjoint components with different manifold structures, and the regression function can have discontinuous changes from one component to another. To handle such geometrically structured data, we approach the regression task by first representing the sample covariate space with a graph, which we call the *skeleton*, to summarize the manifold structures. We then focus on the regression function over the skeleton graph, which incorporates the covariate geometry in a dimension-independent way.

We illustrate our regression framework on the simulated Two Moon data in Figure 3.1.

The covariates of the Two Moon data consist of two 2-dimensional clumps with intrinsically 1-dimensional curve structure, and the regression response increases polynomially with the angle and the radius (Figure 3.1 (a)). We construct the skeleton presentation to summarize the geometric structure (Figure 3.1 (b,c)) and project the covariates onto the skeleton. The regression function on the skeleton is estimated using kernel smoothing (Section 3.3.1, illustrated in Figure 3.1 (d)) and linear spline (Section 3.3.3, illustrated in Figure 3.1 (e)). The estimated regression function can be used to predict new projected covariates. We summarize the overall procedure in Algorithm 2.

Algorithm 2 Skeleton Regression Framework

Input: Observations $(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_n, Y_n)$.

1. **Skeleton Construction.** Construct a data-driven skeleton representation of the covariates preferably assisted with subject knowledge.
2. **Data Projection.** Project the covariates onto the skeleton.
3. **Skeleton Regression Function Estimation.** Fitting regression function on the skeleton using nonparametric techniques such as kernel smoothing (Section 3.3.1), k-Nearest Neighbor (Section 3.3.2), and linear spline (Section 3.3.3).
4. **Prediction.** Project new covariates onto the skeleton and use the estimated regression function for prediction.

3.2.1 Skeleton Construction

A skeleton is a graph constructed from the sample space representing regions of interest. From a statistical perspective, a region is of interest if it encompasses a sufficient measure of probability distribution. For given covariate space $\mathcal{X} \subseteq \mathbb{R}^d$, let $\mathcal{V} = \{V_j \in \mathbb{R}^d : j = 1, \ldots k\}$ be a collection of points of interest and E be a set of edges connecting points in $\mathcal{V}$ such

that an edge $e_{j\ell} \in E$ if the region between V_j and V_ℓ is also of interest. The tuple $(\mathcal{V}, E)$ together forms a graph that represents the focused regions in the sample space. Notably, different from common graph-based regression approaches that take each sample covariate as a vertex, the set $\mathcal{V}$ takes representative points of the covariate space and has size $k \ll n$ where n is the sample size. Moreover, the points on the edges are also part of the analysis as belonging to the regions of interest, which is different from the usual knot-edge graph. While the graph $(\mathcal{V}, E)$ contains the region of interest, it is not easy to work with this graph directly. Thus, we introduce the concept of the skeleton induced by this graph.

Let $\mathcal{E} = \{tV_j + (1-t)V_\ell : t \in (0,1), e_{j\ell} \in E\}$ be the collection of line segments induced by the edge set E. We define the skeleton of $(\mathcal{V}, E)$ as $\mathcal{S} = \mathcal{V} \cup \mathcal{E}$, i.e., $\mathcal{S}$ is the points of interest and the associated line segments representing the regions of interest. Clearly, $\mathcal{S}$ is a collection of one-dimensional line segments and zero-dimensional points so it is independent of the ambient dimension d, but the physical location of $\mathcal{S}$ is meaningful as representing the region of interest. The idea of skeleton regression is to build a regression model on the skeleton $\mathcal{S}$.

A data-driven approach to construct skeleton

The skeleton should ideally be constructed based on the analyst's judgment or prior knowledge of the focus regions. However, this information may be unavailable and we have to construct a skeleton from the data. In this section, we give a brief description of a data-driven approach proposed in Wei and Chen (2023) that constructs the skeleton to represent high-density regions. The method constructs knots as the centers from the k-

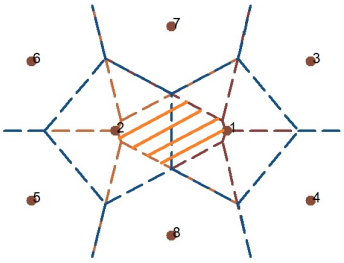


Figure 3.2: Orange shaded area illustrates the 2-NN region between knots 1 and 2.

means clustering with a large number of centers [1]. The edges are connected by examining the sample 2-Nearest-Neighbor (2-NN) region of a pair of knots (V_j, V_ℓ) (see Figure 3.2) defined as

$$B_{j\ell} = \{X_m, m = 1, \ldots, n : \|x - V_i\| > \max\{\|x - V_j\|, \|x - V_\ell\|\}, \forall i \neq j, \ell\}, \tag{3.2.1}$$

where $||.||$ denotes the Euclidean norm, and an edge between V_j and V_ℓ is added if $B_{j\ell}$ is non-empty. The method can further prune edges or segment the skeleton by using hierarchical clustering with respect to the Voronoi Density weights defined as $S^{VD}_{j\ell} = \frac{\frac{1}{n}|B_{j\ell}|}{\|V_j - V_\ell\|}$. We provide more details about this approach in Appendix I.

Remark 5. The idea of using the k-means algorithm to divide data into cells and perform analysis based on the cells has been proposed in the literature for fast computation. Sivic and Zisserman (2003), when carrying out an approximate nearest neighbor search, proposed to divide the data into Voronoi cells by k-means and do a neighbor search only in the same or some nearby cells. Babenko and Lempitsky (2012) adopted the Product Quantization technique to construct cell centers for high-dimensional data as the Cartesian product of

[1] By default $[\sqrt{n}]$. We explore the effect of choosing different numbers of knots with empirical results.

centers from sub-dimensions.

3.2.2 Skeleton-Based Distance

One of the advantages of the physically located skeleton is that it allows for a natural definition of the skeleton-based distance function $d_{\mathcal{S}}(.,.) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+ \cup \{\infty\}$. Let $\boldsymbol{s}_j, \boldsymbol{s}_\ell \in \mathcal{S}$ be two arbitrary points on the skeleton and note that, different from the usual geodesic distance on a graph, in our framework $\boldsymbol{s}_j, \boldsymbol{s}_\ell$ can be on the edges. We measure the skeleton-based distance between two skeleton points as the graph path length as defined below:

- If $\boldsymbol{s}_j, \boldsymbol{s}_\ell$ are disconnected that they belong to two disjoint components of $\mathcal{S}$, we define

$$d(\boldsymbol{s}_j, \boldsymbol{s}_\ell) = \infty \tag{3.2.2}$$

- If $\boldsymbol{s}_j$ and $\boldsymbol{s}_\ell$ are on the same edge, we define the skeleton distance as their Euclidean distance that

$$d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell) = ||\boldsymbol{s}_j - \boldsymbol{s}_\ell|| \tag{3.2.3}$$

- For $\boldsymbol{s}_j$ and $\boldsymbol{s}_\ell$ on two different edges that share a knot V_0, the skeleton distance is defined as

$$d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell) = ||\boldsymbol{s}_j - V_0|| + ||\boldsymbol{s}_\ell - V_0|| \tag{3.2.4}$$

- Otherwise, let knots $V_{i(1)}, \ldots, V_{i(m)}$ be the vertices on a path connecting $\boldsymbol{s}_j, \boldsymbol{s}_\ell$, where $V_{i(1)}$ is one of the two closest knots of $\boldsymbol{s}_j$ and $V_{i(m)}$ is the other closest knots of $\boldsymbol{s}_\ell$. We add the edge lengths of the in-between knots to the distance that

$$d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell) = ||\boldsymbol{s}_j - V_{i(1)}|| + ||\boldsymbol{s}_\ell - V_{i(m)}|| + \sum_{p=1}^{m-1} \left\| V_{i(p)}, V_{i(p+1)} \right\| \tag{3.2.5}$$

and we use the shortest path length if there are multiple paths connecting $\boldsymbol{s}_j$ and $\boldsymbol{s}_\ell$.

An example illustrating the skeleton-based distance is shown in Figure 3.3. Like the shortest path (geodesic) distance that makes a usual knot-edge graph into a metric space, the skeleton-based distance is also a metric on the skeleton graph. In the following sections, we will discuss methods to perform regression on space only with the defined metric.

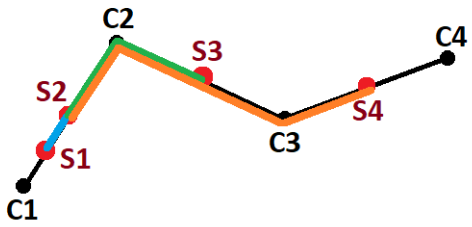


Figure 3.3: Illustration of skeleton-based distance. Let C_1, C_2, C_3, C_4 be the knots, and let S_2, S_3, S_4 be the mid-point on the edges E_{12}, E_{23}, E_{34} respectively. Let S_1 bet the midpoint between C_1 and S_2 on the edge. Let $d_{ij} = \|C_i - C_j\|$ denotes the length of the edge E_{ij}. $d_{\mathcal{S}}(S_1, S_2) = \frac{1}{4}d_{12}$ illustrated by the blue path. $d_{\mathcal{S}}(S_2, S_3) = \frac{1}{2}d_{12} + \frac{1}{2}d_{23}$ illustrated by the green path. $d_{\mathcal{S}}(S_2, S_4) = \frac{1}{2}d_{12} + d_{23} + \frac{1}{2}d_{34}$ illustrated by the orange path.

Remark 6. We may view the skeleton-based distance as an approximation of the geodesic distance on the underlying data manifold. Moreover, to make a stronger connection to the manifold structure, it is possible to define edge lengths through local manifold learning techniques that have better approximations to the local manifold structure. However, using more complex local edge weights can pose additional challenges for the data projection step described in the next section and we leave this as a future direction.

3.2.3 Data Projection

For the next step, we project the sample covariates onto the constructed skeleton. For given covariate $\boldsymbol{x}$, let $I_1(\boldsymbol{x}), I_2(\boldsymbol{x}) \in \{1, \ldots, k\}$ be the index of its closest and second closest knots in terms of the Euclidean metric. We define the projection function $\Pi(.) : \mathcal{X} \to \mathcal{S}$ for $\boldsymbol{x} \in \mathcal{S}$ as (illustrated in Figure 3.4):

Case I: If $V_{I_1(\boldsymbol{x})}$ and $V_{I_2(\boldsymbol{x})}$ are not connected, $\boldsymbol{x}$ is projected onto the closest knot that $\Pi(\boldsymbol{x}) = V_{I_1(\boldsymbol{x})}$

Case II: If $V_{I_1(\boldsymbol{x})}$ and $V_{I_2(\boldsymbol{x})}$ are connected, $\boldsymbol{x}$ is projected with the Euclidean metric onto the line passing through $V_{I_1(\boldsymbol{x})}$ and $V_{I_2(\boldsymbol{x})}$ that, let $t = \frac{\left(\boldsymbol{x} - V_{I_1(\boldsymbol{x})}\right)^T \cdot \left(V_{I_2(\boldsymbol{x})} - V_{I_1(\boldsymbol{x})}\right)}{\left\| V_{I_2(\boldsymbol{x})} - V_{I_1(\boldsymbol{x})} \right\|^2}$ be the projection proportion,

$$\Pi(\boldsymbol{x}) = V_{I_1(\boldsymbol{x})} + \left(V_{I_2(\boldsymbol{x})} - V_{I_1(\boldsymbol{x})}\right) \cdot \begin{cases} 0, \text{ if } t < 0 \\ 1, \text{ if } t > 1 \\ t, \text{ otherwise} \end{cases} \tag{3.2.6}$$

where we constrain the covariates to be projected onto the closest edge.

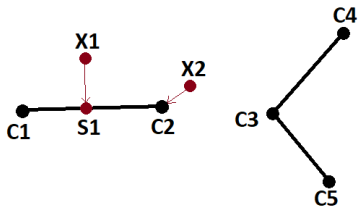


Figure 3.4: Illustration of projection to the skeleton. The skeleton structure is given by the black dots and lines. Data point X_1 is projected to S_1 on the edge between C_1 and C_2. Data point X_2 is projected to knot C_2.

Note that with the projection defined above, a non-trivial volume of points can be projected onto the knots of the skeleton graph as belonging to Case I or due to the truncation

in Case II. This adds complexities to the theoretical analysis of the proposed regression framework and leads to our separate analysis of the different domains of the graph in Section 3.3.1.

3.3 Skeleton Nonparametric Regression

Covariates are mapped onto the skeleton after the data projection step and are equipped with skeleton-based distances. In this section, we apply nonparametric regression techniques to the skeleton graph with projected data points. We study three feasible nonparametric approaches: the skeleton-based kernel regression (S-Kernel), the skeleton-based k-nearest-neighbor method (S-kNN), and the linear spline on the skeleton (S-Lspline). At the end of this section, we discuss the challenges of applying some other nonparametric regression methods in the setting of skeleton graphs.

3.3.1 Skeleton Kernel Regression

We start by adopting kernel smoothing to the skeleton graph. Let $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$ be the projections on the skeleton from $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$, i.e., $\boldsymbol{s}_i = \Pi(\boldsymbol{x}_i)$. With the skeleton-based distances, the skeleton kernel regression makes a prediction at the location $\boldsymbol{s} \in \mathcal{S}$ as

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{i=1}^{N} K(d_{\mathcal{S}}(\boldsymbol{s}_i, \boldsymbol{s})/h) Y_i}{\sum_{j=1}^{N} K(d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s})/h)}, \tag{3.3.1}$$

where $K(\cdot) \geq 0$ is a smoothing kernel such as the Gaussian kernel and $h > 0$ is the smoothing bandwidth that controls the amount of smoothing. In practice, we choose h by cross-validation. Essentially, the estimator $\hat{m}(\boldsymbol{s})$ is the kernel regression applied to a general

metric space (skeleton) rather than the usual Euclidean space. Notably, the kernel function calculation only depends on the skeleton distances and hence is independent of neither the ambient dimension of the original input nor the intrinsic dimension of the manifold structure.

It should be noted that $\hat{m}(\boldsymbol{s})$ only makes predictions on the skeleton $\mathcal{S}$. If we are interested in predicting the outcome at any arbitrary point $\boldsymbol{x} \in \mathcal{X}$, the prediction will be based on the projected point, i.e., $\hat{m}(\boldsymbol{x}) = \hat{m}\left(\Pi(\boldsymbol{x})\right)$, where $\Pi(\boldsymbol{x}) \in \mathcal{S}$. Because of the above projection property, one can think of the skeleton kernel regression as an estimator to the following skeleton-projected regression function

$$m_{\mathcal{S}}(\boldsymbol{s}) = \mathbb{E}(\boldsymbol{Y}|\Pi(\boldsymbol{X}) = \boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}. \tag{3.3.2}$$

We study the convergence of $\hat{m}(\boldsymbol{s})$ to $m_{\mathcal{S}}(\boldsymbol{s})$ in what follows.

Remark 7. Admittedly, the projection of the covariates onto the skeleton as described in Section 3.2.3 introduces the projection error between the true regression function $m(\boldsymbol{x}) = \mathbb{E}(Y|X = \boldsymbol{x})$ and the skeleton-projected regression function. Bounding this projection error involves not only a precise characterization of the underlying manifolds and the data distribution around them but also the physical locations of the skeleton relative to the local manifold structure. Due to such complexity, a theoretical result bounding the projection error under some general conditions requires careful formulation (despite that results are straightforward for particular cases such as having covariates exactly on a 1D circular segment, with example in Appendix L). We leave the in-depth analysis of the projection as future work and focus on generalizing the nonparametric regression methods to the skeleton graph in this work.

Consistency of S-Kernel Regression

Our analysis assumes that the skeleton is fixed and given and focuses on the estimation of the regression function. To evaluate the estimation error, we must first impose some concepts of distribution on the skeleton. However, due to the covariate projection procedure, the probability measures on the knots and edges are different, and we analyze them separately (see Chen and Dobra (2020); Chen (2019) for dealing with singular measures). On an edge, the domain of the projected regression function varies in one dimension, resulting in a standard univariate problem for estimation. For the case of knots, a nontrivial region of the covariate space can be projected onto a knot, leading to a nontrivial probability mass at the knot.

For simplicity, we write $K_h(\boldsymbol{s}_j, \boldsymbol{s}_\ell) \equiv K(d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell)/h)$ for $\boldsymbol{s}_j, \boldsymbol{s}_\ell \in \mathcal{S}$. Let $\mathcal{B}(\boldsymbol{s}, h) = \{\boldsymbol{s}' \in \mathcal{S} : d_{\mathcal{S}}(\boldsymbol{s}', \boldsymbol{s}) < h\}$ be the ball on skeleton centered at the point $\boldsymbol{s} \in \mathcal{S}$ with radius h. We can decompose the kernel regression estimator into edge parts and knot parts as

$$\begin{aligned}
\hat{m}(\boldsymbol{s}) &= \frac{\sum_{j=1}^n Y_j K_h(\boldsymbol{s}_j, \boldsymbol{s})}{\sum_{j=1}^n K_h(\boldsymbol{s}_j, \boldsymbol{s})} \\
&= \frac{\frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}) + \frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V})}{\frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}) + \frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V})} \\
&= \frac{\frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h)) + \frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h))}{\frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h)) + \frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{s}_j, \boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h))}
\end{aligned} \tag{3.3.3}$$

In the last line, we emphasize that the knots and edges in the kernel estimator have a meaningful contribution only within the support of the kernel function. We inspect the different domain cases separately in the following sections.

For the model and assumptions, we let $Y_j = m_{\mathcal{S}}(\boldsymbol{S}_j) + U_j, \boldsymbol{S}_j \in \mathcal{S}$, and $\mathbb{E}(U_j|\boldsymbol{S}_j) = 0$ almost surely. Let $\sigma^2(\boldsymbol{s}) = \mathbb{E}(U_j^2|\boldsymbol{S}_j = \boldsymbol{s})$. Let the density on the skeleton edge be defined as

the 1-Hausdorff density that $g(\boldsymbol{s}) = \lim_{r \downarrow 0} \frac{P(\boldsymbol{S} \in \mathcal{B}(\boldsymbol{s}, r))}{2r}$. Note that $g(\boldsymbol{s}) = \infty$ if $\boldsymbol{s}$ is at a knot point that has a probability mass. We consider the following assumptions:

A1 $\sigma^2(\boldsymbol{s})$ is continuous and uniformly bounded.

A2 The skeleton edge density function $g(\boldsymbol{s}) > 0$ and are bounded and Lipschitz continuous for $\boldsymbol{s} \in \mathcal{E}$.

A3 $m_{\mathcal{S}}(\boldsymbol{s})g(\boldsymbol{s})$ is bounded and Lipschitz continuous for $\boldsymbol{s} \in \mathcal{E}$.

K The kernel function has compact support and satisfies $\int K(x)dx = 1$, $\int K^2(x)dx < \infty$, $\int xK(x)dx = 0$, and $\int x^2K(x)dx < \infty$

Conditions A1 and K are general assumptions that are commonly made in kernel regression analysis. A2 and A3 are mild conditions that can be sufficiently implied by the boundedness and Lipschitz continuity of the density and regression function in the ambient space along with non-overlapping knots that the area of the orthogonal complements have Lipschitz changes. We do not assume the second-order smoothness commonly required for kernel regression because requiring higher-order derivative smoothness would necessitate specifying directions on the graph, which may present difficulties in model formulation. We include further discussions on formulating the derivatives on the skeleton in Section 3.3.4.

Convergence of the Edge Point

We first look at an edge point $\boldsymbol{s} \in E_{j\ell} \in \mathcal{E}$. In this case, as $n \to \infty, h \to 0$, for sufficiently large n, we have $\mathcal{B}(\boldsymbol{s}, h) \subset E_{j\ell}$, and the skeleton distance is the 1-dimensional Euclidean distance for any point within the support. Therefore, we have a convergence rate similar to the 1-dimensional kernel regression estimator (Bierens, 1983; Wasserman, 2006b; Chen

et al., 2017).

Theorem 3 (Consistency on Edge Points). Let $\boldsymbol{s} \in \mathcal{E}$ be a point on the edge. Assume conditions (A1-3) hold for all points in $\mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h)$ and (K) for the kernel function. When $n \to \infty$, $h \to 0$, $nh \to \infty$, we have

$$|\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})| = O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right) \tag{3.3.4}$$

We leave the proof in Appendix K. Theorem 3 gives the convergence rate for a point on the edge of the constructed skeleton. The convergence rate at the bias is $O(h)$, which is the usual rate when we only have Lipschitz smoothness (A2) of $m_{\mathcal{S}}$. One may be wondering if we can obtain a faster rate such as $O(h^2)$ if we assume higher-order smoothness of $m_{\mathcal{S}}$. While it is possible to obtain a faster rate if we have a higher-order smoothness, we note that this assumption will not be reasonable on the skeleton because $m_{\boldsymbol{S}}(\boldsymbol{s}) = \mathbb{E}(Y|\Pi(X) = \boldsymbol{s})$ is defined via projection. The region being projected onto $\boldsymbol{s}$ is continuously changing and may not be differentiable due to the boundary of Voronoi cells. Therefore, the Lipschitz continuity (A2) is reasonable while higher-order smoothness is not.

Convergence of the Knots with Nonzero Mass

We then look at the knots with nonzero probability mass that $\boldsymbol{s} \in \mathcal{V}$ with $p(\boldsymbol{s}) > 0$, where we use $p(\boldsymbol{s})$ to denote the probability mass on a knot. This case mainly occurs for knots with degree 1 on the skeleton graph, when a non-trivial region of points is projected onto such knots. For example, refer to knot C2 in Figure 3.4.

Theorem 4 (Consistency on Knots with Nonzero Mass). Let $\boldsymbol{s} \in \mathcal{V}$ be a point at a knot and the probability mass at $\boldsymbol{s}$ be $P(\Pi_{\mathcal{S}}(X) = \boldsymbol{s}) \equiv p(\boldsymbol{s}) > 0$ and assume $\sigma^2(\boldsymbol{s})$ bounded. Also, assume conditions (A1-3) hold for all points in $\mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h)$ and (K) for the kernel function. When $n \to \infty$, $h \to 0$, and $nh \to \infty$, we have

$$|\hat{m}(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})| = O(h) + O_p\left(\sqrt{\frac{1}{n}}\right) \tag{3.3.5}$$

Theorem 4 gives the convergence result for a knot point with a nontrivial mass of the skeleton. The bias term $O(h)$ comes from the influence of nearby edge points. For the stochastic variation part, instead of having the $O_p\left(\sqrt{\frac{1}{nh}}\right)$ rate as the usual kernel regression and in Theorem 3, we have $O_p\left(\sqrt{\frac{1}{n}}\right)$ rate which comes from averaging the observations projected onto the knots. The proof of Theorem 4 is provided in Appendix K.

Convergence of the Knots with Zero Mass

We now look at a knot point $\boldsymbol{s} \in \mathcal{V}$ with no probability mass that $p(\boldsymbol{s}) = 0$. This can be the case for a knot with a degree larger than 1 like knot C3 in Figure 3.4. Since we define edge sets excluding the knots, there will be no density as well as no probability mass at $\boldsymbol{s}$. Note that, with some reformulation, degree 2 knots can be parametrized together with the two connected edges and, under the appropriate assumptions, Theorem 3 applies, giving consistency estimation with $O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)$ rate. However, density cannot be extended directly to knots with a degree larger than 2, but the kernel estimator still converges to some limits as presented in the Proposition below.

Proposition 5. Let $\boldsymbol{s} \in \mathcal{V}$ be a point at a knot such that the probability mass at $\boldsymbol{s}$ be $P(\Pi_{\mathcal{S}}(X) = \boldsymbol{s}) \equiv p(\boldsymbol{s}) = 0$. Assume conditions (A1-3) hold for all points in $\mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h)$ and (K) for the kernel function. Let $\mathcal{I}$ collect the indexes of edges with one knot being $\boldsymbol{s}$. For $\ell \in \mathcal{I}$ and edge E_ℓ connects $\boldsymbol{s}$ and V_ℓ, let $g_\ell(t) = g((1-t)\boldsymbol{s}+tV_\ell)$ and $g_\ell(0) = \lim_{x \downarrow 0} g_\ell(x)$. Let $m_\ell(t) = m_{\mathcal{S}}((1-t)\boldsymbol{s} + tV_\ell)$ and $m_\ell(0) = \lim_{t \downarrow 0} m_\ell(t)$. When $n \to \infty$, $h \to 0$, and $nh \to \infty$, we have

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{\ell \in \mathcal{I}} m_\ell(0) g_\ell(0)}{\sum_{\ell \in \mathcal{I}} g_\ell(0)} + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right). \tag{3.3.6}$$

Proposition 5 shows that, under proper conditions, the skeleton kernel estimator on a zero-mass knot converges to the weighted average of the limiting regression values of the connected edges, and the convergence rate is the same as the edge points shown in Theorem 3. The proof is included in Appendix K.

Remark 8. The domain $\mathcal{S}$ of the regression function can be seen as bounded, and hence the boundary bias issue can arise. The true manifold structure's boundary can be different from the boundary of the skeleton graph, making the consideration of the boundary more complicated. However, the boundary of the skeleton is the set of degree 1 knots, and, under our formulation, knots have discrete measures, so the consideration of boundary bias may not be necessary for the proposed formulation. However, some boundary corrections can potentially improve the empirical performance and we leave it for future research.

3.3.2 Skeleton kNN regression

The k-Nearest Neighbor (kNN) method can be easily applied to the skeleton using the distance on the skeleton. For a given point on the skeleton at $\boldsymbol{s} \in \mathcal{S}$, we define the distance to the k-th nearest observation on the skeleton as

$$R_k(\boldsymbol{s}) = \min \left\{ r > 0 : \sum_{i=1}^{n} I(d_{\mathcal{S}}(\boldsymbol{s}_i, \boldsymbol{s}) \leq r) \geq k \right\}. \tag{3.3.7}$$

Note that it is possible to have multiple observations being the k-th nearest observation due to observations being projected to the vertices. In this case, we can either randomly choose from them or consider all of them. Here we include all of them in the calculation. The skeleton-based kNN regression (S-kNN) predicts the value of outcome at $\boldsymbol{s}$ as

$$\hat{m}_{SkNN}(\boldsymbol{s}) = \frac{\sum_{i=1}^{k} Y_i I(d_{\mathcal{S}}(\boldsymbol{s}_i, \boldsymbol{s}) \leq R_k(\boldsymbol{s}))}{\sum_{j=1}^{k} I(d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}) \leq R_k(\boldsymbol{s}))}. \tag{3.3.8}$$

Different from the usual kNN regressor with the covariates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, which selects neighbors through Euclidean distance in the ambient space, the S-kNN regressor chooses neighbors with skeleton-based distances after projection onto the skeleton graph. Measuring proximity with the skeleton can improve the regression performance when the dimension of the covariates is large, which we empirically show in Section 3.4.

Remark 9. It is well known that the usual k_nNN regressor can be consistent if we let k_n grow as a function of the sample size n, and under appropriate assumptions, Györfi et al. (2002) give the convergence rate of the k_nNN estimate m_n to the true function m as

$$\mathbb{E} \left\| m_n - m \right\|^2 \leq \frac{\sigma^2}{k_n} + c_1 \cdot C^2 \left(\frac{k_n}{n} \right)^{2/d}$$

Later, Kpotufe (2011) has shown that the convergence rate of kNN regressor depends on

the intrinsic dimension. We expect a similar result with $d = 1$ rate for the skeleton kNN regression at an edge point.

3.3.3 Linear Spline Regression on Skeleton

In this section, we propose a skeleton-based linear spline model (S-Lspline) for regression estimation. By construction, this approach results in a continuous model across the graph. Moreover, we show that the skeleton-based linear spline corresponds to an elegant parametric regression model on the skeleton. As the skeleton $\mathcal{S}$ can be decomposed into the edge component $\mathcal{E}$ and the knot component $\mathcal{V}$, the linear spline regression on the skeleton can be written as the following constrained model:

$$\begin{aligned} f : \mathcal{S} \ \to \ \mathbb{R} \quad \text{such that } 1.\ & f(x) \text{ is linear on } x \in \mathcal{E}, \\ 2.\ & f(x) \text{ is continuous at } x \in \mathcal{V}. \end{aligned} \tag{3.3.9}$$

While solving the above constrained problem may not be easy, we have the following elegant representer theorem showing that a linear spline on the skeleton can be uniquely characterized by the values on each knot.

Theorem 1 (Linear spline representer theorem). Any function satisfying equation (3.3.9) can be characterized by $\{f(v) : v \in \mathcal{V}\}$ and for $x \in \mathcal{E}$, $f(x)$ is linear interpolation between the values on the two knots on the edge that x belongs to.

Proof. Let f be a function satisfying equation (3.3.9). By construct, f is linear for $x \in \mathcal{E}$ and is continuous at $x \in \mathcal{V}$. Let V_j and V_ℓ be two knots that share an edge and

let $E_{j\ell} = \{x = tV_j + (1-t)V_\ell : t \in (0,1)\}$ be the shared edge segment. For any $x \in \mathcal{E}$, there exists a pair (V_j, V_ℓ) such that $x \in E_{j\ell}$. Because f is linear in $E_{j\ell}$, f can be uniquely characterized by the pairs $(f(e_1), e_1), (f(e_2), e_2)$ for two distinct points $e_1, e_2 \in \bar{E}_{j\ell}$, where $\bar{E}_{j\ell} = \{x = tV_j + (1-t)V_\ell : t \in [0,1]\}$ is the closure of $E_{j\ell}$. Thus, we can pick $e_1 = V_j$ and $e_2 = V_\ell$, which implies that f on the segment $E_{j\ell}$ is parameterized by $f(v_j)$ and $f(V_\ell)$, the values on the two knots.

By applying this procedure to every edge segment, we conclude that any function satisfying the first condition in (3.3.9) can be characterized by the values of the knots. The second condition in (3.3.9) will require that every knot has one consistent value. As a result, any function f satisfying (3.3.9) can be uniquely characterized by the values on the knot $\{f(x) : x \in \mathcal{V}\}$ and $f(x)$ will be a linear interpolation when $x \in \mathcal{E}$.

□

Using Theorem 1, we only need to determine the values on the knots. Let $\boldsymbol{\beta} \in \mathbb{R}^k$ be the values of the skeleton linear spline model on each knot with $k = |\mathcal{V}|$ being the number of knots. As is argued previously, the spline model is parameterized by $\boldsymbol{\beta}$, so we only need to estimate $\boldsymbol{\beta}$ from the data. Given $\boldsymbol{\beta}$, the predicted value of each $\boldsymbol{y}_i$ is a linear interpolation depending on the projected location of each $\boldsymbol{x}_i$.

To derive an analytic form of $\boldsymbol{y}_i$, we introduce a transformed covariate matrix $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T \in \boldsymbol{R}^{n \times k}$ as follows:

1. If $\boldsymbol{x}_i$ is projected onto a vertex that $\boldsymbol{s}_i = V_j$ for some j, then

$$\boldsymbol{z}_{ij'} = I(j' = j).$$

2. If $\boldsymbol{x}_i$ is projected onto an edge between knots V_j and V_ℓ, then

$$\boldsymbol{z}_{ij} = \frac{||\boldsymbol{s}_i - V_j||}{||V_j - V_\ell||}, \quad \boldsymbol{z}_{i\ell} = \frac{||\boldsymbol{s}_i - V_\ell||}{||V_j - V_\ell||}, \quad \text{and } \boldsymbol{z}_{ij'} = 0 \text{ for } j' \neq j, \ell.$$

With the above feature transform, the predicted value of $\boldsymbol{y}_i$ by the S-Lspline model is

$$\hat{\boldsymbol{y}}_i = \boldsymbol{\beta}^T \boldsymbol{z}_i. \tag{3.3.10}$$

To see this, if $\boldsymbol{x}_i$ is projected onto a vertex that $\boldsymbol{s}_i = V_j$ for some j, the linear model with transformed covariates gives $\boldsymbol{\beta}^T \boldsymbol{z}_i = \boldsymbol{\beta}_j$, the predicted value on vertex V_j. In the case where $\boldsymbol{x}_i$ is projected onto an edge between knots V_j and V_ℓ, let $\boldsymbol{\beta}_j$ and $\boldsymbol{\beta}_\ell$ be the corresponding predicted values at V_j and V_ℓ, and the linear interpolation between $\boldsymbol{\beta}_\ell$ and $\boldsymbol{\beta}_j$ at $\boldsymbol{s}_i$ can be written as

$$\boldsymbol{\beta}_j + \frac{||\boldsymbol{s}_i - V_j||}{||V_j - V_\ell||} \cdot (\boldsymbol{\beta}_\ell - \boldsymbol{\beta}_j) = \frac{||\boldsymbol{s}_i - V_\ell||}{||V_j - V_\ell||} \cdot \boldsymbol{\beta}_j + \frac{||\boldsymbol{s}_i - V_j||}{||V_j - V_\ell||} \cdot \boldsymbol{\beta}_\ell = \boldsymbol{\beta}^T \boldsymbol{z}_i.$$

To estimate $\boldsymbol{\beta}$, we can apply the least squares procedure to get:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \mathsf{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2 \\ &= \mathsf{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\beta}^T \boldsymbol{z}_i)^2. \end{aligned}$$

So it becomes a linear regression model and the solution can be elegantly written as

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T \boldsymbol{Z})^{-1} \boldsymbol{Z} \boldsymbol{y}.$$

Note that in a sense, the above procedure can be viewed as placing a linear model

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{\beta}^T \boldsymbol{Z},$$

where $\boldsymbol{Z}$ is a transformed covariate matrix from $\boldsymbol{X}$. Note that the S-Lspline model with the graph-transformed covariates does not include an intercept.

Remark 10. An alternative justification of the value-on-knots parameterization is to calculate the degree of freedom. On each graph, the sum of the vertex degrees is twice the number of edges since each edge is counted from both ends. Let e be the number of edges in the graph, let v be the number of vertices, and let r be the sum of all the vertex degrees, we have $r = 2e$. For the S-Lspline model, we construct a linear model with 2 free parameters for each edge, and thus without any constraints, the total number of degrees of freedom is $2e$. For each vertex V_i with degree r_i, the continuity constraint imposes $r_i - 1$ equations, and as a result, the continuity constraints consume a total of $\sum_{i=1}^{v} r_i - 1 = r - v$ degrees of freedom. Combining it, we have $2e - (r - v) = v$ degrees of freedom, which matches the degrees of freedom given by the parametrization of values on the knots.

Regularized Linear Spline Method

Given the formulation of the S-Lspline as a linear regression with transformed data, it is natural to incorporate penalization with this method. In this section, we introduce penalization into the S-Lspline method by making connections to the literature about regularization on graphs, with a particular focus on graph Laplacian smoothing by Smola and Kondor (2003) and graph trend filtering by Wang et al. (2016).

Let B be the (unoriented) incidence matrix of the skeleton graph that

$$B_{ij} = \begin{cases} 1 & \text{if vertex } v_i \text{ is incident with edge } e_j \\ 0 & \text{otherwise.} \end{cases},$$

for $i = 1, \ldots, k$ where k is the number of knots and $j = 1, \ldots, m$ where m is the number of edges in the skeleton graph. Let L denote the Laplacian matrix that $L = D - A = BB^T$

where D is the degree matrix and A is the adjacency matrix of the skeleton graph. The q-th order trend filtering matrix, for $q \in \{0, 1, 2, \dots\}$, is defined as

$$\Delta^{(q+1)} = \begin{cases} L^{\frac{q+1}{2}} & \text{for odd } q, \\ BL^{q/2} & \text{for even } q. \end{cases}$$

The q-th order Laplacian smoothing can be taken as the L_2 penalty with the trend filtering matrix, and we have the regularized problem to be

$$\mathsf{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}\|_2^2 + \lambda \left\|\Delta^{(k+1)}\boldsymbol{\beta}\right\|_2$$

where $\left\|\Delta^{(k+1)}\boldsymbol{\beta}\right\|_2 = \boldsymbol{\beta}^T L^{k+1}\boldsymbol{\beta}$ for Laplacian matrix L, and $\boldsymbol{Z}$ the transformed covariate matrix from $\boldsymbol{X}$. This can be solved as a Generalized Ridge problem[2].

The Trend Filtering regularization similarly applies a L_1 penalty and the problem becomes

$$\mathsf{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}\|_2^2 + \lambda \left\|\Delta^{(k+1)}\boldsymbol{\beta}\right\|_1.$$

We follow Tibshirani and Taylor (2011) to get the solution to the generalized Lasso problem. We include the algorithm in Appendix K for completeness. Empirically, we observe that penalization does not improve the regression results of the S-Lspline model (see Appendix M). To account for this, note that the skeleton graph is a summarizing presentation of the data with a concise structure, and the S-Lspline method assumes a simple piecewise linear model on the skeleton which inherits the simple geometric structure and is not a complex model in nature, and hence adding penalization does not improve the performance of this

[2]Generally if the penalty matrix L^{k+1} is positive definite, the generalized penalty is a non-degenerated quadratic form in β, and hence strictly convex. The analytical solution is then

$$\hat{\beta} = \left(X^T X + \lambda L^{k+1}\right)^{-1}, \left(X^T Y\right)$$

However, the Laplacian matrix is only positive semi-definite, and therefore the loss function need not be strictly convex. Some work suggests adding $\|\boldsymbol{\beta}\|_2^2$ as an additional penalty to address this, but we do not implement that to be consistent with the trend filtering penalization.

method.

3.3.4 Challenges of Other Nonparametric Regression

In this section, we discuss the challenges when applying other nonparametric regression methods to the skeleton. Particularly, the skeleton graph is only equipped with a metric and does not have a well-defined inner product or orientation, which makes many conventional approaches not directly applicable.

Local polynomial regression

Local polynomial regression Fan and Gijbels (2018) is a common generalization of the kernel regression that tries to improve the kernel regression estimator by using higher-order polynomials as local approximations to the regression function. In the Euclidean space, a p-th order local polynomial regression aims to choose $\beta(\boldsymbol{x})$ via minimizing

$$\sum_{i=1}^{n}\left[Y_i-\sum_{j=0}^{p}\beta_j(\boldsymbol{x}_i-\boldsymbol{x})^j\right]^2 K\left(\frac{\boldsymbol{x}_i-\boldsymbol{x}}{h}\right) \tag{3.3.11}$$

and predict $m(\boldsymbol{x})$ via $\hat{\beta}(\boldsymbol{x})$, the first element in the minimizer. Note that when $p=1$, one can show that this is equivalent to the kernel regression.

Unfortunately, the local polynomial regression cannot be easily adapted to the skeleton because the polynomial $(\boldsymbol{x}_i-\boldsymbol{x})^j$ requires a well-defined orientation, which is ill-defined at a knot (vertex). Directly replacing $(\boldsymbol{s}_i-\boldsymbol{s})$ with the distance $d_{\mathcal{S}}(\boldsymbol{s}_i-\boldsymbol{s})$ will make all the polynomials to be non-negative, which will be problematic for odd orders. Unless in some special skeletons such as a single chain structure, the local polynomial regression cannot be

directly applied.

Higher-Order Spline

In Section 3.3.3, we introduce the linear spline model. One may be curious about the possibility of using a higher-order spline (enforcing higher-order smoothness on knots; see, e.g., Chapter 5.5 of Wasserman (2006b)). Unfortunately, the higher-order spline is generally not applicable to the skeleton because a higher-order spline requires derivatives and the concept of a derivative may be ill-defined on a knot because of the lack of orientation. To see this, consider a knot with three edges connecting to it. There is no simple definition of derivative at this knot unless we specify the orientation of these three edges.

One possible remedy is to introduce an orientation for every edge. This could be done by ordering the knots first and, for every edge, the orientation is always from a lower index vertex to the higher index vertex. With this orientation, it is possible to create a higher-order spline on the skeleton but the result will depend on the orientation we choose.

Even with edge directions provided and the derivatives on the skeleton defined, higher-order spline on the skeleton can be prone to overfitting. Classical spline methods use degree $p+1$ polynomial functions to achieve continuity at p-th order derivative. For example, univariate cubic splines use polynomials up to degree 3 to ensure the second-order smoothness of the regression function at each knot. However, on a graph, degree $p+1$ polynomial functions may fail to achieve continuity at p-th order derivative, and on complete graphs, which is the worst case, $2p+1$ degree polynomials are needed instead.

Smoothing Spline

Smoothing spline Wang (2011); Wahba (1975) is another popular approach for curve-fitting that attempts to find a smooth curve that minimizes the square loss in the prediction with a penalty on the curvature (second or higher-order derivatives).

The major difficulty of this method is that the concept of a *smooth* function is ill-defined at a knot even if we have a well-defined orientation. In fact, the 'linear function' is not well-defined in general on a skeleton's knot. To see this, consider a knot V_0 with three edges e_1, e_2, e_3 connecting to V_1, V_2, V_3, respectively. Suppose we have a linear function f_0 and f_0 is linearly increasing on paths $V_1 - V_0 - V_2$ and $V_1 - V_0 - V_3$. However, on the path $V_2 - V_0 - V_3$, the function f_0 will be decreasing $(V_2 - V_0)$ and then increasing $(V_0 - V_3)$, leading to a non-smooth structure.

Orthonormal Basis and Tree

Orthonormal basis approach (see, e.g., Chapter 8 of Wasserman (2006b)) uses a set of orthonormal basis functions to approximate the regression function. In general, it is unclear how to find a good orthonormal basis for a skeleton unless the skeleton is simply a circle or a chain.

Having said that, it is possible to construct an orthonormal basis borrowing the idea from wavelets (Torrence and Compo, 1998). The key idea is that the skeleton is a measurable set that we can measure its (one-dimensional) volume. Thus, we can partition the skeleton $\mathcal{S}$ into two equal-volume sets A_1, A_2. Note that the resulting sets A_1, A_2 are not necessarily

skeletons because we may cut an edge into two pieces. For each set A_j, we can further partition it again into equal volume sets $A_{j,1}, A_{j,2}$. And we can repeat this dyadic procedure to create many equal-volume subsets. We then define a basis as follows:

$$
\begin{aligned}
f_0(s) &= 1, \\
f_1(s) &= I\left(s \in A_1\right) - I\left(s \in A_2\right) \\
f_2(s) &= I(s \in A_{1,1}) - I(s \in A_{1,2}) \\
f_3(s) &= I(s \in A_{2,1}) - I(s \in A_{2,2}) \\
&\vdots
\end{aligned}
$$

After normalization, this set of functions forms an orthonormal basis. With this basis, it is possible to fit an orthonormal basis on the skeleton. However, the above construction creates the partition arbitrarily. The fitting result depends on the particular partition we use to generate the basis and it is unclear how to pick a reasonable partition in practice.

The regression tree Breiman (2017); Loh (2014) is a popular idea in nonparametric regression that fits the data via creating a tree of partitioning the whole sample space whose leaves represent a subset of the sample space and predicts the response using a single parameter at each leaf (region). This idea could be applied to the skeleton using a similar procedure as the construction of an orthonormal basis that we keep splitting a region into two subsets (but we do not require the two subsets to be of equal size). However, unlike the usual regression tree (in Euclidean space) that the split of two regions is often at a threshold at one coordinate, the split of a skeleton may not be easily represented as the skeleton is just a connected subregion of Euclidean space. Therefore, similar to the orthonormal basis,

regression tree may be used in skeleton regression, but there is no simple and principled way to create a good partition.

3.4 Simulations

In this section, we use simulated data to evaluate the performance of the proposed skeleton regression framework. [3] We first demonstrate an example with the intrinsic domain composed of several disconnected components, which we call the Yinyang data (Section 3.4.2). Then, we add noisy observations to the Yinyang data (Section 3.4.3) to show the effectiveness of our method in handling noisy data points. Moreover, we present an example where the domain is a continuous manifold with a Swiss roll shape (Section 3.4.4). In all the simulations in this section, there are random perturbations in the intrinsic dimensions, and we add random Gaussian variables as covariates to increase the ambient dimension.

3.4.1 Analysis Procedure

We apply the following analysis procedure for all the simulations in this section. We randomly generate the dataset for 100 times, and, on each dataset, we use 5-fold cross-validation to calculate the sum of squared errors (SSE) as the performance assessment. We use the skeleton construction method described in Section 3.2.1 to construct skeletons with varying numbers of knots on each training set. In this section, we present results where

[3] R implementation of the proposed skeleton regression methods can be accessed at https://github.com/JerryBubble/skeletonMethods and Python implementation can be accessed at https://pypi.org/project/skeleton-methods/.

the construction procedure cuts the skeleton into a given number of disjoint components according to the Voronoi Density weights (Section 3.2.1). We also empirically tested using different cuts to get skeleton structures with different numbers of disjoint components under the same number of knots and noticed little change in the squared error performance (see Appendix M).

We evaluate the skeleton-based nonparametric regressors introduced in Section 3.3: skeleton kernel regression (S-Kernel), k-NN regressor using skeleton-based distance (S-kNN), and the skeleton spline model(S-Lspline). For S-Kernel and S-kNN methods, To simplify the calculation, we only compute the skeleton-based distances between points in the same or neighboring Voronoi cells. That is the skeleton-based distance between a pair of points is calculated when they share at least one knot from their respective set of two closest knots. For the S-Lspline method, we include the results without additional penalization in this section. We compare the empirical performance of the S-Lspline method with various penalizations discussed in Section 3.3.3 on the simulated datasets and present the results in Appendix M, and we observe that incorporating penalization terms does not improve the empirical performance of the S-Lspline method.

For comparisons, we apply the classical k-nearest-neighbors regression based on Euclidean distances (kNN). For penalization regression methods, we test Lasso and Ridge regression. Among the recent manifold and local regression methods, we include the Spectral Series approach (Lee and Izbicki, 2016) with the radial kernel (SpecSeries) for its superior performance[4]

[4]The Spectral Series approach demonstrates similar empirical performance as the kernel machine learning methods with regularization in RKHS as in Lee and Izbicki (2016).

and readily available R implementation [5]. For kernel machine learning approaches, we include the Divide-and-Conquer Kernel Ridge Regression (Fast-KRR) method as in Zhang et al. (2013). For Fast-KRR, we set the penalization hyperparameter $\lambda = 1/n$ and set the number of random partitions $m = \sqrt{n}$ where n is the size of the training sample, and use the radial kernel where the best bandwidth σ is given by grid search.

For the simulations presented in this section, we add random Gaussian variables to create settings with a large ambient dimension of 1000 to demonstrate that the proposed skeleton regression framework is robust under such challenging scenarios. For completeness, we also include the simulation results on low-dimensional data settings in Appendix M, and the skeleton-based regression methods also show competitive performance in such settings.

3.4.2 Yinyang Data

The covariate space of Yinyang data is intrinsically composed of 5 disjoint structures of different geometric shapes and different sizes: a large ring of 2000 points, two clumps each with 400 points (generated with the `shapes.two.moon` function with default parameters in the `clusterSim` library in R (Walesiak and Dudek, 2020)), and two 2-dimensional Gaussian clusters each with 200 points (Figure 3.6 left). Together there are a total of 3200 observations. Note that the intrinsic structures of the components are curves and points, and, with perturbations, the generated covariates do not lay exactly on the corresponding manifold structures. The responses are generated from a trigonometric function on the ring and

[5] https://projecteuclid.org/journals/supplementalcontent/10.1214/16-EJS1112/supzip_1.zip

constant functions on the other structures with random Gaussian error(Figure 3.6 right). That is, let $\epsilon \sim N(0, 0.01)$ and let θ be the angle of the covariates, then

$$Y = \epsilon + \begin{cases} \sin(\theta * 4) + 1.5 & \text{for points on the outer ring} \\ 0 & \text{for points on the bottom-right Gaussian cluster} \\ 1 & \text{for points on the right clump} \\ 2 & \text{for points on the left clump} \\ 3 & \text{for points on the upper-left Gaussian cluster} \end{cases}$$

To make the task more challenging with the presence of noisy variables, we add independent and identically distributed random $N(0, 0.01)$ variables to the generated covariates. In this section, we increase the dimension of the covariates to a total of 1000 with those added Gaussian variables.

For the Yinyang data, we cut the skeleton into 5 disjoint components during the skeleton construction process according to the Voronoi Density weights. We take the median, 5th percentile, and 95th percentile of the 5-fold cross-validation Sum of Squared Errors (SSEs) for each parameter setting of each method on the 100 datasets. We present the smallest median SSE for each method in Table 3.1 along with the corresponding best parameter setting.

We observed that all the skeleton-based methods (S-Kernel, S-kNN, and S-Lspline) perform better than the standard kNN in this setting. That is, the skeleton better captures the geometric structures of the data and improves the downstream regression performance. The three skeleton-based methods have similar performance on this simulated Yinyang data,

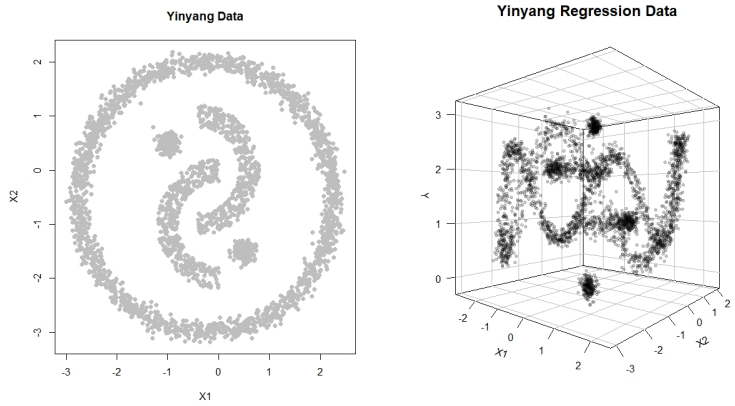


Figure 3.6: Yinyang Regression Data

Method	Median SSE (5%, 95%)	nknots	Parameter
kNN	204.5 (192.3, 221.9)	-	neighbor=18
Ridge	2127.0 (2100.2, 2155.2)		$\lambda = 7.94$
Lasso	1556.8 (1515.4, 1607.9)		$\lambda = 0.0126$
SpecSeries	1506.4 (1469.1,1555.6)	-	bandwidth = 2
Fast-KRR	2404.0 (2370.0, 2440.2)	-	$\sigma = 0.1$
S-Kernel	91.6 (81.6, 103.5)	38	bandwidth = 4 r_{hns}
S-kNN	92.7 (84.5, 102.8)	38	neighbor = 36
S-Lspline	94.4 (87.7, 103.2)	38	-

Table 3.1: Regression results on Yinyang $d = 1000$ data. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5th percentile and 95th percentile of the SSEs from the given parameter settings are reported in brackets.

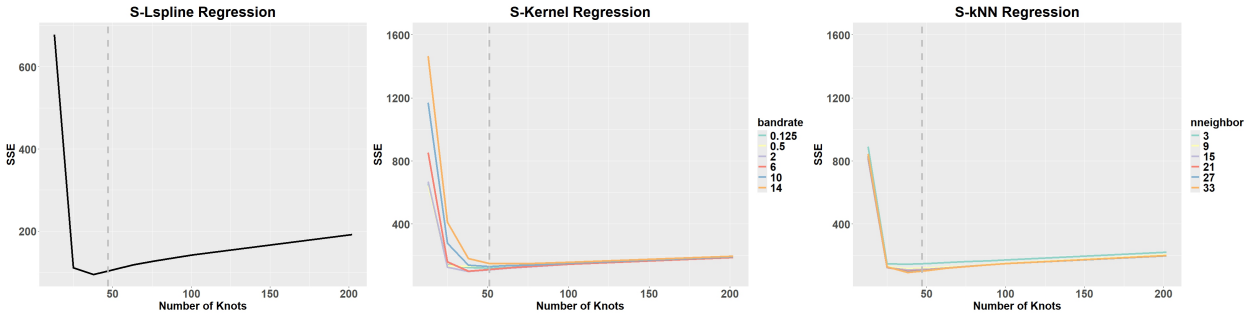


Figure 3.7: Yinyang $d = 1000$ data regression results with varying number of knots. The median SSE across the 100 simulated datasets with each given parameter setting is plotted.

but the S-Lspline method can be preferred in this case in terms of computation as it does not require the skeleton-based distance computations. The spectral method SpecSeries and the kernel machine learning approach Fast-KRR both perform worse than the classical kNN. The underlying data structure being comprised of multiple disconnected components in this case can diminish the power of such manifold learning methods. Ridge and Lasso regression, despite the regularization effect, resulted in relatively high SSEs. Therefore, the skeleton regression framework has the empirical advantage when dealing with covariates that lie around manifold structures.

In Figure 3.7, we present the median SSE of the S-Lspline, S-Kernel, and S-kNN methods on skeletons with various numbers of knots. The vertical dashed line indicates $[\sqrt{n}] = 51$ knots as suggested by the empirical rule, where n is the training sample size. The empirical rule seems to produce satisfactory results in this simulation study, roughly identifying the "elbow" position, but it's advised to use cross-validation for fine-tuning in practice.

3.4.3 Noisy Yinyang Data

To show the robustness of the proposed skeleton-based regression methods, we add 800 noisy observations to the Yinyang data in Section 3.4.2 (20% of a total of 4000 observations). The first two dimensions of the noisy covariates are uniformly sampled from the 2-dimensional square $[-3.5, 3.5]\times[-3.5, 3.5]$ and independent random normal $N(0, 0.01)$ variables are added to make the covariates 1000-dimensional in total. The responses of the noisy points are set as $1.5 + \epsilon$ with $\epsilon \sim N(0, 0.01)$, while the responses on the Yinyang covariates are generated the

same as in the previous example. The first two dimensions of the Noisy Yinyang covariates are plotted in Figure 3.9 left and the Y values against the first two dimensions of the covariates are illustrated in Figure 3.9 right.

To evaluate the robustness of the proposed skeleton-based regression methods, we randomly generate the Noisy Yinyang data 100 times and follow the analysis procedure as in Section 3.4.1, except that we leave the skeleton to be a fully connected graph. We also took the median, 5th percentile, and 95th percentile of the 5-fold cross-validation SSEs for each parameter setting of each method on the 100 datasets. The smallest median SSE for each method is reported in Table 3.2 along with the corresponding best parameter setting.

It can be observed that all the skeleton-based regression methods outperform the standard kNN approach, which indicates that the skeleton regression framework can capture the data structure in the presence of noisy observations and give good regression performance. Among the skeleton-based methods, the S-Kernel has the best performance, and kernel smoothing can be a helpful nonparametric technique to deal with noisy observations. The SpecSeries, Fast-KRR, Ridge, and Lasso regressions again fail to provide good performance on this simulated dataset. The advantage of the skeleton regression framework is more manifesting with noisy observations.

In Figure 3.7, we plot the median SSE of the skeleton-based methods on skeletons with different numbers of knots. Using the empirical rule to construct a skeleton with $[\sqrt{3200}] = 57$ knots results in good regression performance and approximately identifies the "elbow" position in Figure 3.7. However, for some skeleton-based methods, using a number of knots

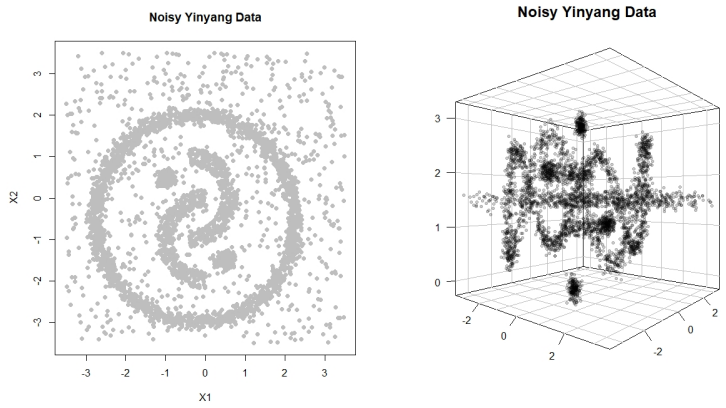


Figure 3.9: Noisy Yinyang Regression Data

Method	Median SSE (5%, 95%)	Number of knots	Parameter
kNN	440.8 (420.4, 463.0)	-	neighbor=18
Ridge	2139.1 (2102.6, 2171.1)	-	$\lambda = 6.31$
Lasso	2029.2 (1988.7, 2071.0)	-	$\lambda = 0.02$
SpecSeries	1532.0 (1490.7, 1563.2)	-	bandwidth = 2
Fast-KRR	2584.6 (2556.3, 2624.5)	-	σ =0.1
S-Kernel	313.5 (293.2, 331.1)	28	bandwidth = 2 r_{hns}
S-kNN	352.9 (332.4, 376.7)	28	neighbor = 15
S-Lspline	376.5 (354.3, 399.2)	57	-

Table 3.2: Regression results on Noisy Yinyang $d = 1000$ data.The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5 percentile and 95 percentile of the SSEs from the given parameter settings are reported in brackets.

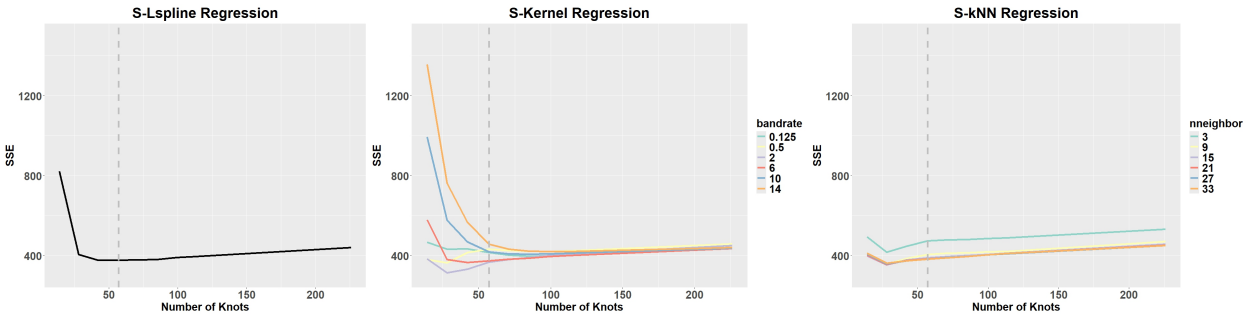


Figure 3.10: Noisy Yinyang $d = 1000$ data regression results with varying number of knots. The median SSE across the 100 simulated datasets with each given parameter setting is plotted.

larger than that given by the empirical rule leads to better regression performance. This improvement is related to the phenomenon observed in Wei and Chen (2023) that when dealing with noisy observations, it's better to have a skeleton with more knots and cut the skeleton into more disjoint components in order to have a cleaner representation of the key manifold structures. Therefore, when facing data with noisy feature vectors, it's advised to empirically tune the number of knots favoring larger values.

3.4.4 SwissRoll Data

The intrinsic components of the covariates in Yinyang data are all well-separated, which, admittedly, can give an advantage to skeleton-based methods. Moreover, the intrinsic dimensions of the structural components for Yinyang data covariates are all lower than or equal to 1 and can be straightforwardly represented by knots and line segments, potentially giving another advantage to skeleton-based methods. To address such concerns, we present another simulated data which has covariates lying around a Swill Roll shape (Figure 3.12 left), an intrinsically 2-dimensional manifold in the 3-dimensional Euclidean space. To make the density on the Swill Roll manifold balanced, we sample points inversely proportional to the radius of the roll in the X_1X_3 plane. Specifically, let u_1, u_2 be independent random variables from Uniform$(0, 1)$ and let the angle in the X_1X_3 plane be generated as $\theta_{13} = \pi 3^{u_1}$. Then for the first 3 dimensions of the covariates we have

$$X_1 = \theta_{13}\cos(\theta_{13}), \quad X_2 = 4u_2, \quad X_3 = \theta_{13}\sin(\theta_{13})$$

The true response has a polynomial relationship with the angle on the manifold if the X_2 value of the point is within some range. Let $\tilde{\theta}_{13} = \theta_{13} - 2\pi$, and let $\epsilon \sim N(0, 0.3)$. Then we set

$$Y = 0.1 \times \tilde{\theta}_{13}^3 \times [I(X_2 < \pi) + I(2\pi < X_2 < 3\pi)] + \epsilon$$

The response versus the angle θ_{13} and X_2 is demonstrated in Figure 3.12 right. Independent random Gaussian variables from $N(0, 0.1)$ are added to make the covariates 1000-dimensional in total, and 2000 observations are sampled to make the Swiss Roll dataset.

We follow the same analysis procedures as in Section 3.4.1 with the skeletons constructed to be fully connected graphs without additional graph cuts. We took the median, 5th percentile, and 95th percentile of the 5-fold cross-validation SSEs across each parameter setting for each method on the 100 datasets, and reported the smallest median SSE for each method along with the corresponding best parameter setting in Table 3.3.

All the proposed skeleton-based methods have better performance than the standard kNN regressor, while the S-Kernel method had the best performance in terms of SSE. Particularly, the methods that utilize the skeleton-based distances, S-Kernel and S-kNN, have significantly better performance compared to the S-Lspline method which only utilizes the knot-edge structure of the skeleton graph. Intuitively, the skeleton-based distances are good approximations to the geodesic distances on the manifold and hence lead to improvements in the regression performance. The spectral and penalization approaches do not demonstrate good performance on this simulated data. Therefore, the proposed skeleton regression framework can also be powerful for data on connected, multi-dimensional manifolds.

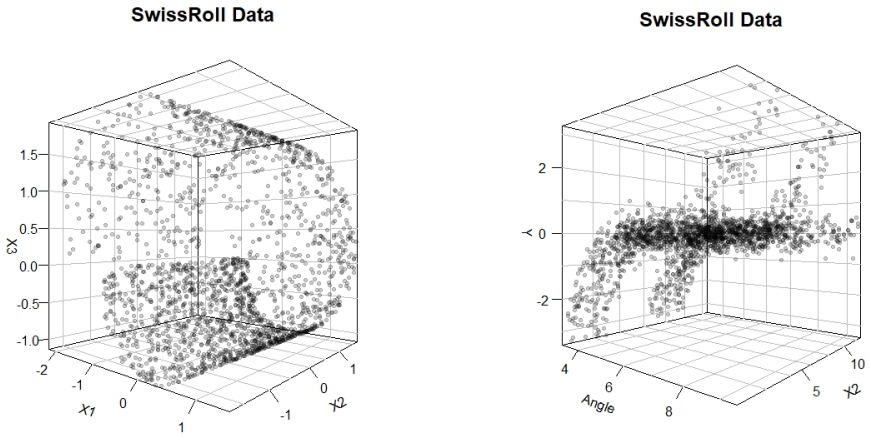


Figure 3.12: SwissRoll Regression Data

Method	Median SSE (5%, 95%)	nknots	Parameter
kNN	648.5 (607.1, 696.0)	-	neighbor=12
Ridge	1513.7 (1394.4, 1616.2)	-	$\lambda = 2.0$
Lasso	1191.4 (1106.7, 1260.7)	-	$\lambda = 0.032$
SpecSeries	1166.5 (1081.4, 1238.8)	-	bandwidth = 2.0
Fast-KRR	1503.5 (1403.2, 1592.9)	-	$\sigma = 0.1$
S-Kernel	458.2 (409.0, 511.8)	30	bandwidth = 2 r_{hns}
S-kNN	474.7 (417.6, 553.4)	30	neighbor = 18
S-Lspline	569.8 (519.5, 645.8)	60	$\lambda = 0$

Table 3.3: Regression results on the Swiss Roll $d = 1000$ data. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5 percentile and 95 percentile of the SSEs from the given parameter settings are reported in brackets.

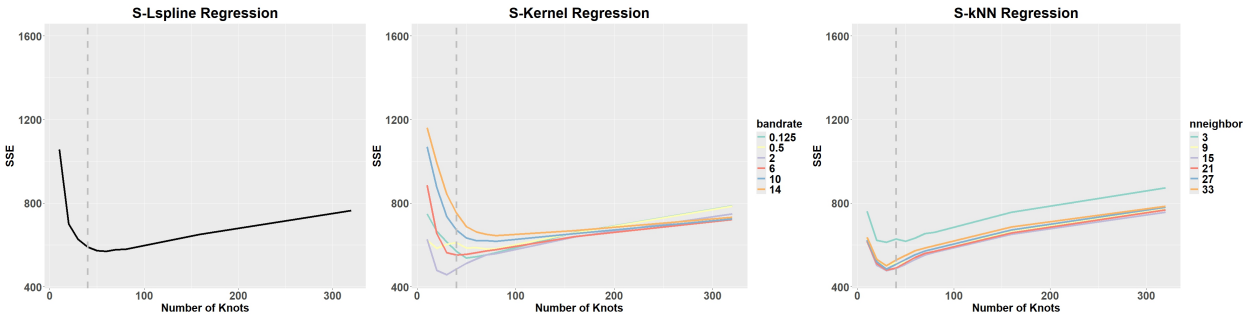


Figure 3.13: SwissRoll $d = 1000$ data regression results with varying number of knots. The median SSE across the 100 simulated datasets with each given parameter setting is plotted.

By plotting the median SSE under skeletons with a varying number of knots in Figure 3.13, we observed that the best performance for all the skeleton-based methods is achieved with the number of knots larger than $[\sqrt{1600}] = 40$ knots. Given the intrinsic structure of the Swiss Roll input space is a 2D plane, having more knots on the plane can give a better representation of the data structure and, therefore, lead to better prediction accuracy. We conjecture that the optimal number of knots should depend on the intrinsic dimension of the covariates, and we plan to discuss this further in future work. However, it's recommended to use cross-validation to choose the number of knots in practice.

3.5 Real Data

In this section, we present analysis results on two real datasets. We first predict the rotation angles of an object in a sequence of images taken from different angles (Section 3.5.1). For the second example, we study the galaxy sample from the Sloan Digital Sky Survey (SDSS) to predict the spectroscopic redshift (Section 3.5.2), a measure of distance from a galaxy to Earth.

3.5.1 Cup Images Data

This dataset consists of 72 gray-scale images of size 128×128 pixels taken from the COIL-20 processed dataset (Nene et al., 1996). They are 2D projections of a 3D cup obtained by rotating the object by 72 equispaced angles on a single axis. Several examples of the images are given in Figure 3.15.

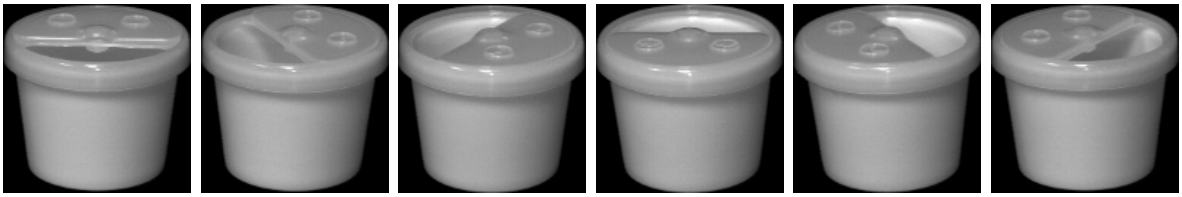


Figure 3.15: A part of the cup images from the COIL-20 processed dataset. Each image is of size 128 pixels.

Method	SSE	Parameter
kNN	1147.2	neighbor=3
Ridge	-	-
Lasso	-	-
SpecSeries	-	-
Fast-KRR	-	-
S-Kernel	1735.0	bandwidth = $2r_{hns}$
S-kNN	2068.8	neighbor = 2
S-Lspline	1073.4	-

Table 3.4: Regression results on cup images data from COIL-20. The best SSE from each method is listed with the corresponding parameters used.

The response in this dataset is the angle of rotation. However, this response has a circular nature where degree 0 is the same as degree 360. To avoid this issue, we removed the last 8 images from the sequence, only using the first 64 images. As a result, our dataset consists of 64 samples from a 1-dimensional manifold embedded in $\mathbb{R}^{16384}$ along with scalar values representing the angle of rotation. To assess the performance of each method, we use leave-one-out cross-validation, that, in each iteration, one image is taken out of the dataset and the regression methods are fitted to the remaining images to estimate the angle of the left-out image.

Similarly to the simulation studies, we use the skeleton construction method with Voronoi weights in Wei and Chen (2023) to construct the skeleton on the training set. In practice,

we found that a small number of knots can still lead to loops in the constructed skeleton structure, and, after some tuning, we fit $2[\sqrt{n}] = 16$ knots to each training set. Additionally, since the underlying manifold should be one connected structure, we do not cut the constructed skeleton structure in this experiment. Due to the high-dimensional nature of the data, Ridge regression, Lasso regressions, and the Spectral Series approach failed to run with the implementations in R. The best result from each method is listed in Table 3.4 along with the corresponding parameters.

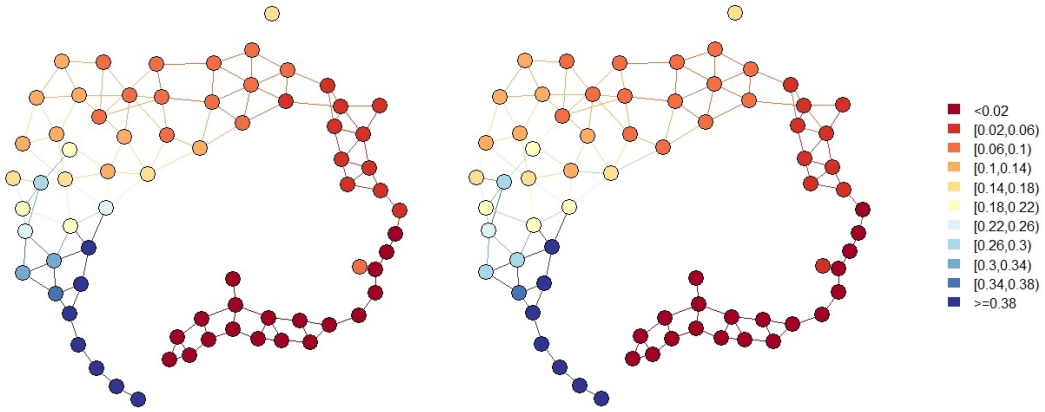
We observe that the S-Lspline method gives outstanding performance on this real data, outperforming the kNN regressor, while the other skeleton-based methods also demonstrate good performance. The lightening conditions of this series of images do not vary much by the rotation angle, which poses challenges to the similarity calculations based on the Euclidean distance and hence limits the performance of the classical kNN method. Note that the S-Kernel and S-kNN methods depend on the skeleton-based distances between data points while the S-Lspline methods do not, and hence the difference between such methods may imply that, although the skeleton graph can capture the data structure which leads to the good performance of the S-Lspline method, the skeleton-based distances can give inaccurate relations between data points compared to the true underlying data structure. However, the skeleton graph still provides information about the data structure as the S-Lspline method has good performance with the simple piecewise linear model assumption on the skeleton.

3.5.2 SDSS Data

In this section, we applied the skeleton regression to a galaxy sample of size 5000, taken from a random subsample of the Sloan Digital Sky Survey (SDSS), data release 12 (York et al., 2000; Alam et al., 2015). We repeat the random data subsampling for 100 times to get 100 different datasets. One dataset consists of 5 covariates measuring apparent magnitudes of galaxies from images taken using 5 photometric filters. These covariates can be understood as the color of a galaxy and are inexpensive to obtain. The response variable is the spectroscopic redshift, which is a very costly but accurate measurement of the distance to the Earth. It is known that the 5 photometric color measurements are correlated with the spectroscopic redshift. So the goal is to use the photometric information to predict the redshift; this is known as the clustering redshift problem in Astronomy literature (Morrison et al., 2017; Rahman et al., 2015).

We construct the skeleton with the same method in the simulation studies. The resulting skeleton graph is shown in Figure 3.17. In the left panel of Figure 3.17, we color the knots by their predicted redshift values according to the S-Lspline method and color the edges by the average predicted values of the two connected knots. For comparison, we color the knots and edges using the true values in the right panel of Figure 3.17. The predictions given by S-Lspline are very close to the true values.

For completeness, we perform the same analysis as in Section 3.4 by comparing the 5-fold cross-validation SSEs of different regression methods on this dataset and include results in Table 3.5. The classical kNN shows superior performance on this dataset, which can imply



(a) S-Lspline (b) True Value

Figure 3.17: SDSS Skeleton Colored by values predicted by S-Lspline (left) and by true values (right).

Method	SSE	nknots	Parameter
kNN	58.6 (46.7, 79.1)		neighbor=6
Ridge	868.4 (771.8, 984.5)		$\lambda = 0.001$
Lasso	861.7 (750.3, 993.8)		$\lambda = 0.0013$
SpecSeries	73.0 (54.1, 114.0)		bandwidth = 5
Fast-KRR	312.3 (242.5, 396.9)		$\sigma = 0.1$
S-Kernel	78.6 (71.5, 92.4)	126	bandwidth = $4r_{hns}$
S-kNN	83.1 (73.9, 98.7)	126	neighbor = 9
S-Lspline	75.9 (69.0, 90.4)	126	$\lambda = 0$

Table 3.5: Regression results on SDSS data. The best SSE from each method is listed with the corresponding parameters used. The 5 percentile and 95 percentile of the SSEs from the 100 runs are reported in brackets.

that the kNN method adapts nicely to the complex structure of the data. Notably, the Spectral Series regression shows good performance in this low-dimensional setting. However, note that the Spectral Series regression has a large variation in its performance ranging over the different subsampled datasets, with SSEs a 5 percentile of 54.1 to 95 percentile of 114.0. Overall, kNN and SpecSeries methods work well in this data and both methods can adapt to the underlying manifold, while the skeleton-based regression methods also show comparable results. The Fast-KRR approach demonstrates performance better than the usual Ridge and Lasso regression, demonstrating the effectiveness of kernel tricks in this setting. While skeleton approaches do not provide the best prediction accuracy, the skeleton structure obtained in Figure 3.17 shows a clear one-dimensional structure in the underlying covariate distribution and an approximate monotone trend in the response. Thus, even if our method does not provide the best prediction accuracy, the skeleton itself can be used as a tool to investigate the structure of the covariate distribution, which can be valuable for practitioners.

3.6 Conclusion

In this work, we introduce the skeleton regression framework to handle regression problems with manifold-structured inputs. We generalize the nonparametric regression techniques such as kernel smoothing and splines onto graphs. Our methods provide accurate and reliable prediction performance and are capable of recovering the underlying manifold structure of the data. Both theoretical and empirical analyses are provided to illustrate the effectiveness

of the skeleton regression procedures.

In what follows, we describe some possible future directions:

- **Generalizing skeleton graphs to a simplicial complex.** From a geometric perspective, the skeleton graph constructed in this work only focuses on 0-simplices (points) and 1-simplices (line segments). Additional geometric information can be encoded using higher-dimensional simplices. Recent research in deep learning has explored the use of simplicial complices for tasks such as clustering and segmentation (Bronstein et al., 2017; Bodnar et al., 2021). Higher-dimensional simplicies offer a finer approximation to the covariate distribution but have a higher computational cost and a more complex model. Thus, it is unclear if using a higher-dimensional simplex will lead to better prediction accuracy. We will explore the possibility of extending skeleton graphs to the skeleton complex in the future.
- **Connection to Topological Data Analysis.** Topological data analysis (TDA) has long studied the extraction of topological invariants such as homology from discrete data measures, with two main approaches: One way use localized kernels to construct weighted graphs and derive discrete operators that converges to the Laplace-Beltrami operator of the manifold; Another approach is the persistent homology that produces a series of unweighted graphs to reconstruct topology at different scales. Berry and Sauer (2019) proposed to use a multi-scale metric called Continuous k-Nearest Neighbors (CkNN) to construct a single unweighted graph, and they proved that it is a consistent representation of the geometry of the underlying manifold in the limit of large data

and captures topological features at multiple scales simultaneously. We may adopt such results for skeleton graph construction and potentially provide further theoretical properties of the resulting graph.

- **Nonparametric smoothers on graphs.** Some other nonparametric smoothing can be performed on graphs. For example, Wang et al. (2016) generalized the concept of trend filtering (Kim et al., 2009; Tibshirani, 2014) to graphs and compared it to Laplacian smoothing and Wavelet smoothing. In contrast to our work, these regression estimators for graphs are applied to data where both the inputs and responses are located on the vertices of a given graph. As a result, these graph smoothers, which include different regularizations, can only fit values on the vertices and do not model the regression function on the edges.

  As Wang et al. (2016) mentioned the possibility of linear interpolation with trend filtering, it is possible to generalize these methods to the skeleton by constructing responses on the knots in the skeleton graph as the mean values of the corresponding Voronoi cell, and then graph smoothers can be applied. Some interpolation methods can again be used to predict the responses on the edge, and this can lead to another skeleton-based regression estimator.

  It can also be interesting to apply statistical machine learning methods onto the skeleton graphs. Particularly, we note that the skeleton graph as constructed is a general metric space, and Steinwart and Scovel (2012) generalizes Mercer's Representation Theorem to some general domains. This can serve as a starting point for some

statistical learning theory on the skeleton graph.

- **Time-varying covariates and responses.** A possible avenue for future research is to extend the skeleton regression framework to handle time-varying covariates and responses. Specifically, covariates collected at different times could be used together to construct knots in a skeleton. The edges in the skeleton can change dynamically according to the covariate distribution at different times, providing insight into how the covariate distributions have evolved. Additionally, representing the regression function on the skeleton would make it simple to visualize how the function changes over time.
- **Streaming data and online skeleton update.** As streaming data becomes increasingly common, a potential area of future research is to investigate methods for updating the skeleton structure and its regression function in a real-time or online fashion. Reconstructing the entire skeleton can be computationally costly, but local updates to edges and knots can be more efficient. We plan to explore ways to develop a simple yet reliable method for updating the skeleton in the future.

Chapter 4

Network Measurement Error and Non-robustness of Diffusion Estimates

4.1 Introduction

Researchers and policymakers studying the spread of ideas, technology, or disease often estimate models of diffusion using network data on how individuals interact. Examples include (i) quantifying the extent of illness or technology take-up; (ii) summarizing diffusion dynamics (e.g., the reproduction number $\mathcal{R}_0$ of a disease); (iii) targeting interventions (e.g., where to seed new information to maximize spread, where to lockdown to prevent spread); (iv) and estimating counterfactuals (e.g., in estimates of peer effects, as we show in an empirical example). See Anderson and May (1991), Jackson (2009), Jackson and Yariv (2011), and Sadler (2023) and references within for all three classes of topics (as well as an account of how such models are used in the case of strategic behavior).

In this paper, we focus on a setting where the econometrician has an imperfect measurement of either the initial seeding or the interaction network, and wants to estimate models of diffusion or generate forecasts. Importantly, we let this measurement error be very small: when we study errors in identifying the initial seed or links, we hold the other as known (which is generous to the econometrician) and assume that the error being studied is small enough to vanish in the limit. We focus on an intermediate time horizon, where the econometrician is not focused on predictions on "day 2" of a diffusion, nor are they focused on "long run" predictions since by that point the diffusion would have saturated. In the long run, forecasts and predictions of where the diffusion goes are much less consequential for policy. Our preferred environment captures the setting where an econometrician is equipped with the richest possible data on individuals and interactions and is interested in making predictions about the diffusion process in the "medium run" which is critical for policy.

We show that this tiny mismeasurement significantly affects the predictions from the estimated diffusion model. We show four key results: (i) predictions of where diffusion goes is considerably sensitive to *local* uncertainty of the initial seeding; (ii) predictions of diffusion counts will be grossly under-estimated with even vanishingly small measurement error of the network; (iii) while aggregated estimated quantities such as the basic reproductive number $\mathcal{R}_0$ can be estimated correctly despite the measurement error, it provides limited information for more disaggregated targets; (iv) because the measurement error is so small, most data augmentation (either estimating the measurement error or conducting additional data collection) will be ineffectual. These seemingly pessimistic results, however, provide

clarity on possible positive strategies to be explored: the extremely high returns to utilize widespread strategies early in a diffusion.

The key insight in our results is that diffusions are extremely susceptible to measurement error because missed links create opportunities for the process to propagate out-of-view and have knock-on effects that eventually overwhelm the econometricians' estimated predictions. To give intuition, consider a common network formation where connections occur with a higher probability for people with some observable commonality (e.g., geography, school, work) or latent factors (e.g., Hoff et al. (2002)). With a perfectly measured graph, when a diffusion process is seeded, we can draw a ball around the (known) initial seed that will exhaustively enumerate the number of nodes possibly activated[1] by the process. This ball will expand over time, with the ball's radius defined by the distance from the initial seed.

Assume there is a small set of idiosyncratic links in this network. Note that even if the network is fully known, initial seeds that are nearby can effectively have the balls drawn around each expand somewhat differently—there will be overlap, but there will be non-trivial divergence. So, small perturbations in the initial seed can lead to misleading conclusions as to where the disease goes. Now, imagine the seed is known, but a small set of idiosyncratic links are missed. If any of these missed links reach further than the ball drawn around the seed in the base graph, the diffusion process can escape past the econometricians' determined set of possibly impacted nodes. Since the link is outside of the ball, it spreads even more quickly because it has the largest possible set of unexposed units to diffuse to. This jump need not

[1]Since the model applies to diseases, technology adoption, social learning, and other diffusion settings, we use the term *activated* to nest the application-specific terms such as "infected," "informed," or "adopted" (Jackson and Yariv, 2007).

be far – it simply needs to be a link that creates diffusion unexpected by the econometrician.[2] Our results are general: each node can link to a (vanishing) fraction of the population and this can be arbitrary in structure. This nests cases of only local mismeasurement: e.g., only missing links to "nearby" locations.

Missing links in the measurement of networks is a common concern (Wang et al., 2012; Sojourner, 2013; Chandrasekhar and Lewis, 2010; Advani and Malde, 2018; Griffith, 2022), but our paper highlights the dramatic impact of even the smallest errors when attempting to forecast diffusion. Mismeasurement can happen for several reasons. The first is practical: many analyses using empirical data (including one of our own empirical examples) do some amount of aggregation into groups with measured amounts of interaction. For example, individuals may be binned into groups location-by-age-by-occupation, and the interactions between these groups are approximated based on underlying microdata. Using this data on individuals and interactions to construct compartments and forecast diffusion processes implicitly assumes that connections occur with a much higher probability for people with some observable commonality within the bin (Acemoglu et al., 2021; Farboodi et al., 2021; Fajgelbaum et al., 2021b). These choices may match the average interaction pattern, but

[2]Like all work in this space, we are indebted to Watts and Strogatz (1998), the seminal paper on small worlds, demonstrating that small probabilities of rewiring links in lattice-like graphs can yield drastic reductions in path length and time to saturation of a simple diffusion process. Our analysis is related but distinct. First and foremost, we do not require that the missed links could go anywhere in the network. Our most general results allow for nodes to have mismeasurement to potentially only a vanishing share of nodes in the graph. In our environment, the key condition of polynomial expansion is a joint property of the graph and diffusion process and not a property of the graph alone. This distinct assumption allows for analytic analysis of the diffusion processes, while also allowing for a much wider array of graph structures (including expansive networks). Further, much of the work on small world graphs and diffusion focuses on phase transitions of the process (e.g., Newman and Watts (1999)), but we compare shifts within the same (critical) phase. And, of course, our focus is on forecasts of the extent and location of the diffusion, sensitivities to perturbation of the initial seed, and possible solutions to the identified problems.

miss underlying heterogeneity, and may also mismeasure cross-compartment connections. Second, the mismeasurement of the network may occur because the sampling process for the network is imperfect. Studies surveying individuals may focus on local connections (e.g. within a school or village), and ignore other connections. Or it may be that certain connections are not mentioned, despite mattering to the diffusion process. Third, it may be that a rich snapshot of a network does not capture the relevant links for diffusion by the time the process reaches an individual.

We proceed as follows. Section 4.4 considers problems with forecasting. Theorem 3 shows that the econometrician's estimates of the diffusion count will be of lower order of magnitude than the true counts in the intermediate run – the prediction will be dominated by the error. Second, in Theorem 2, we show that diffusion on G_n – *even when the error network is completely known* – is not stable with respect to the location of the initial activation. Section 4.5.1 demonstrates we can consistently estimate both the activation rate p_n and the basic reproductive number $\mathcal{R}_0$, despite the aforementioned problems with forecasting diffusion[3]. In Section 4.5.2, we consider two possible solutions: (i) estimating the idiosyncratic links through supplementary data collection and (ii) widespread node-level sampling (e.g., testing). In our assumed regime neither solution works. The sample size required to estimate β_n is unrealistically large, and the fraction of correctly identified locations with positive tests will be bounded below one in the short run. Section 4.6 contains an extension about the case

[3] Alimohammadi et al. (2023) makes a similar point. They study a SIR model on a network and design an estimation strategy for the parameters and the trajectory of epidemics. They consider a local estimation algorithm based on sampled network data and show that asymptotically they identify the correct proportions of nodes that will eventually be in the SIR compartments. These results are analogous to our finding that one can estimate p_n and $\mathcal{R}_0$ in a straightforward manner.

wherein the econometricians' dataset exhibits exponential expansion, and Theorem 5 shows that forecasts will still be inaccurate in this scenario.

Empirically, we first examine versions of our main theorems on simulated networks. In our Monte Carlo exercises, we generate networks that match known features of empirical data. We set the measurement error probability to be small ($\beta_n \approx 1/10n$) and find that forecasts become problematic: underestimates of the diffusion count range from 22% to 83% across the simulations. We also demonstrate extreme sensitivity to initial conditions. When we perturb the initial seed in a neighborhood comprising 1% (or 5%) of the graph, the expected overlap share of activated nodes over perturbations is only 40% (or 13%) by the time the diffusion could potentially have saturated the network.

We then turn to the analysis of real data. In our first example, we construct a real-world mobility network from California and Nevada and examine mismeasurement due to "pruning" – where links between locations are only included if a sufficient number of people move between them. We find that changing the threshold from five to six people traveling between Census tracts causes the policymaker to underestimate the extent of diffusion by nearly 56%. In addition to pruning, we induce errors by removing i.i.d. random links and find more extreme underestimation by more than 76%. As a second example, we show that similar patterns hold in a viral marketing experiment in rural India (Banerjee et al., 2019). We also document extreme sensitive dependence on the seed set: when we move only one single seed to one of its neighbors, the intersection is only 61% of the activations encompassed by both diffusions. Finally, we show how our results relate to the estimation of peer effects,

focusing on the uptake of insurance in China (Cai et al., 2015).

4.2 Model

Environment For a given set of observed nodes V_n with the number of nodes n, we model the network through a random, undirected, unweighted graph $G_n \coloneqq (V_n, L_n \cup E_n)$ where L_n consists of the "base" links and E_n collects the missing links[4]. Generally, we assume that L_n is fixed and known perfectly, and all the links in L_n are true links. Each link in E_n is constructed independently following $\text{Ber}(\beta_{ij,n})$ where these can be heterogeneous at the pair level. The links in E_n are random and not observed, and hence the randomness of the true graph G_n only comes from the random realizations of E_n. Particularly, in our model we only consider the mismeasurement caused by missingness and there are no falsely added links. With an abuse of notation, we may use L_n and E_n respectively to denote the undirected and unweighted graph with the base links and the missing links.

The diffusion process spreads over the network G_n following a standard Susceptible-Infected-Removed (SIR) process with i.i.d. passing probability p_n. Each node is activated for a single period and has the opportunity to transmit the process with i.i.d. probability p_n to each of its neighbors. After nodes are activated for a single period, they are removed and cannot be re-activated. To better represent the randomness in the diffusion process, we define $P_n(G_n)$ as a random percolation on the graph G_n, which is a directed, binary graph

[4]We focus on the case of missing links, as we believe this issue will be the primary one in practice (see Griffith (2022) for several empirical examples). In the case where the econometrician both misses some links and incorrectly assumes others exist, the problem becomes much more complex. Globally, the net rate of missing or added links seems to be the key factor; locally, forecasts could either over or underestimate the volume of diffusion.

with each directed link based on G_n activated i.i.d. with probability p_n. The diffusion process is equivalent to a deterministic process emanating from some initial seed through $P_n(G_n)$, and hence the randomness in the diffusion process is captured by the random realization of $P_n(G_n)$. Similarly, we let $P_n(L_n)$ as the percolation on the graph L_n given by restricting $P_n(G_n)$ with the edges in L_n.

We conduct asymptotic analysis, taking limits as both n, the number of nodes, and T, the number of time periods, become large. We consider a sequence of graphs $\{G_n\} = \{L_n, E_n\}$, where E_n are drawn randomly, that grows with n, and consider $T := T(n)$ where T is an increasing function in n. More details on exactly how T grows are discussed below. We generally suppress the dependence of T on n for simplicity.

We define the expected activation for a diffusion on L_n set as

$$\mathcal{E}_t = \mathbb{E}\left|\{x \in V_n \mid x \text{ ever activated by the diffusion on } L_n\}\right|.$$

To set up our first results, we impose the following conditions on the diffusion process.

Assumption 1. For some constant $q > 1$ and all discrete-time t, $\mathcal{E}_t = \Theta(t^{q+1})$ and $\mathcal{S}_t = \mathcal{E}_t - \mathcal{E}_{t-1} = \Theta(t^q)$, where $a_n \in \Theta(b_n)$ means a_n is bounded both above and below by b_n asymptotically in Bachmann-Landau notation. Furthermore, $p_n \in \left((\log n)^{-q/(2q+2)}, 1\right]$. [5]

We write this assumption over the diffusion process rather than on the graph structure of L_n to allow for more generality. We could have simply assumed that L_n itself has a polynomial expansion[6], and, together with the appropriate p_n and i.i.d. draw assumptions, Assumption 1 follows. But, we allow for more general settings. For example, Assumption

[5] We assume p_n is not changing with time in this case and generalize that in Appendix S.

[6] For an arbitrary node in the graph, the number of nodes within r geodesic distance from the reference node increases polynomially with r.

1 covers cases of L_n with non-polynomial expansion and i.i.d. draws of p_n, but with a sub-critical passing probability or short time horizons. The lower bound on p_n is to ensure that the diffusion process spreads with sufficient speed – otherwise, the diffusion may halt before the medium time horizon that we study. For the substance of the assumption, firstly this condition implies that, as the diffusion progresses, a growing number of nodes become activated in expectation.[7] Secondly, this condition governs both the structure of the graph and the diffusion process. As an example, consider a latent space network where nodes form links locally in a Euclidean space (Hoff et al., 2002). Since volumes in Euclidean space expand at a polynomial rate, this ensures that Assumption 1 will be satisfied.[8] Thirdly, note the geometric relationship between $\mathcal{E}_t$ and $\mathcal{S}_t$ — $\mathcal{E}_t$ governs the total volumetric expansion of the diffusion, while $\mathcal{S}_t$ governs the shells of the diffusion (e.g., the boundary at time t). We explore the case where $\mathcal{E}_t$ has exponential growth for completeness in Section 4.6.

We then put specific constraints on the time horizon considered. The first condition restricts the time so that the diffusion does not reach the edge of the graph.[9] The second condition ensures we are making a forecast about a time period appreciably far in the future. Let a be any positive constant satisfying $2a > 1/(q+1)$; smaller a is permitted for more expansive (larger q) graphs.

[7] The basic reproductive number, which is the expected number of cases directly generated by one case in a population where all individuals are susceptible to infection, must be greater than 1 on L_n.

[8] As another example, consider the case where the latent space is equipped with hyperbolic, rather than Euclidean, geometry (Lubold et al., 2023). While volumes in the space expand at an exponential rate, Assumption 1 may still be satisfied for some T and p_n. In the case of sufficiently small p_n, this situation corresponds to the case when the diffusion simply spreads slowly because it has a low passing probability. In the case of sufficiently small T, this situation corresponds to the diffusion not having enough time to reach a large portion of the graph.

[9] Formally, this assumption makes sure that the diffusion does not reach the edge of particular subgraphs. Our proof strategy relies on the construction of independent subgraphs to simplify computations, so we adjust the upper bound on T to compensate.

Assumption 2. T_n has for each n, $T_n \in [\underline{T}_n, \overline{T}_n]$ where the following holds: (1) $\overline{T}_n = n^{\frac{1}{q+1}}$ and (2) $\underline{T}_n = (\log n)^a$.

Our main results will hold for any $T_n \in [\underline{T}_n, \overline{T}_n]$. To get a sense of scale, consider California with a population of 39 million, and assume the parameters are set at the day level. If a geographic-type network ($q = 2$) is a good model of expansion, this has an upper bound of 11 months, and if a slightly more expansive model represents the state ($q = 3$) then the upper bound is 3 months. The lower bound can be close to 1 in either case.

We set up a general structure on the distribution of E_n. We allow a given node i to only potentially link to a fraction δ_n of the nodes through E_n, each with independent linking probability β_n. The share δ_n controls the support of potential links in E_n and will do so in an unstructured way. We take *no stand* as to which js constitute this δ_n-share for any i. The maximum value of δ_n is clearly 1, unchanging with n. In this case, both "long-range" and "short-range" (from the perspective of L_n) links are permitted. But a smaller δ_n can disallow long-range links – it may be that the entire δ_n share of nodes are in a highly localized neighborhood of i. It is useful to provide some notation to provide a lower bound on δ_n. We use the same constant a as in Assumption 2, with $2a > 1/(q+1)$. Let $\nu := a - 1/(2q+2)$ simply be the difference. The lower bound on the share of nodes that can be linked to is given by $\underline{\delta}_n = (\log n)^{-q\nu}$. If we consider California, with a population of 38.9 million, and if it were thought of as a geographic-type network ($q = 2$) or even a more expansive one ($q = 3$) it is easy to see that rates such as $\underline{\delta}_n < 0.001$ becomes permissible (as do even smaller rates). Given the unstructured nature, this allows for topologies such as only very rare, local (in L)

links being formed.

Assumption 3. For every n, i, j, $E_{ij} \sim \text{Ber}(\beta_n)$ for up to some share δ_n of the n nodes and is zero otherwise. Further,

1. $\delta_n \in (\underline{\delta}_n, 1]$
2. $\beta_n \in \left(\frac{1}{p_n T^q \delta_n n}, \frac{1}{n}\right)$.

We can examine β_n, fixing a given value of δ_n. First, note that both the upper and lower bounds for β_n go to zero as n grows large. Second, Assumptions 1 and 2 ensure that $p_n \delta_n T^q \in (1, n)$. This restriction ensures that there are some links in E_n, with probability one, as $n \to \infty$. Third, the upper bound on β_n imposes that E_n is sparse: with probability one, E_n is not a connected graph, nor will it contain a giant component as $n \to \infty$. Given these restrictions, the large forecasting errors we characterized below are not a function of a dense set of links unobserved by the econometrician. Instead, they are caused by a small (and disconnected) set of idiosyncratic links that can have an unstructured pattern. While the forecast errors would also clearly happen if the econometrician missed a dense graph or a giant component, we focus on a regime where the mismeasurement is sparse, making the results more surprising.

Econometrician's Forecasting Problems We study two policy objectives: identify where the diffusion goes and how much diffusion there is by time T. The first target is to estimate which individuals have been activated by time T for a diffusion process that starts at node i_0 with percolation $P_n(G_n)$. Let y_{jt} be an indicator which denotes if node j has ever been activated through time t, and we generally suppress the dependence of this on

G_n, $P_n(G_n)$, and i_0. The ever-activated set can be written as

$$I_{P_n(G_n)}(i_0, T) := \{j \in V_n \text{ s.t. } y_{jT} = 1 | G_n, P_n(G_n), i_0\}.$$

We then consider the functions of this set. We will generally assume that the econometrician is not an oracle: while they may have a (potentially large) amount of information, the realization of P_n is not known. Therefore, we will study the distribution (or moments thereof) induced by the random P_n. As we discuss each objective function, we will make clear what is known to the econometrician. Throughout, we will assume T, q, and L_n are known perfectly. These are optimistic assumptions that give advantages to the econometrician. Therefore, our results can be thought of as modeling policy objectives in a best-case scenario.

4.3 Sensitive Dependence on the Seed Set

We first show how the diffusion process is sensitive to the exact starting location. We consider a local perturbation of the initial activation of the diffusion. We make this comparison to show a structural lack of robustness of the model caused by mismeasurement in the initial seed: if seeds differ only slightly, this is enough to generate very different diffusion patterns. The setup of the result is motivated by a policy-relevant consideration. Given a slightly incorrect assessment of the seed (e.g., patient zero), even knowing the true network G_n, the policymaker is likely to see large differences as to both *where* the diffusion jumps and *who* is activated.

To define notation, we first fix a percolation, P_n, for the diffusion process and vary only the initial seed between i_0 and some nearby j_0. This removes the randomness from the

diffusion and holds fixed the set of possible paths that it can take as we vary the initial seed. The percolation is a useful construct because we can study the resulting activated sets, given percolation $P := P_n(G_n)$, when seeding with some i_0 versus some j_0. Recall that $I_P(i_0, T)$ and $I_P(j_0, T)$ denote the ever-activated sets by period T for the two seeds respectively.

For some node e that is activated at time T, if the diffusion process continues for t more periods, then the *catchment area* is defined as the maximal set of nodes that can be activated beginning with e, $B_e(t)$, which is the ball centered at e with radius t relative to the true graph G_n. In what follows, we will find that, given the extreme sparsity of E_n, for any two nodes e_1 and e_2 which have edges in E_n (i.e., there exist nodes e_1', e_2' that $e_1e_1',\ e_2e_2' \in E_n$), the catchment areas (over t periods of transmission) typically will not intersect: $B_{e_1'}(t) \cap B_{e_2'}(t) = \emptyset$ with probability tending to one. We call e_1' an alter of e_1 in E_n as it is linked to e_1 in E_n. Intuitively, the catchment areas of these alters in E_n, e_1', and e_2', can be thought of as analogous to geographically distinct areas (though the network is not constrained to geographic structure). Each region has potential size $\mathcal{E}_t$ in expectation and is bounded above in size by the total number of nodes in a t radius ball around the seed, where t is the number of periods post-seeding.

We define a sequence of *local neighborhoods relative to a realized graph.* Let $U_{n,i_0} = B_{i_0}(a_n)$ be a ball of radius a_n around the reference node, possibly growing, with $a_n/T_n \to 0$. Relative to the total expansion of the diffusion process over T periods, the local neighborhood about i_0 we consider is vanishing.

We make use of the fact that relative to seed i_0, there are two nodes, e_1 and e_2, which

are the closest and second closest nodes to i_0 that have a link to some respective alters in E_n. In what follows, we condition on the sequence of events $\{[P_n^T]_{j_0 e_2} > 0\}$ that there exists at least one path between j_0 and e_2 in the percolated graph. The construct helps us rule out pathologies and instead focus on cases where escapes are possible. In general, percolation problems with changes in linkages (e.g., bond percolation) are extremely complicated and not our focus (see, e.g., Smirnov and Werner (2001); Borgs et al. (2006)). So, we consider sequences under general conditions of interest here.[10] It is useful to also condition on the event $\{|I_P(i_0, T) \cap I_P(j_0, T)| > 0\}$ that i_0 and j_0 are connected in the percolation because otherwise, the problem is uninteresting since the diffusions never overlap. So, we define

$$\Gamma_n \coloneqq \{[P_n^T]_{j_0 e_2} > 0\} \cap \{|I_P(i_0, T) \cap I_P(j_0, T)| > 0\}.$$

We will use a version of the Jaccard index (Jaccard, 1901) to compare the expected set of nodes that are ever activated by both the diffusion processes starting at i_0 and starting at j_0 relative to the expected number of nodes that are activated by either initial node process. We call this discrepancy measure $\Delta_n(i_0, j_0)$— the relative expected number of nodes ever activated by only one of the epidemics to the expected number activated by both, defined as

$$\Delta_n(i_0, j_0) \coloneqq \left\{ \frac{|(I_P(i_0, T) \cap I_P(j_0, T))|}{|I_P(i_0, T) \cup I_P(j_0, T)|} \;\middle|\; \Gamma_n \right\},$$

Note that Γ_n eliminates events where the Jaccard index mechanically takes a value of zero, which makes the result that this object will be strictly less than one stronger. If $\Delta_n(i_0, j_0)$ is small for a nearby pair i_0 and j_0, then, on average with a fixed percolation, a large set of

[10] To see an example, with infill asymptotics, one can construct sequences where Γ_n occurs with probability tending to zero just by virtue of adding more independent paths in L_n at a sufficiently high rate relative to p_n.

nodes is activated through the process by only one diffusion process, and not the other.

Theorem 2. Let Assumptions 1, 2, and 3 hold. Let the stochastic sequence $\{G_n\}_n$ comprised of a fixed sequence of $\{L_n\}_n$ and random $\{E_n\}_n$ and consider i_0 be an arbitrary initial seed. Let $U_{n,i_0} = B_{i_0}(a_n)$ be a ball on G_n of radius a_n around i_0 with $a_n/T_n \to 0$. Then with probability approaching one over draws of (E_n, P_n), the following holds: There exists a sequence of time periods $\{T_n\}_n$, local neighborhoods $\{U_{n,i_0}\}_n$ vanishing relative to the overall time length that $|U_{n,i_0}|/T_n \to 0$, and a sequence of shift node sets $\{J_{n,i_0}\}$ with $J_{n,i_0} \subset U_{n,i_0}$ for each n that

1. $|J_{n,i_0}|/|U_{n,i_0}| > C$ for some positive fraction C independent of n, and
2. the number of catchment regions disjoint from $B_{i_0}(T) \cup B_{j_0}(T)$ activated under seeding with $j_0 \in J_{i_0}$ rather than i_0 is at least

$$n\beta_n p_n s_n^q > 1,$$

for growing s_n, and may be order constant or even diverge in n.

Further, for any $j_0 \in J_{i_0}$,

$$\Delta_n(i_0, j_0) > c$$

for some fraction c independent of n.

All proofs are in Appendix 4.10 unless otherwise noted.

This result shows that for a non-trivial share of nodes near i_0, if the seed were counterfactually shifted that, we get a disjoint set of locations activated and the overall overlap is potentially low. The key idea is that for a given i_0, the realized E_n may generate links to other points in the network, which then may be traversed in a given draw of P_n. When considering a

diffusion pattern starting at a nearby j_0, we must consider whether a percolation would activate a different shortcut than that beginning with i_0. We show that there will always exist some j_0 and time period for which this is true. The intuition comes from fixing the second closest "shortcut" link in E_n to i_0: before a diffusion pattern from i_0 can reach this shortcut, the diffusion from j_0 will reach this shortcut. This will induce two effects. First, a non-trivial share of activations will be different due to variation in seed. Figure 4.1 shows a heuristic construction of the set J_{i_0}. Second, there will be jumps in the number of distinct catchment regions activated in the network. The portion of the proof that tracks the number of catchment areas serves as a key lemma used in a number of results throughout the paper and we state it as Lemma 1 below. The intuition is that the number of new catchment regions activated at each time step will be closely related to the shells of diffusion on L_n.

Lemma 1. Let Assumptions 1, 2, and 3 hold. Let X_s be the number of catchment regions activated in time step s. Then, the following holds:

$$\mathbb{E}_{P_n(G_n),E_n}[X_{s_n}] \geq n\beta_n\delta_n p_n s_n^q$$

The intuition is that the number of new catchment regions activated at each time step will be closely related to how many nodes are in the shell of diffusion on L_n. Note that expectations are taken over both the diffusion process $P_n(G_n)$ and realizations of E_n. With an application of Hoeffding's inequality, Lemma 1 yields the final portion of Theorem 2.

Stepping back, we can also note that j_0 is close to i_0 in the sense that their network distance is small relative to the length of the diffusion, and yet these problems occur. Further, these alternative seeds are not isolated: the first part of the theorem shows that a non-trivial

fraction of the location neighborhood about i_0 contains such problematic alternative seeds. Our simulations quantify examples to show how extreme the problem can get in even realistic setups.

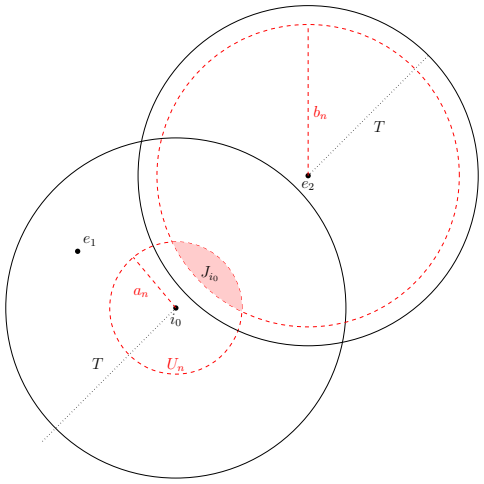


Figure 4.1: A heuristic construction of J_{i_0} using $\mathbb{R}^2$ to represent L_n. Let e_1 and e_2 be the closest and second closest nodes in L_n that also have a link in E_n. The smaller red dotted circle denotes $U_{n,i_0} := B_{i_0}(b_n)$, while the larger denotes $B_{e_2}(a_n)$. The intersection gives the set J_{i_0}.

4.4 Forecasting Difficulties

We now show how tiny measurement error leads to large forecasting errors in the diffusion process. First, we show how using the observed network L_n to make forecasts with a known seed can greatly underestimate the average extent of diffusion on the true network G_n. Second, we show how diffusion patterns can be very different even with small perturbations to the location of the initial seed, even when the true graph G_n is known without measurement error.

We assume i_0 and L_n are known perfectly. The error we study is one in which the

econometrician uses the observed L_n as a stand-in (mistakenly assuming $E_n \equiv 0$),

$$\hat{Y}_T(L_n) := \mathbb{E}_{P_n(L_n)}\left[\sum_{j=1}^{n} y_{jT} \;\middle|\; L_n, i_0\right]$$

where the expectation is taken with respect to the diffusion process $P_n(L_n)$ on L_n. We focus on this specific estimator for several reasons. First, it captures what is often done in practice both out of convenience and feasibility. Surveys about interaction and contact tracing both face survey fatigue and/or top-coding in data collection. Mobility data routinely use thresholding to define connections. They also leave out interactions that are not traceable through phones: e.g., in the developing world where households may have a single phone, movements of all members without a phone are simply dropped. In information diffusion, studies about social learning on social media may leave out person-to-person interaction, SMS off the platform, and so on. Unless one models the entire span of such missed interaction, the econometrician is really in the situation described here: effectively dropping E_n. Second, a consequence of some of the results below is that recovering the distribution of E_n to integrate over it may be practically impossible. Even in the simple case of a homogenous $\beta_{ij,n} = \beta_n$ for every i, j, the imposed rates on β_n make it difficult to identify enough links in E_n to precisely estimate β_n.

Nonetheless, a reasonable if not forgiving benchmark for $\hat{Y}_T(L_n)$ is setting the target as integrating over E_n rather than treating it as known.

$$\hat{Y}_T(G_n) := \mathbb{E}_{E_n, P_n(G_n)}\left[\sum_{j=1}^{n} y_{jT} \;\middle|\; L_n, i_0\right]$$

where the expectation is taken with respect to $P_n(G_n)$ and realizations of E_n, with known i_0 and L_n. If we compare the econometrician who ignores E_n entirely to one who uses E_n to the

full extent, they will surely do worse[11]. Comparatively, the case where the econometrician knows the distribution of E_n and integrates over it is a more fair comparison. It also demonstrates the value of understanding the error distribution (even if it may be difficult to assess in practice).

We now give the econometrician perfect knowledge of not only L_n but i_0 as well. As before, the econometrician is assumed to have perfect knowledge of T and q. However, despite these advantages, the econometrician's forecast error will swamp the forecast as $n \to \infty$.

Theorem 3. Under Assumptions 1, 2, and 3, as $n \to \infty$, $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \to 0$.

We briefly give some intuition of why the forecast error dominates the predicted error in magnitude. Small errors caused by the error network E_n recursively compound on themselves, creating massive forecast errors. When considering the diffusion process in period t, it is helpful to consider a volume around the initial node (what we call a "shell") of size t. This shell contains all the nodes that could possibly be activated by the process according to our *observed* network, L_n. As time grows, this shell grows in size, and as one might expect, the likelihood of hitting a low probability mismeasured link in E_n increases. This leads to the creation of a new shell elsewhere on the graph.[12] This initial missed jump to other locations in the network does not generate forecasting issues of any consequence. What creates the issue is that these jumps recursively explode. In totality, these new shells

[11] Furthermore, computing the expectation treating E_n is known to be NP-Complete (Shapiro and Delgado-Eckert, 2012).

[12] With i.i.d. E_n, the shell is almost guaranteed to be far, but as we show later, this is not necessary for our results – we simply need that sufficient number of new shells form that has no overlap with the existing shells.

caused by the propagating error dwarf the diffusion captured by the observed graph L_n.

The proof strategy formalizes this intuition, using a key lemma from the proof of Theorem 2. We first compute a lower bound on the number of expected new "shells" in each time period. To generate a lower bound on expected activations, we introduce a tiling of the graph and count how many tiles are activated in expectation. We then calculate this number and scale by the number of nodes activated in each tile.

In Appendix S, we show a similar version of Theorem 3 that allows the diffusion process to slow over time with decay at a polynomial rate: the resulting structure is equivalent to considering a diffusion process with a lower q. Doing so requires slightly different assumptions: we require an earlier time window and a higher rate of missing links. The intuition is that we need additional missing links to "compensate" for the slowing diffusion process to get the same result.

4.5 Estimation and Possible Solutions

We now consider several estimation procedures in our setting. First, we consider how the econometrician can estimate the underlying structural parameters like p_n successfully, despite our pathological results above. Second, we show that what seems like a natural solution to our results on forecasting – estimating β_n, our error rate, and adjusting for it – is almost impossible in reasonable samples, because the error rate is so small. Third, we consider a widespread testing regime and show that the detected number of regions that have activated nodes willunderestimate the true number of activated regions.

4.5.1 Estimating Parameters of the Process

We now show that despite the aforementioned pathologies, some core parameters of the process can be consistently estimated. We assume that the econometrician has perfect detection that they see all true activations. Using knowledge of observed L_n and $y_{j,t-1}$, the econometrician will be able to derive the exact number of expected activations for a given value of p_n and hence consistently estimate $\hat{p}$ using the observed y_{it}.[13] It then follows that the econometrician will be able to consistently estimate $\mathcal{R}_0$, the basic reproduction number [14] as $\hat{\mathcal{R}}_0 = \hat{p}d_L$ where d_L is the (observed) mean degree of L_n, whereas in actuality it is $\mathcal{R}_0(G_n) = p_n d_L + \beta_n n p_n$.

Remark 11. Assume that the policymaker has a consistent estimator $\hat{p}$ of p_n and knows d_L, that $\mathcal{R}_0$ is constant, and Assumptions 1, 2, and 3 hold. Consider the estimator $\hat{\mathcal{R}}_0 = \hat{p}d_L$. Then, we have $\frac{\hat{\mathcal{R}}_0}{\mathcal{R}_0(G_n)} \to_p 1$.

Proof. Note that $\mathcal{R}_0(G_n) = d_L p_n + \beta_n \delta_n n p_n = d_L p_n \left(1 + \frac{\beta_n \delta_n n}{d_L}\right) = \mathcal{R}_0(L_n)(1 + o(1))$, where the final equality follows by assumption. $\hat{\mathcal{R}}_0$ is a consistent estimator of $\mathcal{R}_0(L_n)$, as d_L can be computed directly and the econometrician has access to a consistent estimator of p_n. An application of the continuous mapping theorem completes the result. □

This means that while the econometrician can consistently estimate $\mathcal{R}_0$ they will still be unable to accurately forecast the location or volume of diffusion as shown in Theorems 2 and 3.

[13] We do not solve a general formulation, as solving the generic problem is known to be NP-Hard (Shapiro and Delgado-Eckert, 2012). Rather, we show an (inefficient) estimator.

[14] The number of nodes, in expectation, activated by the first seed in an activation-free equilibrium.

We give an example of one way an econometrician might estimate p_n consistently. Let $\mathcal{I}(i,t)$ be the set of neighbors of i activated at period t. Then at time T, a consistent (though inefficient) estimator of p_n will be

$$\hat{p} \coloneqq \sum_{t=1}^{T}\sum_{i=1}^{n} y_{it}\mathbb{1}\{y_{it-1}=0, |\mathcal{I}(i,t-1)|=1\} / \sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{1}\{y_{it-1}=0, |\mathcal{I}(i,t-1)|=1\}.$$

Note that by restricting attention to susceptible nodes with exactly one activated neighbor, activations occur independently with probability p_n, and hence $\hat{p}/p_n \to_p 1$. Note that this estimator makes use of perfect knowledge of L_n via the sets $\mathcal{I}(i,t)$.

4.5.2 Possible Solutions

We explore two possible solutions that a policymaker might pursue. First, they might estimate β_n, the connection rate for the E_n graph, using supplementary measurements. Second, they might use widespread testing.

Estimating β_n Given the prior results, one approach for the econometrician might be to estimate β_n, and use the estimate in order to inform forecasts. Assume the econometrician already has L_n, but is able to obtain follow-up data that they sample m_n nodes uniformly at random out of the n, and query whether or not each ij link exists in G_n. In this way, they can potentially find links in E_n to supplement the information of the known L_n. Note that a sample of size m_n nodes will deliver $\binom{m_n}{2}$ possible links.

We show that, in practical settings, this strategy will not be feasible. Specifically, our above theorems have demonstrated forecasting difficulties under extremely small levels of measurement error, and such small β_n poses challenges for estimation. Throughout, we

assume that $\delta_n = 1$. We view this as a best-case scenario for the policymaker, in the sense that it makes it as easy as possible to find missing links. In fact, there are two regimes. First, with a large, growing sample the probability that one does not find a single E_n link tends to one, even though the rate of β_n is high enough to cause all the problems previously discussed. Second, one may find some missed links with a (potentially unrealistically) larger sample, but one will not be able to develop a consistent estimator.

Proposition 6. Under Assumption 3 with $\delta_n = 1$, if:

1. $m_n = o(\sqrt{n})$, $\mathbb{P}\left(\text{No links amongst } \binom{m_n}{2} \text{ found}\right) \to 1$.
2. $m_n = O(1/\sqrt{\beta_n})$, there exists $\epsilon > 0$ and $c \in (0,1)$ such that $\Pr(|\hat{\beta}_n/\beta_n - 1| < \epsilon) < c$.

To give a sense of scale, say that n is equal to one million. Consider a case where $\beta_n = \frac{1}{n(\log n)^2}$ (which is valid for $T = \log n$ and $q = 2$, which are allowable parameters under Assumption 2). Then, with constant p_n, having $m_n = o\left(\log(n) \times \sqrt{n}\right)$ samples would still deliver nearly no information, even when this corresponds to a (perfect) survey of more than 13,800 people out of n equal to a million. For another example, if $\beta_n = \frac{\log n}{np_n n^{q/(1+q)}}$ (which is admissible under Assumption 2), then with constant p_n, under any $m_n = O\left(n \times \sqrt{\frac{1}{n^{1/(1+q)} \log n}}\right)$, an estimator for $\hat{\beta}$ is not consistent, even if there is information gained in the survey. To illustrate numerically, keep the population as one million and take $q = 4$ a perfect survey of nearly 68,000 people would still generate an inconsistent estimator. In practice, surveys of 15,000 people, let alone 70,000 people in a city, are uncommon. It is unlikely that this is an obstacle that can feasibly be overcome in most policy settings.

Widespread Testing Another potential solution is the use of widespread testing. Say that a policymaker wishes to estimate where in society activated agents reside at a given time period, in order to track regions with a disease or locations susceptible to problematic rumors or where certain technologies have been adopted. We show that the number of true regions that are activated at some time period will be grossly underestimated.

Specifically, we assume that the policymaker conducts random tests instantaneously and uniformly throughout the entire society of n nodes and detects the activations with i.i.d. probability α_n. Under this widespread testing regime, we can calculate the probability that a region is correctly identified as having been seeded by period T with the diffusion process.

Theorem 4. Let Assumptions 1, 2, and 3 hold. Consider a test with detection probability $\alpha_n \to 0$ with n, such that $T < (1/\alpha_n)^{1/(q+1)}$. Let $K_T^\star$ be the expected number of regions with an activated agent at time step T and let $\hat{K}_T$ be the expected number of regions with an observed activated agent at time step T. Assume each activated individual is observed i.i.d. with probability α_n. Then as $n \to \infty$,

$$\frac{\hat{K}_T}{K_T^\star} \leq \alpha_n T^{q+1} < 1.$$

This result demonstrates that in the short run, widespread testing will be bounded away from full effectiveness. The result holds because many regions will have few activations, making it harder to accurately detect them, but they will comprise a non-trivial fraction of activated regions. In practice, wide testing can become infeasible with a large population that asymptotically $\alpha_n \to 0$. Even with an accurate test, there may not be enough tests for the full population, or it may be hard to make testing compulsory.

4.6 Extension to the Exponential Case

We now turn to the case of exponential expansion, included for completeness. If there was significant exponential expansion throughout the network, diffusion would happen so quickly that from a policy perspective, forecasting would become moot and sensitive dependence unnecessary as the process will spread through the graph immediately. Nonetheless, we explore the implications of small mismeasurement even in this case. We make assumptions that correspond to Assumption 3 and 2, to account for the faster-moving diffusion process. As before, we assume that each node i can link to a fraction of nodes δ_n of the graph through E_n.

Assumption 4. For some constant $q > 1$ and all t, $\mathcal{E}_t = \Theta(q^t)$ and $\mathcal{S}_t = \mathcal{E}_t - \mathcal{E}_{t-1} = \Theta(q^t)$. In addition, we assume that $p_n\delta_n > \frac{1}{\log n}$.

Assumption 5. T_n has for each n, $T_n \in [\underline{\mathrm{T}}_n, \overline{T}_n]$ where the following holds: (1) $\overline{T}_n = \log(n)$ and (2) $\underline{\mathrm{T}}_n = \log(\log(n))$.

Assumption 6. For every n, i, j, $E_{ij} \sim \mathrm{Ber}(\beta_n)$ for up to some share δ_n of nodes and is zero otherwise. Further:

$$\beta_n = \Omega\left(\frac{1}{p_n\delta_n n}\right)$$

We can then note the differences in the bounds on T: we impose a smaller lower bound and a larger upper bound than for a polynomial diffusion process. The smaller lower bound on T is intuitive: because the diffusion spreads more quickly, the seeds from idiosyncratic links can cause the diffusion to explode much more quickly.

Theorem 5. Under Assumptions 4, 6 and 5, as $n \to \infty$ we have that $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \to 0$.

We make a few comparisons to our previous results. Relative to Theorem 3, we impose a stronger lower bound on β_n – in order for similar results to hold, we require a larger probability of idiosyncratic links. This change follows from the structure of the proof – the key comparison is the expansion in all of the areas "seeded" via the idiosyncratic links compared to the expansion of the original diffusion process. When the original diffusion process is faster moving, it means that more idiosyncratic links are needed to overwhelm the original diffusion.

Second, we note that if $p_n\delta_n < 1$, then the condition on β_n implies that as $n \to \infty$, E_n will contain a giant component almost surely. This condition will hold generically, in contrast to the case where the diffusion follows a polynomial process, which generally does not need E_n to contain a giant component asymptotically. While the fraction of links missed by the policymaker still goes to zero, the policymaker still misses a large amount of structure.

In both cases, we give the policymaker access to perfect local forecasting, though it plays distinct roles in each case. We get a similar result as in Theorem 3. Perfect local forecasting cannot save the policymaker from only identifying a vanishing fraction of expected activations.

Partial Converse With additional structure on L_n we prove a partial converse to Theorem 5.

Proposition 7. Assume that L_n is made up of K_n independent regions, which each fulfill Assumption 4. Furthermore, assume that Assumption 5 holds. Then if $\beta_n = O\left(\frac{1}{p_n\delta_n n}\right)$, we have that $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \to 1$.

This result is positive for the econometrician – they correctly identify a fraction of activated nodes that asymptotically goes to 1. This follows from the fact that the initial activation creates too many activations for the additional "seeded" activations through E_n to overwhelm. Note E_n will not contain a giant component asymptotically. Combined with Theorem 5, this tells us that for a more expansive diffusion, the forecasts made by the policymaker will not be "accurate" if and only if E_n contains a giant component. The giant component within E_n will have a tree-like structure meaning that the policymaker is missing a highly expansive structure. Perfect local forecasting plays a positive role – it is what allows the policymaker to be arbitrarily accurate. But given the nature of the network structure itself, a very large share of the population becomes activated very quickly.

4.7 Simulations

We present a number of simulations to illustrate our results in finite samples and explore how variation in parameters affects things quantitatively. We simulate a Susceptible-Infected-Removed process on a network with one period of activation before removal, analogous to the processes that we study theoretically. We give an overview of each part of the simulations in the relevant subsections, with full details in the Online Appendix O.

Throughout, we fix L_n, the graph observed by the policymaker and design it to mimic the sparsity and clustering structure in real data. We first generate L_n by placing nodes in a q-dimensional lattice on $[0,1]^q$. The remainder of nodes are placed uniformly at random throughout $[0,1]^q$. Nodes then link to nearby nodes, with a radius of connection chosen

to ensure both that the lattice is connected and that all randomly placed nodes will be connected to the graph. As an illustrative example, we simulate two different networks with $n = 4,000$ nodes: one with $q = 4$ and one with $q = 2$. For the SIR process on the graph, we set $\mathcal{R}_0 = 2.5$, and then compute p_n by dividing $\mathcal{R}_0$ by the mean degree in L_n. Summary statistics are shown for both graphs (along with average summary statistics for the corresponding G_n) in Appendix A.10.

We choose simulation time length T to be twice the diameter of L_n – meaning that for $q = 4$, it is chosen to be 38, while for $q = 2$ it is chosen to be 184. This value is chosen to cover both periods early on in the diffusion process, and as well as past the time period covered by our asymptotic theory.[15] Since the asymptotic theory we consider cannot speak to long-run, we simulate to the point when the diffusion extends well past the diameter of the graph, at which point we would expect the diffusion to conclude.

Forecast Errors We begin by simulating a version of Theorem 3. To do so, we simulate the error network, E_n, as an Erdos-Renyi graph with links that are i.i.d. with probability $\beta_n = \frac{1}{10n} = \frac{1}{40000}$. We simulate 2,500 iterations of the SIR process on both the fixed L_n and $G_n = L_n \cup E_n$, with E_n re-drawn in each simulation. We do so for the L_n generated with both $q = 4$ and $q = 2$. Average graph statistics for each G_n are shown in Table A.10. Note that the degree distribution stays quite similar, as the average additional degree from E_n is 0.100 for both sets of simulations. The initial seed i_0 is chosen uniformly at random and held fixed throughout the simulations. We then compute the empirical analogue of $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)}$,

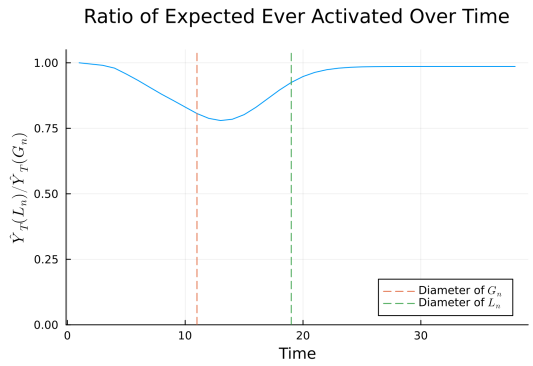
[15] Recall the time period bounds from Assumption 2 of Theorem 3.

the ratio of the expected number of ever-activated nodes under each process.

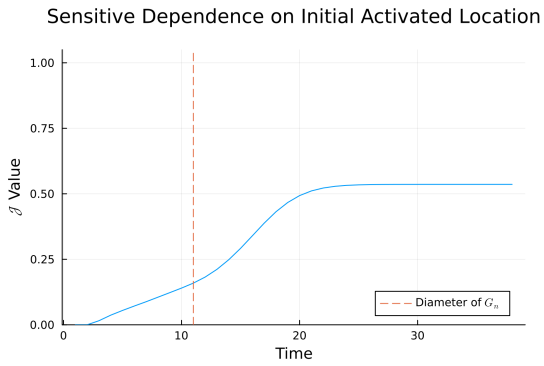
In Figures 4.2a and 4.2c, we plot the simulated values of $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)}$ over time for each graph. For $q = 4$, the minimum ratio is attained at $T = 13$ with a value of 0.780, meaning the policymaker would underestimate the extent of the diffusion by 22%. Once the diffusion on G_n reaches the diameter of the graph, the ratio increases towards a value just below one. For $q = 2$, the minimum ratio is attained at $T = 28$, taking a value of 0.169. With a lower-dimension diffusion process, the simulations are much more sensitive to additional links in E_n. In the Appendix O, we show that with $q = 2$ and $\beta_n = \frac{1}{100n}$, the minimum ratio of $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)}$ is still much smaller than the values attained with $q = 4$. The shape of the curves in Figure 4.2a and 4.2c are similar to our theoretical results, since our results focus on asymptotic results where the diffusion cannot reach the edge of the network. Hence, the ratio in our theoretical results will continue to decline. Appendix Figure A.49 shows exactly this phenomenon by separating the ratio into separate curves for $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$ – the separation between the two curves is maximized just after the diameter of G_n is reached.[16] Consequentially, the decline in the period prior to reaching the diameter of G_n lines up exactly with the results anticipated by Theorem 3.

Sensitive Dependence Next, we investigate Theorem 2 in simulation by looking at perturbations of an initial seed within local balls covering 1% to 5% of the overall number of nodes. We fix L_n and a particular instance of E_n to form G_n, and set i_0 as the center of the lattice. Then, we construct J_{i_0}, the set of possible alternate seeds, and choose a $j_0 \in J_{i_0}$

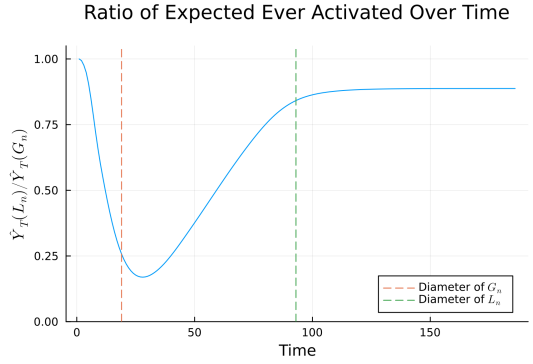
[16]Note that the ratio asymptotes with T to a value just below 1, as the additional links in G_n allow for there to be more overall activations in expectation than in L_n.



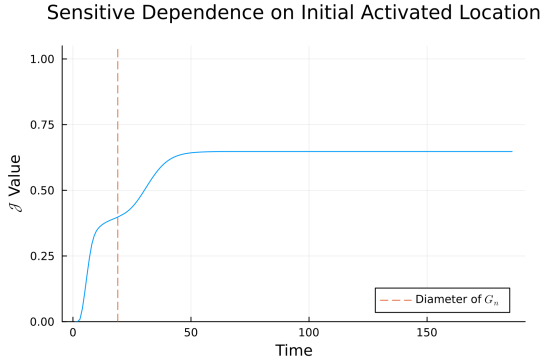
(a) $q = 4$



(b) $q = 4$



(c) $q = 2$



(d) $q = 2$

Figure 4.2: Panels 4.2a and 4.2c show simulations of Theorem 3, while Panels 4.2b and 4.2d show simulations of Theorem 2. In Panels 4.2a and 4.2c, we simulate 2,500 iterations of the diffusion process on both L_n and G_n for each value of q, re-drawing E_n for each simulation. We then track the expected number of ever-activated nodes under each simulation at each time period, and then take the ratio. We plot the diameters of both L_n and the average G_n. Panels 4.2b and 4.2d each fix a separate draw of E_n, then each choose a fixed j_0. We then simulate 2,500 diffusion processes while tracking the Jaccard index after perturbing the initial seed location. Alternate initial location j_0 is chosen nearby to i_0, in accordance with Theorem 2.

uniformly at random. To construct J_{i_0}, we first find the depth of the second closest links in E_n to i_0 – call this distance d_{e_2}. Then, nodes are included in J_{i_0} if they are at distance $d_{e_2}+1$ from i_0. Empirically, for $q=4$, $d_{e_2}=2$ meaning that the distance from i_0 to j_0 is 3. The local neighborhood around i_0, U_{i_0} (which contains all nodes at or within distance $d_{e_2}+1$) of this size makes up 5.3% of the total nodes in the graph, while J_{i_0} makes up 64.6% of the local neighborhood. For $q=2$, the distance from i_0 to j_0 is 4, while the local neighborhood of this size makes up 1.05% of the graph and the set of j_0 makes up 31.0% of the local neighborhood.

To approximate $\Delta_n(i_0, j_0)$, we fix the underlying percolation and examine the set of ever-activated nodes infected by an epidemic that begins from i_0 and j_0. We exploit the connection between percolations and the one-period SIR process, predetermining which links in the network will transmit. However, we do not condition on the event that there is *some* overlap between the diffusions (in Theorem 2, this is encoded in the object Γ_n and is assumed), and do not take expectations over E_n. We call this version of the Jaccard index $\mathcal{J}$. We generate a single draw of E_n and then hold it fixed. We simulate the process 2,500 times, and then take the average over simulations at each time period to get $\mathcal{J}(T)$. Results are shown in Figures 4.2b and 4.2d.

Figures 4.2b and 4.2d indicate that there is generally little overlap between the diffusions until the process has reached the diameter of the graph and saturated the network. Recall that when $\mathcal{J}(T)$ is close to zero, this implies that the share of nodes that would be activated by both starting conditions as a share of the total activations is small. Hence, this implies

that the activation paths are following very different portions of the network. This lack of overlap is despite the fact that i_0 and j_0 are extremely local. For $q = 4$, at $T = 5$ (the halfway point to the diameter of G_n), the value of $\mathcal{J} = 0.055$ indicates almost entirely distinct processes. For $q = 2$, at $T = 9$ (again half of the diameter of G_n), the value of $\mathcal{J} = 0.32$. These results are consistent with the theoretical results: there exist time periods early on in which the diffusions are almost entirely disjoint. Empirically, these results demonstrate that the diffusions remain disjoint for a relatively long period of time.

While it is clear that our simulations are highly sensitive to measurement error, regardless of whether $q = 2$ or $q = 4$, the changes in sensitivity are instructive. Comparing $q = 2$ to $q = 4$, the simulations demonstrate that the diffusion process is much more sensitive in terms of the extent of diffusion with lower dimension, rather than the location. This is because $q = 2$ ensures that a greater fraction of connections are "local" – therefore, there can be less local perturbation. However, i.i.d. connections lead to many more activations. Nonetheless, we note that there is still severe sensitive dependence on initial conditions with $q = 2$ – in the short run only a third of the diffusion overlaps on average.

4.8 Empirical Applications

We consider three empirical applications. The first examines the COVID-19 pandemic, which showcases our results in a large-scale setting. In addition, it demonstrates how only local linking can still cause errors in diffusion – though we show that the problems are much worse in the idiosyncratic case. The second example studies mobile phone marketing in India,

which showcases our results in a much smaller scale setting. Here, sensitive dependence on initial location has much more dramatic results – volumes of diffusion are more robust in this setting because the networks themselves are much smaller. Finally, we consider the diffusion of a weather insurance product in China. Here, we consider how errors in a diffusion model could impact statistical power when estimating peer effects.

4.8.1 Data from the COVID-19 Pandemic

Kang et al. (2020) introduces a dynamic human mobility flow data set across the United States, with data starting from January 1st, 2019. By analyzing millions of anonymous mobile phone users' movements to various places, the daily and weekly dynamic origin-to-destination population flows are computed at three geographic scales: census tract, county, and state. We study tract-to-tract flows on March 1st, 2020, at the start of the COVID-19 pandemic in the United States. Note that this date was before the WHO declared COVID-19 a pandemic and before the United States declared a national state of emergency. For the sake of computational tractability, we focus on a region in the Southwest of the United States that contains all of California and Nevada, along with a small portion of Arizona.

We use this real-world dataset to simulate disease transmission as in Section 4.7. One approach would be to construct a network with unweighted edges between two census tracts if at least one person moves between them. However, this results in an extremely dense graph. The resulting graph has a diameter of 4, a mean degree of 143.82, and a max degree of 991. The dense network will result in the epidemic spreading everywhere in a very short

time, negating the need for forecasting.[17]

Realistically, researchers may decide to "prune" the network by only including links where there is sufficient traffic between two census tracts. In this case, a missing link implies a (potentially large) flow of people between two places, rather than missing a single individual contact. Hence, we construct the observed L_n by linking tracts if the average flow between them (averaging over directions) is greater than six trips (the 93rd percentile of all flows). We then consider two ways to define the "true" base graph G_n. The first, denoted G_n^{92} links tracts if the average flow exceeds five trips (the 92nd percentile), meaning that E_n^{92} includes links of exactly 6 trips. Further discussion of the pruning procedure is given in Appendix P. The other, G_n^{β}, adds links i.i.d. with probability $\beta_n = \frac{1}{0.32n}$ corresponding exactly to the extra links missed going from the 5 trips to 6 trips, with these links now placed idiosyncratically. Properties of the resulting L_n and G_n are shown in Table A.12.

We begin by simulating Theorem 3 and calculating the share of $Y_t(L_n)/Y_t(G_n)$ for our two G_n measures. In the first, we look at $G_n^{93} = L_n$, where L_n amounts to pruning about 18 percent from the G_n^{92} graph. Here, because G_n^{92} is a (non-stochastic) function of the data, we hold it fixed and take expectations only over the path of the epidemic.[18] In the second, we generate G_n^{β} via $L_n \cup E_n$, where E_n has i.i.d. links to generate the same density as the error graph in the pruning procedure. In both cases, we choose i_0 uniformly at random and

[17]The researcher may use the dense network and assume that p_n is very small. However, with the dense network, the resulting disease process will look like an Erdos-Renyi random graph, which still follows an exponential diffusion process, rendering the forecast exercise pointless. Formally, consider the case where G_n is a complete network. Then, the resulting diffusion outcome can be modeled by dropping links in G_n with i.i.d. probability $1 - p_n$. The result will then be an Erdos-Renyi random graph generated with probability p_n, which induces exponential diffusion.

[18]In the rest of the paper we consider expectations for Theorem 3 over both the epidemic and error graph.

hold it fixed across simulated epidemics.

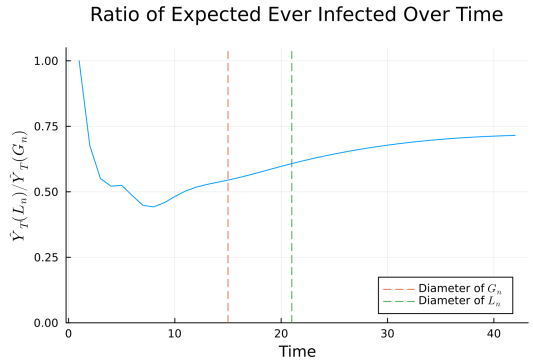
We plot $Y_t(L_n)/Y_t(G_n)$ over time in Figures 4.3a and 4.3c. For G_n^{92}, the pruned network, the minimum ratio of 0.442 is achieved at $T = 8$. We note that this ratio has the same qualitative pattern as in the simulated graph in Section 4.7 – the ratio achieves a minimum just before reaching the diameter of G_n^{92}, and then slowly increases. When compared to the previous simulations, the ratio increases much more slowly. This result comes from the larger dispersion in degrees – it takes longer for the disease to fully saturate the network, because there are more nodes with very few links. When compared to the i.i.d. errors in G_n^{β}, the minimum ratio of 0.234 is achieved at $T = 9$. One explanation for i.i.d. errors leading to additional underestimation follows from the missing mechanism. The pruning procedure induces spatially clustered errors – so for the same level of error, the spatially clustered additional links in G_n^{92} will not jump as far as G_n^{β}, leading to fewer "new" shells of infection.

Next, we simulate a version of Theorem 2. We follow a similar procedure as with Section 4.7, tracking $\mathcal{J}(T)$. We choose j_0 in a conservative fashion – after fixing a i_0 uniformly at random, we choose the set of potential j_0, J_{i_0}, to be all nodes at distance two from i_0[19]. In G_n^{92}, the local neighborhood containing all potential j_0, U_{i_0}, makes up 1.57% of the graph, while the set of J_{i_0} makes up 81.68% of the local neighborhood. In G_n^{β}, U_{i_0} contains all j_0 comprises 2.93% of the graph, and J_{i_0} makes up 93.46% of U_{i_0}.

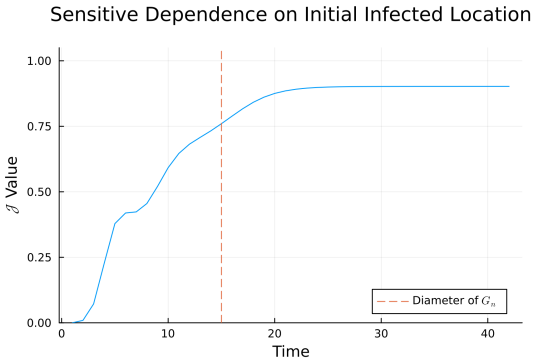
We plot $\mathcal{J}(t)$, the amount of overlap between percolations over time, in Figures 4.3b and 4.3d. These results follow the same qualitative pattern as before – $\mathcal{J}(t)$ stays close to zero

[19]We found that when choosing J_{i_0} based on the location of links in E_n, the distance from i_0 to the set of potential j_0 was typically three. Therefore, our choice of nodes at distance two is truly conservative, in the sense that we choose j_0 to be closer to i_0 than what is used in the theory.

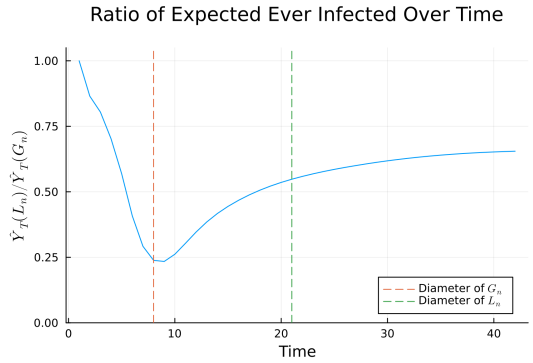
for the first few time steps while the epidemics are almost entirely distinct, but then slowly increases. For the first few time periods, this graph shows dramatic sensitive dependence on the initial starting point of the epidemic. For the pruning procedure, halfway to the diameter of G_n^{92}, $\mathcal{J} = 0.42$. For the i.i.d. procedure, halfway to the diameter of G_n^{β}, $\mathcal{J} = 0.023$.



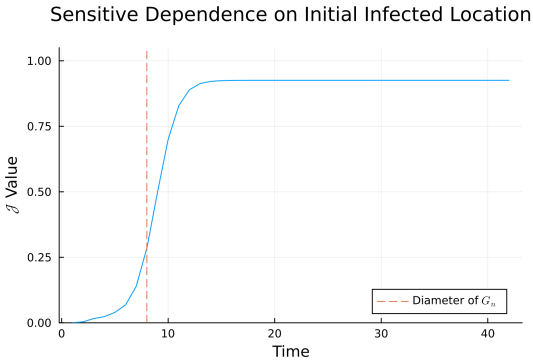
(a) Pruning



(b) Pruning



(c) I.I.D.



(d) I.I.D.

Figure 4.3: Simulated version of Theorems 3 and 2 on L_n and G_n generated from Census tract flow data in California and Nevada. Panels (A) and (C) show simulations of Theorem 3, while Panels (B) and (D) show simulations of Theorem 2.

4.8.2 Diffusion in Mobile Phone Marketing

As a second empirical exercise, we study the diffusion of high-value information in Indian villages. The goal of this exercise is to highlight how the measurement issues can crop up in

settings with much smaller networks, and how the initial seed condition plays a much larger role here. In Banerjee et al. (2019), one of this article's authors, along with collaborators, conducted a randomized controlled trial wherein randomly selected people in villages in Karnataka were given information on a program where they could receive a high-value cell phone or smaller cash prizes if they participated. The information about the program then diffused throughout the village.

We use this data to study the robustness of the diffusion process in an information setting. Details on how the graphs are constructed are in Appendix Q. In a change from the prior simulations and analysis, many of the villages have multiple initial seeds. There are on average 3.26 seeds per village and 196 nodes per village.

We first estimate the passing probability p_n for the diffusion process. Villagers could indicate they heard about the cell phone program by making a free call to the researchers. While we observe data on the sampled networks connecting households, we only observe the total number of calls received by the researchers in each village, and we do not observe whether a given household made a call. Hence, we back out the passing probability $\hat{p}_n$ using the method of simulated moments. Formally, we consider the following problem. Let $V = 69$ be the number of villages in our data (for which we have network data) and let C_v be the number of calls received in village v. We treat the number of calls as the number of ever-activated nodes. We then simulate a SIR process with passing probability p and record the number of simulated calls after T periods. Let $\hat{C}_v^s(p)$ be the simulated number of calls in

simulation s under passing probability p. Then, we choose $\hat{p}_n$ as follows:

$$\hat{p}_n = \text{argmin}_p \left(\frac{1}{V} \sum_{v=1}^{69} \left(C_v - \frac{1}{s} \sum_s C_v^s(p) \right) \right) \left(\frac{1}{V} \sum_{v=1}^{69} \left(C_v - \frac{1}{s} \sum_s C_v^s(p) \right) \right)$$

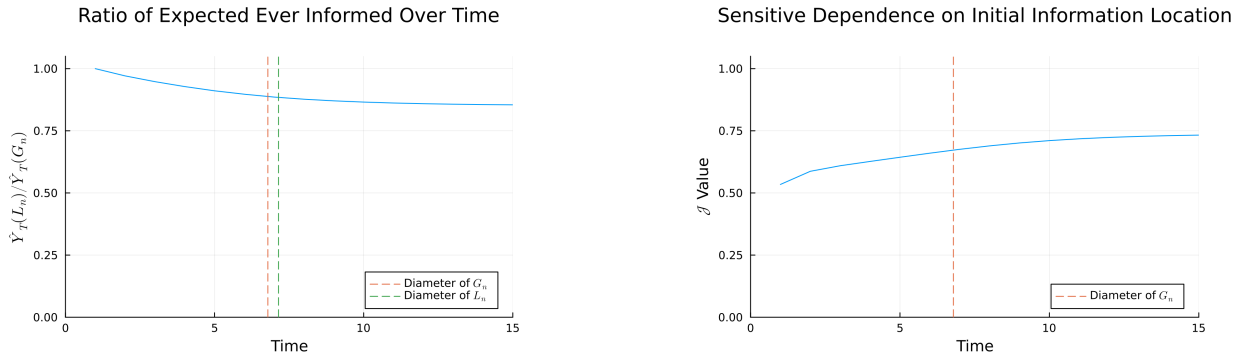
We set $T = 15$, just larger than twice the average diameter of a village graph and use 2,500 simulation iterations. We estimate a value of $\hat{p}_n = 0.13$, meaning that each household transmits the information with roughly one in six chance. We then use this estimated $\hat{p}_n$ to conduct simulations.

Next, we consider the error structure E_n on our observed network L_n. Since our data has many separate villages, we consider a slightly more complex structure for E_n. Let n_v be the number of households in village v. Then, we form E_n by taking the union over draws of Erdos-Renyi random graphs in each village, where $\beta_n^v = \frac{1}{2n_v}$ changes in each village to keep measurement error proportional to village size. We choose a proportionally larger value of β_n because there are multiple seeds – because the graph becomes saturated much more quickly, measurement error has less time to become a problem.

We first simulate a version of Theorem 3. We simulate 2,500 diffusion processes across each village, adding up the total number of households who ever get the information and averaging across simulations. We run this both on L_n, the set of village graphs, and G_n constructed as above (with a new draw of G_n in each simulation iteration). As shown in Figure 4.4a, the ratio $\hat{Y}_T(L_n)/\hat{Y}_T(G_n)$ monotonically decreases over time, taking value 0.854 at $T = 15$. Despite the village-level networks being relatively small, in aggregate, the econometrician still underestimates the extent of diffusion by nearly 15 percentage points.

To simulate a version of Theorem 2, we choose a modified seed set for each village. Recall

that most villages have multiple seeds. Here, we perturb the seed set in each village in a conservative manner. Say that a seed set is comprised of $\{i_0, j_0, k_0\}$ in some village. We choose one element of the seed set at random, say k_0, and then replace k_0 in the seed set with a neighbor chosen uniformly at random. This corresponds to a local neighborhood of 3.5% of the entire network on average. Despite the conservative perturbation, we still find similar results (Figure 4.4b). As before, we track $\mathcal{J}(t)$, the Jaccard index for the aggregate patterns of diffusion across all villages over time. While the value of $\mathcal{J}(t)$ does not start at 0 (as in the prior simulations), given the multiple seeds and that we conservatively only perturb one, it remains below 0.75, indicating that despite the conservative perturbation, there is still not complete overlap in the perturbed diffusion processes. Halfway to the diameter of G_n, the average value of $\mathcal{J} = 0.61$ indicates a lack of overlap.



(a) (b)

Figure 4.4: Simulations of Theorems 3 and 2 on village networks from Karnataka, India. Panel (A) shows a version of Theorem 3. We take 2,500 diffusion simulations on L_n and G_n, where G_n is constructed at the village level with $\beta_n = \frac{1}{2n_v}$. n_v is the number of households in the village. Panel (B) shows a version of Theorem 2. We perturb one seed uniformly at random by a single set in each village. Then, we simulate 2,500 diffusion processes on a fixed draw of G_n, computing the average Jaccard index of the process.

4.8.3 Treatment Effects with Spillovers in Networks

As a third empirical exercise, we study the uptake of insurance in rural China. The goal of this exercise is to illustrate how the problems we identify in diffusion could affect conclusions from an estimated model of peer effects. If nodes are seeded with information, then the take-up behavior of a product may be a function of "exposure to information" through the diffusion process. A typical peer effects regression would consider the outcome regressed on this exposure to treatment as defined through a diffusion; our analysis suggests that results could be biased and estimators could lose considerable power.

In Cai et al. (2015), they give farmers information about a weather insurance product, a product with low adoption rates that is highly valuable. Intensive information sessions were randomly given to some farmers. The authors then measure the take-up by other people in the same village, who were not part of the first set of information sessions. We consider a measure of exposure to treatment based on a model of information flows.

We first take the data from Cai et al. (2015) and build the village networks.[20] We convert the directed networks from the paper to undirected networks, where household i is linked to household j in our construction of the data either if i reports j as a link, j reports i as a link, or both.[21] The resulting graph is denoted $G_{n,v}$ for village v. Graph statistics for the villages are shown in Table A.15.

[20] In their data collection, the authors "top-code" the number of links each household has, by only recording five outgoing links. This possibly generates measurement error as well, since it creates an artificial upper bound for all high-degree nodes, but we ignore it for our illustrative analysis (as do they in their empirical analysis).

[21] Studying an OR network may be more robust in capturing exposures due to measurement error (Banerjee et al., 2013).

We consider an exposure measure based on a model of information flows. For a generic graph, let A be the corresponding adjacency matrix. Let s be a vector of indicators, with an entry equal to one if the household attended an information session. For a given p_n and T, we define the vector of "diffusion exposure" as,

$$DE^A = \left(\sum_{t=1}^{T} (p_n A)^t \right) s,$$

which calculates the expected number of times that each individual hears information through repeated passing over T periods (Banerjee et al., 2019). We imagine that the take-up of insurance in Cai et al. (2015) increases in such exposure to treatment: hearing more about the product through conversation makes one more likely to take up.[22] Note that this exposure measure, based on how often a person hears about the product, is slightly different than a typical SIR model. It considers the eventual outcome as depending on the total number of times person i hears about the topic through T periods, rather than a once-and-for-all decision the first time someone hears about the product. This model is perhaps a more realistic description of the take-up of an insurance product. Nonetheless, the mechanics of error we outline in the paper have analogs for this kind of model.

We then simulate an experiment. We treat the data from Cai et al. (2015) as the true network G_n. We then regress insurance take-up ($y_{i,v}$) on the exposure measure ($DE^G_{i,v}$), a set of household controls ($X_{i,v}$), and village fixed effects (μ_v),

$$y_{i,v} = \alpha + \gamma DE^G_{i,v} + X'_{i'v}\delta + \mu_v + \epsilon_{i,v},$$

[22] Following Banerjee et al. (2019), we compute this measure within each village, setting T equal to the diameter of the village network. We set p_n to be equal to one divided by the maximum eigenvalue of the village adjacency matrix. This is the critical value of p_n such that for p_n less than this value, entries of $(p_n A)^t$ tend to zero as $t \to \infty$, and some entries diverge if p_n is larger.

where i indexes household and v indexes village. To do so, we subset the data to only households who did not receive the initial informational intervention. We standardize the exposure measure to have mean zero and standard deviation one for the sake of interpretability. Results are shown in Table 4.1. A one standard deviation increase in diffusion exposure increases insurance uptake by 2.9 percentage points (s.e. 1.2 percentage points, $p = 0.02$), relative to a mean of 45.9%, in a linear probability model.[23]

Table 4.1: regression of diffusion exposure on insurance uptake

	Insurance Uptake
Diffusion Exposure	0.029
	(0.012)
Household Controls	Yes
Village FE	Yes
Num Obs.	2676
Uptake Mean	0.459

A regression of diffusion exposure on insurance uptake, with diffusion exposure computed from the networks collected in Cai et al. (2015). Standard errors are clustered at the village level.

We then drop links in G_n with i.i.d. probability β_n and construct L_n. That is, we imagine that there is a small measurement error in our survey process (or network construction process) and for this exercise we allow the error to be fully i.i.d. Errors may be correlated with factors such as geography, place of work, etc., which may be emphasized or used in constructing network data. Our simulation corresponds to what the researcher would have observed had information flowed over G_n, but they instead measured L_n.

For each village v, we drop links with probability $\beta_{v,n}$, operationalized by intersecting

[23]This estimated value is almost exactly half of the value reported by Cai et al. (2015) of 5.8 percentage points. Given that we use a different specification, the difference is not surprising, but it is reassuring that the results are of a similar order of magnitude.

the corresponding village graph with an Erdos-Renyi random graph with links that form with probability $1 - \beta_n$. We vary the value of $\beta_{v,n} = \frac{1}{k\bar{d}_v}$, where $\bar{d}_v$ is the village average degree and k is a specified constant.[24] We vary k from 5 to 15 or $\beta_{v,n}$ ranging from 0.037 to 0.0123 and recompute the diffusion exposure ($DE^L_{i,v}$), re-estimate the regression, and record the point estimate and p-values. We repeat this 2,500 times for each value of k. Let $\hat{\gamma}(G_n)$ and $\hat{\gamma}(L_n)$ be the coefficients of interest from the two regressions.

Figure 4.5 plots the joint distribution of the bias percentage—the percentage difference between $\hat{\gamma}(L_n)$ and $\hat{\gamma}(G_n)$—and the rejection level (one-to-one with the p-value) of the null of the coefficient $\hat{\gamma}(L_n)$ being equal to zero. While on average the bias is small, for any given draw, we see large dispersion in the difference between $\gamma(G_n)$ and $\gamma(L_n)$ even when a very small fraction of links are dropped. This is striking. Notice that in the real world, the econometrician observes only a single draw—one instance of this phenomenon. The result shows that enormous biases are likely in *any single draw*. Here, even with the smallest $\beta = 0.012$, we find the bias still has a large standard deviation of nearly 8 percentage points. With $\beta = 0.037$, biases upwards of 20% in magnitude are common.

We also see a range of p-values: as we decrease β, we would expect to see the p-values converge to the true value. Specifically, with no noise we know $p = 0.02$ and so for very small β we might imagine that we reject the null of no peer effect at the 95% level ($0.02 < 0.05$). However, with $\beta = 0.037$, we fail to reject (at the 95% level) the null of no peer effects over 15% of the time. And in the even more extreme case of $\beta = 0.012$, we still fail to reject the

[24] We scale β_n by the mean degree, rather than the number of nodes, for the following reason. In order to drop a link, two things must occur: the link must exist in the first place, and that indicator must be equal to 0. In order to ensure we actually β_n percent of links, we must scale by degree – because the graphs are sparse, if we scale by n_v, we drop fewer links than intended.

null of no peer effects 4.5% of the time. This means that even though with no measurement error we have $p = 0.02$, with a very small error anywhere between roughly 5% to 15% of the time we may be unable to reject a null at the 95% level.

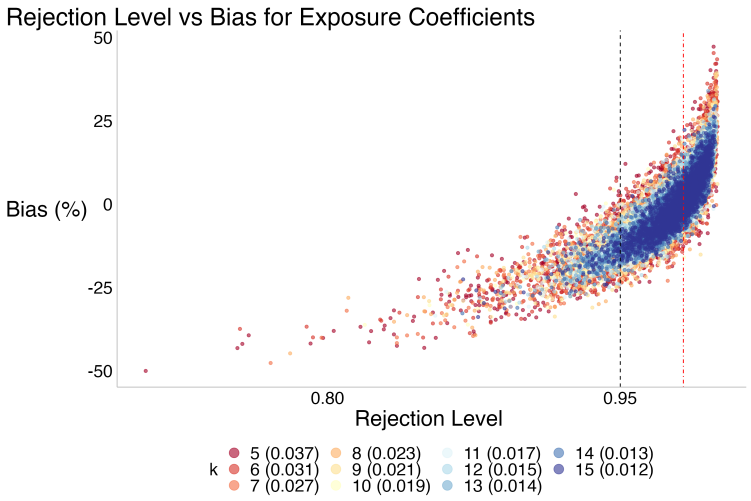


Figure 4.5: The joint distribution of the difference in $\hat{\gamma}(L_n)$ and $\hat{\gamma}(G_n)$ (in percentage terms) and the level at which we can reject the null that $\hat{\gamma}(L_n) = 0$ for different values of k. As k increases, $\beta_{v,n}$ decreases. In parenthesis, we include the average value of the corresponding β_n across villages. The red, dashed, vertical line denotes the level at which we can reject $\hat{\gamma}(G_n) = 0$. The black dotted line shows rejection at the 95 percent level.

4.9 Discussion

We have studied the lack of robustness to extremely small quantities of mismeasurement in SIR diffusion models on networks. Such models are widely used to conceptualize epidemics, information flow, and technology adoption, among other applications. For the bulk of the paper we analyze what we call polynomial diffusion over these time horizons, capturing the idea that if it were globally exponential then the diffusion would blanket the society almost immediately. These reflect real-world contagion processes where geography, homophily,

transport infrastructure, and community interactions shape the diffusion.

We have seen that a number of quantities of interest to policymakers, such as diffusion forecasts, estimates of where the diffusion has occurred in the network, and the efficacy of further data collection or widespread testing may all be problematic in the face of extremely small measurement errors in the network. In many cases, network data collection is known to be imperfect, but the econometrician may have knowledge of the structure of errors. In the case where the error has a known structure, the econometrician can correct for sampling error by integrating over the error (Chandrasekhar, 2016). However, this approach will only work in the case of exactly correct specification of the error model; any departure could potentially generate missing links, meaning our results become relevant. In the case where one knew β_n, one could use this strategy to integrate over E_n. However, given Theorem 2, the set of possibly activated nodes in this exercise could be geographically disparate.

In fact, we have shown that even if the missed links constitute not only a vanishing share of the overall links at a very rapid rate but also are only concentrated locally to any node in question the problems persist. This means that the problems are not consequences of long-range shortcuts and transitioning polynomial-like diffusion to exponential-like diffusion as in the small worlds literature. Rather, the point is that even small infrequent errors that are entirely localized wind up aggregating throughout the SIR process, thereby generating all of the aforementioned problems.

Our work demonstrates the general care needed in identifying the limits of what models can reasonably predict to inform policy. Tools must be used for exactly what they are

developed. Aggregate concepts geared towards retrospective calculations may be good for just that purpose– certain aggregates, e.g., $\mathcal{R}_0$, may better be used as descriptive rather than prescriptive tools.

This raises practical concerns for any normative work that builds on the scaffolding of such models. Almost certainly the failure of robustness would propagate to welfare calculations, which often rely on the extent of diffusion or the locations (or composition or compartments) of diffusion, if not both (Acemoglu et al., 2021; Fajgelbaum et al., 2021b). It is possible, though requires future work, that the susceptibility to small measurement error presents an argument for policymakers to respond earlier and much more aggressively. Barnett et al. (2023) make the point that in an uncertain world, policymakers may want to pursue more aggressive containment policies to guard against worst-case scenarios. The full decision theory exercise is beyond the scope of this paper, but it should be clear that this is the thrust of the statistical force given the massive uncertainty we document.

This paper is specific to SIR models on graphs, but the phenomenon need not be. In fact, the same sort of perturbation robustness failure may impact general models of treatment effects with spillovers (e.g., Aronow and Samii (2017) and Athey et al. (2018)). The final empirical example that we presented, using the insurance take-up data from Cai et al. (2015), suggests this is exactly the case. An examination of perturbation robustness failure in general models of treatment effects with spillovers is likely worth studying in its own right which we leave to future work.

4.10 Proofs

Proof of Lemma 1. We can start by partitioning L_n into K disjoint "tiles", which generates strictly fewer activations than if the tiles were still connected. The tiling is a counting device – instead of counting overall activations, we count the number of tiles that are activated and then scale those values by the number of periods for which the diffusion spreads. Each tile is composed of a subset of L_n that is disjoint from every other tile.

Let $\tilde{L}_n$ be L_n divided into K evenly sized tiles – note that K will depend on both n and T, along with the other model primitives. We suppress this dependence for the sake of compact notation. Note that $\tilde{L}_n$ is not connected, by definition. We define $\mathcal{X}_T := \mathbb{E}_{P_n(G_n),E_n}[X_t]$, the expected number of tiles that are activated in time step T. We impose the following condition in the construction of the tiling for some constant $C \in [0, 1)$: $C \leq \sum_{t=1}^{T-1} \mathcal{X}_t/K$ for all T. This ensures that there are inactive tiles for all T, such that we do not have saturation of the network by the diffusion. We can always construct a tiling where this is the case – by subdividing L_n into balls of radius T and growing n sufficiently quickly relative to T this will be possible. This restriction on the tiling is not entirely without loss. Instead of imposing that the diffusion does not reach the edge of L_n, we need to impose a bound so that it does not reach the edge of any of the tiles in $\tilde{L}_n$ – as shown in the proof, this is implied by Assumption 2.

For the sake of tractable computations, we construct a lower bound by only tracking diffusion spread in each tile that is the result of the first seed in each tile. For this simplified

computation, we can compute, for $T \geq 1$:

$$\begin{aligned}
\mathcal{X}_T &= \underbrace{\beta_n \delta_n p_n}_{\text{Diffusion Jumps}} \times \underbrace{\mathcal{K}_T}_{\text{Nodes in Tiles to Jump To}} \times \underbrace{\sum_{t=0}^{T-1} \mathcal{X}_t \mathcal{S}_{T-t}}_{\text{Weight by past spread}} \\
&= \beta_n \delta_n p_n \left(n - \frac{n}{K} \sum_{t=1}^{T-1} \mathcal{X}_t \right) \sum_{t=0}^{T-1} \mathcal{X}_t \mathcal{S}_{T-t} \\
&= \beta_n p_n n \left(1 - \frac{1}{K} \sum_{t=1}^{T-1} \mathcal{X}_t \right) \sum_{t=0}^{T-1} \mathcal{X}_t \mathcal{S}_{T-t} \\
&\approx \beta_n \delta_n p_n n \sum_{t=0}^{T-1} \mathcal{X}_t \mathcal{S}_{T-t}
\end{aligned}$$

where the approximation holds up to a constant by the construction of the tiling. We can begin by substituting in:

$$\begin{aligned}
\mathcal{X}_T &= \beta_n \delta_n p_n n \sum_{t=0}^{T-1} \mathcal{X}_t (T-t)^q \\
&= \beta_n \delta_n n p_n \left[T^q + \left(\sum_{t_1=1}^{T-1} (T-t_1)^q \left(\beta_n \delta_n p_n n \left[t_1^q + \sum_{t_2=1}^{t_1-1} (t_1-t_2)^q \left(\beta_n \delta_n p_n n \left[t_2^q + \sum_{t_3=1}^{t_2-1} (t_2-t_3)^q (\beta_n \delta_n p_n n \times \right. \right. \right. \right. \right.
\end{aligned}$$

Note that the nested summation must be polynomial in T, despite the multiplicative structure. While we have combinatorial growth in the number of terms, we are only multiplying polynomials of T together. As polynomials are closed under multiplication, the result will be a polynomial in T, with the lead term to be $T^q \beta_n \delta_n n p_n$.

To complete the proof, we verify the validity of the tiling with the given assumptions. We verify compatibility with Assumption 2. First, note that to have links in E_n, in expectation, we must have:

$$p_n \delta_n T^q < n \Rightarrow T < \left(\frac{n}{p_n \delta_n} \right)^{1/q}$$

Second, recall the assumption we made in the tiling: we have to be able to divide L_n, the

base graph, into enough tiles. We can collect the relevant conditions:

$$K(T,n) \geq \sum_{t=0}^{T-1} \mathcal{X}_t \geq \mathcal{X}_{T-1} \geq \beta_n \delta_n p_n n (T-1)^q,$$

$$n > K(T,n)\mathcal{E}_T \Rightarrow \frac{n}{T^{q+1}} > K(T,n)$$

The first statement holds by construction and evaluation based on prior computations. The second statement enforces that the total expected number of activations in all tiles must be less than n – mechanically, this enforces that not all nodes are activated in expectation. We can combine inequalities to get $\frac{n}{T^{q+1}} > \beta_n \delta_n p_n n (T-1)^q$. Given that $\beta_n > \frac{1}{p_n \delta_n n T^q}$, asymptotically this gives us that $T < n^{\frac{1}{q+1}}$. This is the stricter of the two upper bounds on T, so it binds (and is exactly the upper bound of Assumption 2).

We can consider the resulting structure of the tile level graph, despite E_n not necessarily being connected. This will give us a lower bound on T, as we implicitly assume that the tile level graph to be connected with probability one. We imposed that there are $v(T,n) = n/K(T,n)$ nodes per tile. Given β_n, the probability of connection between two *tiles* will be $1-(1-\beta_n\delta)^{v(T,n)^2} \approx \beta_n \delta v^2(T,n)$. We want this quantity to be at least $\log n/n$. Re-writing our expression for the tile link rate in terms of K yields the following expression.

$$\beta_n \delta_n \frac{n^2}{K(T,n)^2} > \frac{\log n}{n} \Rightarrow \beta_n \delta_n > \frac{\log n}{n^3} K(T,n)^2.$$

We can then consider this expression when β_n is as small as possible, and $K(T,n)$ is as large as possible, and note that this is consistent with Assumption 2 that

$$\frac{1}{p_n n T^q} > \frac{\log n}{n} \frac{1}{T^{2q+2}} \implies T > (p_n \log n)^{1/(q+2)}$$

Note that this is a (much) stricter lower bound than what is imposed by Assumption 2. Thus

the tighter lower bound will still give the desired properties. This completes the proof. $\square$

Proof of Theorem 2. Fix the percolation P_n and recall in what follows Γ_n is respected. All distances are with respect to $P_n \cap G_n$, meaning the intersection of the realized graph and the realized percolation. Recall that e_1 is the closest node to i_0 in P_n that also has a link in E_n. Let e_2 be the second closest such node.

Define $r \coloneqq d(i_0, e_2)$, the distance between i_0 and e_2. Set $T = \kappa \cdot r$ for some $\kappa > 0$, which determines the diffusion duration. Then let $a_n = o_p(r)$ growing in n be a distance and $U_n \coloneqq B_{i_0}(a_n)$. Note $|U_n|/T_n^{q+1} \to_p 0$ by construction, meaning that U_n is a sequence of local neighborhoods vanishing relative to the diffusion. Then pick $b_n = r - ca_n$ for $c \in (0, 1)$, constant in n. Notice the lens formed, $\ell(a_n, b_n; r) \coloneqq U_n \cap B_{e_2}(b_n)$ is of constant order relative to U_n. Let $J_{i_0} \coloneqq \ell(a_n, b_n; r)$, completing the construction of J_{i_0}. This proves the first part of the theorem.

We can show that $\Delta_n(i_0, j_0) < c < 1$ for some positive fraction independent of n. For any P, the distance between the two nodes is order b_n, so the lens between them has order b_n^q as does the disjoint set. But this is the same order as s_n^q which we saw as the volume of the activations emanating from alter e_2'. So the result follows as this holds for any P that respects Γ_n.

We can then prove the final part of the Theorem. Every $j_0 \in J_{i_0}$ reaches e_2 with at least $s_n = cb_n - 1$ more steps. At that point, at least s_n^q activations occur about alter e_2' of e_2. We can think of a new diffusion starting at e_2 for at least s_n periods. The region around the alter of e_2 will be the first region seeded, and there will be potentially more in expectation,

depending on the parameters. By Lemma 1, the number of regions activated in expectation will be at least: $n\beta_n\delta_n p_n s_n^q$. Recall that this result relies on choosing a tiling with K regions – we take the regions to be the catchment areas themselves. Note that K is growing in n. Then, it follows that:

$$\mathbb{P}(X_{s_n} - n\beta_n\delta_n p_n s_n^q \geq \epsilon) \leq \exp\left(-\frac{\epsilon}{K}\right) \to 0$$

Via an application of Hoeffding's inequality to the set of indicators for if a catchment region has been activated. This completes the proof. □

Proof of Theorem 3. We can first note that the numerator is exactly $\hat{Y}_T(L_n) = \mathcal{E}_T$ and can be bounded from above using Assumption 1. Then, we can construct a tiling and apply Lemma 1.

Formally:

$$\hat{Y}_T(G_n) = \mathbb{E}\left[\sum_{j=1}^{n} y_{jT} = 1 \;\middle|\; E_n + L_n\right] \geq \mathbb{E}\left[\sum_{j=1}^{n} y_{jT} = 1 \;\middle|\; E_n + \tilde{L}_n\right].$$

The lower bound comes from ignoring the spread between tiles – instead, we only allow for inter-tile spread through E_n. We will lower bound the expression further by only counting the first activation in each tile.

Note that Lemma 1 provides a lower bound for the number of *tiles* seeded in each period (only tracking first activations), but we want the number of nodes ever activated. This will be $\sum_{s=0}^{T} \mathcal{X}_s \mathcal{E}_{T-s}$, where we weight the spread in each period $\mathcal{E}_{T-s}$ by the number of tiles seeded for the first time in that period. We must weigh the number of tiles by the volume

of (expected) spread given the initial activation time. Therefore we have the following:

$$\begin{aligned}\hat{Y}_T(G_n) &= T^{q+1} + \sum_{s=0}^{T-1} \mathcal{X}_s (T-s)^{q+1} \\ &\geq T^{q+1} + \beta_n p_n n \sum_{s=0}^{T-1} s^q (T-s)^{q+1} \\ &\geq T^{q+1} + \frac{1}{4^{2q+1}} \beta_n p_n n T^{2q+1}\end{aligned}$$

where the second bound comes from taking only the term corresponding to $\frac{T}{2}$ from the sum, which will be the largest individual term.[25]

Now we can consider our object of interest using these bounds:

$$\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \leq \frac{T^{q+1}}{T^{q+1} + T^{2q+1}\beta_n \delta_n p_n n / 4^{2q+1}} = \frac{1}{1 + T^q \beta_n \delta_n p_n n / 4^{2q+1}}$$

Then, by Assumption 3, this quantity will go to 0 as $n \to \infty$ and $T \to \infty$. $\square$

Proof of Proposition 6. For (1), We note as $m_n = o(\sqrt{n})$ and $\beta_n \in \left(\frac{1}{p_n n T^q}, \frac{1}{n}\right)$, then $\beta_n m_n = o\left(\frac{1}{\sqrt{n}}\right)$, $\beta_n m_n^2 = o(1)$. Then we have that

$$\begin{aligned}\mathbb{P}\left(\text{No links amongst } \binom{m_n}{2} \text{ found}\right) &= (1-\beta_n)^{\binom{m_n}{2}} \approx 1 - \beta_n \binom{m_n}{2} \\ &= 1 - \beta_n \frac{m_n^2 - m_n}{2} = 1 - o(1) + o\left(n^{-1/2}\right) \to 1,\end{aligned}$$

where we use the binomial approximation. Note that this will tend to 1 even in the most adversarial case, where β_n is as large as possible ($m_n = o(\sqrt{n})$).

For (2), it suffices to show that a necessary condition for the law of large numbers fails. Let e_{ij}^n denote a potential edge in E_n and $z_{ij}^n = e_{ij}^n / \beta_n$ which is a normalized version. Then

[25] We assume for the sake of more compact notation that T is even – if odd, simply take the floor of $T/2$ and the order of magnitude and thus the proof is preserved.

we can calculate, for s_{ij} a dummy for the pair being sampled,

$$\text{var}\left(\frac{2}{m_n(m_n-1)}\sum_{i,j:s_{ij}=1} z_{ij}^n\right) = \frac{1}{\beta_n^2}\frac{2}{m_n(m_n-1)}\beta_n(1-\beta_n) = \frac{2(1-\beta)}{m_n^2\beta_n - m_n\beta_n}.$$

For the law of large numbers to apply we need the variance to go to zero and therefore we need $m_n^2\beta_n$ to diverge, and this fails under the hypothesized condition. □

Proof of Theorem 4. We assume the policymaker observes an activated agent with a known probability α_n. The total number of activations can be accurately estimated by dividing the observed total count by α_n. Say that a region has x activations: then the probability of at least one activation being detected will be $1-(1-\alpha_n)^x \approx \alpha_n x$. Because this expression is approximately linear, the probability of detecting at least one activation in period t will be $\Theta(\alpha_n t^{q+1})$ via Assumption 1. We then want to scale by the number of regions activated in each period. This is exactly analogous to Lemma 1. Here, we take the tiles used in the proof to be the regions themselves. Recall that at time T there will be at least $\beta_n\delta_n np_nT^q$ regions activated in expectation – lower bounding K_T^*. So we have that

$$\frac{\hat{K}_T}{K_T^*} \leq \frac{\alpha_n\beta_n\delta_n np_nT^{2q+1}}{\beta_n\delta_n np_nT^q} + \alpha_n\frac{o\left(T^q\right)}{\beta_n\delta_n np_nT^q} \leq \alpha_n T^{q+1} < 1.$$

as $n\to\infty$, which completes the proof. □

Proof of Theorem 5. We can begin with a similar computation to the polynomial case, though the exponential nature of $\mathcal{E}_t$ makes exact computations possible. We begin with the analogue of Lemma 1, again working with a tiling of L_n. Again assuming that $K(T)$, the number of tiles, grows sufficiently quickly we can compute:

$$\mathcal{X}_T \geq \beta_n p_n n\sum_{t=0}^{T-1}\mathcal{X}_t\mathcal{S}_{T-t} = \beta_n\delta_n p_n n\sum_{t=0}^{T-1}\mathcal{X}_t q^{T-t} = \beta_n p_n\delta_n n(1+\beta_n\delta_n p_n n)^{T-1}q^T$$

Then, we can compute:

$$\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \leq \frac{q^T}{q^T + \sum_{s=0}^{T-1} \beta_n \delta_n p_n n (1+\beta_n p_n n)^{s-1} q^T q^{T-s}} = \frac{1}{1 + \frac{(1+\beta_n \delta_n n p_n)^T - 1}{1+\beta_n \delta_n n p_n}}$$

This quantity then goes to zero by Assumption 6.

We can then verify the validity of the tiling. We begin with our conditions on the tiling and that not all nodes are activated in expectation.

$$K(T,n) \geq \sum_{t=0}^{T-1} \mathcal{X}_t \geq \mathcal{X}_{T-1} = \beta_n \delta_n p_n n (1+\beta_n \delta_n p_n n)^{T-2} q^{T-1}, \text{ and } \frac{n}{q^T} > K(T,n)$$

in an identical fashion to the proof of Theorem 3. We can chain inequalities to get:

$$\frac{n}{q^T} > \beta_n \delta_n p_n n (1+\beta_n \delta_n p_n n)^{T-2} q^{T-1}$$

$$\log(n) > \log(\beta_n \delta_n p_n n) + (T-2)\log(1+\beta_n \delta_n p_n n) + (2T-1)\log(q)$$

By Assumption 6, we have that $\beta_n \delta_n p_n n > \varepsilon > 0$ so the bound reduces to $T = O(\log n)$. This restriction is exactly the first part of Assumption 5. For the second part of the bound, we repeat the same computation from the proof of Theorem 3, ensuring that the tile level graph is connected almost surely. We know that the following must hold:

$$\beta_n \delta_n > \frac{\log n}{n^3} K(T,n)^2 \implies \frac{1}{p_n n} > \frac{\log n}{n} \frac{1}{q^{2T}} \implies q^{2T} > p_n \log n$$

$$2T \log(q) > \log(p_n) + \log\log(n) \implies T > \frac{\log p_n}{2\log q} + \frac{\log\log n}{2\log(q)}$$

So the key condition is $T = \Omega(\log\log(n))$, which is exactly the second condition on T from Assumption 5. Note that we use Assumption 4 so that this bound is well-defined. This completes the proof of the Theorem. □

Proof of Proposition 7. Recall that under Assumption 4, and L_n being divided into $K(T,N)$ independent tiles, we can compute the expected number of regions activated at

time T via a recursion in the same way as before: $\mathcal{X}_T = \beta_n p_n n(1+\beta_n p_n n)^{T-1} q^T$. Note that because we assume L_n is divided into tiles, the computation is exact rather than a lower bound. Note that tracking secondary activations preserves the same order of magnitude. Then, by the same computation as before we have:

$$\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \geq \left(1 + \frac{(1+\beta_n \delta_n n p_n)^T - 1}{1+\beta_n n \delta_n p_n}\right)^{-1}$$

which goes to 1 as $\beta_n = O\left(\frac{1}{p_n n}\right)$. Verification of the tiling strategy proceeds in much the same way as in the proof of Theorem 5, which completes the proof. $\square$

Chapter 5

Summary and Conclusion

In this work, we use graphs to study relations from two perspectives. First, locational relations between the data points encode information about the underlying geometric structure of the data, and graphs as geometric objects can help characterize such data structures. For this line of work, we propose the skeleton framework that constructs a representational weighted graph, referred to as the skeleton, to encode the geometric structures in the data and utilize the learned graph to assist the downstream analysis such as clustering and regression. For the skeleton clustering, we make connections to density-based clustering literature and prototype approaches and adopt these methods to data regimes with large dimensionality and complex structures. A key construct in the skeleton clustering framework is the surrogate density measure and in particular the Voronoi density is derived from the geometric intuition of the in-between region and has simple and fast estimation. For the skeleton regression work, we combine the skeleton graph with the nonparametric regression approaches and manifold learning techniques and discussed how these methods can be applied to a general metric

space.

One main topic for future work along the skeleton framework is to provide better theoretical characterizations of the skeleton construction. This involves analyzing the locations of the knot and the locations of the edges relative to the underlying manifold accounting for the randomness in the construction process. In the current theoretical analysis, we assume that the knots and edges are fixed to simplify the problem, but in practice, knots are computed from the sample data with inherent uncertainty. For clustering, the locations of knots directly impact the Voronoi cells, which changes the value of the similarity measures and consequently the cluster label assignments. In particular, observations on the boundary of clusters will be more sensitive to any perturbations in the locations of knots. For regression, the proposed skeleton-based models are defined on the knots and edges, and hence variations in the skeleton fundamentally change the regression models. Moreover, better characterization of the skeleton is needed to quantify the projection error between the true regression function and the skeleton-projected regression.

To start with, there are two major technical challenges when dealing with random knots. First, the randomness of knots may be correlated with the randomness of estimated edge weight, so the calculation of rates is much more complicated. Second, while there are established theories for k-means algorithm (Graf and Luschgy, 2000, 2002; Hartigan and Wong, 1979), these results only apply to the global minimum of the objective function. In reality, we are unlikely to obtain the global minimum, but instead, our inference is based on a local minimum. It is unclear how to properly derive a theoretical statement based on local

minima. For the next step of the skeleton construction, the edges are connected according to the approximate Delaunnay Triangulation using the Voronoi density. While we study the convergence of the Voronoi density to its theoretical true quantity, some results on how such convergence affects the edge construction can be further developed.

With the skeleton constructed and fixed, we project the covariates onto the skeleton for regression purposes as described in Section 3.2.3, which inevitably introduces the projection error between the true regression function $m(\boldsymbol{x}) = \mathbb{E}(Y|X = \boldsymbol{x})$ and the skeleton-projected regression function. Bounding this projection error requires more than understanding the skeleton, but also a precise characterization of the underlying manifolds and the data distribution around them and further the relative location of the skeleton with respect to the local manifold structure. Therefore, although we have results on the simple examples in Appendix L, bounding the projection error under some general conditions requires careful formulation with high complexity.

Despite the challenges of rigorously analyzing the skeleton construction process, we still see possibilities to further develop the skeleton framework, particularly the potential to generalize skeleton graphs to a simplicial complex. From a geometric perspective, the skeleton graph constructed in this work only focuses on 0-simplices (points) and 1-simplices (line segments). Additional geometric information can be encoded using higher-dimensional simplices, which offer a finer approximation to the covariate distribution but entail a higher computational cost and a more complex model. Recent research in deep learning has explored the use of simplicial complices for tasks such as clustering and segmentation, and some

intuitions can potentially be relevant.

On network graphs, we have studied the lack of robustness to vanishingly small quantities of local mismeasurement in SIR diffusion models, which are widely used to conceptualize epidemics, information flow, and technology adoption, among other applications. We have seen that a number of quantities of interest to policymakers, such as diffusion forecasts, estimates of where the diffusion has occurred in the network, and the efficacy of further data collection or widespread testing may all be problematic in the face of extremely small measurement errors in the network. Our work demonstrates that network diffusion models can be limited and tools should be used for exactly what they are developed to inform policy. Particularly, aggregate concepts geared towards retrospective calculations may be good for just that purpose– certain aggregates, e.g., $\mathcal{R}_0$, may better be used as descriptive rather than prescriptive tools.

However, although our work raises practical concerns for network diffusion estimates along with the consequent policy implications and further discusses the limitation of some potential solutions, there is no need to be nihilistic. The problematic scenarios we discussed in this work all happen in a medium-time regime where the estimates are off after running one fixed model for a sufficient amount of time. This presents an argument for policymakers to respond to a network diffusion earlier and more aggressively without relying on medium to long-run estimations. In an uncertain world, policymakers may want to pursue more aggressive containment policies to guard against worst-case scenarios on network dynamics. This is the thrust of the statistical force given the massive uncertainty we document and the

full decision theory exercise can be interesting to pursue in the future.

For this work, we study the robustness of SIR models from the aspect of missing links on a polynomial expansion network. This paper is specific to SIR models on graphs, but the phenomenon need not be. In fact, the same sort of perturbation robustness failure may impact general models of treatment effects with spillovers (e.g., Aronow and Samii (2017) and Athey et al. (2018)). Other diffusion models on the network with other types of network mismeasurements like fictitious edges can be studied. Particularly, the examination of perturbation robustness failure in causal graphical models can be interesting in its own right. Future research can also formulate the network structure in different ways, particularly how the geometric properties of the network can affect the robustness of particular diffusion models on it. Overall, robustness analysis from diverse perspectives can help understand the limitations of models in different scenarios and hence better inform policymaking.

Appendix A

Appendix

Chapter 2 Appendices

A Computational Complexity

Knots construction. The first step of skeleton clustering is choosing knots, and in this work, we take overfitting k-means as the default method. The k-means algorithm of Hartigan and Wong (Hartigan and Wong, 1979) has time complexity $O(ndkI)$, where n is the number of points, d is the dimension of the data, k is the number of clusters for k-means, and I is the number of iterations needed for convergence. When using overfitting k-means to choose knots, the reference rule is $k = \sqrt{n}$, and hence the complexity is $O(n^{3/2}dI)$. This is a time-consuming step of our clustering framework, and the complexity increases linearly with d. Therefore, preprocessing the data with dimension reduction techniques or using subject knowledge to choose knots can be helpful to speed up this process.

Edges construction. For the edge construction step, we approximate the Delaunay Triangulation with $\hat{DT}(\mathcal{C})$ by looking at the 2-NN neighborhoods (the Voronoi Density regions in 2.3.1). Hence the main computational task for our edge construction step is the 2-nearest knot search. We used the k-d tree algorithm for this purpose, which gives the worst-case complexity of $O(ndk^{(1-1/d)})$, while a brute-force search line search gives a decent complexity of $O(ndk)$. Notably, the computation complexity at this step is at the worst linear in d, which is a much better rate than computing the exact Delaunay Triangulation (exponential dependence on d), and our empirical studies have illustrated the effectiveness of such approximation.

Edge weight construction: VD. Next, we consider the computation complexity of the different edge weight measurements. For the VD, its numerator can be computed directly from the 2-NN search when constructing the edges and hence no additional computation is needed. The denominators are pairwise distances between knots and can be computed with the worst-case complexity of $O(dk^2)$ because the number of nonzero edges is less than $\frac{k(k-1)}{2}$. With $k = \sqrt{n}$, we have the total time complexity of computing the VD to be $O(nd)$.

Edge weight construction: FD. For the Face density, we calculate the projected KDE at the middle point for each pair of neighboring Voronoi cells. The projection of one data point onto one central line can be done by matrix multiplication with complexity $O(d)$. Recall that we only use data points in local Voronoi cells for FD calculation, and the local sample size would be at $n_{loc} = O(\sqrt{n})$ under the conditions in Section 2.4 and the reference rule $k = [\sqrt{n}]$. Together it takes $O(d\sqrt{n})$ to calculate the projected data for one edge. With the

projected data, KDE calculation has a time complexity $O(c \log c)$ where $c = \max_{j \neq \ell}\{n_j + n_\ell\}$ for any pair of knot indexes j, ℓ. Again we have $c = O(n/k) = O(\sqrt{n})$ under the previously mentioned conditions. We need to do KDE for each edge in the skeleton, which gives the overall time complexity of FD weights to $O(k^2 d\sqrt{n} + k^2 c \log c) = O(n^{3/2}d + n^{3/2} \log n)$.

Edge weight construction: TD. For Tube density, we similarly perform a projected KDE for each edge. Let η be the maximum number of points in a tube region $\eta = \max_{j,\ell} |\{X_i : \|\Pi_{j\ell}(X_i) - X_i\| \leq R\}|$, the data projection again takes $O(\eta d)$ complexity. Suppose the minimum density is obtained by a grid search with m grid points, the KDE step takes a total of $O(m\eta \log \eta)$ for one edge. To compute the whole edge weights matrix with $k = \sqrt{n}$, we have the complexity to be $O(n\eta d + nm\eta \log \eta)$. Under conditions where the tube regions for TD estimations are also of size $\eta = O(n/k) = O(\sqrt{k})$, we have the overall complexity for VD weights calculation to be $O(k^2 d\sqrt{n} + k^2 c \log c) = O(n^{3/2}d + mn^{3/2} \log n)$, which is larger than that for FD due to the grid search for minimum density.

Knots segmentation. In this work, we segment the learned weighted skeleton using hierarchical clustering. With links that can be updated by Lance-Williams update (Lance and Williams, 1967) and satisfy the reducibility condition (Gordon, 1987), hierarchical clustering can be carried out with computation complexity $O(N^2)$, where N is the number of points to start the algorithm with (Murtagh, 1983). For our empirical results, we favored single linkage and average linkage, and both satisfy the requirements for an efficient hierarchical clustering algorithm. We perform hierarchical clustering on the $k = \sqrt{n}$ knots, and hence the computation complexity for segmenting the skeleton structure is $O(k^2) = O(n)$.

B Theory for Face Density

Here we derive the convergence rate of the Face Density estimator. Recall that μ_d is the Lebesgue measure on the d-dimensional Euclidean space and $F_{j\ell} = \mathbb{C}_\ell \cap \mathbb{C}_j$ is the face region between knots c_j, c_ℓ. Let $\partial F_{j\ell}$ be the boundary of $F_{j\ell}$. We consider the following assumptions:

(D1) (Density conditions) The PDF p has compact support $\mathcal{X}$, is bounded away from zero that $\inf_{x\in\mathcal{X}} p(x) \geq p_{\min} > 0$, $\sup_{x\in\mathcal{X}} p(x) \leq p_{\max} < \infty$, and is Lipschitz continuous.

(B2) (Bounded face region) There exist constants c_0, c_1 such that the face area

$$\frac{c_0}{k^{1-\frac{1}{d}}} \leq \min_{(j,\ell)\in E} \mu_{d-1}(F_{j\ell}) \leq \max_{(j,\ell)\in E} \mu_{d-1}(F_{j\ell}) \leq \frac{c_1}{k^{1-\frac{1}{d}}}$$

(B3) (Boundary of face bounded) There exists a constant c_2 such that

$$\max_{(j,\ell)\in E} \mu_{d-2}(\partial F_{j\ell}) \leq \frac{c_2}{k^{1-\frac{2}{d}}},$$

(B4) (Intersecting angle condition) There is an angle $\theta_0 < \pi$ such that, for every pair of intersecting face regions F_{ij} and $F_{j\ell}$, the maximal principle angle between the two subspaces $\theta_{ij,j\ell}$ satisfies $\theta_{ij,j\ell} \leq \theta_0$

(K1) (Kernel function conditions) The kernel function K is a positive and symmetric function satisfying $\int K^2(x)dx < \infty, \quad \int |x|K(x)dx < \infty, \quad \int x^2K(x)dx < \infty$.

Assumption (D1) is commonly assumed for the density estimation problem, but usually with higher-order smoothness conditions. Notably, for consistency of the FD estimator we require only the Lipschitz condition since the bias of the sample estimator will be dominated by a geometric difference even if we have a higher-order smoothness (see the discussion after Theorem 8 and Appendix D for more detail). Condition (B2) restricts the shared boundary

of two Voronoi cells to scale at the rate of $O(k^{1-\frac{1}{d}})$. While this condition may seem abstract, it is a mild condition. To illustrate this, suppose we have $k = m^d$ points that are on a uniform grid of $[0,1]^d$ for some integer m. We form the Voronoi cells of these grid points. The $(d-1)$-dimensional volume of the shared boundary of two neighboring Voronoi cells will scale at rate $O(k^{1-\frac{1}{d}})$ as $k \to \infty$. (B3) requires the boundaries of the face regions to scale at most at a rate of $O(k^{1-\frac{2}{d}})$, and (B4) requires that we cannot have two nearby faces to be parallel to each other. Assumptions (B3) and (B4) are needed when bounding the geometric difference between the estimator and the population quantity and are both mild conditions: When the knots form a spherical packing of a smooth region, these conditions hold. Notably, (D1) and (B2) imply (B1), and hence the consistency of FD requires more conditions than the consistency of VD. The condition (K1) is a common assumption on the kernel function (Wasserman, 2006a; Scott, 2015) satisfied by many common kernel functions, including the Gaussian kernel.

Theorem 8 (Face Density). Assume (D1), (K1), and (B2-B4). With $h \to 0$, $k \to \infty$, $hk^{1/d} \to 0$, $\frac{nh}{k^{1-\frac{1}{d}}} \to \infty$, then for any pair $j \neq \ell$, we have

$$\left| \frac{\hat{S}_{j\ell}^{FD}}{S_{j\ell}^{FD}} - 1 \right| = O\big(hk^{1/d}\big) + O_p\left(\sqrt{\frac{k^{1-\frac{1}{d}}}{nh}}\right) \tag{A.0.1}$$

Theorem 8 shows the convergence rate of estimating the FD. Roughly speaking, the rate is similar to a 1-dimensional density estimation problem. With $d \to \infty$, we have the rate to be $O\big(h\big) + O_p\left(\sqrt{\frac{k}{nh}}\right) = O\big(h\big) + O_p\left(\sqrt{\frac{1}{n_{loc}h}}\right)$, where $n_{loc} = O\big(\frac{n}{k}\big)$ is the local effective sample size. Therefore, the effect of the ambient dimension is negligible when d is large, and

this is because we are estimating a 'projected' density on the central line, which reduces to a 1-dimensional problem.

Noticeably, the bias term in Theorem 8 is of the order $O(h)$. While this rate is optimal under the Lipschitz smoothness (D1) for density estimation problem, it is slower than the conventional rate $O(h^2)$ when we have a bounded second-order derivative of p. One may be wondering if higher-order smoothness of p is assumed, can we improve the convergence rate? Unfortunately, even if p is very smooth, the bias rate will still stay the same at $O(h)$. This is because there are two sources of bias. The first one is the usual bias from kernel smoothing, which can be improved to higher order if we have high-order derivatives of p. The other source of bias comes from the different geometric shapes of the Voronoi cells $\mathbb{C}_j$ and $\mathbb{C}_\ell$ (for illustration see Figure A.1 in Appendix D). Consider the characterization of central line as $c_j + t(c_\ell - c_j)$ for $t \in [0, 1]$, and the boundary will occur at $t = \frac{1}{2}$. Regions projected onto the central line will be different depending on the value of t. Specifically, when $t > \frac{1}{2}$, the projected region is from $\mathbb{C}_\ell$ whereas when $t < \frac{1}{2}$, the projected region is from $\mathbb{C}_j$, and those projected regions can have shapes different from the face region. This difference leads to an additional geometric bias of the order $O(h)$ and cannot be improved by higher-order smoothness of p. In a sense, this bias $O(h)$ is similar to the boundary bias in that the density function is continuous but not differentiable. However, since the non-differentiability is caused by the geometric difference in two nearby Voronoi cells, it is unclear if we can use the conventional boundary-correction kernels (Jones, 1993) to correct this bias.

From Theorem 8, one can see that the optimal bandwidth scales at rate $h \asymp \left(\frac{k^{1-3/d}}{2n}\right)^{1/3}$.

Recall that our reference rule sets $k = \sqrt{n}$ so that $n_{loc} = \frac{n}{k} = \sqrt{n}$ is the average number of observations per each knot. When d large, $\frac{3}{d}$ is negligible. Thus, the optimal bandwidth is given by $h \asymp \left(\frac{k}{n}\right)^{1/3} = n_{loc}^{-1/3}$. While our empirical rule $n_{loc}^{-1/5}$ is not optimal in this case, it still gives a consistent estimator and our empirical analysis shows that such choice leads to reliable clustering results; see Appendix F.

One may notice that a small k in Theorem 8 leads to a better convergence rate, which suggests to use a small k. While this is true from the perspective of estimation, overall a small k may lead to a poor representation of the data and result in a bad clustering performance. Empirical results show that we need a sufficiently large number of knots to represent the data in order for the skeleton clustering to perform appropriately. Therefore, our reference rule with $k = \sqrt{n}$ is a suitable balance between the trade-off between representation and estimation. We include an empirical analysis of the effect of k on clustering performance in Appendix F.

C Theory for Tube Density

In this section we derive the convergence rate of the Tube Density estimator. We consider the following assumptions, which are slightly stronger than the corresponding ones in the case of the FD:

(D2) (Density conditions) The PDF p has a compact support and is 3-Hölder and $\inf_{x \in \mathcal{X}} p(x) \geq f_{\min} > 0$.

(D3) (Disk Density conditions) For any pair c_j, c_ℓ, the minimum disk density location $t^* =$

$\mathsf{argmin}_{t\in[0,1]}\mathsf{pDisk}_{j\ell,R}(t) \in (0,1)$ is unique and the second derivative of the disk density $\mathsf{pDisk}^{(2)}_{j\ell,R}(t^*) \geq c_{\min} > 0$.

(K2) (Kernel function conditions) The kernel function K is a positive and symmetric function satisfying $\int x^2 K^{(\alpha)}(x)dx < \infty, \int (K^{(\alpha)}(x))^2 dx < \infty$, for all $\alpha = 0, 1, 2$, where $K^{(\alpha)}$ denotes the α-th order derivative of K.

(D2) is a stronger version of (D1) that we require an additional smoothness condition of p. We need the 3-Hölder class (slightly weaker than the requirement of third-order derivatives) to obtain the rate of estimating the minimum (Chacón et al., 2011; Chen et al., 2016). Also, a stronger condition (K2) on the kernel function is needed to ensure the gradient estimation is consistent. Fortunately, common kernel functions such as the Gaussian kernel satisfy these conditions.

Theorem 9 (Tube Density Consistency). Assume (D2), (D3), and (K2). Let $h \to 0$, $k \to \infty$, $R \to 0$, $nh^3 \to \infty$,$nhR^{d-1} \to \infty$. Suppose that for every pair c_j, c_ℓ, $\inf_{t\in[0,1]} \mathsf{pDisk}_{j\ell,R}(t)$ and $\inf_{t\in[0,1]} \widehat{\mathsf{pDisk}}_{j\ell,R}(t)$ do not occur at the boundary $t = 0, 1$. Then for any pair $j \neq \ell$ that shares an edge, we have

$$\mathsf{pDisk}_{j\ell,R}(t) = O(R^{d-1}), \tag{A.0.2}$$

$$\left|\frac{\hat{S}^{TD}_{j\ell}}{S^{TD}_{j\ell}} - 1\right| = O(h^2) + O_p\left(\sqrt{\frac{1}{nhR^{d-1}}}\right) + O_p\left(\frac{1}{nh^3}\right) \tag{A.0.3}$$

Theorem 9 shows that the TD estimator converges to the population TD with a rate consisting of three components. We allow $R \to 0$ as $n \to \infty$ but this result also applies to scenarios where R is fixed. The first component $O(h^2)$ is the usual smoothing bias. The

second component $O_p\left(\sqrt{\frac{1}{nhR^{d-1}}}\right)$ is similar to the stochastic variation part from usual KDE but with an additional dependence on R^{d-1}. This is due to the fact that, when $R \to 0$, we are using fewer and fewer observations to perform smoothing, and nR^{d-1} serves as the effective sample size. The third component $O_p\left(\frac{1}{nh^3}\right)$ is due to the error of estimating the location of the minimum. It is a squared term because the density behaves like a quadratic function around its minimum due to (D3).

While the convergence rate of TD requires stronger conditions (D2) and (K2) compared to the conditions (D1) and (K1) when estimating the FD, the TD estimator has a smaller bias than the FD estimator (comparing Theorem 8 and 9). This is because the TD is evaluated on a "regular shape", which leads to a smoother quantity being estimated.

For the stochastic variation part, the second term in Theorem 9 gives $O_p\left(\sqrt{\frac{1}{nhR^{d-1}}}\right)$ while the second term in Theorem 8 gives $O_p\left(\sqrt{\frac{k^{1-\frac{1}{d}}}{nh}}\right)$. Note that empirically we choose R to be the average of the root mean squared distances of each Voronoi cell (Section 2.3.3), which is of order $O(k^{-1/d})$ with cell sizes to have the same rates. Hence $k^{1-1/d}$ and $\frac{1}{R^{d-1}}$ are at the same rate and the stochastic variation part is comparable for TD and FD estimators. However, for TD we have another source of variation coming from the uncertainty of the location of minimum, which can cause TD to have larger variation than the FD estimator.

Based on the above reasoning, our choice of R leads to $\frac{1}{R^{d-1}} \asymp k^{1-1/d}$, which implies the rate $O(h^2) + O_p\left(\sqrt{\frac{k^{1-1/d}}{nh}}\right) + O_p\left(\frac{1}{nh^3}\right)$. Under our reference rule $k = \sqrt{n}$ the optimal bandwidth is $h \asymp n^{-\frac{1}{10}(1+\frac{1}{d})}$. Recall that the local sample size is about $n_{loc} = n/k = \sqrt{n}$ and hence the optimal bandwidth is $h \asymp n_{loc}^{-\frac{1}{5}(1+\frac{1}{d})}$. When $d \to \infty$, this leads to $h \asymp n_{loc}^{-1/5}$, which

is the same rate on sample size as given by Silverman's rule of thumb.

Remark 12. Similar uniform bounds of the Face and Tube density can be derived with an extra $\log k$ factor in the rates through the concentration bound for kernel density estimator (Giné and Guillou, 2002). Also, similar concentration bounds on the Adjusted Rand Indexes can be achieved for partition based on the Face and Tube density.

D Proofs

Voronoi Density Consistency

We restate the assumption:

(B1) There exists a constant c_0 such that the minimal knot size $\min_{(j,\ell)\in E} \mathbb{P}(A_{j\ell}) \geq \frac{c_0}{k}$ and $\min_{(j,\ell)\in E} \|c_j - c_\ell\| \geq \frac{c_0}{k^{1/d}}$, where $A_{j\ell}$ is the 2-NN region of knots c_j, c_ℓ as defined in Equation 2.3.1.

Proof of Theorem 1.

For given knots c_j, c_ℓ, the distance $||c_j - c_\ell||$ is also given. We denote the numerator of $S^{VD}_{j\ell}$ as

$$p_{j\ell} = \mathbb{P}(A_{j\ell}) = \mathbb{E}I(X_i : d(X_i, c_m) > max\{d(X_i, c_j), d(X_i, c_\ell), \forall m \neq j, l\})$$

and note that the numerator of $\hat{S}^{VD}_{j\ell}$ is

$$\hat{P}_n(A_{j\ell}) = \frac{1}{n}\sum_{i=1}^{n} I(X_i : d(X_i, c_m) > max\{d(X_i, c_j), d(X_i, c_\ell), \forall m \neq j, l\}),$$

which is a sum of binary variables and has variance $\sigma^2_{j\ell} = \frac{p_{j\ell}(1-p_{j\ell})}{n}$. By the Chebyshev's

inequality,

$$\left|\hat{P}_n(A_{j\ell}) - p_{j\ell}\right| = O_p(\sigma_{j\ell}^{1/2}) = O_p\left(\left[\frac{p_{j\ell}(1-p_{j\ell})}{n}\right]^{1/2}\right)$$

Note that the region $A_{j\ell}$ is changing with respect to k. The ratio is then

$$\begin{aligned}\left|\frac{\hat{S}_{j\ell}^{VD}}{S_{j\ell}^{VD}} - 1\right| &= \left|\frac{\hat{P}_n(A_{j\ell})}{\mathbb{P}(A_{j\ell})} - 1\right| = \frac{1}{p_{j\ell}} O_p\left(\left[\frac{p_{j\ell}(1-p_{j\ell})}{n}\right]^{1/2}\right) \\ &= O_p\left(\left[\frac{(1-p_{j\ell})}{np_{j\ell}}\right]^{1/2}\right) = O_p\left(\left[\frac{(1-c_0/k)}{nc_0/k}\right]^{1/2}\right) = O_p\left(\left(\frac{k}{n}\right)^{1/2}\right)\end{aligned}$$

by assumption (B1) that $\min_{(j,\ell)\in E}\mathbb{P}(A_{j\ell}) \geq \frac{c_0}{k}$, which completes the proof for Equation 2.4.1.

To get the uniform bound, we first start with the concentration bound. Note that $\left(I(X_i \in A_{j\ell}) - p_{j\ell}\right)$ has zero mean and $|I(X_i \in A_{j\ell}) - p_{j\ell}| \leq 1$. Hence by Bernstein's inequalities, we have

$$\begin{aligned}\mathbb{P}\left\{\left|\frac{\hat{P}_n(A_{j\ell})}{p_{j\ell}} - 1\right| > \varepsilon\right\} &= \mathbb{P}\left\{\left|\hat{P}_n(A_{j\ell}) - p_{j\ell}\right| > \varepsilon p_{j\ell}\right\} \\ &= \mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} I(X_i \in A_{j\ell}) - p_{j\ell}\right| > \varepsilon p_{j\ell}\right\} \\ &= 2\mathbb{P}\left\{\sum_{i=1}^{n}\left(I(X_i \in A_{j\ell}) - p_{j\ell}\right) > n\varepsilon p_{j\ell}\right\} \\ &\leq 2\exp\left\{-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n^2}{\sum_{i=1}^{n}\mathbb{E}\left[\left(I(X_i \in A_{j\ell}) - p_{j\ell}\right)^2\right] + \frac{1}{3}\varepsilon p_{j\ell} n}\right\} \\ &= 2\exp\left\{-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n^2}{np_{j\ell}(1-p_{j\ell}) + \frac{1}{3}\varepsilon p_{j\ell} n}\right\} \\ &= 2\exp\left\{-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n}{p_{j\ell}(1-p_{j\ell}) + \frac{1}{3}\varepsilon p_{j\ell}}\right\}\end{aligned}$$

Note that plugging in the $p_{j\ell} = \Omega\left(\frac{1}{k}\right)$ rate to above concentration bound we can recover the

$O_p\left(\sqrt{\frac{k}{n}}\right)$ rate in Equation 2.4.1. Then by union bound, we have

$$\begin{aligned}
\mathbb{P}\left\{\max_{(j,\ell)\in\mathcal{S}}|\hat{S}_{j\ell}/S_{j\ell}-1|>\varepsilon\right\} &\leq \mathbb{P}\left\{\max_{j,\ell}|\hat{S}_{j\ell}/S_{j\ell}-1|>\varepsilon\right\} \\
&\leq \sum_{j,\ell}\mathbb{P}\left\{|\hat{S}_{j\ell}/S_{j\ell}-1|>\varepsilon\right\} \\
&\leq \frac{k(k-1)}{2}\max_{j,\ell}\mathbb{P}\left\{\left|\frac{\hat{P}_n(A_{j\ell})}{p_{j\ell}}-1\right|>\varepsilon\right\} \\
&\leq k(k-1)\max_{j,\ell}\left\{\exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n}{p_{j\ell}(1-p_{j\ell})+\frac{1}{3}\varepsilon p_{j\ell}}\right)\right\} \\
&\leq k(k-1)\exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1-p_{min})+\frac{1}{3}\varepsilon}\right)
\end{aligned}$$

where $p_{min}=\min_{j\ell}p_{j\ell}$. Therefore we can derive the uniform error bound that

$$\max_{j,\ell}\left|\frac{\hat{S}_{j\ell}^{VD}}{S_{j\ell}^{VD}}-1\right| = O_p\left(\sqrt{\frac{k}{n}}\log k\right),$$

when $n\to\infty, k\to\infty, \frac{n}{k}\to\infty$.

□

Proof. of Theorem 2 (Performance guarantee for Voronoi density) We note that, assuming (P1),

$$\begin{aligned}
\mathbb{P}\left\{ARI(\mathcal{L}^*,\hat{\mathcal{L}})<1\right\} &\leq \mathbb{P}\left\{\text{there exists at least one wrongly cut edge}\right\} \\
&= \mathbb{P}\left\{\max_{(j,\ell)\in\mathcal{S}}|\hat{S}_{j\ell}/S_{j\ell}-1|>\varepsilon\right\} \\
&\leq k(k-1)\exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1-p_{min})+\frac{1}{3}\varepsilon}\right)
\end{aligned}$$

□

by the uniform bound derived above.

Face Density Consistency

Let $p(x)$ be the density function of the data distribution, let μ_d be the Lebesgue measure on the d-dimensional Euclidean space, let $F_{j\ell} = \bar{\mathbb{C}}_\ell \cap \bar{\mathbb{C}}_j$ denote the face between knots c_j, c_ℓ, and let $\partial F_{j\ell}$ be the boundary of $F_{j\ell}$. We consider the following assumptions: Again, we recall the assumptions:

(D1) (Density conditions) The PDF p has compact support $\mathcal{X}$, is bounded away from zero that $\inf_{x\in\mathcal{X}} p(x) \geq p_{\min} > 0$, $\sup_{x\in\mathcal{X}} p(x) \leq p_{\max} < \infty$, and is Lipschitz continuous.

(B2) There exist constants c_0, c_1 such that the face area

$$\frac{c_0}{k^{1-\frac{1}{d}}} \leq \min_{(j,\ell)\in E} \mu_{d-1}(F_{j\ell}) \leq \max_{(j,\ell)\in E} \mu_{d-1}(F_{j\ell}) \leq \frac{c_1}{k^{1-\frac{1}{d}}}$$

(B3) There exists a constant c_2 such that $\max_{(j,\ell)\in E} \mu_{d-2}(\partial F_{j\ell}) \leq \frac{c_2}{k^{1-\frac{2}{d}}}$,

(B4) There is an angle $\theta_0 < \pi$ such that, for every pair of intersecting face regions F_{ij} and $F_{j\ell}$, the maximal principle angle between the two subspaces $\theta_{ij,j\ell}$ satisfies $\theta_{ij,j\ell} \leq \theta_0$

(K1) (Kernel function conditions) The kernel function K is a positive and symmetric function satisfying $\int K^2(x)dx < \infty, \quad \int |x|K(x)dx < \infty, \quad \int x^2K(x)dx < \infty$.

Proof of Theorem 8.

Our analysis starts with the usual bias-variance decomposition that

$$\hat{S}^{FD}_{j\ell} - S^{FD}_{j\ell} = \underbrace{\hat{S}^{FD}_{j\ell} - \mathbb{E}(\hat{S}^{FD}_{j\ell})}_{\textbf{stochastic variation}} + \underbrace{\mathbb{E}(\hat{S}^{FD}_{j\ell}) - S^{FD}_{j\ell}}_{\textbf{bias}}.$$

We analyze the two terms separately. Before we start our proof, we first recall some useful notations.

Recall that the face region between two knots c_j, c_ℓ is $F_{j\ell} \equiv \overline{\mathbb{C}}_j \cap \overline{\mathbb{C}}_l$ and $c_* = c_j + \frac{1}{2}(c_\ell -$

$c_j) = \frac{1}{2}(c_\ell + c_j)$ and $\mathbb{L}_{j\ell} = \{c_j - a(c_\ell - c_j) : a \in [0,1]\}$ is the central line passing through c_j and c_ℓ, and for a value $a \in [0,1]$. The face $F_{j\ell} = \left\{x \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_l : \Pi_{j\ell}(x) = c_*\right\}$, where $\Pi_{j\ell}$ denotes the projection onto $\mathbb{L}_{j\ell}$. The quantity $\mu_s(dx)$ denotes the integration with respect to s-dimensional volume. We now reparametrize any point in $\mathbb{L}_{j\ell}$ using a unit distance t. Let $T_{j\ell,t} = \left\{x \in \mathcal{X} : \Pi_{j\ell}(x) = c_* + t\frac{c_\ell - c_j}{||c_\ell - c_j||}\right\}$ be the subspace orthogonal to $\mathbb{L}_{j\ell}$ at the point $c_* + t\frac{c_\ell - c_j}{||c_\ell - c_j||}$. t is 1-dimensional distance to c_* along the line passing through c_j and c_ℓ. Let

$$q_{j\ell}(t) = \int_{(\overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) \cap T_{j\ell,t}} p(x)\mu_{d-1}(dx)$$

With these quantities, $S_{j\ell}^{FD} = q_{j\ell}(0)$ and that $q_{j\ell}(t)$ is a 1-dimensional quantity. Our estimator is

$$\hat{S}_{j\ell}^{FD} = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - c_*}{h}\right) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell).$$

Bias: We study the bias part first. A direct computation shows that

$$\mathbb{E}[\hat{S}_{j\ell}^{FD}] = \mathbb{E}\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - c_*}{h}\right) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\right) \tag{A.0.4}$$

$$= \frac{1}{h}\int_{x\in\mathcal{X}} K\left(\frac{\Pi_{j\ell}(x) - c_*}{h}\right) I(x \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) p(x)\mu_d(dx) \tag{A.0.5}$$

$$= \frac{1}{h}\int_{\mathbb{L}_{j\ell}} K\left(\frac{c_* + t\frac{c_\ell - c_j}{||c_\ell - c_j||} - c_*}{h}\right)\left(\int_{(\overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) \cap T_{j\ell,t}} p(y)\mu_{d-1}(dy)\right) d\left(c_j + t\frac{c_\ell - c_j}{||c_\ell - c_j||}\right) \tag{A.0.6}$$

$$= \frac{1}{h}\int_{\mathbb{R}} K\left(\frac{\left\|t\frac{c_\ell - c_j}{||c_\ell - c_j||}\right\|}{h}\right) q_{j\ell}(t)dt \tag{A.0.7}$$

$$= \frac{1}{h}\int_{\mathbb{R}} K\left(\frac{t}{h}\right) q_{j\ell}(t)dt \tag{A.0.8}$$

$$= \int_{\mathbb{R}} K(u) q_{j\ell}(hu)du, \tag{A.0.9}$$

where for the third equality, we split the integration with respect to $c_j + t\frac{c_\ell - c_j}{||c_\ell - c_j||} \in \mathbb{L}_{j\ell}$

and the integration with respect to the subspace orthogonal to $\mathbb{L}_{j\ell}$ at $c_j + t\frac{c_\ell - c_j}{||c_\ell - c_j||}$. This is possible because all the points in $T_{j\ell,t}$ have the same projection onto $\mathbb{L}_{j\ell}$. For the fourth equality, we used the symmetry of the kernel function. the property of the kernel function that $K(x) = K(\|x\|)$. For the last equality, we used the change of variable that $u = \frac{t}{h}$ and got the simplified form.

The expansion of

$$q_{j\ell}(t) = \int_{(\overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) \cap T_{j\ell,t}} p(y)\mu_{d-1}(dy)$$

is more involved when $t \approx 0$. Let

$$W_{j\ell}(t) = (\overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) \cap T_{j\ell,t}$$

$$= \begin{cases} \overline{\mathbb{C}}_j \cap T_{j\ell,t}, & t < 0, \\ \overline{\mathbb{C}}_\ell \cap T_{j\ell,t}, & t > 0, \\ (\overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) \cap T_{j\ell,0} = F_{j\ell}, & t = 0 \end{cases}$$

be the region that leads to $q_{j\ell}(t)$. For a face $F_{j\ell}$ and a real number $t \in \mathbb{R}$, we denote

$$F_{j\ell} \oplus t = \left\{ x + t\frac{c_\ell - c_j}{||c_\ell - c_j||} : x \in F_{j\ell} \right\}.$$

By the above notation, we can decompose

$$W_{j\ell}(t) = [F_{j\ell} \oplus t] \cup \Delta_{j,\ell}(t),$$

where $\Delta_{j,\ell}(t)$ is the additional region when moving away from $t = 0$; see Figure A.1 for an example.

Thus, the difference

$$q_{j\ell}(hu) - q_{j\ell}(0) = \int_{W_{j\ell}(hu)} p(y)\mu_{d-1}(dy) - \int_{W_{j\ell}(0)} p(y)\mu_{d-1}(dy)$$

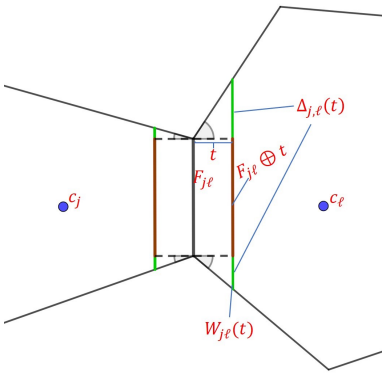


Figure A.1: Decomposition of $W_{j\ell}(t)$. The dark red segment is $F_{j\ell} \oplus t$, which has the same shape as $F_{j\ell}$. The green segments consist of $\Delta_{j,\ell}(t)$, the part leading to geometric bias.

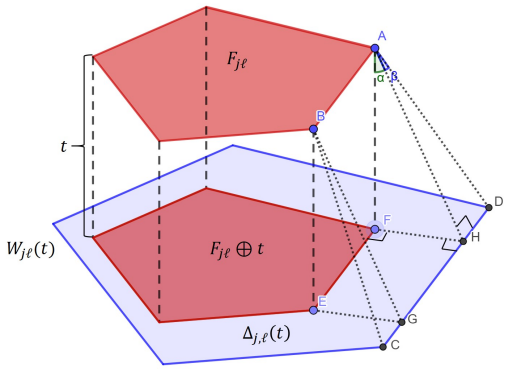


Figure A.2: Decomposition of $W_{j\ell}(t)$. The red regions are $F_{j\ell}$ and the projected $F_{j\ell} \oplus t$, while the blue band region denotes $\Delta_{j,\ell}(t)$. All the α angles such as $\angle FAH$ and all the β angles such as $\angle HAD$ are bounded by θ_0 from assumption (B4).

$$= \underbrace{\int_{F_{j\ell} \oplus hu} p(y)\mu_{d-1}(dy) - \int_{F_{j\ell}} p(y)\mu_{d-1}(dy)}_{(I)} + \underbrace{\int_{\Delta_{j\ell}(hu)} p(y)\mu_{d-1}(dy)}_{(II)}.$$

(I) is the usual bias caused by the change in density. Note that the Lipchitz condition in (D1) implies that there is a constant C_g such that $|p(x_1) - p(x_2)| \leq C_g|x_1 - x_2|$. Since every point can be matched nicely between $F_{j\ell} \oplus hu$ and $F_{j\ell}$, it can be bounded by

$$|(I)| \leq \mu_{d-1}(F_{j\ell})C_g h|u|.$$

(II) is the bias due to the change of volume, so we call it a geometric bias. With an upper bound of the density, (II) can be bounded by $(II) \leq \mu_{d-1}(\Delta_{j,\ell}(hu)) \cdot p_{max}$. Thus, we only

need to bound the volume $\mu_{d-1}(\Delta_{j,\ell}(hu))$.

$\Delta_{j,\ell}(t)$ is illustrated by the blue region in Figure A.2. The width of the band region like FH will all be bounded by $t\tan(\theta_0) = O(t)$, and as $t \to 0$ the surface area (circumference) will be bounded by $O\big(\mu_{d-2}(\partial F_{j\ell})\big)$.

Thus, the volume of the blue region $\mu_{d-1}(\Delta_{j,\ell}(t)) \leq O\big(\mu_{d-2}(\partial F_{j\ell})t\big)$, which leads to the bound

$$(II) \leq O\big(h|u| \cdot \mu_{d-2}(\partial F_{j\ell})\big) \cdot p_{max}.$$

Putting altogether, we have

$$\big|q_{j\ell}(hu) - q_{j\ell}(0)\big| \leq \mu_{d-1}(F_{j\ell})C_g h|u| + p_{max}h|u| \cdot O\big(\mu_{d-2}(\partial F_{j\ell})\tan(\theta_0)\big) \tag{A.0.10}$$

This, together with equation (A.0.9), implies that

$$\begin{aligned}
|\mathbb{E}[\hat{S}^{FD}_{j\ell}] - \underbrace{q_{j\ell}(0)}_{=S^{FD}_{j\ell}}| &= \left|\int_{\mathbb{R}} K(u)[q_{j\ell}(hu) - q_{j\ell}(0)]du\right| \\
&\leq \int_{\mathbb{R}} K(u)|q_{j\ell}(hu) - q_{j\ell}(0)|du \\
&\leq h\left[\int_{\mathbb{R}} |u|K(u)du\right] \times \left[\mu_{d-1}(F_{j\ell})C_g + p_{max}O\big(\mu_{d-2}(\partial F_{j\ell})\big)\right] \\
&\overset{(B2\text{–}3)}{=} O\left(h \cdot \left[\frac{1}{k^{1-1/d}}\right]\right) + O\left(h \cdot \left[\frac{1}{k^{1-2/d}}\right]\right)
\end{aligned}$$

As a result,

$$|\mathbb{E}[\hat{S}^{FD}_{j\ell}] - S^{FD}_{j\ell}| = O\left(\frac{h}{k^{1-1/d}}\right) + O\left(\frac{h}{k^{1-2/d}}\right) \tag{A.0.11}$$

Moreover, note that

$$\frac{h}{k^{1-1/d}} \times \frac{k^{1-2/d}}{h} = \frac{1}{k^{1/d}} \to 0 \tag{A.0.12}$$

since $k \to \infty$. Therefore the bias given by the geometric difference (II) dominates the bias

given by the change in density (I). Even if we assume a higher-order derivative, the bias in (II) will still dominate the component in (I).

Therefore, the overall bias can be expressed as reduces to

$$|\mathbb{E}[\hat{S}_{j\ell}^{FD}] - S_{j\ell}^{FD}| = O\bigg(\frac{h}{k^{1-2/d}}\bigg) \tag{A.0.13}$$

Stochastic variation: For the stochastic variation part, we have

$$\begin{aligned} Var(\hat{S}_{j\ell}^{FD}) &= Var\bigg(\frac{1}{nh}\sum_{i=1}^{n} K\bigg(\frac{\Pi_{j\ell}(X_i) - c_*)}{h}\bigg) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\bigg) \\ &\leq \frac{1}{nh^2}\mathbb{E}\bigg[K^2\bigg(\frac{\Pi_{j\ell}(X_i) - c_*}{h}\bigg) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\bigg] \\ &\leq \frac{1}{nh}\int K^2(u)\bigg(q_{j\ell}(0) + \mu_{d-1}(F_{j\ell})C_g + p_{max}h|u|\mu_{d-2}(\partial F_{j\ell})\tan(\theta_0)\bigg)du \\ &\leq \frac{1}{nh}\int K^2(u)\bigg(q_{j\ell}(0) + O\bigg(\frac{h}{k^{1-1/d}}\bigg) + O\bigg(\frac{h}{k^{1-2/d}}\bigg)\bigg)du \end{aligned} \tag{A.0.14}$$

by the same decomposition in (A.0.9) and the bound in (A.0.10) and the assumptions (K1). Note that similar to(A.0.12), the second term in (A.0.14) is at a slower rate than the third term, so we can simplify it as

$$Var(\hat{S}_{j\ell}^{FD}) = O\bigg(\frac{q_{j\ell}(0)}{nh}\bigg) + O\bigg(\frac{1}{nk^{1-2/d}}\bigg). \tag{A.0.15}$$

Combining (A.0.11) and (A.0.14), we conclude that for $\forall j, \ell$,

$$|\hat{S}_{j\ell}^{FD} - S_{j\ell}^{FD}| = O\bigg(\frac{h}{k^{1-2/d}}\bigg) + O_p\bigg(\sqrt{\frac{q_{j\ell}(0)}{nh}}\bigg) + O_p\bigg(\sqrt{\frac{1}{nk^{1-2/d}}}\bigg) \tag{A.0.16}$$

Note that the volume of face region $F_{j\ell}$ decreases when k increases. By assumption (D1) and (B2), we have

$$q_{j\ell}(0) = S_{j\ell}^{FD} \geq p_{\min} \min_{(j,\ell)\in E} \mu_{d-1}(F_{j\ell}) = p_{\min}\frac{c_0}{k^{1-\frac{1}{d}}}. \tag{A.0.17}$$

For the theorem, we again take the ratio between the estimated and the true face density

to accommodate the fact that the true face density decreases with the number of knots, and we have that This implies that

$$\left|\frac{\hat{S}_{j\ell}^{FD}}{S_{j\ell}^{FD}} - 1\right| = O\big(hk^{1/d}\big) + O_p\left(\sqrt{\frac{k^{1-\frac{1}{d}}}{nh}}\right) + O_p\left(\sqrt{\frac{k}{n}}\right) \tag{A.0.18}$$

When $hk^{1/d} \to 0$,

$$\frac{k^{1-\frac{1}{d}}}{nh} \times \frac{n}{k} = \frac{1}{hk^{1/d}} \to \infty, \tag{A.0.19}$$

so the second term dominates the third term in (A.0.18) and the rate reduces to

$$\left|\frac{\hat{S}_{j\ell}^{FD}}{S_{j\ell}^{FD}} - 1\right| = O\big(hk^{1/d}\big) + O_p\left(\sqrt{\frac{k^{1-\frac{1}{d}}}{nh}}\right), \tag{A.0.20}$$

which completes the proof.

□

Tube Density Consistency

We consider the following assumptions, which are slightly stronger than those in the case of the FD:

(D2) (Density conditions) The PDF p has compact support, is in the 3-Hölder class, and $\inf_{x\in\mathcal{X}} p(x) \geq f_{\min} > 0$.

(D3) (Disk Density conditions) For any pair c_j, c_ℓ, the minimum disk density location $t^* = \mathsf{argmin}_{t\in[0,1]}\mathsf{pDisk}_{j\ell,R}(t) \in (0,1)$ is unique and satisfies $\mathsf{pDisk}^{(2)}_{j\ell,R}(t^*) \geq c_{\min} > 0$.

(K2) (Kernel function conditions) The kernel function K is a positive and symmetric function satisfying $\int x^2 K^{(\alpha)}(x)dx < \infty, \int (K^{(\alpha)}(x))^2 dx < \infty$, for all $\alpha = 0, 1, 2$, where $K^{(\alpha)}$ denotes the α-th order derivative of K.

Proof of Theorem 9.

Let $t^* = \mathsf{argmin}_t \mathsf{pDisk}_{j\ell,R}(t)$ and $\hat{t}^* = \mathsf{argmin}_t \hat{\mathsf{pDisk}}_{j\ell,R}(t)$. Then the tube densities

$$S_{j\ell}^{TD} = \inf_{t\in[0,1]} \mathsf{pDisk}_{j\ell,R}(t) = \mathsf{pDisk}_{j\ell,R}(t^*),$$

$$\hat{S}_{j\ell}^{TD} = \inf_{t\in[0,1]} \hat{\mathsf{pDisk}}_{j\ell,R}(t) = \hat{\mathsf{pDisk}}_{j\ell,R}(\hat{t}^*).$$

Since the ratio difference

$$\frac{\hat{S}_{j\ell}^{TD}}{S_{j\ell}^{TD}} - 1 = \frac{1}{S_{j\ell}^{TD}} \left(\hat{S}_{j\ell}^{TD} - S_{j\ell}^{TD} \right),$$

we will focus on the difference $\hat{S}_{j\ell}^{TD} - S_{j\ell}^{TD}$.

The difference admits the following decomposition:

$$\begin{aligned}
\hat{S}_{j\ell}^{TD} - S_{j\ell}^{TD} &= \hat{\mathsf{pDisk}}_{j\ell,R}(\hat{t}^*) - \mathsf{pDisk}_{j\ell,R}(t^*) \\
&= \underbrace{\hat{\mathsf{pDisk}}_{j\ell,R}(\hat{t}^*) - \hat{\mathsf{pDisk}}_{j\ell,R}(t^*)}_{(I)} + \underbrace{\hat{\mathsf{pDisk}}_{j\ell,R}(t^*) - \mathbb{E}(\hat{\mathsf{pDisk}}_{j\ell,R}(t^*))}_{(II)} \\
&\quad + \underbrace{\mathbb{E}(\hat{\mathsf{pDisk}}_{j\ell,R}(t^*)) - \mathsf{pDisk}_{j\ell,R}(t^*)}_{(III)}.
\end{aligned}$$

It is easier to start with term (III) then term (II) and then term (I).

Recall that

$$q_{v,R}(y) = \int_{Disk(y,R,v)} p(x)dx,$$

and hence $\mathsf{pDisk}_{j\ell,R}(t) = q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j))$.

(III): Bias. Note that the kernel weights $w(x) = K\big(\frac{\Pi_{j\ell}(x) - c_j - t(c_\ell - c_j)}{h}\big)$ is the same for all $x \in Disk(c_j - t(c_\ell - c_j), R, c_\ell - c_j)$. Let $\mathbb{L}_{j\ell} = \{c_j - t(c_\ell - c_j) : t \in \mathbb{R}\}$ be the line passing

through c_j and c_ℓ. Then

$$\begin{aligned}
\mathbb{E}[\widehat{\mathsf{pDisk}}_{j\ell,R}(t)] &= \mathbb{E}\bigg(\frac{1}{nh}\sum_{i=1}^{n} K\bigg(\frac{\Pi_{j\ell}(X_i) - c_j - t(c_\ell - c_j)}{h}\bigg) I\big(||X_i - \Pi_{j\ell}(X_i)|| \le R\big)\bigg) \\
&= \frac{1}{h}\int_{x\in\mathcal{X}} K\bigg(\frac{\Pi_{j\ell}(x) - c_j - t(c_\ell - c_j)}{h}\bigg) I(||x - \Pi_{j\ell}(x)|| \le R) p(x)\mu_d(dx) \\
&= \frac{1}{h}\int_{\mathbb{L}_{j\ell}} K\bigg(\frac{z - c_j - t(c_\ell - c_j)}{h}\bigg)\bigg(\int_{Disk(z,R,c_\ell - c_j)} p(y)\mu_{d-1}(dy)\bigg)dz \\
&= \frac{1}{h}\int_{\mathbb{L}_{j\ell}} K\bigg(\frac{z - c_j - t(c_\ell - c_j)}{h}\bigg) q_{c_\ell - c_j,R}(z)dz \\
&= \frac{||c_j - c_\ell||}{h}\int_{\mathbb{L}_{j\ell}} K\bigg(\frac{(s-t)||c_j - c_\ell||}{h}\bigg) q_{c_\ell - c_j,R}(c_j - s(c_\ell - c_j))ds
\end{aligned}$$

where for the third equality we split the integration with respect to $z \in \mathbb{L}_{j\ell}$ and the integration with respect to $y \in Disk(z, R, c_\ell - c_j)$, and for the last equality we set $z = c_j - s(c_\ell - c_j)$ and utilized the symmetry of the kernel function K.

Then by another change of variable that $u = \frac{(s-t)||c_\ell - c_j||}{h}$ and Taylor expansion, we have

$$\begin{aligned}
\mathbb{E}[\widehat{\mathsf{pDisk}}_{j\ell,R}(t)] &= \int K(u) q_{c_\ell - c_j,R}\bigg(c_j - t(c_\ell - c_j) - hu\frac{c_\ell - c_j}{||c_j - c_\ell||}\bigg)du \\
&= \int K(u)\bigg(q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j)) + hu\cdot g_1 + \frac{1}{2}h^2u^2\cdot g_2 + O(h^2)\bigg)du
\end{aligned}$$

where

$$\begin{aligned}
g_1 &= \bigg(\frac{c_\ell - c_j}{||c_j - c_\ell||}\bigg)^T \cdot \nabla q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j)) \\
g_2 &= \bigg(\frac{c_\ell - c_j}{||c_j - c_\ell||}\bigg)^T \cdot \nabla\nabla q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j))\bigg(\frac{c_\ell - c_j}{||c_j - c_\ell||}\bigg)
\end{aligned}$$

When $R \to 0$, assumption (D2) implies that there is a constant C_{d-1} that

$$2p_{\min}C_{d-1}R^{d-1} \le \mathsf{pDisk}_{j\ell,R}(t) \le 2p_{\max}C_{d-1}R^{d-1} = O(R^{d-1}) \tag{A.0.21}$$

where $0 < p_{\min} \le \inf_{x\in\mathcal{X}} p(x)$, $\sup_{x\in\mathcal{X}} p(x) \le p_{\max} < \infty$. Since the disk density is shrinking at rate $O(R^{d-1})$, one can easily verify that the gradient and Hessian of the disk density

function are also at rate $O(R^{d-1})$. Namely,

$$g_1 = O(R^{d-1}), \qquad g_2 = O(R^{d-1}).$$

By assumption (**D2**) we have g_1 and g_2 to be bounded and therefore Thus,

$$\begin{aligned}
\mathbb{E}[\widehat{\mathsf{pDisk}}_{j\ell,R}(t)] &= q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j)) \int K(u)du + h\left[\int uK(u)du\right] \cdot g_1 \\
&\quad + \frac{1}{2}h^2\left[\int u^2 K(u)du\right] \cdot g_2 + O(h^2 R^{d-1}) \\
&= q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j)) + O(h^2 R^{d-1}) \\
&= \mathsf{pDisk}_{j\ell,R}(t) + O(h^2 R^{d-1}),
\end{aligned}$$

where for the second equality we used, by assumption (K)

$$\int K(u)du = 1, \int uK(u)du = 0, \int u^2 K(u)du < \infty$$

so we conclude that $|\mathbb{E}[\widehat{\mathsf{pDisk}}_{j\ell,R}(t)] - \mathsf{pDisk}_{j\ell,R}(t)| = O(h^2 R^{d-1})$

(II): Stochastic variation.

$$\begin{aligned}
Var(\widehat{\mathsf{pDisk}}_{j\ell,R}(t)) &= Var\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - c_j - t(c_\ell - c_j)}{h}\right) I(||X_i - \Pi_{j\ell}(X_i) \le R)\right) \\
&\le \frac{1}{nh^2}\mathbb{E}\left[K^2\left(\frac{\Pi_{j\ell}(X_i) - c_j - t(c_\ell - c_j)}{h}\right) I(||X_i - \Pi_{j\ell}(X_i) \le R)\right] \\
&= \frac{1}{nh}\int K^2(u)\left(q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j)) + hu \cdot g_1 + O(h^2)\right)du \\
&= O\left(\frac{1}{nh}\right)
\end{aligned}$$

by the same analysis procedure as for Face Density and the assumptions (D1), (K1).

Now, by assumption (D2), the face density $q_{c_\ell - c_j,R}(c_j - t(c_\ell - c_j)) = O(R^{d-1})$, which leads to

$$Var(\widehat{\mathsf{pDisk}}_{j\ell,R}(t)) = O\left(\frac{R^{d-1}}{nh}\right).$$

Therefore,

$$|\widehat{\mathsf{pDisk}}_{j\ell,R}(t) - \mathbb{E}[\widehat{\mathsf{pDisk}}_{j\ell,R}(t)]| = O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right)$$

and

$$|\widehat{\mathsf{pDisk}}_{j\ell,R}(t) - \mathsf{pDisk}_{j\ell,R}(t)| = O(h^2 R^{d-1}) + O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right). \tag{A.0.22}$$

(I): Change in position. Finally, we bound the term

$$(I) = \hat{\mathsf{pDisk}}_{j\ell,R}(\hat{t}^*) - \hat{\mathsf{pDisk}}_{j\ell,R}(t^*).$$

Note that the minimizer $\hat{t}^*$ satisfies the gradient condition

$$\hat{\mathsf{pDisk}}'_{j\ell,R}(\hat{t}^*) = 0.$$

By a simple Taylor expansion at $\hat{t}^*$, we obtain

$$\begin{aligned}
(I) &= -(\hat{\mathsf{pDisk}}_{j\ell,R}(t^*) - \hat{\mathsf{pDisk}}_{j\ell,R}(\hat{t}^*)) \\
&= -(t^* - \hat{t}^*)\underbrace{\hat{\mathsf{pDisk}}'_{j\ell,R}(\hat{t}^*)}_{=0} - \frac{1}{2}(t^* - \hat{t}^*)^2 \hat{\mathsf{pDisk}}''_{j\ell,R}(\hat{t}^*) + O(|t^* - \hat{t}^*|^3) \\
&= O(|t^* - \hat{t}^*|^2).
\end{aligned}$$

Thus, we only need to derive the rate of $t^* - \hat{t}^*$.

Now by the fact that t^* solves the population gradient condition $\mathsf{pDisk}'_{j\ell,R}(t^*) = 0$, we have

$$\begin{aligned}
\hat{\mathsf{pDisk}}'_{j\ell,R}(t^*) - \mathsf{pDisk}'_{j\ell,R}(t^*) &= \hat{\mathsf{pDisk}}'_{j\ell,R}(t^*) - \hat{\mathsf{pDisk}}'_{j\ell,R}(\hat{t}^*) \\
&= \hat{\mathsf{pDisk}}''_{j\ell,R}(t^*)(t^* - \hat{t}^*) + O(|t^* - \hat{t}^*|^2).
\end{aligned}$$

Because $\hat{\mathsf{pDisk}}''_{j\ell,R}(t^*) \xrightarrow{P} \mathsf{pDisk}''_{j\ell,R}(t^*)$ from the analysis of terms (II) and (III), we conclude

that

$$\hat{t}^* - t^* = O(\hat{\mathsf{pDisk}}'_{j\ell,R}(t^*) - \mathsf{pDisk}'_{j\ell,R}(t^*)) = O(h^2 R^{d-1}) + O_P\left(\sqrt{\frac{R^{d-1}}{nh^3}}\right).$$

Note that the above rate analysis follows from the same analysis as term (II) and (III) except that we are using gradient rather than density.

As a result, we conclude that

$$(I) = O(|t^* - \hat{t}^*|^2) = O(h^4 R^{2d-2}) + O_P\left(\frac{R^{d-1}}{nh^3}\right).$$

Combining together, we have

$$\begin{aligned}
|\hat{S}_{j\ell}^{TD} - S_{j\ell}^{TD}| &= (I) + (II) + (III) \\
&= O(h^4 R^{2d-2}) + O_p\left(\frac{R^{d-1}}{nh^3}\right) + O(h^2 R^{d-1}) + O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right) \\
&= O(h^2 R^{d-1}) + O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right) + O_p\left(\frac{R^{d-1}}{nh^3}\right).
\end{aligned}$$

Using the fact that $S_{j\ell}^{TD} \geq 2p_{\min} C_{d-1} R^{d-1}$ from equation (A.0.21), we conclude that

$$\left|\frac{\hat{S}_{j\ell}^{TD}}{S_{j\ell}^{TD}} - 1\right| = O(h^2) + O_p\left(\sqrt{\frac{1}{nhR^{d-1}}}\right) + O_p\left(\frac{1}{nh^3}\right),$$

which completes the proof.

□

E Choice of Linkage

In this section, we use different simulations to investigate the effect of different linkage criteria under our skeleton clustering framework. We start with the same Yinyang data to illustrate how different linkages cope with well-separated clusters in Appendix E. Next, we add noisy observations to the Yinyang data and make the comparison again in Appendix

E. Moreover, we repeat this comparison using different simulation scenarios when there are overlapping clusters; the comparisons in Appendix E, E, E, and E.

Except for the linkage criterion, all other procedures are the same with the following settings: we use k-means clustering with $k = \sqrt{n}$ to find knots and use the Voronoi density as the density-aided similarity measure. We vary the total number of final clusters from 1 to 40 and compare the adjusted Rand Index (ARI) to the actual cluster label. The entire procedure is repeated 100 times for the comprehensive comparison of various linkage methods from the `hclust` function in R. The medium performances of the resulting clusterings are summarized in Table A.1. For datasets without noisy points, we only present the medium ARI at the true number of clusters, while for data with noisy points, we show the best medium ARI across different S and record the corresponding S in the bracket. The best linkages for each data scenario are in bold.

From Table A.1, either average linkage or single linkage achieve the best and most reliable performance. Thus, we recommend using one of them as the linkage criterion. We include a more detailed analysis of each dataset in the following subsections and we plot the 5th percentile, medium, and 95th percentile of the adjusted Rand index for single linkage, average linkage, and complete linkage. Plots comparing all the linkages on the different datasets are deferred to Appendix E.

Yinyang Data

We begin by comparing the different linkage methods on the Yinyang datasets with different numbers of noisy dimensions (same data as in Section 2.5.1). The results are shown

	average	centroid	complete	mcquitty	median	minimax	single	Ward
Yinyang,d=10	**1.000**	0.119	-0.017	**1.000**	0.111	0.027	**1.000**	**1.000**
Yinyang,d=100	**1.000**	0.098	-0.008	**1.000**	0.097	0.055	**1.000**	**1.000**
Yinyang,d=500	0.560	0.074	-0.028	0.587	0.054	0.062	**1.000**	0.526
Yinyang,d=10000	0.533	0.107	-0.029	0.555	0.021	0.106	**1.000**	0.456
MixMickey,d=10	**0.731**	-0.005	0.017	0.380	0.007	0.010	-0.004	0.194
MixMickey,d=100	**0.740**	-0.005	0.005	0.341	0.010	0.043	-0.001	0.129
MixMickey,d=500	**0.710**	-0.003	0.003	0.356	0.013	-0.003	-0.004	0.180
MixMickey,d=10000	**0.692**	-0.006	-0.014	0.297	0.011	-0.045	-0.006	0.217
MixStar,d=10	**0.763**	0.0001	0.00532	0.510	0.001	0.0488	0.0001	0.424
MixStar,d=100	**0.763**	0.0001	0.007	0.540	0.001	0.0503	0.0001	0.415
MixStar,d=500	**0.762**	0.0001	0.004	0.537	0.001	0.039	0.0001	0.444
MixStar,d=1000	**0.721**	0.0001	0.005	0.533	0.001	0.050	0.0001	0.418
NoisyYinyang,d=10	0.875(S=4)	0.182(4)	0.102(35)	0.397(3)	0.180(13)	0.132(28)	**0.968(16)**	0.535(4)
NoisyYinyang,d=100	0.875(S=3)	0.182(6)	0.103(35)	0.798(2)	0.242(20)	0.135(23)	**0.999(14)**	0.695(4)
NoisyYinyang,d=500	0.875(S=3)	0.121(10)	0.107(28)	0.783(3)	0.209(20)	0.143(21)	**0.999(11)**	0.539(4)
NoisyYinyang,d=1000	0.875(S=3)	0.176(7)	0.111(27)	0.875(3)	0.193(28)	0.149(19)	**0.998(10)**	0.372(5)
NoisyMixMickey,d=10	**0.686(S=5)**	0.119(34)	0.093(29)	0.413(6)	0.077(39)	0.157(15)	0.501(31)	0.235(5)
NoisyMixMickey,d=100	**0.700(S=5)**	0.141(37)	0.094(29)	0.358(6)	0.095(39)	0.158(16)	0.506(31)	0.221(6)
NoisyMixMickey,d=500	**0.697(S=5)**	0.095(37)	0.091(30)	0.359(7)	0.098(39)	0.155(17)	0.502(31)	0.232(6)
NoisyMixMickey,d=1000	**0.692(S=5)**	0.122(36)	0.091(29)	0.386(6)	0.104(39)	0.153(17)	0.497(31)	0.241(5)
NoisyMixStar,d=10	**0.783(S=10)**	0.109(40)	0.221(30)	0.613(11)	0.140(40)	0.330(17)	0.623(31)	0.476(4)
NoisyMixStar,d=100	**0.779(S=9)**	0.129(40)	0.220(28)	0.627(10)	0.171(40)	0.334(18)	0.667(30)	0.487(4)
NoisyMixStar,d=500	**0.788(S=8)**	0.115(40)	0.220(29)	0.604(9)	0.158(40)	0.328(16)	0.651(30)	0.498(4)
NoisyMixStar,d=1000	**0.791(S=9)**	0.113(40)	0.219(29)	0.599(9)	0.150(40)	0.333(15)	0.621(30)	0.476(4)

Table A.1: Comparison of the linkage methods across different simulated datasets. All reported values are mediums of 100 random simulations. For datasets without noisy points, the performance at the true number of clusters is reported ($S = 5$ for Yinyang, $S = 3$ for Mix Mickey and Mix Star). For datasets with noisy points, we report the best performance across different numbers of clusters and include the number of clusters at which the max is achieved in the bracket.

in Figure A.3. For each dimension ($d = 10, 100, 500, 1000$), the medium adjusted Rand index of the 100 runs is plotted with the solid line, and the 5 percentile to 95 percentile range is depicted with a lighter color band. The true number of clusters $S = 5$ is shown as the red dotted vertical line.

We observe that single linkage and average linkage have similar performance for lower dimensions $d = 10$ and $d = 100$, with medium performance achieving nearly perfect clustering at the true number of clusters. However, the clustering results returned by single linkage are more stable, having a narrower band while the band of average linkage is much wider. For cases with higher dimensions $d = 500, 1000$, we observe single linkage still stably achieves nearly perfect clustering at $k = 5$, which corroborates our results in Section 2.5.1, but

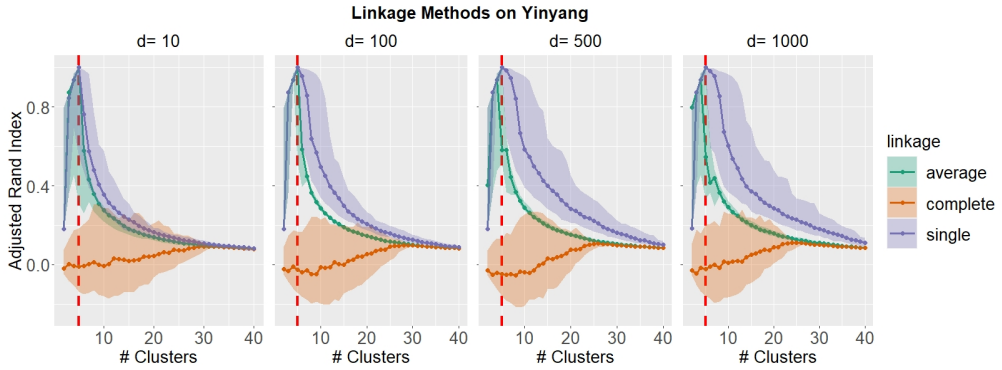


Figure A.3: Clustering results with different linkage methods across different numbers of final clusters on Yinyang data. Line for medium and band from 5th percentile to 95th percentile. The vertical red dashed line indicates the true number of 5 clusters.

average linkage fails to get such good clustering performance when dimensions get higher. Therefore, single linkage has superior performance on the Yinyang data, arguably because the true manifold of the data has well-separated clusters that single linkage is suitable for separation.

Noisy Yinyang Data

To create additional noise, we added 640 (20% of the number of signals) noisy points to the Yinyang dataset, sampled uniformly from $[-3, 3] \times [-3, 3]$ in the first two dimensions, with random Gaussian variables in the other dimensions the same way we generated Yinyang data. The adjusted Rand indexes are calculated only for the true signal data points and the results are shown in Figure A.4.

Average linkage can achieve slightly better performance than single linkage around the true number of clusters $S = 5$ for lower dimensions ($d = 10, 100$), but fails to achieve satisfactory clustering performance when dimensionality gets higher ($d = 500, 1000$). The performance of single linkage improves with S being slightly larger than the actual number

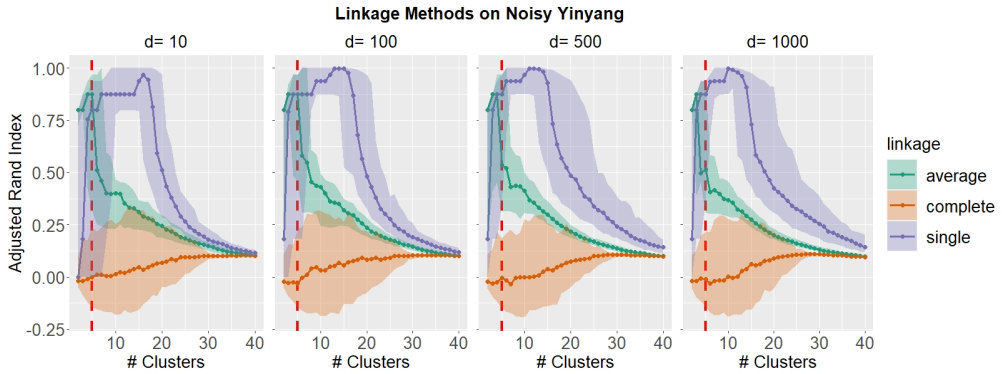


Figure A.4: Clustering results with different linkage methods across different numbers of final clusters on Yinyang data with noisy points. The vertical red dashed line indicates the true number of 5 clusters.

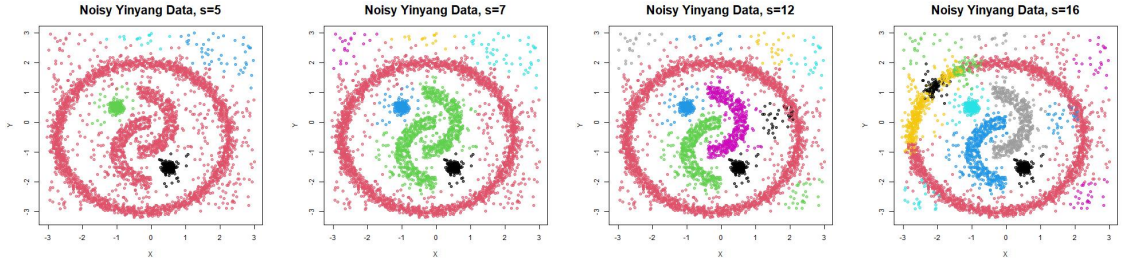


Figure A.5: The clustering results with single linkage in skeleton clustering with a different number of final clusters S for Noisy Yinyang data, $d = 1000$.

5 and can yield nearly perfect clusters with S being around 15 to 20. A further investigation reveals that large S will group noisy points into separate clusters and hence improve the clustering performance; see Figure A.5. This suggests that our framework may be used for anomaly detection.

Mix Mickey Data

The well-separated structures in the Yingyang data may provide advantages to the single linkage. To investigate the effect of linkage criteria on the overlapping clusters, we consider a three-Gaussian mixture model in a 2D case that we call the Mix Mickey data. The large cluster is centered at $(0, 0)$ with the covariance matrix being a diagonal matrix of 2 and

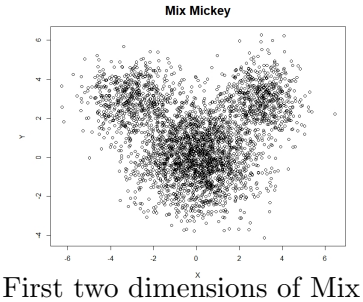


Figure A.6: First two dimensions of Mix Mickey data.

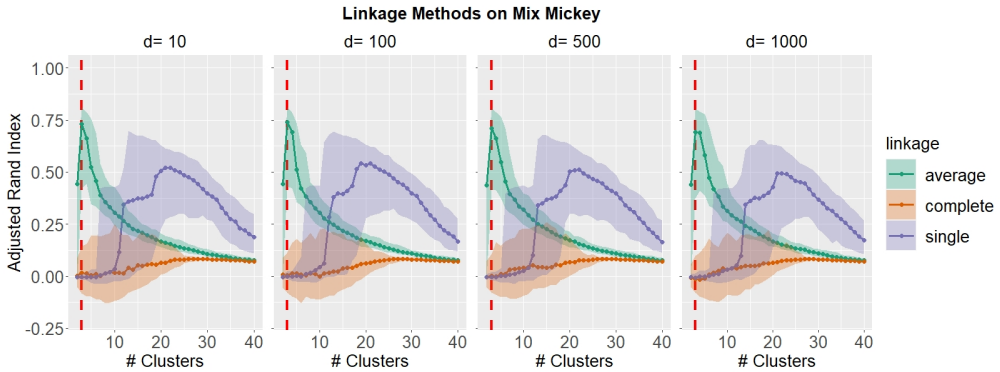


Figure A.7: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data. The vertical red dashed line indicates the true number of 3 clusters.

has 2000 points. The two smaller clusters are centered at $(3, 3)$ and $(-3, 3)$ respectively, and both have a covariance matrix being a diagonal matrix of 1, and each has 600 points. Random Gaussian variables are added to make the data $d = 10, 100, 500, 1000$ dimensions in the same way we generate the Yinyang data. Figure A.6 presents a scatter plot of the first two dimensions; the three clusters have a substantial amount of overlap so that it is difficult for clustering methods to separate them into three distinct clusters. The results under the same linkages analysis pipeline are shown in Figure A.7.

Remark 13. GMM can be favored in this data example but is unstable and cannot work with too many noisy dimensions. We present some comparisons including GMM in Appendix F.

We observe that average linkage gives good performance at $S = 3$ (the true number of

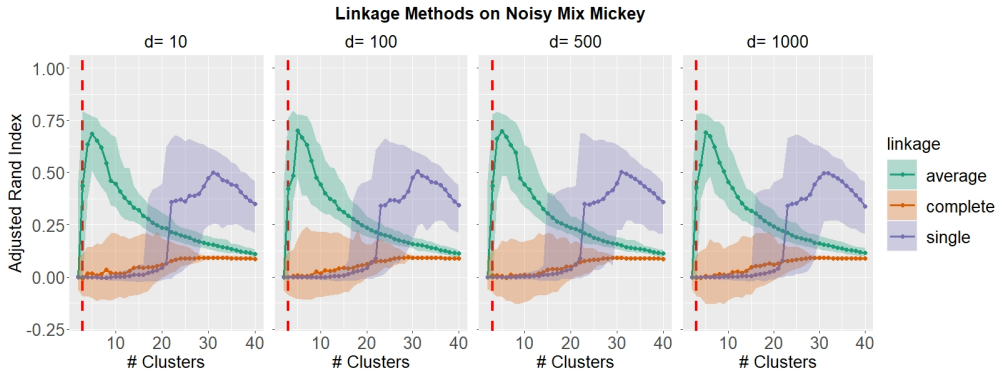


Figure A.8: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data with Noise. The vertical red dashed line indicates the true number of 3 clusters.

clusters) and single linkage fails to give a satisfying performance under this scenario, giving non-informative clusters at low S (only extracting small clusters) and too fragmented clusters at high S. The average linkage is a criterion that tends to create spherical clusters with similar sizes and hence is better suited for this simulated data. Thus, our experiment shows that, for data containing overlapping clusters with roughly spherical shapes, the average linkage criterion in the knots segmentation step is preferred.

Noisy Mix Mickey Data

In this section, we experiment with a scenario with both overlapping clusters and noisy observations. We added 640 (20% of the number of signals) noisy points to the Mix Mickey dataset, sampled uniformly from $[-6, 6] \times [-5, 6]$ in the first two dimensions, with random Gaussian noises in the other dimensions the same way as in Mix Mickey data. The adjusted Rand indices are measured only on the true signal data points with the results shown in Figure A.8.

Average linkage still gives good performance and is superior to the single linkage, which

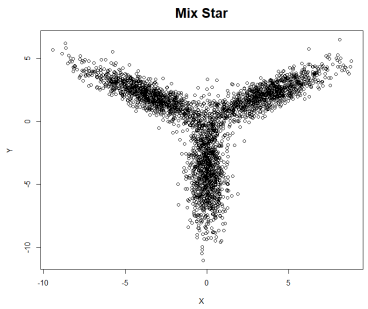


Figure A.9: First two dimensions of the Mix Star data.

fails to give reasonable clustering performance under a decent number of clusters. Notably, average linkage achieves the best performance with the S being slightly higher than 3 due to the introduction of noisy data points.

Mix Star Data

We present here the Mix Star dataset, another 3-GMM data but with a more elongated shape as illustrated in Figure A.9. The three clusters are all generated as 2D Gaussian with 5 and 0.3 on the diagonal of the covariance matrix with respective centers at $(4, 0)$, $(-4, 0)$, and $(0, -4)$, and then are rotated to get a star-like shape. Each cluster has 1000 sample points, and random Gaussian variables with standard deviation 0.1 are added to make the data $d = 10, 100, 500, 1000$ dimensions. There is still a decent overlap among clusters, but each cluster is more distinct compared to Mix Mickey. We apply the same analysis pipeline as the Yinyang and Mix Mickey data and compare different linkage criteria. Figure A.10 displays the median clustering performance. Again, we see that average linkage has the best performance.

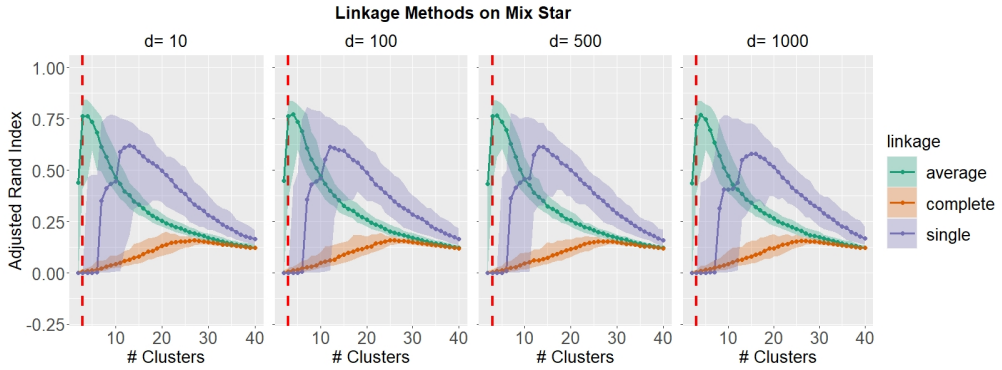


Figure A.10: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data. The vertical red dashed line indicates the true number of 3 clusters.

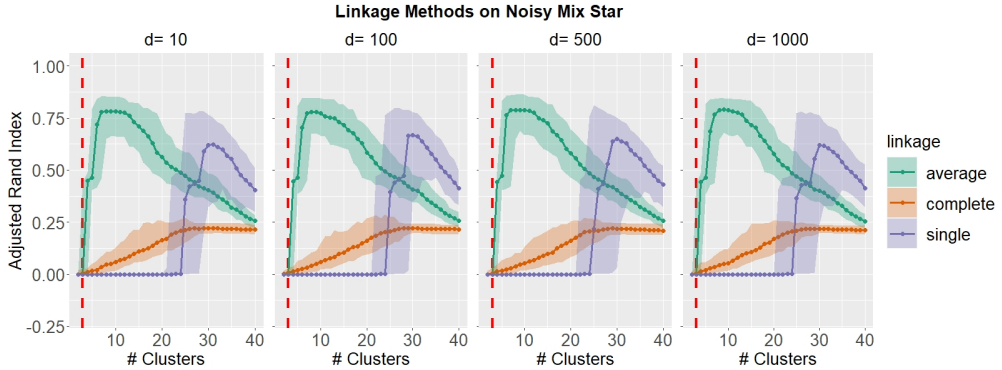


Figure A.11: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data with Noise.

Noisy Mix Star

To investigate the effect of added noises, we make the data similar to the Noisy Mix Mickey by adding 600 (20% of the number of signals) noisy points to the Mix Star dataset, sampled uniformly from $[-10, 10] \times [-10, 5]$ in the first two dimensions, with random Gaussian noises in the other dimensions generated the same way. The results of the linkage comparison results are shown in Figure A.11. Average linkage still gives the best clustering results in this scenario.

In summary, as illustrated by all the simulations in this section, our skeleton clustering

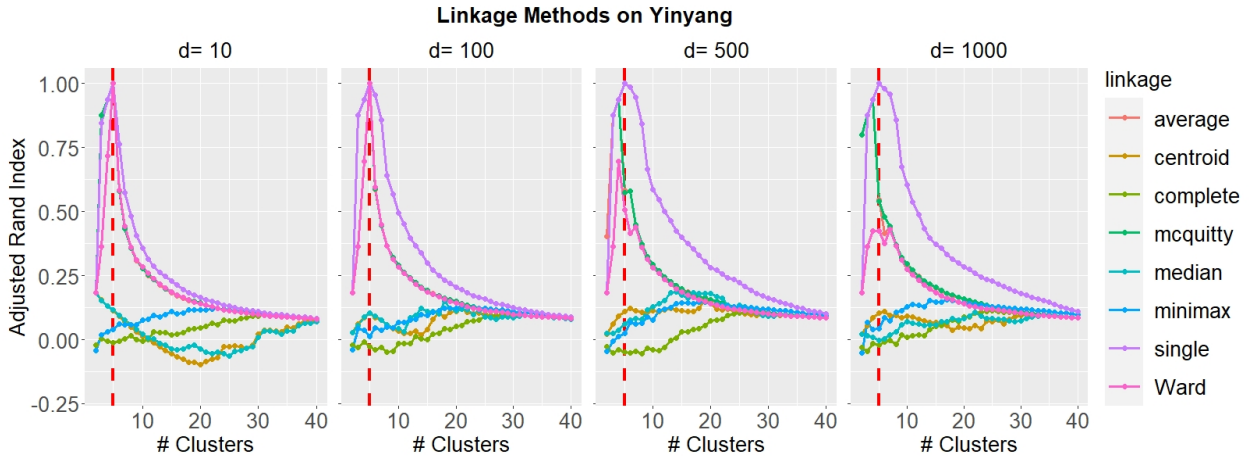


Figure A.12: Clustering results with different linkage methods across different numbers of final clusters on Yinyang Data.

framework is able to handle noisy data points by tuning the number of final clusters and can cope with overlapping clusters by choosing appropriate linkage criteria for skeleton segmentation. Broadly speaking, the appropriate choice of linkage method depends on the intrinsic geometric structure of the data and may require subject matter knowledge or exploratory analysis. Specifically, if the intrinsic clusters are well-separated, the single linkage is preferred as it gives clear cuts for disjoint components. However, if the clusters are believed to have some degree of overlapping with each cluster approximately spherically shaped, the average linkage criterion can lead to better performance.

All Linkage Comparisons

Figures A.12 and A.13 display the median clustering performances of all linkage methods under different numbers of clusters using Yinyang and noisy Yinyang data. We see that average linkage and single linkage dominate all other methods, while single linkage is superior in those two cases.

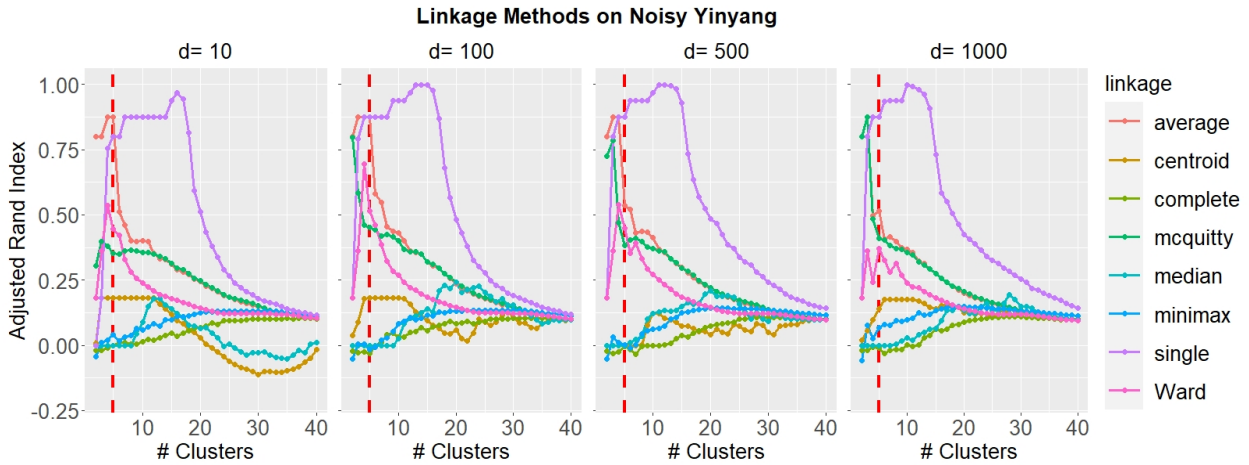


Figure A.13: Clustering results with different linkage methods across different numbers of final clusters on Noisy Yinyang Data.

Figures A.14 and A.15 present the median clustering performance under different numbers of clusters for the Mix Mickey and noisy Mix Mickey data (same setup in Section E). Similar to the case of Yinyang data, we observe that average linkage and single linkage dominate all other methods, while average linkage is superior among the two.

To further investigate what the clusters will be like in high dimensions, we present 2D scatterplot of clustering results under $S = 3$ (final number of clusters is 3) of the first two coordinates in Figure A.16. We use the data with $d = 1000$ and color the clusters using red, green, and blue. Clearly, average linkage successfully recovers the actual clusters while other methods fail to recover. Note that single linkage does not perform well because clusters overlap with each other.

Figures A.17 and A.18 present the median clustering performance under different numbers of clusters for the Mix Star and noisy Mix Star data. We observe that average linkage and single linkage dominate all other methods.

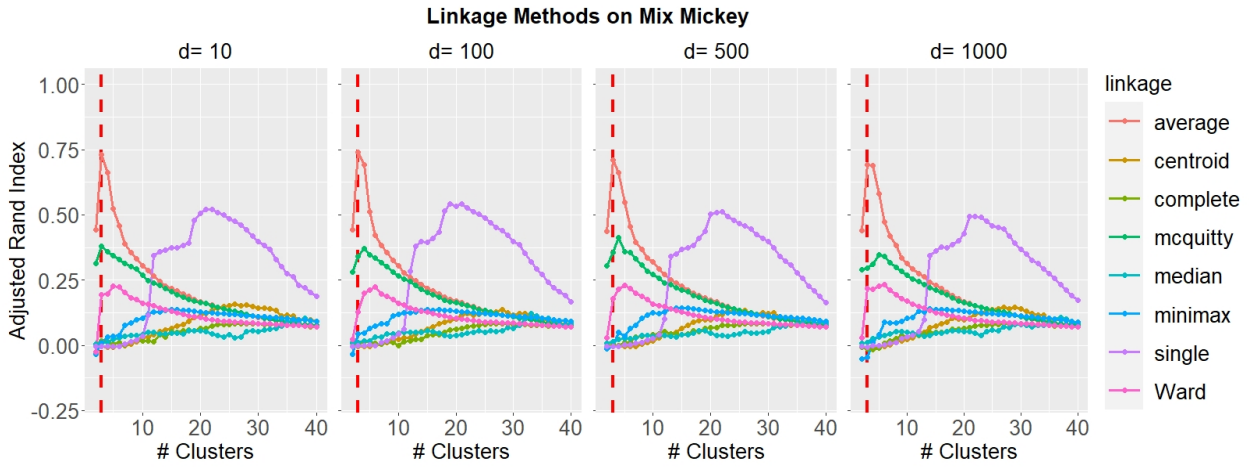


Figure A.14: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data.

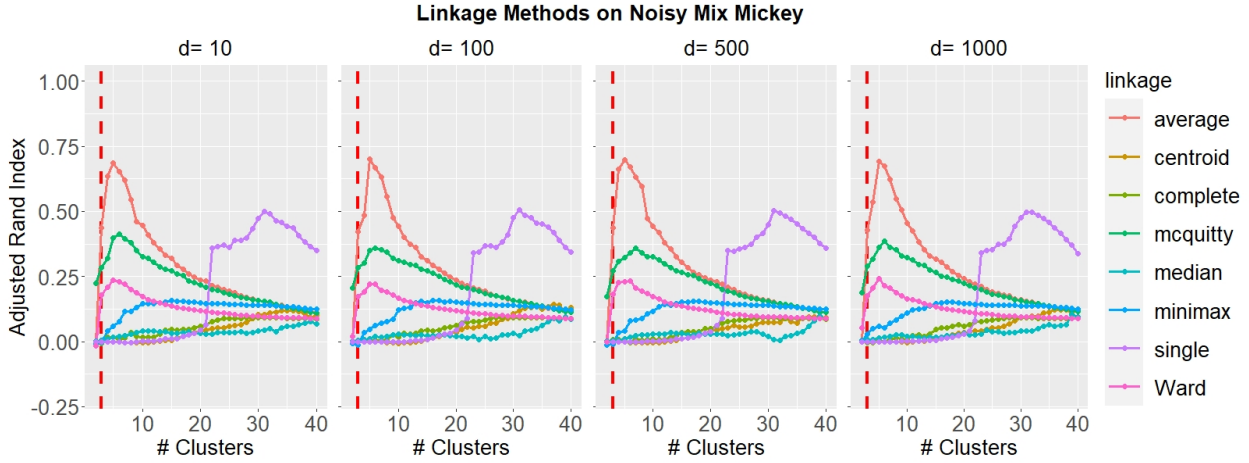


Figure A.15: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data with Noise.

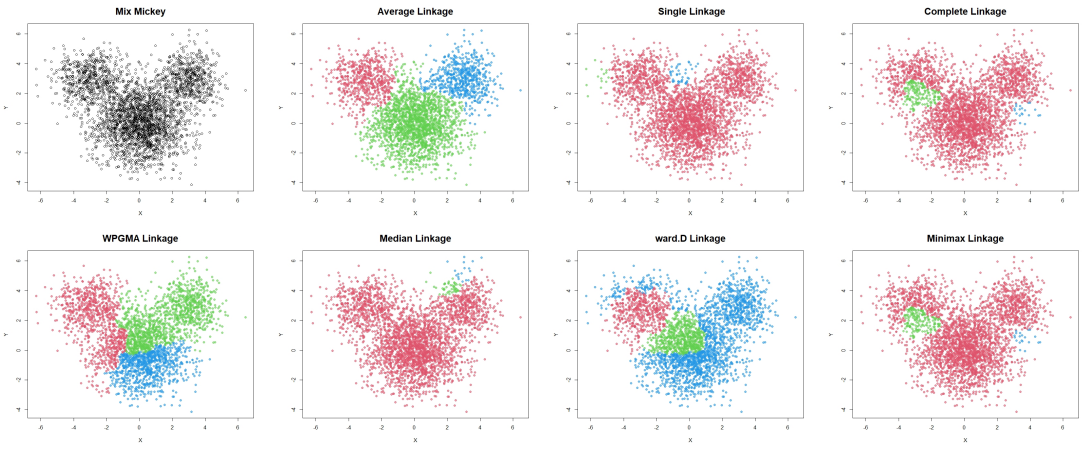


Figure A.16: Comparing linkage criteria in segmentation on the Mix Mickey data, $d = 1000$.

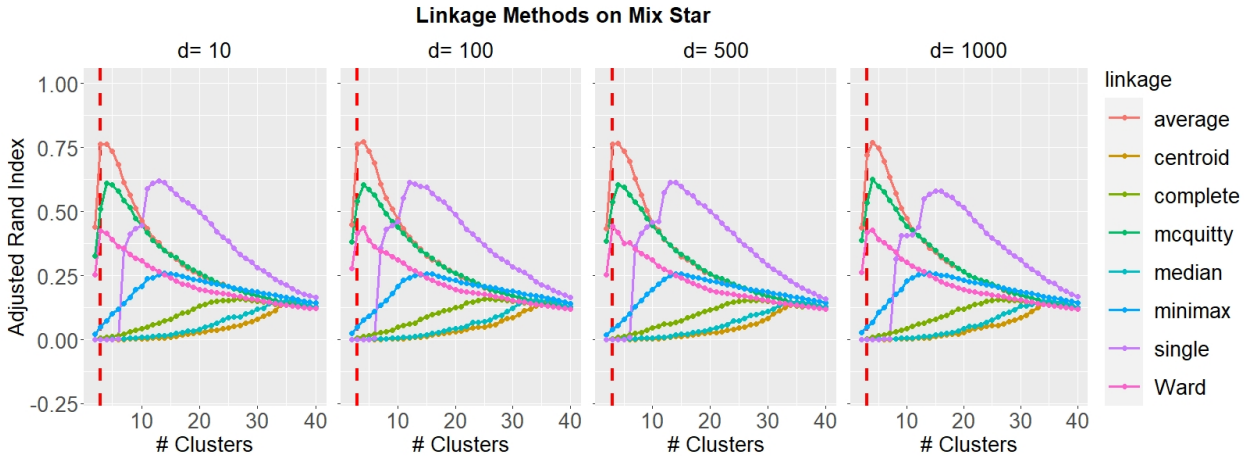


Figure A.17: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data.

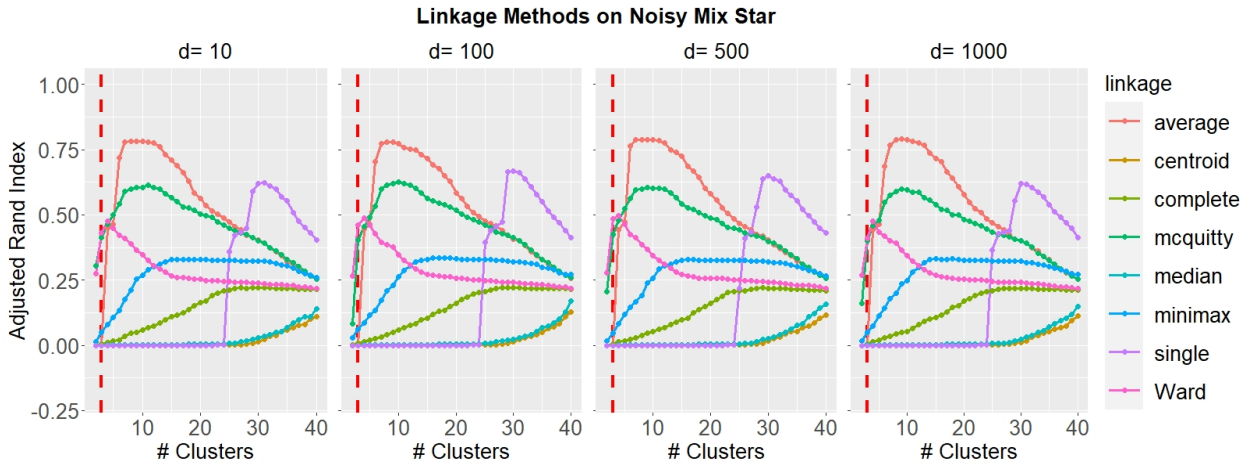


Figure A.18: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data with Noise.

F Additional Data Analysis

Performance with Different Number of Knots

We analyze how the number of knots would affect the performance of the skeleton clustering. We empirically test the effect of the number of knots, k, on the final clustering performance on Yinyang data with dimensions $10, 100, 500$ and 1000. For each dimension, we simulated the Yinyang data 100 times, and for each simulated data we carried out the default skeleton clustering procedure with single linkage and different k (other steps the same as in Section 2.5.1). Figure A.19 displays the median adjusted Rand index given by each method across different k, where the reference rule with $k = 57$ is marked by the vertical dash line. We see that as long as k is sufficiently large, skeleton clustering works well.

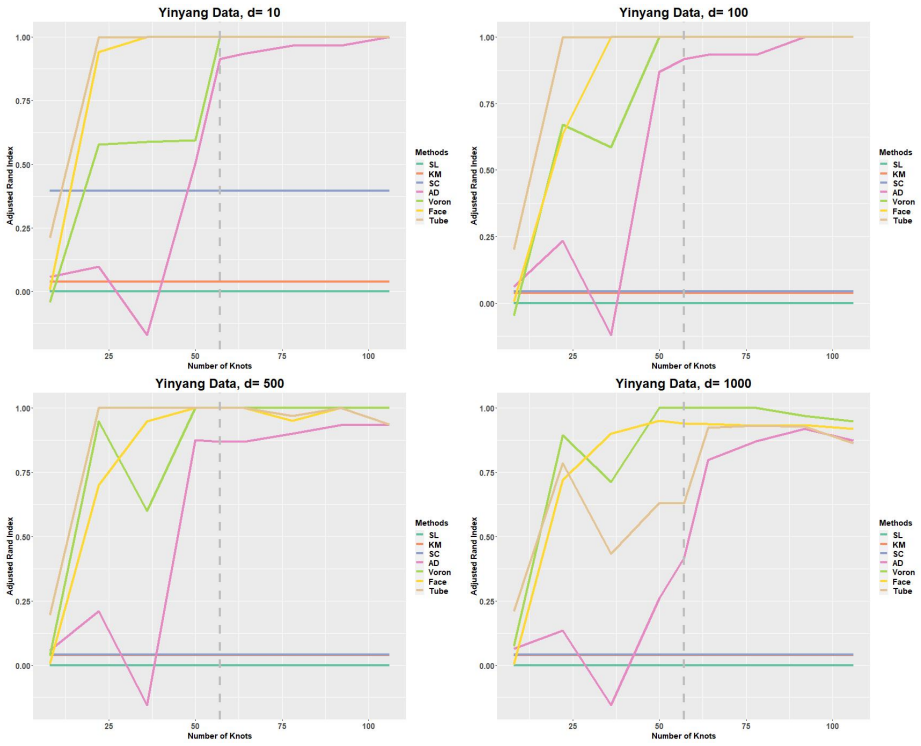


Figure A.19: Adjusted Rand indexes of different clustering methods against different numbers of knots on 100 simulated Yinyang data.

Self-Organizing Map

The Self-Organizing Map (SOM) is another popular prototype clustering method and can be used as an alternative to k-means clustering in finding knots. Thus, here we conduct a simple experiment to examine the performance of using SOM to find knots. We examine the performance using Yingyang data with $d = 10$ to $d = 1000$. The identical procedure as in Section 2.5.1 is applied except that the knots are now detected by the SOM rather than overfitting k-means. The total number of grid points in the SOM is the total number of knots we obtain and, to be comparable to k-means with $k = \sqrt{n}$ knots, we used $\lceil n^{1/4} \rceil$ breaks for each dimension of the SOM grid, giving a total of $\lceil n^{1/4} \rceil^2$ initial grid points. However, the SOM may return knots with very tiny sample sizes, on which the density-aided similarity measures cannot be calculated. Therefore, we remove knots with less than 3 data points and use the remaining ones for skeleton construction.

Figure A.20 summarizes the result. The top left panel shows the knots from the SOM (after post-processing), which are located around the main data structures and are representative to the original data as well. The dendrogram shows the cluster structure of the SOM knots using Voronoi density on one 100-dimensional Yinyang data. In the bottom row, we display the adjusted Rand indices from the clustering methods. Compared to the results of Figure 2.6, the adjusted Rand indices given by the skeleton clustering with SOM knots are similarly good when the dimension is not so high ($d = 10$ and 100). But when the data dimension becomes high ($d = 500, 1000$), knots constructed by SOM lead to worse clustering results. Therefore, overfitting k-means is favored in this work. Another limitation of SOM is

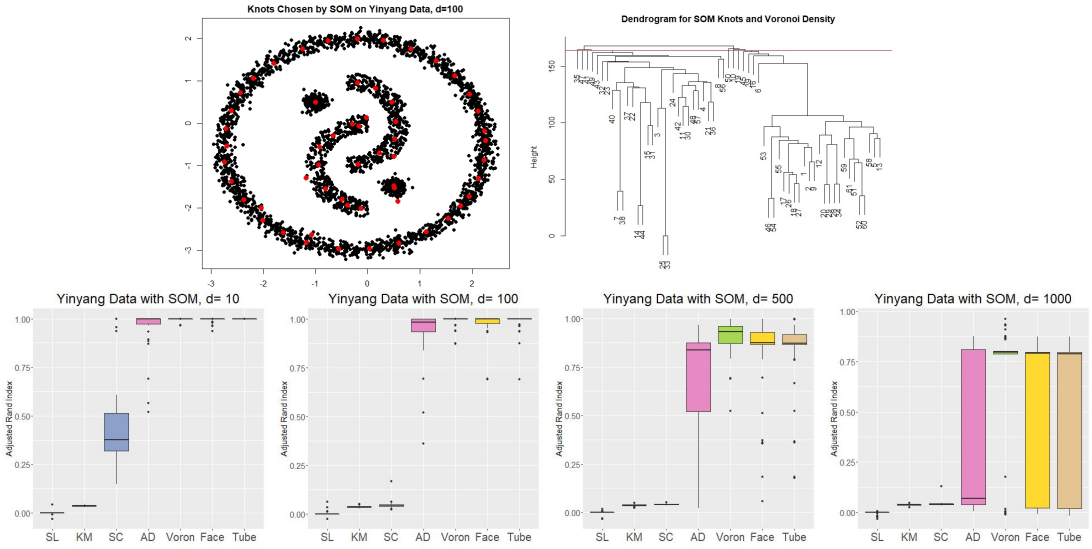


Figure A.20: Adjusted Rand indexes using SOM for knots selection on Yinyang data.

that we need to perform some post-processing to remove tiny knots; in the case of k-means, we do not need such a procedure.

Bandwidth Selection Yinyang Data

The estimations of the FD and the TD involve the use of the projected kernel density estimation, for which the type of kernel and the bandwidth need to be specified. Similar to the usual KDE, the kernel function does not affect the final performance much, so by default we use the Gaussian kernel in all of our empirical studies. It is worth noting that using the uniform kernel can save some computation since it has compact support, but empirically we find using the Gaussian kernel leads to better final clustering results. In what follows, we focus on the bandwidth selection.

It is known that the bandwidth is a pivotal parameter that can significantly affect the estimation result of a kernel density estimator. In Figure A.21, we conduct a simulation using

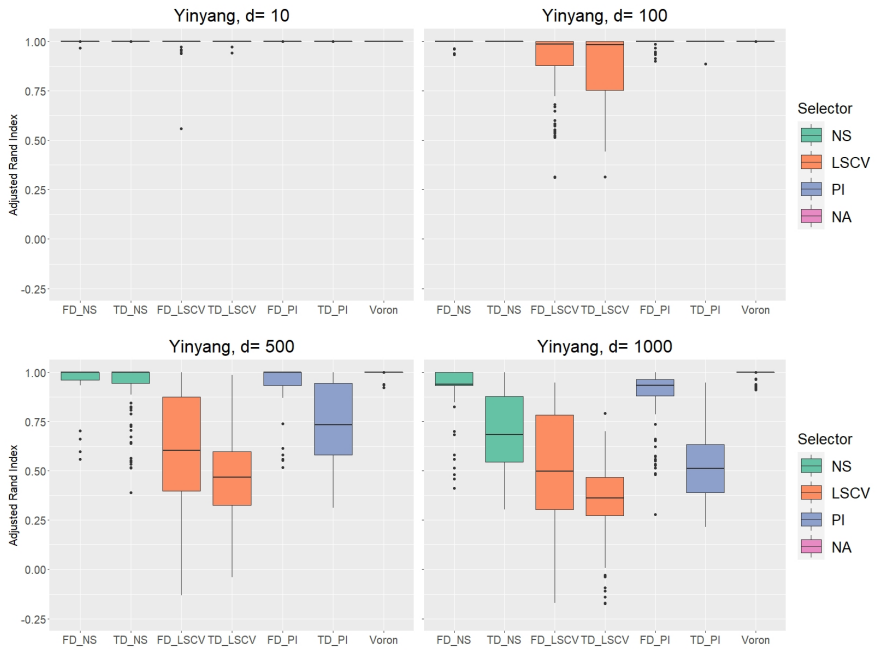


Figure A.21: Performance of skeleton clustering on Yinyang data $d = 10, 100, 500, 1000$ with Face and Tube density by different bandwidth selectors. Voronoi density result is included for comparison.

the Yinyang data with different dimensions of noisy Gaussian variables (see Section 2.5.1 for more details) and compare the performance of three common bandwidth selectors: the normal scale bandwidth (NS) (Chacón et al., 2011), the least-squared cross-validation (LSCV) (Bowman, 1984; Rudemo, 1982), and the plug-in approach (PI) (Wand and Jones, 1994). Each edge is allowed to have its own bandwidth. Voronoi density performance results are also included for comparison. We found that the NS performs reliably well while the others may have unstable performance. A similar comparison of the bandwidth selectors on another dataset is presented in Appendix F and the NS also performs relatively better than the other bandwidth selectors. As a result, we recommend using the NS as the default bandwidth selector. Additionally, since the density estimations are all 1-dimensional, in practice it is possible to examine the estimated density to assess the degree of oversmoothing

or undersmoothing and manually adjust the bandwidth.

In addition to different bandwidth selectors, we also study how the bandwidth should depend on the sample size for clustering purposes. In 1-dimensional data, the normal scale bandwidth agrees with Silverman's rule of thumb (Silverman, 1986) giving the bandwidth as $h = \frac{4}{3}^{1/5} \hat{\sigma} n_{loc}^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the sample used in the edge weight calculation, and n_{loc} the number of sample points used. Empirically we tested the clustering performance with FD and TD calculated under bandwidth with rates on n_{loc} from $-1/3$ to $-1/10$ (see Appendix F). We found that the clustering performance with FD and TD generally stays stable with varying bandwidth rates, although a larger bandwidth (slower rate than $O\big(n_{loc}^{-1/5}\big)$) may give better clustering results with TD when the dimension of the data is high.

Bandwidth Selection with Mix Mickey

We present additional results comparing different bandwidth selectors on the Mix Mickey dataset generated the same way as in Section E. We use average linkage for all the included skeleton clustering approaches. The results are presented in Figure A.22. The selectors have similar performances on this Mix Mickey dataset, but NS again seems to perform better with larger dimensions, which corroborates our default choice of using NS for bandwidth.

Performance under Different Bandwidth Rate

In this section we present empirical results on how changing the bandwidth rate affects the performance of clustering. We consider the Yinyang data in Section 2.5.1 with $d =$

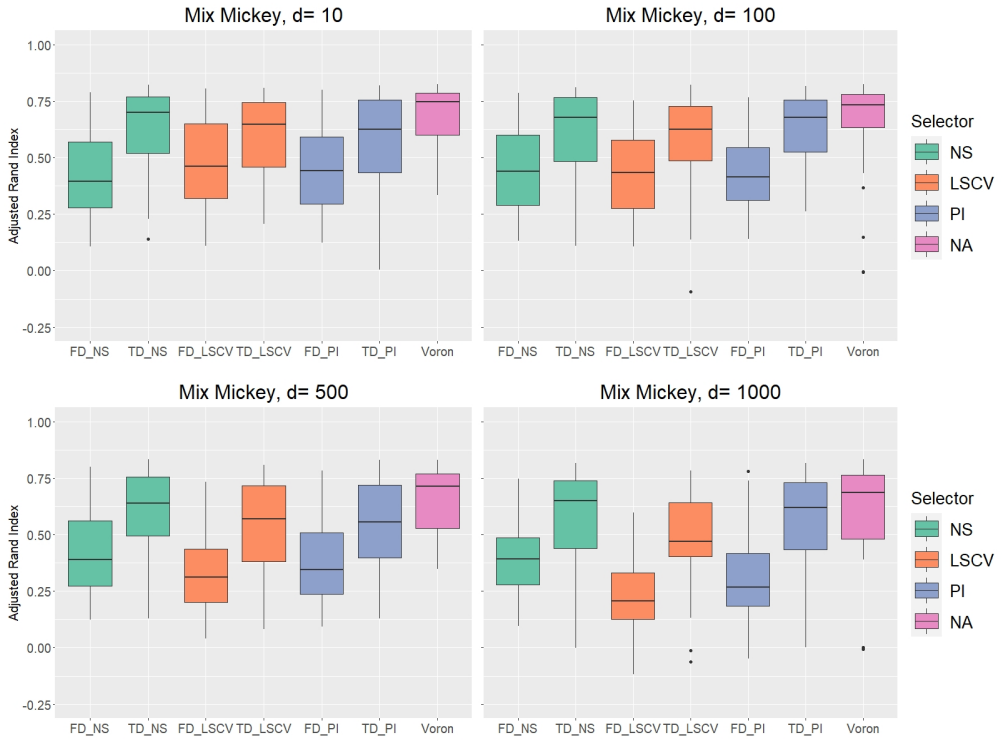


Figure A.22: Performance of skeleton clustering on Mix Mickey data $d = 10, 100, 500, 1000$ with Face and Tube density by different bandwidth selectors. Voronoi density result is included for comparison.

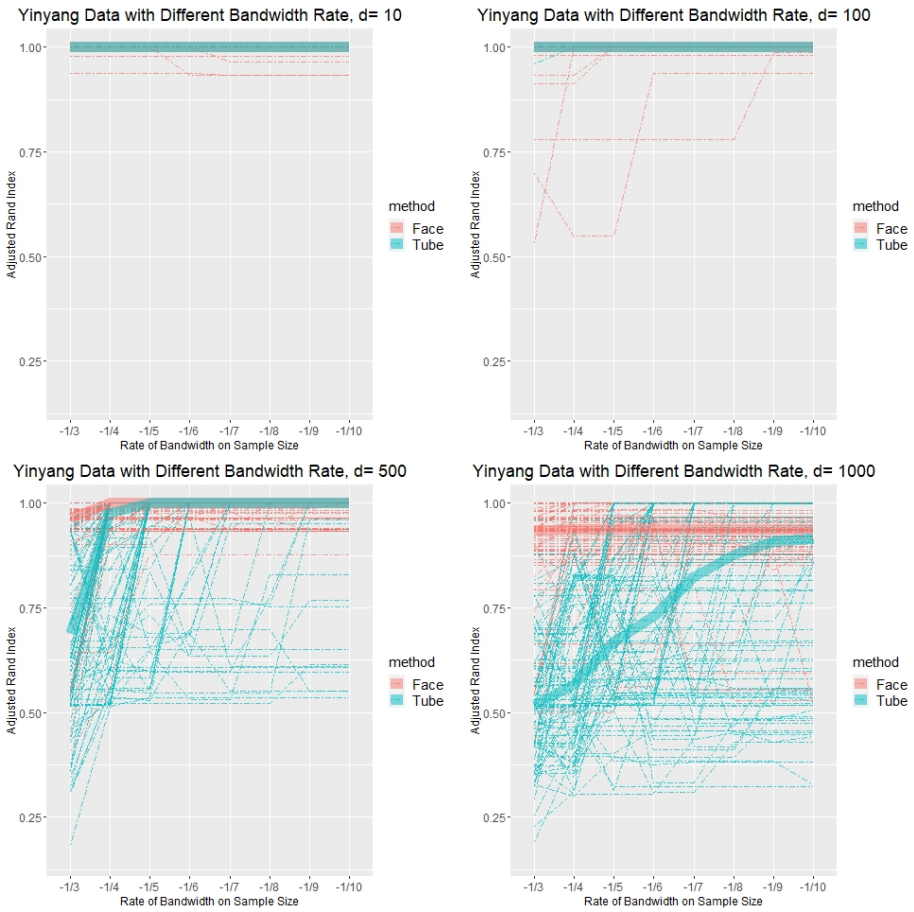


Figure A.23: Adjusted Rand indexes of skeleton clustering with Face and Tube density under different bandwidth rates on 100 simulated Yinyang datasets. The thick lines indicate the median adjusted Rand index of a given method.

$10, 100, 500, 1000$. We compare the Face and Tube density where the bandwidth is selected by Silverman's rule of thumb with different rates, ranging from $n_{loc}^{-1/3}$ to $n_{loc}^{-1/10}$. Note that the original Silverman's rule of thumb will be at rate $n_{loc}^{-1/5}$. We repeat the experiment 100 times and record the adjusted Rand index in Figure A.23.

When the dimension is low (top panels), all bandwidth within this range works well. When the dimension is large (bottom panels), a slower rate (larger bandwidth) seems to be showing a better performance for the TD. Interestingly, the face density yields a robust

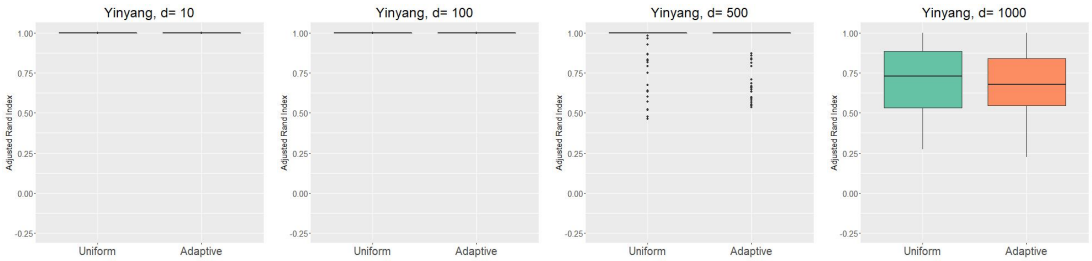


Figure A.24: Comparison of radius choices on Yinyang data with dimensions 10, 100, 500, 1000.

result across different rates of bandwidth. Note that for the TD, the theory (Theorem 9) suggests the choice at rate $h \asymp n_{loc}^{-1/5}$ is optimal for estimation in large d, the same rate may not lead to the optimal clustering performance. Figure A.23 bottom-right panel suggests that the choice $h \asymp n_{loc}^{-1/10}$ may have a better clustering performance in this case.

Adaptive Radius for Tube Density

We compare the clustering performance of Tube density when using fixed radius and that when using adaptive radius as described in Section 2.3.3. The data is the same Yinyang data in Section 2.5.1 and the results are presented in Figure A.24. The two approaches (adaptive and fixed radius) have a similar performance.

Higher Standard Deviations for Noisy Dimensions

We investigate how changing the noise level of the added noisy dimensions of our simulation examples changes the clustering performance. Here we simulate Yinyang data with different standard deviations of the added dimensions. We apply the same analysis procedure as in Section 2.5.1 is applied. The adjusted Rand indexes of each clustering method on 100 simulated datasets with under setting are presented in Figure A.25.

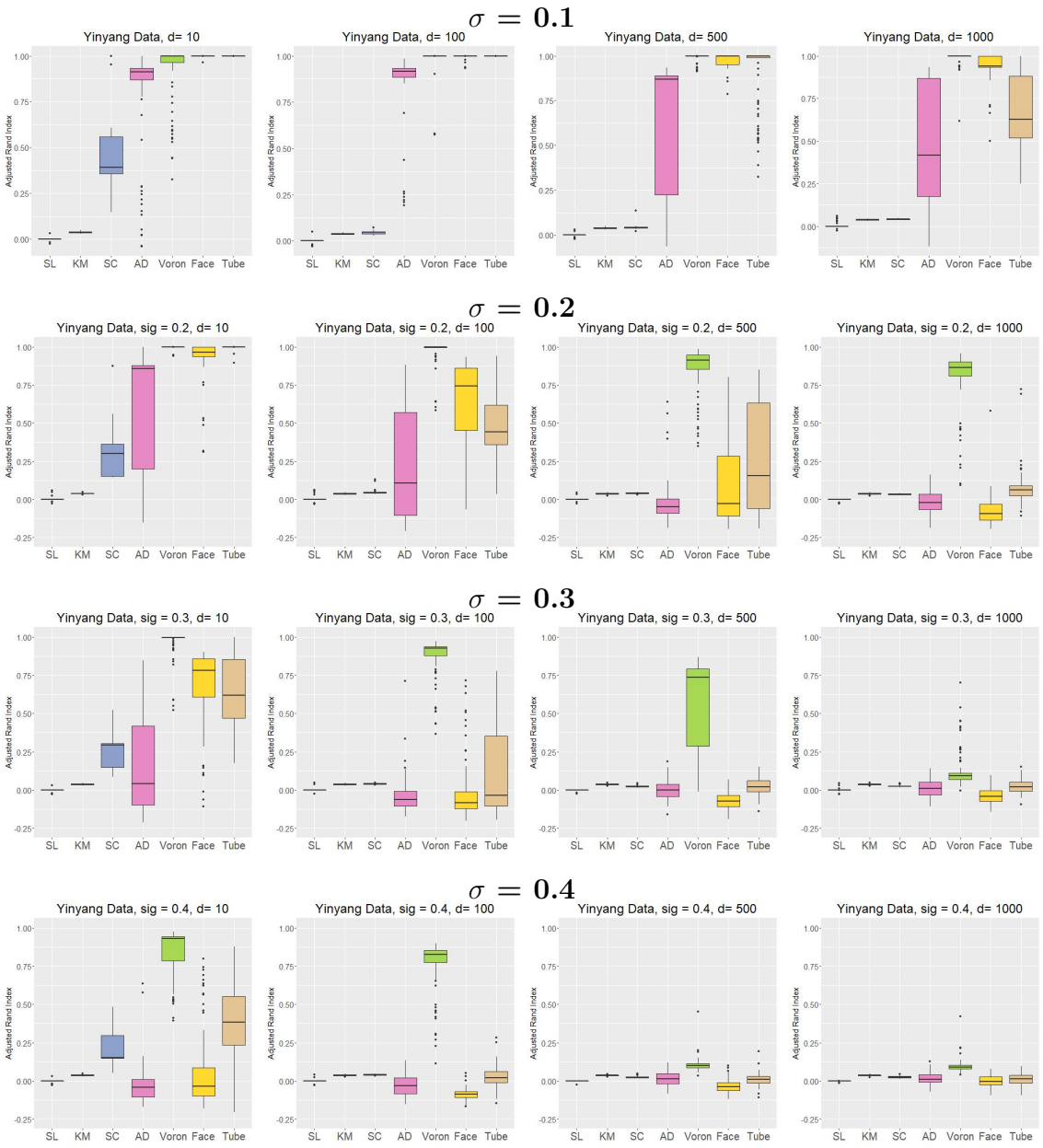


Figure A.25: Adjusted Rand index performance of clustering methods on Yinyang data with different standard deviations for added dimensions.

We observe that increasing the standard deviation of the noisy dimensions (noise level) has a stronger impact than adding more noisy variables. For example, increasing $\sigma = 0.1 \to 0.2$ scales the standard deviation by a factor of 2 (scales the variance 4 times), but the clustering performance with $\sigma = 0.2, d = 100$ is worse than that with $\sigma = 0.1, d = 500$. However, we still observe that the skeleton clustering with Voronoi density similarity measure can give good clustering performance even under the setting with $\sigma = 0.4$ and $d = 100$.

Mix Mickey with GMM

We compare the performance of Gaussian Mixture Models (GMMs) to our methods using the Mix Mickey data same as in Section E. Unfortunately, the GMM method from `clusterR` package in R cannot work with dimension 500 and 1000 case because of too many noisy dimensions, so we only compare the case of dimension 10 and 100. For the skeleton clustering, we use average linkage for the segmentation step the same as in Section E. Because this data is generated from 3-GMM and we fit the GMM with 3 components, the GMM naturally has the best performance. However, our proposed approaches may achieve comparable performance to the GMM and are capable of handling high dimensional data ($d = 500, 1000$).

Graphical Representation of GvHD Data Clusters

We visualize the skeleton structure of the clusters identified on the GvHD dataset in Section 2.6. These graph representations are generated by the `igraph` package in R. Cluster 6 only has 1 knot with 17 corresponding data points and is hence omitted in Figure A.27.

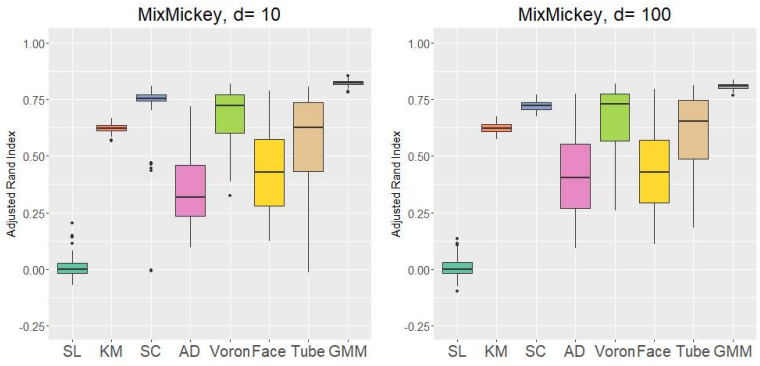
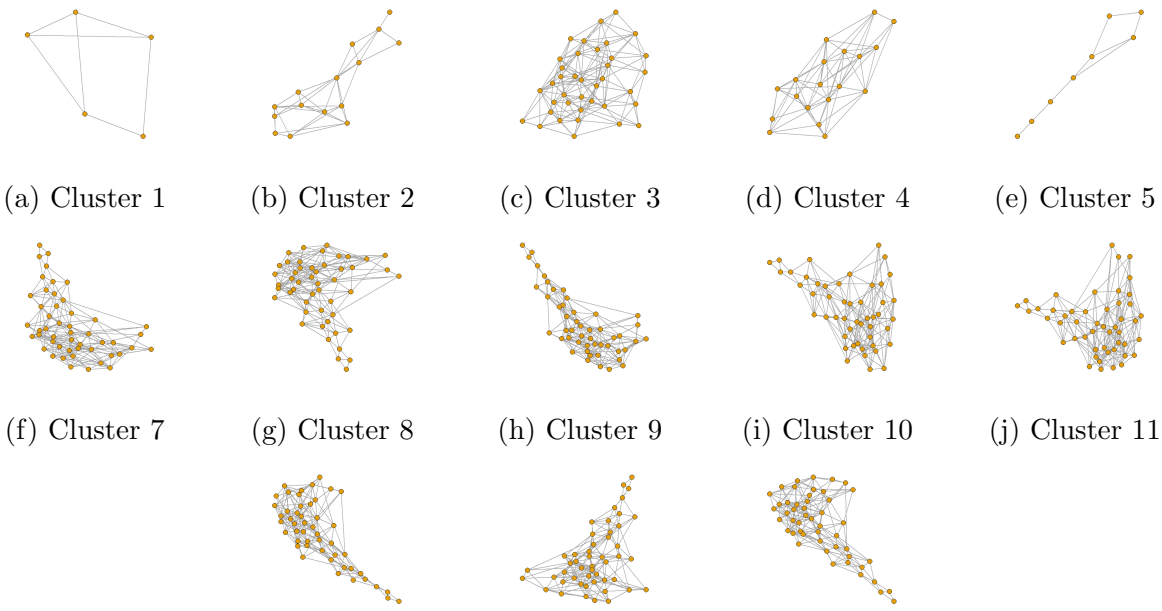


Figure A.26: Comparison of clustering methods on Mix Mickey data $d = 10, 100$ with GMM included.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3 (d) Cluster 4 (e) Cluster 5

(f) Cluster 7 (g) Cluster 8 (h) Cluster 9 (i) Cluster 10 (j) Cluster 11

(k) Cluster 12 (l) Cluster 13 (m) Cluster 14

Figure A.27: Skeleton structures of the clusters identified for the GvHD dataset in Section 2.6

We observe that most clusters display a hammer-like structure, which is non-spherical and not favorable for some classical clustering methods. Only Cluster 3 has a spherical shape in this data.

G Additional Simulated Data Examples

Manifold Mixture Data

In the Yinyang data and the Mix Mickey data experiments, the underlying components are all two-dimensional structures. Here we consider the data composed of structures of different intrinsic dimensions called the manifold mixture data. The simulated manifold mixture data, as illustrated in the left panel of Figure A.28, consists of a 2-dimensional plane with 2000 data points, a 3-dimensional Gaussian cluster with 400 data points, and an essentially 1-dimensional ring shape with 800 data points. There are a total of 3200 observations and we choose $k = [\sqrt{3200}] = 57$ knots. Similar to the other two simulations, we include Gaussian noise variables to make the data high-dimensional ($d = 10, 100, 500, 1000$) and make comparisons between the same set of clustering methods. The true number of components $S = 3$ is provided to all the clustering algorithms.

Figure A.29 summarizes the performance of each method. Traditional methods (SL, KM, and SC) do not perform well when $d > 10$ while all methods of skeleton clustering perform very well when $d \leq 500$. Notably, the skeleton clustering with VD still has a perfect performance even when $d = 1000$, whereas skeleton clustering based on other similarity measures gives satisfying results.

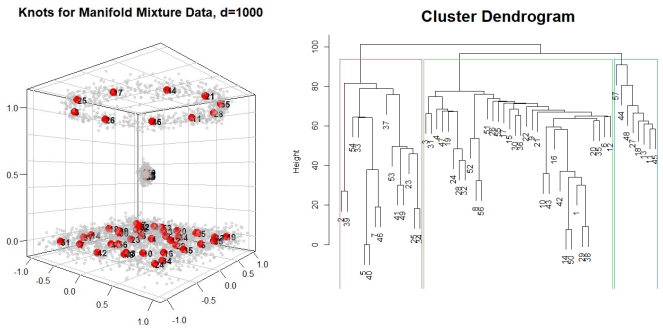


Figure A.28: Results on Manifold Mixture data with dimension 100.

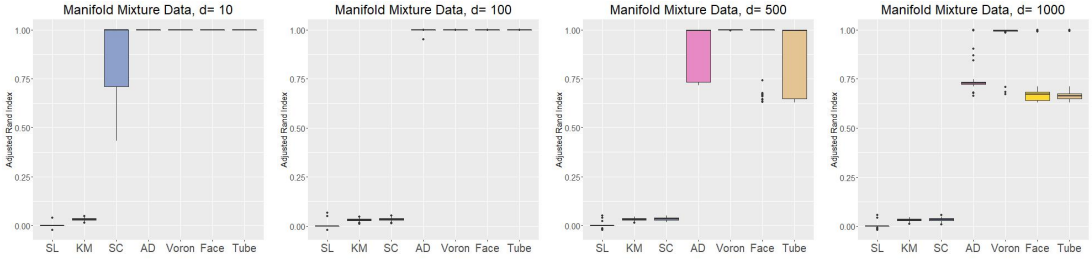


Figure A.29: Comparison of adjusted Rand index using different similarity measures on Manifold Mixture data with dimensions 10, 100, 500, 1000.

Ring Data

The ring data is constructed by a mixture distribution such that with a probability of $\frac{1}{6}$ we sample from the ring structure and with a probability of $\frac{5}{6}$ we sample from the central part. The ring structure is generated by a uniform distribution over the ring $\{(x_1, x_2) : x_1^2 + x_2^2 = 1\}$ and is corrupted with an additive Gaussian noise $N(0, 0.2^2\mathbf{I}_2)$. The central part is simply a Gaussian $N(0, 0.2^2\mathbf{I}_2)$. We generate a total of $n = 1200$ points from the above mixture and add the high dimensional noise with the same procedure as in Section 2.5.1. The same skeleton clustering approaches are applied as well as the classical approaches, with the final number of clusters chosen to be 2. The result is displayed in Figure A.31. Again, the density-based skeleton clustering methods work well even when the dimension is large.

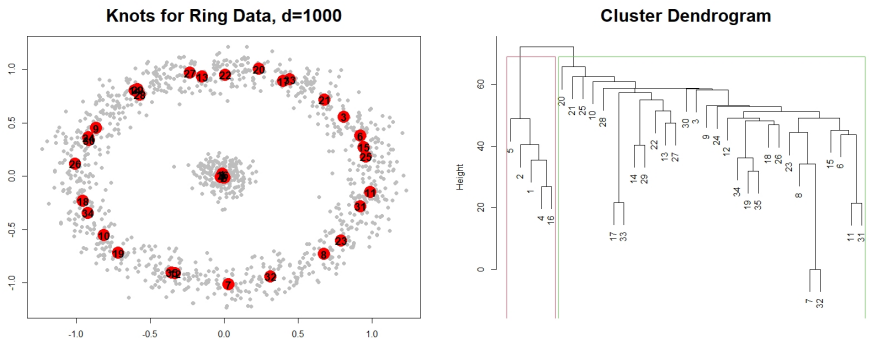


Figure A.30: Results on Ring data with dimension 1000.

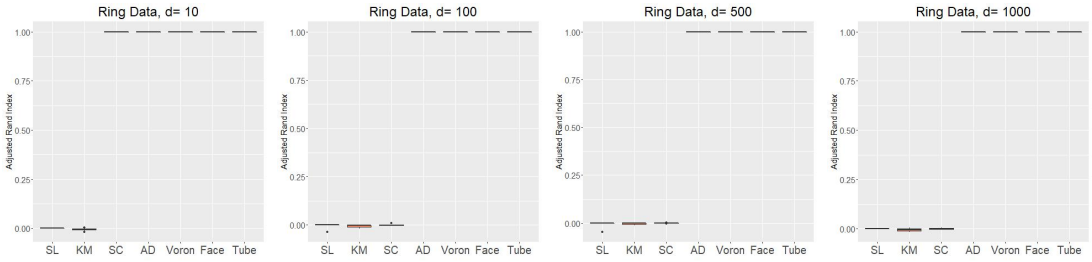


Figure A.31: Comparison of the rand index using different similarity measures on Ring data with dimensions 10, 100, 500, 1000. Medium of 100 repetitions.

H Additional Real Data Examples

Zipcode Data

This dataset consists of $n = 2000$ 16×16 images of handwritten Hindu-Arabic numerals from (Stuetzle and Nugent, 2010). We use the overfitting k-means to find $k = 45$ knots. Similar to the procedure in Section 2.5.1, we consider four similarity measures to obtain the edge weight: VD, FD, TD, and AD. We use single linkage for the four skeleton clustering approaches and compare them to three traditional methods: direct single linkage hierarchical clustering (SL), direct k-means clustering (KM), and spectral clustering (SC).

The result is shown in the left panel of Figure A.32 with the adjusted Rand index plotted against different numbers of total cluster S. The gray vertical line indicates $S = 10$, which is the actual number of digits. The skeleton clustering with VD (Voron) gives the best clustering result in terms of adjusted Rand index at the true 10 clusters and gives good clustering results when the number of clusters is specified to be larger than the truth. However, we note that spectral clustering (SC) and naive k-means clustering (KM) give comparably good results with a small number of clusters.

The right panel of Figure A.32 is the "denoised" version of the digits. We estimate the density of each observation by $[\sqrt{n}]$-nearest-neighbor density estimator and remove the observations with the lowest 10% density. We see that all clustering results are slightly improved, but such improvement may come from the decreased total sample size after denoising. Notably, the skeleton clustering with Tube density (Tube) generates significantly better clustering results after denoising the data, giving adjusted Rand indexes comparable

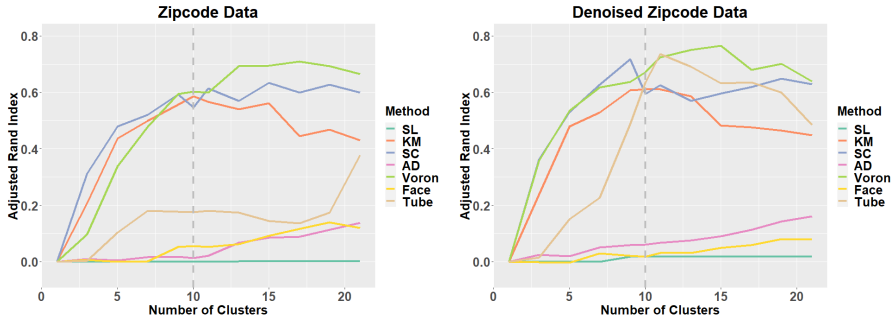


Figure A.32: Comparison of different similarity measures on all Zipcode Data.

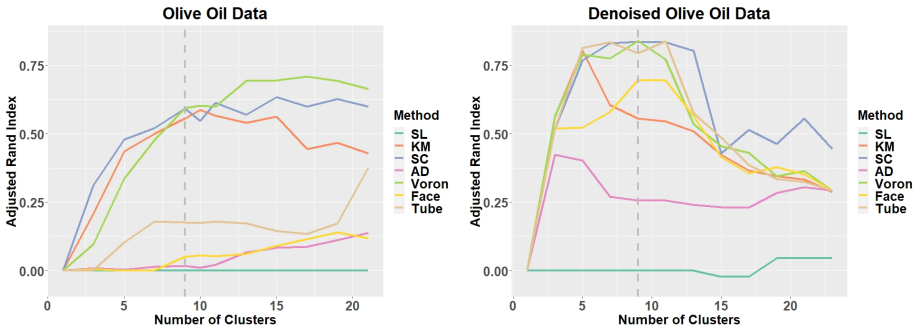


Figure A.33: The clustering performance under different numbers of final clusters of the Olive oil data.

to skeleton clustering with Voronoi density. This shows skeleton clustering with Tube density can be sensitive to noises in real data but still has the potential to give insightful clustering results.

Olive Oil Data

We consider another real dataset: the Olive Oil data (Tsimidou et al., 1987), a popular dataset for cluster analysis. This data set represents $d = 8$ chemical measurements on different specimens of olive oil produced in 9 different regions in Italy (northern Apulia, southern Apulia, Calabria, Sicily, inland Sardinia, and coast Sardinia, eastern and western Liguria, Umbria). There are a total of $n = 572$ observations in the dataset.

The same comparison procedure as in Section H is employed. The performance of different similarity measures is presented in Figure A.33. Different color denotes different similarity measures and the gray vertical line indicates the actual number of clusters 9. Overall, the skeleton clustering with Voronoi density and Tube density works well; the spectral clustering also performs well in this case. The fact that average distance fails to capture clusters in the data highlights the importance of using a density-aided similarity in this case. Note that we also include the clustering performance on the 'denoised' data, in which we remove the 10% observation with the lowest $\sqrt{n}$-Nearest-Neighbor density estimate.

Chapter 3 Appendices

I Skeleton Construction with Voronoi Density

In this section, we provide a more detailed description of the procedures for constructing the skeleton and computing the density-aided edge weight called the Voronoi density, following the work in Wei and Chen (2023).

Knots Construction

The knots in the skeleton serve as reference points within the data, allowing us to focus our attention from the overall data to these specific locations of interest. We utilize the k-means algorithm with a relatively large value of a number of knots k to create these knots in a data-driven way. The number of knots is a crucial parameter in this procedure as it governs the trade-off between the summarizing power of the representation and the

preservation of information. Empirical evidence from Wei and Chen (2023) suggests that setting k to around $\sqrt{n}$ can be a helpful reference rule, while the dimensionality of the data should be taken into consideration when choosing k.

In practice, since the k-means algorithm may not always find the global optimum, we repeat it $1,000$ times with random initial points and select the result corresponding to the optimal objective. We also advise pruning knots with only a small number of with-in-cluster observations. Additionally, it can be helpful to preprocess or denoise the data by removing observations in low-density areas to address issues that could arise for k-means clustering.

Edges Construction

We denote the given knots as $c_1, \cdots, c_k$ and represent their collection as $\mathcal{C} = c_1, \cdots, c_k$. An edge is added between two knots if they are neighbors, which is determined by whether their corresponding Voronoi cells share a common boundary. The Voronoi cell associated with a knot c_j is defined as the set of points in $\mathcal{X}$ whose distance to c_j is the smallest among all knots. That is,

$$\mathbb{C}_j = \{x \in \mathcal{X} : d(x, c_j) \leq d(x, c_\ell) \ \ \forall \ell \neq j\}, \tag{A.0.23}$$

where $d(x, y)$ is the usual Euclidean distance. We add an edge between knots (c_i, c_j) if their Voronoi cells have a non-empty intersection. This graph is referred to as the Delaunay triangulation of $\mathcal{C}$, denoted as $DT(\mathcal{C})$.

Although the Delaunay triangulation graph is conceptually intuitive, the computational complexity of the exact Delaunay triangulation algorithm has an exponential dependence on

the ambient dimension d, making it unfavorable for multivariate or high-dimensional data settings. To overcome this issue, we approximate the Delaunay triangulation with $\hat{DT}(\mathcal{C})$ by examining the 2-nearest knots of the sample data points. We query the two nearest knots for each data point and add an edge between c_i, c_j if there is at least one data point whose two nearest neighbors are c_i, c_j. The computational complexity of this sample-based approximation depends linearly on the dimension d, making it suitable for high-dimensional settings.

Voronoi Density

The Voronoi density (VD) measures the similarity between a pair of knots (c_j, c_ℓ) based on the number of observations whose 2-nearest knots are c_j and c_ℓ. We first define the Voronoi density based on the underlying probability measure and then introduce its sample analog. Given a metric d on $\mathbb{R}^d$, the 2-Nearest-Neighbor (2-NN) region of a pair of knots (c_j, c_ℓ) is defined in Equation 3.2.1 as

$$B_{j\ell} = \{X_m, m = 1, \ldots, n : \|x - V_i\| > \max\{\|x - V_j\|, \|x - V_\ell\|\}, \forall i \neq j, \ell\}.$$

Figure 3.2 provides an illustration of an example 2-NN region of a pair of knots. If two knots c_j, c_ℓ are in a connected high-density region, then we expect the 2-NN region of c_j, c_ℓ to have a high probability measure. Therefore, the probability $\mathbb{P}(B_{j\ell}) = P(X_1 \in B_{j\ell})$ can measure the association between c_j and c_ℓ. Based on this insight, the Voronoi density measures the edge weight of (c_j, c_ℓ) as

$$S_{j\ell}^{VD} = \frac{\mathbb{P}(B_{j\ell})}{|c_j - c_\ell|}. \tag{A.0.24}$$

The Voronoi density adjusts for the fact that 2-NN regions have different sizes by dividing the probability of the in-between region by the mutual Euclidean distance.

In practice, we estimate $S_{j\ell}^{VD}$ by a sample average. The numerator $\mathbb{P}(B_{j\ell})$ is estimated by $\hat{P}_n(B_{j\ell}) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \in B_{j\ell})$, and the final estimator for the VD is:

$$\hat{S}_{j\ell}^{VD} = \frac{\hat{P}_n(B_{j\ell})}{|c_j - c_\ell|}. \tag{A.0.25}$$

Calculating the Voronoi density is fast. The numerator, which only depends on 2-nearest-neighbors calculation, can be computed efficiently by the k-d tree algorithm. For high-dimensional space, space partitioning search approaches like the k-d tree can be inefficient, but a direct linear search still gives a short run-time.

Graph Segmentation

After obtaining the weighted skeleton graph, it can be helpful to prune certain edges that are not of interest or segment the skeleton into disconnected components. The edge weights defined above can be utilized to achieve this. We start by first converting the edge weights into dissimilarity measures. Specifically, let $s_{ij} i \neq j$ be the edge weights, where only connected pairs can take non-zero entries, and let $s\max = \max_{i\neq j} s_{ij}$. We then define the corresponding dissimilarities as $d_{ij} = 0$ if $i = j$, and $d_{ij} = s_{\max} - s_{ij}$ otherwise. Next, we apply hierarchical clustering using these distances. The choice of linkage criterion for hierarchical clustering depends on the underlying geometric structure of the data. Single linkage is recommended when the components are well-separated, while average linkage works better

when there are overlapping clusters of approximately spherical shapes. To determine the resulting segmented skeleton graph, dendrograms can be useful in displaying the clustering structure at different resolutions, and analysts can experiment with different numbers of final clusters and choose a cut that preserves meaningful structures based on the dendrograms. However, it is important to note that the presence of noisy data points may require a larger number of final clusters S to achieve better clustering results.

J Computational Complexity

In this section, we briefly analyze the computational costs of the proposed skeleton regression framework. The first main computational burden of the proposed regression procedure is at the skeleton construction step. Wei and Chen (2023) has provided the computational analysis on this. In particular, when constructing knots, the k-means algorithm of Hartigan and Wong Hartigan and Wong (1979) has time complexity $O(ndkI)$, where n is the number of points, d is the dimension of the data, k is the number of clusters for k-means, and I is the number of iterations needed for convergence. For the edge construction step, the approximate Delaunay Triangulation only depends on the 2-NN neighborhoods, and the k-d tree algorithm for the 2-nearest knot search gives the worst-case complexity of $O(ndk^{(1-1/d)})$. For the edge weights with Voronoi density, the numerator can be computed directly from the 2-NN search without additional computation, and the denominators as pairwise distances between knots can be computed with the worst-case complexity of $O(dk^2)$.

Given the skeleton, we then project original feature vectors onto the skeleton, which is

not very time-consuming. Finding the edge to project onto depends on identifying the two nearest knots, which is provided in the skeleton construction step. The projection takes inner product computations and takes $O(nd)$ for all the covariates.

The next computational task is to calculate the skeleton-based distance between points on the skeleton. Note that this step is not needed for the S-Lspline method but is necessary for S-Kernel and S-kNN. To find the shortest path on a graph between two faraway knots, the general version of Dijkstra's algorithm Dijkstra (1959) takes $\Theta(|\mathcal{E}| + |\mathcal{V}|^2) = \Theta(k^2)$ for each run. However, in practice, we don't need the $\frac{n(n-1)}{2}$ pairwise distances between all the projected points as the skeleton-based regressors proposed can perform with distances in local neighborhoods, which do not require path-finding algorithm for the skeleton-distance calculation.

With all the pairwise skeleton-based distances between projected feature points given, the S-kernel estimate at one point takes n_{loc} kernel weights computation where n_{loc} refers to the local support of the kernel function. S-Lspline takes $O(n)$ time to transform the data and then a single run of matrix multiplication and inversion to get the coefficients.

K Proofs

Kernel Regression: Convergence on Edge Point (Theorem 3)

PROOF. Let $\mathcal{B}(\boldsymbol{s}, h) \subset \mathcal{S}$ be the support for the kernel function $K_h(.)$ at point $\boldsymbol{s} \in \mathcal{S}$ with bandwidth h. For an edge point $\boldsymbol{s} \in E_{j\ell} \in \mathcal{E}$, where $\mathcal{E}$ is the overall set of edges defined as open sets. As $n \to \infty, h \to 0$, for sufficiently large n, by the property of an open set, we

have

$$\mathcal{B}(\boldsymbol{s}, h) \subset E_{j\ell}$$

and by our definition of skeleton distance, for two points $\boldsymbol{s}, \boldsymbol{s}' \in E_{j\ell}$ on the same edge in the skeleton, $d_{\mathcal{S}}(\boldsymbol{s}, \boldsymbol{s}') = \|\boldsymbol{s} - \boldsymbol{s}'\|$ where $\|.\|$ denotes the Euclidean distance and is 1-dimensional as parametrized on the same edge. Also we have

$$K_h(\boldsymbol{s}_j, \boldsymbol{s}_\ell) \equiv K(d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell)/h) = K(\|\boldsymbol{s}_j - \boldsymbol{s}_\ell\|/h) = K\left(\frac{\boldsymbol{s}_j - \boldsymbol{s}_\ell}{h}\right)$$

Consequently, the skeleton-based kernel regression estimator reduces to

$$\hat{m}_n(\boldsymbol{s}) = \frac{\frac{1}{nh}\sum_{j=1}^n Y_j K(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h})}{\frac{1}{nh}\sum_{j=1}^n K(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h})} \tag{A.0.26}$$

and we can use the classical asymptotic results for kernel regression in the continuous case Bierens (1983); Wasserman (2006b); Chen et al. (2017).

Let $\hat{g}_n(\boldsymbol{s}) = \frac{1}{nh}\sum_{j=1}^n K\left(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h}\right)$. We express the difference as

$$\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s}) = \frac{[\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})]\hat{g}_n(\boldsymbol{s})}{\hat{g}_n(\boldsymbol{s})} = \frac{\frac{1}{nh}\sum_{j=1}^n [Y_j - m_{\mathcal{S}}(\boldsymbol{s})]K(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h})}{\frac{1}{nh}\sum_{j=1}^n K(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h})} \tag{A.0.27}$$

and we analyze the denominator and numerator below.

Let $g(\boldsymbol{s})$ be the density at point $\boldsymbol{s}$ on the skeleton. For the denominator, we start with the bias:

$$\begin{aligned}
|\mathbb{E}\hat{g}_n(\boldsymbol{s}) - g(\boldsymbol{s})| &= \left|\frac{1}{h}\int K\left(\frac{\boldsymbol{s} - y}{h}\right)g(y)dy - g(x)\int K(y)dy\right| \\
&= \left|\int K(z)[g(\boldsymbol{s} - hz) - g(\boldsymbol{s})]dz\right| \\
&\leq \int K(z)C_1 |hz|\, dz = C_1 h \int K(z)\, |z|\, dz = O(h),
\end{aligned}$$

where C_1 is the Lipschitz constant of the density function. For the variance, we have

$$\begin{aligned}
\mathrm{V}ar\left(\hat{g}_n(\boldsymbol{s})\right) &\leq \frac{1}{nh^2}\int K^2\left(\frac{\boldsymbol{s}-y}{h}\right)g(y)dy \\
&= \frac{1}{nh}\int K^2(z)g(\boldsymbol{s}-hz)dz \\
&\leq \frac{1}{nh}\int K^2(z)[g(\boldsymbol{s})+C_1\left|hz\right|]dz \\
&= \frac{1}{nh}\left[g(\boldsymbol{s})\int K^2(z)dz + C_1 h\int K^2(z)\left|z\right|dz\right] \\
&= \frac{1}{nh}g(\boldsymbol{s})\int K^2(z)dz + o\left(\frac{1}{nh}\right).
\end{aligned}$$

Putting it all together, we have

$$\left|\hat{g}_n(\boldsymbol{s})-g(\boldsymbol{s})\right| = O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right).$$

Note that we only assume Lipschitz continuity and hence have the bias of rate $O(h)$ rather than the usual $O(h^2)$ rate with second-order smoothness. Higher-order smoothness of g may not improve the overall estimation rate due to the fact that we only have Lipschitz continuity of the regression function.

Now we analyze the numerator of equation (A.0.27). We start with the decomposition

$$\begin{aligned}
[\hat{m}_n(\boldsymbol{s})-m_{\mathcal{S}}(\boldsymbol{s})]\hat{g}(\boldsymbol{s}) = &\underbrace{\frac{1}{nh}\sum_{j=1}^{n}U_j K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)}_{q_1(\boldsymbol{s})} \\
&+\underbrace{\frac{1}{n}\sum_{j=1}^{n}\left\{[m_{\mathcal{S}}(\boldsymbol{s}_j)-m(\boldsymbol{s})]K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)\frac{1}{h}-\mathbb{E}\left[[m_{\mathcal{S}}(\boldsymbol{s}_j)-m_{\mathcal{S}}(\boldsymbol{s})]K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)\frac{1}{h}\right]\right\}}_{q_2(\boldsymbol{s})} \\
&+\underbrace{\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[[m_{\mathcal{S}}(\boldsymbol{s}_j)-m_{\mathcal{S}}(\boldsymbol{s})]K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)\frac{1}{h}\right]}_{q_3(\boldsymbol{s})}.
\end{aligned}$$

First, we show that

$$q_1(\boldsymbol{s}) = O_p\left(\sqrt{\frac{1}{nh}}\right).$$

Let

$$v_{n,j}(\boldsymbol{s}) = U_j K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)\frac{1}{\sqrt{h}}$$

and we have

$$\sqrt{nh}q_1(\boldsymbol{s}) = \frac{1}{\sqrt{n}}\sum_{j=1}^{n} v_{n,j}(\boldsymbol{s}).$$

Thus, its mean is

$$\mathbb{E}v_{n,j}(\boldsymbol{s}) = \mathbb{E}\left\{U_j K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)\frac{1}{\sqrt{h}}\right\} = 0$$

and the variance is

$$\begin{aligned}\mathbb{E}[v_{n,j}(\boldsymbol{s})^2] = \mathbb{E}U_j^2 K\left(\frac{\boldsymbol{s}-\boldsymbol{s}_j}{h}\right)^2\frac{1}{h} &= \int \sigma_u^2(\boldsymbol{s}-hz)g(\boldsymbol{s}-hz)K(z)^2dz \\ &\to \sigma_u^2(\boldsymbol{s})g(\boldsymbol{s})\int K(z)^2dz = O(1),\end{aligned}$$

where for the second equality we use the change of variable and by assumption, we have $\int K(z)^2dz < \infty$. Therefore,

$$q_1(\boldsymbol{s}) = O_p\left(\sqrt{\frac{1}{nh}}\right).$$

For the second term, note that $\mathbb{E}(q_2(\boldsymbol{s})) = 0$ and the variance is

$$\begin{aligned}\mathbb{E}\left[\sqrt{nh}q_2(\boldsymbol{s})\right]^2 &= \int [m_{\mathcal{S}}(\boldsymbol{s}-hz) - m_{\mathcal{S}}(\boldsymbol{s})]^2 g(\boldsymbol{s}-hz)K(z)^2dz \\ &\quad - h\left\{\int [m_{\mathcal{S}}(\boldsymbol{s}-hz) - m_{\mathcal{S}}(\boldsymbol{s})]g(\boldsymbol{s}-hz)K(z)dz\right\}^2 \\ &\to 0\end{aligned}$$

when $h \to 0$, and hence,

$$q_2(\boldsymbol{s}) = o_p\left(\sqrt{\frac{1}{nh}}\right).$$

For the last term, note that we have

$$\begin{aligned} q_3(\boldsymbol{s}) &= \int [m_{\mathcal{S}}(\boldsymbol{s} - hz) - m_{\mathcal{S}}(\boldsymbol{s})] g(\boldsymbol{s} - hz) K(z) dz \\ &= \int [m_{\mathcal{S}}(\boldsymbol{s} - hz) g(\boldsymbol{s} - hz) - m_{\mathcal{S}}(\boldsymbol{s}) g(\boldsymbol{s})] K(z) dz - m_{\mathcal{S}}(\boldsymbol{s}) \int [g(\boldsymbol{s} - hz) - g(\boldsymbol{s})] K(z) dz \\ &\leq C_1 h \int |z| K(z) dz + C_2 h \int |z| K(z) dz \end{aligned}$$

where C_1 is the Lipschitz constant for $m(\boldsymbol{s})g(\boldsymbol{s})$ and C_2 is the Lipschitz constant for $g(\boldsymbol{s})$. Therefore,

$$q_3(\boldsymbol{s}) = O(h)$$

Putting all three terms together, $[\hat{m}(\boldsymbol{s}) - m(\boldsymbol{s})]\hat{g}(\boldsymbol{s}) = O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)$. As a result, equation (A.0.27) becomes

$$\begin{aligned} \hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s}) = \frac{[\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})]\hat{g}(\boldsymbol{s})}{\hat{g}(\boldsymbol{s})} &= \frac{O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)}{g(\boldsymbol{s}) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)} \\ &= O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right) \end{aligned}$$

by the Taylor expansion of the fraction.

□

Kernel Regression: Convergence on Knot with Zero Mass (Proposition 5)

For the ease of proof, we first prove Proposition 5 and then prove Theorem 4.

Proof. Let $\boldsymbol{s} \in \mathcal{V}$ be a knot with no mass, i.e., $P(\boldsymbol{s}_j = \boldsymbol{s}) = 0$. The kernel regression

can be decomposed as

$$\begin{aligned}\hat{m}(\boldsymbol{s}) &= \frac{\frac{1}{n}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V}\cap\mathcal{B}(\boldsymbol{s},h))}{\frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V}\cap\mathcal{B}(\boldsymbol{s},h))} \\ &= \frac{\frac{1}{n}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} Y_j I(\boldsymbol{s}_j = \boldsymbol{s})}{\frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} I(\boldsymbol{s}_j = \boldsymbol{s})} \\ &= \frac{\varepsilon_{1,n}(\boldsymbol{s}) + \nu_{1,n}(\boldsymbol{s})}{\varepsilon_{2,n}(\boldsymbol{s}) + \nu_{2,n}(\boldsymbol{s})}.\end{aligned}$$

Because $\boldsymbol{s}$ is a point without probability mass, $\nu_{1,n}(\boldsymbol{s}) = \nu_{2,n}(\boldsymbol{s}) = 0$, so the above can further reduce to

$$\hat{m}(\boldsymbol{s}) = \frac{\frac{1}{nh}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h))}{\frac{1}{nh}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h))}.$$

However, different from the case on edges, the support of the kernel intersects with multiple edges even when $h \to 0$, so we study the contribution of each edge individually. Note that when $h \to 0$, the only knot that exists in the intersection $\mathcal{B}(\boldsymbol{s},h)\cap\mathcal{E}$ is $\boldsymbol{s}$. So we only need to consider contributions of edges adjacent to $\boldsymbol{s}$.

Let $\mathcal{I}$ collect all the edge indices with one knot being $\boldsymbol{s}$, i.e., $\ell \in \mathcal{I}$ implies that there is an edge between $\boldsymbol{s}$ and $\boldsymbol{v}_\ell \in \mathcal{V}$. Let E_ℓ be the edge connecting $\boldsymbol{s}$ and $\boldsymbol{v}_\ell$. The indicator function $I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) = \sum_{\ell\in\mathcal{I}} I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h))$. With this, we can rewrite $\hat{m}(\boldsymbol{s})$ as

$$\begin{aligned}\hat{m}(\boldsymbol{s}) &= \frac{\sum_{\ell\in\mathcal{I}}\frac{1}{nh}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in E_\ell\cap\mathcal{B}(\boldsymbol{s},h))}{\sum_{\ell\in\mathcal{I}}\frac{1}{nh}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in E_\ell\cap\mathcal{B}(\boldsymbol{s},h))} \\ &= \frac{\sum_{\ell\in\mathcal{I}} \hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s})}{\sum_{\ell\in\mathcal{I}} \hat{g}_{n,\ell}(\boldsymbol{s})}.\end{aligned}$$

where

$$\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh}\sum_{j=1}^{n} K\left(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h}\right) I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h)),$$

$$\hat{m}_{n,\ell}(\boldsymbol{s})\cdot\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh}\sum_{j=1}^{n} Y_j K\left(\frac{\boldsymbol{s}_j - \boldsymbol{s}}{h}\right) I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h)).$$

Thus, we will analyze $\hat{g}_{n,\ell}(\boldsymbol{s})$ and $\hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s})$. For a point $\boldsymbol{s}_j$ on the edge E_ℓ, we can reparamterize it as $\boldsymbol{s}_j = T_j\boldsymbol{v}_\ell + (1-T_j)\boldsymbol{s}$ for some $T_j \in (0,1)$. The location $\boldsymbol{s}$ corresponds to the case $T_j = 0$ and any $\boldsymbol{s}_j \in E_\ell$ will be mapped to $T_j > 0$. With this reparameterization, we can write

$$\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh}\sum_{j=1}^{n} K\left(\frac{T_j}{h}(\boldsymbol{v}_\ell - \boldsymbol{s})\right) I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h)),$$

$$\hat{m}_{n,\ell}(\boldsymbol{s})\cdot\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh}\sum_{j=1}^{n} Y_j K\left(\frac{T_j}{h}(\boldsymbol{v}_\ell - \boldsymbol{s})\right) I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h)).$$

To study the limiting behavior when $h \to 0$, let $g_\ell(t) = g((1-t)\boldsymbol{s} + t\boldsymbol{v}_\ell)$, $g_\ell(0) = \lim_{x\downarrow 0} g_\ell(x)$; $m_\ell(t) = m_{\mathcal{S}}((1-t)\boldsymbol{s} + t\boldsymbol{v}_\ell)$, $m_\ell(0) = \lim_{t\downarrow 0} m_\ell(t)$; and $\sigma_\ell^2(t) = \mathbb{E}(|U_j|^2|\boldsymbol{s}_j = (1-t)\boldsymbol{s} + t\boldsymbol{v}_\ell)$, $\sigma_\ell^2(0) = \lim_{t\downarrow 0}\sigma_\ell^2(t)$. Then with the new notations, we can write

$$\begin{aligned}\mathbb{E}(f(T_j(\boldsymbol{v}_\ell - \boldsymbol{s}))I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h))) &= \mathbb{E}(f(\boldsymbol{s}_j - s)I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h))) \\ &= \int_{t>0} f(t)g_\ell(t)dt\end{aligned}$$

for any integrable function f. The bias of the denominator can be written as

$$\begin{aligned}\left|\mathbb{E}\hat{g}_{n,\ell}(\boldsymbol{s}) - \frac{1}{2}g_\ell(0)\right| &= \left|\frac{1}{h}\int_{t>0} K\left(\frac{t}{h}\right)g_\ell(t)dt - g_\ell(0)\int_{z>0} K(z)\right| \\ &= \left|\int_{z>0} K(z)[g_\ell(hz) - g_\ell(0)]dz\right| \\ &\le \int_{z>0} K(z)C_1 hz dz \\ &= C_1 h\int_{z>0} K(z)z dz = O(h).\end{aligned}$$

For stochastic variation, we have

$$\begin{aligned}\mathrm{V}ar\left(\hat{g}_{n,\ell}(\boldsymbol{s})\right) &\le \frac{1}{nh^2}\int_{t>0} K^2\left(\frac{t}{h}\right)g_\ell(t)dt \\ &= \frac{1}{nh}\int_{z>0} K^2(z)g(hz)dz\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{nh}\int_{z>0} K^2(z)[g(0)+C_1\,|hz|]dz \\
&= \frac{1}{nh}\left[g(0)\int_{z>0} K^2(z)dz + C_1 h\int_{z>0} K^2(z)\,|z|\,dz\right] \\
&= O\left(\frac{1}{nh}\right).
\end{aligned}
$$

Thus,

$$
\hat{g}_n(\boldsymbol{s}) = \sum_{\ell\in\mathcal{I}} \hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{2}\sum_{\ell\in\mathcal{I}} g_\ell(0) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)
$$

For the numerator,

$$
\begin{aligned}
\hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s}) = &\underbrace{\frac{1}{nh}\sum_{j=1}^{n} U_j K\left(\frac{t_j}{h}\right) I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h))}_{Q_1} \\
&+ \underbrace{\frac{1}{nh}\sum_{j=1}^{n} m_{\mathcal{S}}(\boldsymbol{s}_j) K\left(\frac{t_j}{h}\right) I(\boldsymbol{s}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s},h))}_{Q_2},
\end{aligned}
$$

where $U_j = Y_j - m_{\mathcal{S}}(\boldsymbol{s}_j)$. Using the fact that $\mathbb{E}(U_j|\boldsymbol{s}_j) = 0$, $\mathbb{E}(Q_1) = 0$, and the variance is

$$
\begin{aligned}
\mathrm{V}ar(Q_1) &\leq \frac{1}{nh^2}\int_{t>0} \sigma_\ell^2(t) K^2\left(\frac{t}{h}\right) g_\ell(t)dt \\
&= \frac{1}{nh}\int_{z>0} \sigma_\ell^2(hz) K^2(z) g_\ell(hz)dz \\
&= \frac{1}{nh}\int_{z>0} \sigma_\ell^2(0) K^2(z) g_\ell(0)dz + O\left(\frac{1}{nh}\right) = O\left(\frac{1}{nh}\right).
\end{aligned}
$$

For Q_2, we have

$$
\begin{aligned}
\left|\mathbb{E}(Q_2) - \frac{m_\ell(0)g_\ell(0)}{2}\right| &= \left|\frac{1}{h}\int_{t>0} m_\ell(t)K(t/h)g(t)dt - m_\ell(0)g_\ell(0)\int_{z>0} K(z)dz\right| \\
&= \left|\int_{z>0} m_\ell(hz)K(z)g_\ell(hz)dz - m_\ell(0)g_\ell(0)\int_{z>0} K(z)dz\right| \\
&\leq \int_{z>0}\left\{\left[m_\ell(0)+C_2hz\right]\left[g_\ell(0)+C_1hz\right] - m_\ell(0)g_\ell(0)\right\}K(z)dz \\
&\leq [C_1 m_\ell(0) + C_2 g_\ell(0)]h\int_{z>0} K(z)zdz + o(h) = O(h).
\end{aligned}
$$

The variance of Q_2 is bounded via

$$\begin{aligned}
\mathrm{V}ar(q_2) &\leq \frac{1}{nh^2}\int_{t>0} m_\ell^2(t)K^2\left(\frac{t}{h}\right)g_\ell(t)dt \\
&= \frac{1}{nh}\int_{z>0} m_\ell^2(hz)K^2(z)g_\ell(hz)dz \\
&\leq \frac{1}{nh}\int_{z>0} \left\{m_\ell(0)+C_2\left|hz\right|\right\}^2 K^2(z)\left\{g_\ell(0)+C_1\left|hz\right|\right\}dz \\
&= \frac{1}{nh}\left\{m_\ell^2(0)g_\ell(0)\int_{z>0} zK^2(z)dz + O(h)\right\} \\
&= O\left(\frac{1}{nh}\right)
\end{aligned}$$

Putting the terms Q_1 and Q_2 together, we have

$$\hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{2}m_\ell(0)g_\ell(0) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right).$$

As a result, we conclude that

$$\begin{aligned}
\hat{m}(\boldsymbol{s}) &= \frac{\sum_{\ell\in\mathcal{I}}\hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s})}{\sum_{\ell\in\mathcal{I}}\hat{g}_{n,\ell}(\boldsymbol{s})} \\
&= \frac{\frac{1}{2}\sum_{\ell\in\mathcal{I}} m_\ell(0)g_\ell(0) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)}{\frac{1}{2}\sum_{\ell\in\mathcal{I}} g_\ell(0) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)} \\
&= \frac{\frac{1}{2}\sum_{\ell\in\mathcal{I}} m_\ell(0)g_\ell(0)}{\frac{1}{2}\sum_{\ell\in\mathcal{I}} g_\ell(0)} + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right) \\
&= \frac{\sum_{\ell\in\mathcal{I}} m_\ell(0)g_\ell(0)}{\sum_{\ell\in\mathcal{I}} g_\ell(0)} + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right),
\end{aligned}$$

which completes the proof. □

Kernel Regression: Convergence on Knot with Nonzero Mass (Theorem 4)

Proof.

Let $\boldsymbol{s} \in \mathcal{V}$ be a point where $P(\boldsymbol{s}_j = \boldsymbol{s}) = p(\boldsymbol{s}) > 0$. Recall that the kernel regression can

be expressed as

$$\begin{aligned}\hat{m}(\boldsymbol{s}) &= \frac{\frac{1}{n}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V}\cap\mathcal{B}(\boldsymbol{s},h))}{\frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{V}\cap\mathcal{B}(\boldsymbol{s},h))} \\ &= \frac{\frac{1}{n}\sum_{j=1}^{n} Y_j K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} Y_j I(\boldsymbol{s}_j = \boldsymbol{s})}{\frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{s}_j,\boldsymbol{s}) I(\boldsymbol{s}_j \in \mathcal{E}\cap\mathcal{B}(\boldsymbol{s},h)) + \frac{1}{n}\sum_{j=1}^{n} I(\boldsymbol{s}_j = \boldsymbol{s})} \\ &= \frac{\varepsilon_{1,n}(\boldsymbol{s}) + \nu_{1,n}(\boldsymbol{s})}{\varepsilon_{2,n}(\boldsymbol{s}) + \nu_{2,n}(\boldsymbol{s})}.\end{aligned}$$

We look at each term individually and note that we have the edge components terms identical to the proof of Proposition 5, so

$$\begin{aligned}\varepsilon_{1,n}(\boldsymbol{s}) &= h\left\{\sum_{\ell\in\mathcal{I}} m_\ell(0) g_\ell(0) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)\right\} = O(h) + O_p\left(\sqrt{\frac{h}{n}}\right), \\ \varepsilon_{2,n}(\boldsymbol{s}) &= h\left\{\sum_{\ell\in\mathcal{I}} g_\ell(0) + O(h) + O_p\left(\sqrt{\frac{1}{nh}}\right)\right\} = O(h) + O_p\left(\sqrt{\frac{h}{n}}\right).\end{aligned}$$

For the terms on the knots, they are just a sample average, so

$$\nu_{2,n}(\boldsymbol{s}) = p(\boldsymbol{s}) + O_p\left(\sqrt{\frac{1}{n}}\right)$$

and similarly

$$\begin{aligned}\nu_{1,n}(\boldsymbol{s}) &= \frac{1}{n}\sum_{j=1}^{n} (m_{\mathcal{S}}(\boldsymbol{s}) + U_j) I(\boldsymbol{s}_j = \boldsymbol{s}) \\ &= m_{\mathcal{S}}(\boldsymbol{s}) p(\boldsymbol{s}) + O_p\left(\sqrt{\frac{1}{n}}\right).\end{aligned}$$

With the fact that $O_p\left(\sqrt{\frac{1}{n}}\right)$ dominates $O_p\left(\sqrt{\frac{h}{n}}\right)$, we conclude

$$\begin{aligned}\hat{m}(\boldsymbol{s}) &= \frac{O(h) + O_p\left(\sqrt{\frac{h}{n}}\right) + m_{\mathcal{S}}(\boldsymbol{s}) p(\boldsymbol{s}) + O_p\left(\sqrt{\frac{1}{n}}\right)}{O(h) + O_p\left(\sqrt{\frac{h}{n}}\right) + p(\boldsymbol{s}) + O_p\left(\sqrt{\frac{1}{n}}\right)} \\ &= \frac{O(h) + O_p\left(\sqrt{\frac{1}{n}}\right)}{O(h) + O_p\left(\sqrt{\frac{1}{n}}\right) + p(\boldsymbol{s})} + \frac{m_{\mathcal{S}}(\boldsymbol{s}) p(\boldsymbol{s})}{O(h) + O_p\left(\sqrt{\frac{1}{n}}\right) + p(\boldsymbol{s})}\end{aligned}$$

$$
\begin{aligned}
&= \frac{O(h) + O_p\left(\sqrt{\frac{1}{n}}\right)}{p(\boldsymbol{s})} + O\left[\left(\frac{O(h) + O_p\left(\sqrt{\frac{1}{n}}\right)}{p(\boldsymbol{s})}\right)^2\right] \\
&\quad + m_{\mathcal{S}}(\boldsymbol{s})p(\boldsymbol{s})\left\{\frac{1}{p(\boldsymbol{s})} + \frac{O(h) + O_p\left(\sqrt{\frac{1}{n}}\right)}{p(\boldsymbol{s})^2}\right\} \\
&= m_{\mathcal{S}}(\boldsymbol{s}) + O(h) + O_p\left(\sqrt{\frac{1}{n}}\right),
\end{aligned}
$$

which completes the proof.

□

Dual Path Algorithm for Generalized Lasso Problem

For the generalized Lasso problem:

$$\mathsf{minimize}_{\boldsymbol{\beta}} \left\|y - X\beta\right\|_2^2 + \lambda \left\|D\beta\right\|_1$$

If D is invertible or the matrix D has dimension $m \times p$ with $\text{rank}(D) = m$, this can be converted into a standard Lasso problem by setting $\theta = D\beta$, and the problem reduces to

$$\mathsf{minimize}_{\boldsymbol{\beta}} \left\|y - XD^{-1}\theta\right\|_2^2 + \lambda \left\|\theta\right\|_1$$

However, this is not the case for the incidence matrix with the number of edges larger than the number of nodes. Hence, we turn to the Lagrange dual problem. Let $\text{rank}(D) = m$, then we want to solve

$$
\begin{aligned}
&\text{minimize}_{u \in \mathbb{R}^m} \frac{1}{2}\left(X^T y - D^T u\right)\left(X^T X\right)^+\left(X^T y - D^T u\right) \\
&\qquad\qquad \text{subject to } \left\|u\right\|_\infty \leq \lambda, D^T u \in \text{row}(X)
\end{aligned}
$$

We then follow the dual path algorithm by Tibshirani and Taylor (2011). For notation, use A^+ to denote the Moore-Penrose pseudo-inverse of matrix A, and use subscript $-\mathcal{B}$ to index over all rows or coordinates except those in set $\mathcal{B}$. The algorithm is described in Algorithm 3.

Algorithm 3 Dual path algorithm for generalized Lasso problem

Start with $k = 0, \lambda_0 = \infty, \mathcal{B} = \emptyset, s = \emptyset$. While $\lambda_k > 0$:

1. Compute a solution at λ_k by least squares as

$$\hat{u}_{\lambda_k, -\mathcal{B}} = \left(D_{-\mathcal{B}} D_{-\mathcal{B}}^T\right)^+ D_{-\mathcal{B}} \left(y - \lambda_k D_{\mathcal{B}}^T s\right) \tag{A.0.28}$$

2. Compute the next hitting time h_{k+1} by

$$t_i^{(hit)} = \frac{\left[\left(D_{-\mathcal{B}} D_{-\mathcal{B}}^T\right)^+ D_{-\mathcal{B}} y\right]_i}{\left[\left(D_{-\mathcal{B}} D_{-\mathcal{B}}^T\right)^+ D_{-\mathcal{B}} D_{-\mathcal{B}}^T s\right]_i \pm 1} \tag{A.0.29}$$

where only one of $+1$ or -1 will yield a value in $[0, \lambda_k]$, and this is the "hitting time" of coordinate i. Hence the next hitting time is

$$h_{k+1} = \max_i t_i^{(hit)} \tag{A.0.30}$$

3. Compute the next leaving time ℓ_{k+1} by first defining

$$c_i = s_i \cdot \left[D_{\mathcal{B}} \left[I - D_{-\mathcal{B}}^T \left(D_{-\mathcal{B}} D_{-\mathcal{B}}^T\right)^+ D_{-\mathcal{B}}\right] y\right]_i, \tag{A.0.31}$$

$$d_i = s_i \cdot \left[D_{\mathcal{B}} \left[I - D_{-\mathcal{B}}^T \left(D_{-\mathcal{B}} D_{-\mathcal{B}}^T\right)^+ D_{-\mathcal{B}}\right] D_{\mathcal{B}}^T s\right]_i \tag{A.0.32}$$

and then the leaving time of the ith boundary coordinate is

$$t_i^{(leave)} = \begin{cases} c_i/d_i, \text{if } c_i < 0 \text{ and } d_i < 0, \\ 0, \text{otherwise} \end{cases} \tag{A.0.33}$$

Therefore, the next leaving time is

$$\ell_{k+1} = \max_i t_i^{(leave)} \tag{A.0.34}$$

4. Set $\lambda_{k+1} = \max\{h_{k+1}, \ell_{k+1}\}$. If $h_{k+1} > \ell_{k+1}$, then add the hitting coordinate to $\mathcal{B}$ and its sign to s, otherwise remove the leaving coordinate to $\mathcal{B}$ and its sign from s. Set $k = k + 1$.

L Preliminary Theory on Skeleton Projection

An essential step in the proposed regression framework is the projection onto the constructed skeleton. In this section, we provide some preliminary theoretical analysis of this projection step.

Characterizing Skeleton Projection

To begin with, for simplification for analysis, we let all the data points be lying exactly on an underlying manifold $\mathcal{M}$ with intrinsic dimension q and let $d_{\mathcal{M}}(.,.)$ denote the geodesic distance between two points on $\mathcal{M}$. Note that for the empirical results, we allow intrinsic noise in the data structure that the data points are lying around the manifolds rather than exactly on the manifold, and the ability to deal with such data is an advantage of the proposed regression framework. The exact manifold assumption is made here for ease of theoretical analysis. To make a distinction, we use $\|\boldsymbol{x}-\boldsymbol{y}\|$ to denote the Euclidean distance between two points in the ambient space $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \subseteq \mathbb{R}^d$. Also, for notation we have $\Pi^{-1} : \mathcal{S} \to \mathbb{R}^d$ that the reverse projection gives the normal space to a point on the skeleton. For $j = 1, \ldots, k$, let $\mathcal{M}_j = \mathcal{C}_j \cap \mathcal{M}$ be the j-th segmentation of the manifold within the ambient space Voronoi cells $\mathcal{C}_j = \{\boldsymbol{x} \in \mathcal{X} | d_{\mathcal{X}}(\boldsymbol{x}, V_j) < d_{\mathcal{X}}(\boldsymbol{x}, V_\ell), \ell \neq j\}$. We let $\bar{\mathcal{A}}$ denote the closure of a set $\mathcal{A}$. Denote the diameter of $\mathcal{M}_j$ as $Diam_j = \sup_{a,b \in \mathcal{M}_j} d_{\mathcal{M}}(a,b)$ and let the reach of $\mathcal{M}$ be $\tau = \sup\{t > 0 : \forall x \in \mathbb{R}^d, \exists! y \in \mathcal{M} \;\; s.t. \;\; dist(x, \mathcal{M}) = \|x - y\|\}$. Intuitively, you can roll a ball with radius τ freely along the manifold.

M1 (Local Manifold Structure) Assume each $\bar{\mathcal{M}}_j, \quad j = 1, \ldots, k$, is a compact, connected,

q_j-dimensional Riemannian submanifold with $q_j \leq q$ embedded in $\mathbb{R}^d$ for some $0 < q \leq d$. Denote the diameter of $\mathcal{M}_j$ as $Diam_j$ and the reach of $\mathcal{M}$ be τ and assume that $Diam_j < \tau$.

M1 indicates that locally within each Voronoi cell, the underlying manifold satisfies usual regularity conditions and that locally the closest point is well-defined. For a large enough number of segments k, the local regions can be small and the local manifold structure within each segmentation can be well-structured.

B1 (Bounding Diameter of Local Manifold) For each $j = 1, \ldots, k$, there exists a constant C_1 such that

$$Diam_j \leq \frac{C_1}{k^{1/q}}$$

Assumption B1 states that the diameter of the segments on the manifold according to the geodesic distance decreases with rates depending on the intrinsic dimension. This rate makes sense if the manifold is segmented in a balanced way that each local submanifold has volume at rate $O(1/k)$.

Remark 14. We employ k-means to perform data segmentation in the proposed framework for its empirical performance and existing theoretical guarantees. Previous works have viewed the k-means clustering objective as equivalent to finding a measure $\tilde{\mu}_k$ supported on at most k points such that $W_2(\mu, \tilde{\mu}_k)$ is small and have provided concentration properties in this regard. Canas and Rosasco (2012) shows that for sufficiently large k and $\mathcal{X}$ a compact, smooth d-dimensional manifold, there exists constants C and C' and a measure $\tilde{\mu}_k$ such that

$$W_2(\mu, \tilde{\mu}_k) \leq C\tau k^{-1/d} \text{ with probability } 1 - e^{-\tau^2}$$

on the basis of $n = C'k^{2+4/d}$ samples. Weed and Bach (2019) further shows that, in high dimensions with $d > 4$, the empirical measure $\hat{\mu}_k$ satisfies

$$\mathbb{E}W_2\left(\mu, \hat{\mu}_k\right) \leq C_2 k^{-1/s}$$

for any $s > d$ and C'' a constant and that

$$W_2\left(\mu, \hat{\mu}_k\right) \leq C\tau k^{-1/s} \text{ with probability } 1 - e^{-\tau^4}$$

that clustering on the basis of k i.i.d. samples from μ is asymptotically optimal.

Proposition 10 (Fréchet Mean and Knot Distance). Assume conditions **M1** and **B1** hold. For arbitrary q-Hausdorff measure on $\mathcal{M}_j$, let $\tilde{V}_j$ be the Fréchet mean of $\mathcal{M}_j$ with respect to the geodesic distance on the manifold. Let the knots V_j be constructed within the convex hull of data points in $\mathcal{C}_j$. For every $j = 1, \ldots, k$, we have

$$\left\|\tilde{V}_j - V_j\right\| \leq \frac{C_1}{k^{1/q}} \tag{A.0.35}$$

where C_1 is the same constant as in Condition **B1**.

Proof of Proposition 10. Let $Conv(\mathcal{M}_j)$ be the convex hull of $\mathcal{M}$ in the ambient space and note that

$$Diam(Conv(\mathcal{M}_j)) = Diam(\mathcal{M}_j) \leq \frac{C_1}{k^{1/q}}$$

The knot V_j given by the k-Means algorithm is a convex linear combination of points from $\mathcal{M}_j$ and hence $V_j \in Conv(\mathcal{M}_j)$, and as $\tilde{V}_j \in \mathcal{M} \subset Conv(\mathcal{M}_j)$, hence

$$\left\|\tilde{V}_j - V_j\right\| \leq Diam_{\mathcal{X}}(Conv(\mathcal{M}_j)) \leq \frac{C_1}{k^{1/q}}$$

□

The above proposition ensures that the knots we constructed are not far from the Fréchet

mean centroid on the manifold. A direct corollary by triangular inequality is that

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}-\Pi(\boldsymbol{x})\| \leq \left\|\boldsymbol{x}, V_{j(\boldsymbol{x})}\right\| \leq d_{\mathcal{M}}(\boldsymbol{x}, \tilde{V}_{j(\boldsymbol{x})}) + \left\|\tilde{V}_{j(\boldsymbol{x})}, V_{j(\boldsymbol{x})}\right\| \leq \frac{2C_1}{k^{1/q}}$$

so the projection distance is bounded with the rate decreasing with respect to the intrinsic dimension. This is a loose bound and note that there is no particularity about $\tilde{V}_j$ that the distance between V_j and any point in $\mathcal{M}_j$ can achieve this same rate. Potentially the bound can be improved with some additional conditions and to argue for the specialty of the Fréchet mean, and we leave this as future work.

By projecting onto the skeleton large dimensional space is essentially projected onto 1-dimensional and 0-dimensional structures, and the next lemma characterizes the projection sets.

Lemma 11. For any $\boldsymbol{z} \in \mathcal{M}_j$ for $j = 1, \ldots, k$, we have

$$\Pi^{-1}(\Pi(\boldsymbol{z})) \subseteq \mathcal{C}_j$$

and hence $\Pi^{-1}(\Pi(\boldsymbol{z}))\cap\mathcal{M} \subseteq \mathcal{M}_j$, where $\Pi^{-1} : \mathcal{S} \to \mathcal{X}$ such that $\Pi^{-1}(\boldsymbol{s}) = \{\boldsymbol{x} \in \mathcal{X} | \Pi(\boldsymbol{z}) = \boldsymbol{s}\}$ for $\boldsymbol{s} \in \mathcal{S}$.

This lemma essentially shows that the points that can be projected onto the same point on the skeleton belong to the same Voronoi cell, and therefore are located in the same manifold $\mathcal{M}_j$.

Proof of Lemma 11. By the definition of projection onto the skeleton, we discuss the two cases of projections separately.

If $\boldsymbol{z}$ is projected onto the knot V_j, then as only points with the closest knot being V_j can be projected onto V_j, trivially we have $\Pi^{-1}(V_j) \subseteq \mathcal{C}_j$.

For $\boldsymbol{z}$ projected onto an edge in the skeleton, the edge must be connecting V_j and the second closest knot from $\boldsymbol{z}$, which, without loss of generality, we denote as V_ℓ. Since $\boldsymbol{z} \in \mathcal{M}_j$, by definition we have $\|V_j - \boldsymbol{z}\| \leq \|V_\ell - \boldsymbol{z}\|$, and as we are doing orthogonal projection from $\boldsymbol{z}$ onto the edge, we have $\|V_j - \Pi(\boldsymbol{z})\| \leq \|V_\ell - \Pi(\boldsymbol{z})\|$ that the projection is on the edge closer to V_j. Although $\Pi(\boldsymbol{z})$ on the edge may not be contained in V_j in some special cases, we still have $\Pi^{-1}(\Pi(\boldsymbol{z})) \subseteq \mathcal{M}_j$ as argued below. Further, we have $\|V_j - \boldsymbol{y}\| \leq \|V_\ell - \boldsymbol{y}\|$ for all $\boldsymbol{y} \in \Pi^{-1}(\Pi(\boldsymbol{z}))$. Also note that, by the definition of skeleton projection, for a point $\boldsymbol{y}$ to be projected onto the edge connecting V_j and V_ℓ, the closest knot from $\boldsymbol{y}$ must be either V_j or V_ℓ. Combining above, we have for all $\boldsymbol{y} \in \Pi^{-1}(\Pi(\boldsymbol{z}))$, $\|V_j - \boldsymbol{y}\| \leq \|V_i - \boldsymbol{y}\|, i \neq j$. Hence $\Pi^{-1}(\Pi(\boldsymbol{z})) \subseteq \mathcal{M}_j$.

□

Bounding Projection Error

In this section, we provide a bound on the projection error between the true regression function

$$m(\boldsymbol{x}) = \mathbb{E}(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$$

and the skeleton-projected regression function

$$m_{\mathcal{S}}(\boldsymbol{s}) = \mathbb{E}\left(\boldsymbol{Y}|\boldsymbol{X} \in \Pi^{-1}(\boldsymbol{s})\right), \boldsymbol{s} \in \mathcal{S}$$

where $\Pi^{-1}(\boldsymbol{s}) = \{\boldsymbol{x} \in \mathcal{X} | \Pi(\boldsymbol{x}) = \boldsymbol{s}\}$.

L1 (Smoothness of the Regression Function) The true regression function is Lipschitz continuous with respect to the geodesic distance on the manifold. That is, there exists

a constant L such that, for arbitrary j in $1, \ldots, k$, for any $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{M}$,

$$|m(\boldsymbol{x}) - m(\boldsymbol{z})| \leq L \cdot d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{z}). \tag{A.0.36}$$

Proposition 12 (Projection Error Bound for Data Projected to Knot). Assume conditions **M1**, **B1**, and **L1** and assume all the data points are lying on the manifold $\mathcal{M}$. Then we have

$$\sup_{\boldsymbol{x} \in \mathcal{X}} |m(\boldsymbol{x}) - m_{\mathcal{S}}(\boldsymbol{x})| \leq L \frac{C_1}{k^{1/q}} \tag{A.0.37}$$

Proof. For $\boldsymbol{z} \in \mathcal{M}$, with a bit abuse of notation we let $m_{\mathcal{S}}(\boldsymbol{z}) := m_{\mathcal{S}}(\Pi(\boldsymbol{z}))$. and by the tower rule of expectation, we have

$$\begin{aligned} m_{\mathcal{S}}(\Pi(\boldsymbol{z})) &= \mathbb{E}\left(\boldsymbol{Y} | \Pi^{-1}(\Pi(\boldsymbol{z}))\right) \\ &= \mathbb{E}\left[\mathbb{E}\left(\boldsymbol{Y} | \boldsymbol{X} \in \Pi^{-1}(\Pi(\boldsymbol{z})), \boldsymbol{X} = \boldsymbol{x}\right) | \boldsymbol{X} \in \Pi^{-1}(\Pi(\boldsymbol{z}))\right] \\ &= \mathbb{E}\left[m(\boldsymbol{x}) | \boldsymbol{X} \in \Pi^{-1}(\Pi(\boldsymbol{z}))\right] \end{aligned}$$

Without loss of generality, we let $\boldsymbol{z} \in \mathcal{M}_j \subset \mathcal{C}_j$. By Lemma 11, we have $\Pi^{-1}(\Pi(\boldsymbol{z})) \subset \mathcal{M}_j$. Therefore,

$$\begin{aligned} |m(\boldsymbol{x}) - m_{\mathcal{S}}(\boldsymbol{x})| &\leq \sup_{\boldsymbol{z} \in \Pi^{-1}(\Pi(\boldsymbol{x}))} |m(\boldsymbol{x}) - m(\boldsymbol{z})| \\ &\leq L \cdot \sup_{\boldsymbol{z} \in \Pi^{-1}(\Pi(\boldsymbol{x}))} d_{\mathcal{M}}\left(m(\boldsymbol{x}) - m(\boldsymbol{z})\right) \\ &\leq L \cdot Diam_j \leq L \frac{C_1}{k^{1/q}} \end{aligned}$$

and hence

$$\sup_{\boldsymbol{x} \in \mathcal{X}} |m(\boldsymbol{x}) - m_{\mathcal{S}}(\boldsymbol{x})| \leq L \frac{C_1}{k^{1/q}}$$

□

Theorem 12 provides a bound with the rate depending on the intrinsic dimension for the projections error between the true regression function on the manifold and the projected regression function on the skeleton. Note that this bound also applies to points projected onto the skeleton edges, but generally, there can be some improvement in the convergence rate for edge points.

Lemma 13 (Submanifold Dimension). Assume condition **M1**. For $\boldsymbol{z} \in \mathcal{M}_j$ such that $\Pi(\boldsymbol{z}) \in \mathcal{E}_{j\ell}$ that $\boldsymbol{z}$ is projected onto the edge $j\ell$ of the skeleton. Let $N_{\boldsymbol{x}}(\mathcal{M}_j)$ denotes the normal space to $\mathcal{M}_j$ at $\boldsymbol{x} \in \mathcal{M}_j$ and let $N_{\Pi(\boldsymbol{z})}(\mathcal{E}_{j\ell})$ be the subspace normal to the line extending from the edge $\mathcal{E}_{j\ell}$. Assume for every j, ℓ and every $\boldsymbol{x} \in \mathcal{M}_j \cap \Pi^{-1}(\Pi(\boldsymbol{z}))$, we have $N_{\boldsymbol{x}}(\mathcal{M}_j) \cap N_{\boldsymbol{x}}(\Pi^{-1}(\Pi(\boldsymbol{z}))) = \{0\}$. Then we have $\Pi^{-1}(\Pi(\boldsymbol{z})) \cap \mathcal{M}_j$ to be a $(q-1)$-dimensional submanifold of $\mathcal{M}_j$.

Proof. Note that $\Pi^{-1}(\Pi(\boldsymbol{z})) = N_{\Pi(\boldsymbol{z})}(\mathcal{E}_{j\ell})$. Then the lemma follows from tangent space arguments. $\mathcal{M}_j$ and $\Pi^{-1}(\Pi(\boldsymbol{z}))$ respectively have codimensions $d-q$ and 1. So $\Pi^{-1}(\Pi(\boldsymbol{z}))$ is defined (locally) as the set of zeros of a differentiable function $f_2 : U \to \mathbb{R}^{n-1}$ defined on an open neighbourhood U of $\boldsymbol{x}$ with surjective derivative df_2. The kernel of df_2 is the tangent space to $\Pi^{-1}(\Pi(\boldsymbol{z}))$ which is l dimensional. If we now look at $f_2 \restriction \mathcal{M}_j$, then $d(f_2 \restriction M_1) : T_x(M_1) \to \mathbb{R}^{n-l}$ is again surjective since the kernel of df_2 is the tangent space to $N_{\Pi(\boldsymbol{z})}(\mathcal{E}_{j\ell})$. The intersection of the two tangent spaces has dimension $q+1-d$ so the rank nullity theorem gives the dimension of the image as $q-(q+1-d) = d-1$. Thus the zero set of $f_2 \restriction M_1$ is the a submanifold of $\mathcal{M}_j$ and is equal to $\Pi^{-1}(\Pi(\boldsymbol{z})) \cap \mathcal{M}_j$ with dimension $(q-1)$. $\square$

Note that the assumption in Lemma 13 is essentially saying that the edges connected to the knot V_j are not within the normal bundle of the manifold $\mathcal{M}_j$. With a carefully constructed skeleton, this should not be an issue. Lemma 13 shows that, generally, the set of points projected onto a point on the skeleton edge is generally of 1 dimension lower than the intrinsic dimension of the manifold. However, further assumptions and analysis are needed to translate this lower-dimensional space into a lower convergence rate. We show below a special case that for the underlying data structure to be a 1D manifold, the edge point on the skeleton has zero projection error, and leave the general theory comparing the rate of the edge point to the knot point as future work.

Projection Error for 1D Manifold

In this section, we look at the special case that the underlying manifold $\mathcal{M}$ has Hausdorff dimension $q = 1$ and provide the bounds for the projection errors onto knot points and edge points.

Corollary 14 (Projection Error onto Knots for Data on 1D Manifold). Assume conditions **M1**, **B1**, and **L1** and assume all the data points are lying on the manifold $\mathcal{M}$ with Hausdorff dimension 1. Then we have

$$\sup_{\boldsymbol{x}\in\mathcal{X}} |m(\boldsymbol{x}) - m_{\mathcal{S}}(\boldsymbol{x})| \leq L\frac{C_1}{k} \tag{A.0.38}$$

This follows directly from Proposition 12 with $q = 1$. Note that, with $k = \sqrt{n}$, the projections error between the true regression function on the manifold and the projected regression function on the skeleton has rate $O\left(n^{-1/2}\right)$.

Then we show that, on the 1D manifold, projection onto the edge point enjoys zero error, which is a lower error compared to the projection onto a knot point.

Proposition 15 (Zero Projection Error for 1D Edge Point). Assume the data manifold has Hausdorff dimension 1, conditions **M1**, and the conditions in Lemma 13 hold. Then for $\Pi^{-1}(\mathcal{E}) \equiv \{\boldsymbol{x} \in \mathcal{X} : \Pi(\boldsymbol{x}) \in \mathcal{E}\}$, we have

$$\sup_{\boldsymbol{x} \in \Pi^{-1}(\mathcal{E})} |m(\boldsymbol{x}) - m_{\mathcal{S}}(\boldsymbol{x})| = 0 \tag{A.0.39}$$

Proof. By Lemma 13, for a 1D manifold, the projection set of an edge point has dimension $q - 1 = 0$, so can only be composed of points. Also by assumption **M1** each submanifold has a diameter smaller than the reach so that the closest point is unique. Therefore, if the data point is projected onto an edge of the skeleton, then that projection is unique and has a one-to-one correspondence between the manifold point and the edge point. □

Intuitively, for a data point on the 1D manifold that is projected onto an edge of the skeleton, such projection is bijective and there is no loss in information with the projection. Particularly, compared to Proposition 14 when bounds the projection error onto knots of the skeleton, when bounding the projection error from 1D manifold onto the skeleton edge, Proposition 15 does not depend on condition **B1** which assumes the shrinking rate on the size of the submanifolds and condition **L1** which imposes smoothness on the true regression function.

Remark 15. To gather the previous propositions into an overall theorem for the projection error for data on a 1D manifold, we need to measure the set of points projected onto the

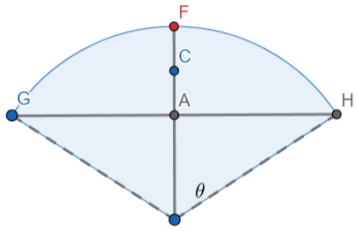


Figure A.34: Example of the difference between the knot C and the Fréchet mean F on a circular segment.

skeleton knots. However, we need a more precise characterization of the skeleton graph for this purpose. Briefly, for a knot with degree 0 or 1, we can bound the measure of the corresponding knot projection set by the volume of the submanifold as $O(1/k)$, but then we need the count of how many knots are of degree 0 or 1. For a knot with a degree greater than or equal to 2, then the knot projection set depends on the principle angles between the edges in the ambient space and the shape of the local manifold. Due to such challenges, we leave the general result on this as future work.

Example of 1D Circular Manifold in 2D Euclidean Space

Facing the challenges for a general theory as discussed above, here we look at a particular example that the manifold is a 1D circular segment embedded in the 2D Euclidean space. In particular, let $\mathcal{M}_j$ be a segment of the 1D circle with radius r embedded in $\mathbb{R}^2$ satisfying condition **M1**, so that the segment is smaller than a half circle. See Figure $A.34$ for illustration.

We first analyze the distance between the knot and the circular manifold. Under this setup, we know the Fréchet mean (F) is the midpoint of the curve, and the knot C is on the

line bisecting the curve. Let $\ell = diam_j$ be the length of the curve, and we know half of the angle for the circular sector is $\theta = \frac{\ell/2}{r}$. Then the distance from the Fréchet mean (F) to the line segment between the two endpoints is

$$d(F, A) = r - r\cos\theta = r\left(1 - 1 + \frac{\theta^2}{2} - \frac{\theta^4}{24} + \dots\right) \leq r\frac{\theta^2}{2} = \frac{\ell^2}{8r}.$$

Therefore,

$$d(F, C) = \frac{1}{2}d(F, A) \leq \frac{\ell^2}{16r}$$

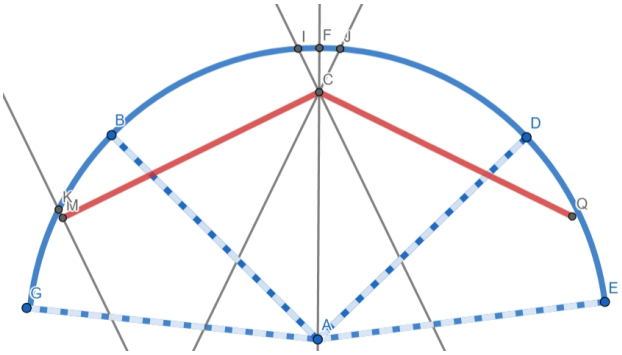
that the distance decreases at the rate of the square of the diameter. Then bounding the diameter with the rate in Assumption **B1** we have the following result:

Proposition 16 (Knot to Circular Manifold Distance). In this circular manifold setting and under assumption **M1, B1**, we have

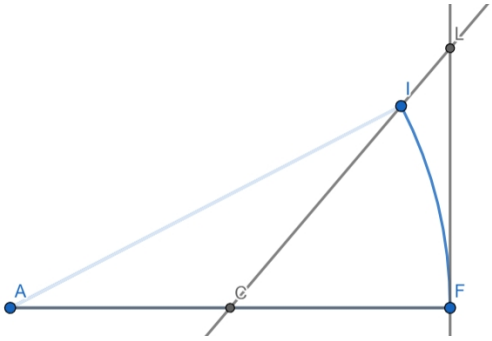
$$d(F, C) \leq \frac{C_1'}{k^{2/q}}.$$

Note that this is the square of the rate as in Proposition 10 and only applies to the Fréchet mean. Hence, under appropriate conditions, a general result with the format in Proposition 10 can be possible.

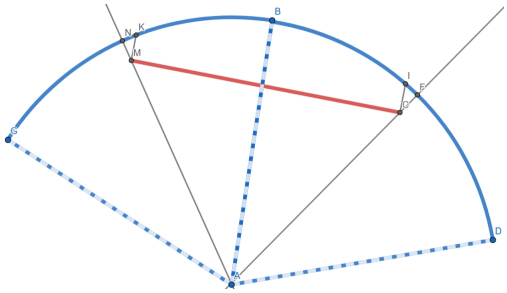
Then we analyze the volume of the set of points on the circular manifold that are projected onto the skeleton knots rather than edges. For this analysis we provide another illustration as shown in Figure A.35. Let the submanifolds be the circular segments between GB, BD, and DE, the knots of the submanifold are M, C, and Q, and the skeleton edges are line segments between MC and CQ. Line $IC \perp CM$ and intersect the manifold at I. Line $JC \perp CQ$ and intersect the manifold at J. The set of points on the manifold that are projected onto the knot C is the circular segment between I and J.



(a) Illustration of the knot project set of a circular segment. The submanifolds are the circular segments between BD, GB, and DE (colored solid blue). The Skeleton is colored in red.



(b) Illustration for the calculation of knot project set.



(c) Illustration for the calculation of knot project set.

Figure A.35: Illustration of the knot project set of a circular segment.

To calculate the length of this knot projection set, we start with some notations and give a more detailed illustration in Figure A.35b. Let the radius for the circular manifold (such as $d(F, A)$) given to be r, let the distance between the knot C and the circular manifold $d(F, C) \coloneqq e$, and let the angle $\angle FCI \coloneqq \theta$. To find the curve length ℓ_{FI} between F, I, we need the angle of $\angle FAI \coloneqq \alpha$, and denote the angle $\angle AIC \coloneqq \beta$. Trivially we have $\alpha + \beta = \theta$. Then by the law of sines, we know

$$\frac{d(A, I)}{\sin(\angle ACI)} = \frac{d(A, C)}{\sin(\angle AIC)}$$

which leads to

$$\beta = \arcsin\left[\frac{r-e}{r}\sin(\pi-\theta)\right] = \arcsin\left[\frac{r-e}{r}\sin(\theta)\right]$$

and therefore

$$\alpha = \theta - \beta = \theta - \arcsin\left[\frac{r-e}{r}\sin(\theta)\right]$$

By the Taylor expansion of $\arcsin(x) = x + \frac{x^3}{6} + \frac{3x^5}{40} + \cdots \geq x$, and $\sin\theta \geq \theta - \frac{\theta^3}{6}$, we know the curve length

$$\begin{aligned}
\ell_{FI} = r\cdot\alpha &= r\cdot\left[\theta - \arcsin\left[\frac{r-e}{r}\sin(\theta)\right]\right] \\
&\leq r\cdot\left[\theta - \frac{r-e}{r}\sin(\theta)\right] \\
&\leq r\cdot\left[\theta - \frac{r-e}{r}\left(\theta - \frac{\theta^3}{6}\right)\right] = e\theta + (r-e)\frac{\theta^3}{6}
\end{aligned}$$

To bound the angle $\angle FCI \coloneqq \theta$, let $\ell_{max} = \max_j\{diam_j\} = O\left(k^{-1}\right)$, and let $\gamma = Reach(\mathcal{M})$. For an illustration of angle bound refer to Figure A.35c. The angle

$$\angle BAD = \frac{\ell_{BD}}{r} = \frac{diam_j}{r} \leq \frac{\ell_{max}}{\gamma}$$

and this bound similarly holds for the angle $\angle GAB$. Note that the knots are on the angular bisecting line of the corresponding circular segments, and therefore we have

$$\angle MAC \leq \frac{\ell_{max}}{\gamma}$$

As $\theta \coloneqq \angle ICF$ and let $\angle NMK = \eta \geq 0$, and we know that

$$(\pi/2-\theta) + (\pi/2-\eta) + \angle MAC = \pi$$

and hence

$$\theta + \eta = \angle MAC$$

For a loose bound we have

$$\theta \leq \angle MAC \leq \frac{\ell_{max}}{\gamma} \leq \frac{C''}{\gamma} k^{-1}$$

Combining with the above, we have the curve length between F, I

$$\ell_{FI} \leq e\theta + (r-e)\frac{\theta^3}{6} = O\left(k^{-2}\right) \times O\left(k^{-1}\right) + O(1) \times O(k^{-3}) = O(k^{-3})$$

This bound also applies to the length of the circular segment ℓ_{FJ}. Therefore, the measure of the set of points projected onto knot C is

$$\ell_{IJ} = \ell_{FJ} + \ell_{FI} = O(k^{-3}) + O(k^{-3}) = O(k^{-3}).$$

Proposition 17 (Knot Projection Set for 1D circular Manifold). Let the underlying manifold be a 1D circular segment. Let the submanifolds partition the circular segment and the skeleton graph has a chain-like structure. Assume conditions **M1, B1** hold. Then for any knot V with degree 2, we have

$$\mu_{\mathcal{M}}\left(\Pi^{-1}(V)\right) = O(k^{-3}).$$

That is, the measure of the projection set of an in-between knot decreases with a rate of the cubic of k, which is fast. Then, as a corollary to Proposition 17, we can bound the total measure of all the knot-projecting points.

Corollary 18. Let the underlying manifold be a 1D circular segment. Let the submanifolds partition the circular segment and the skeleton graph has a chain-like structure. Assume conditions **M1, B1** hold. Then we have

$$\mu_{\mathcal{M}}\left(\cup_{V \in \mathcal{V}} \Pi^{-1}(V)\right) = O\left(k^{-1}\right).$$

Proof. Overall, by Proposition 17, the set of points projected onto any connecting knot

has a measure smaller than or equal to $(k-2) \times O(k^{-3}) = O(k^{-2})$.

For end knots (with degree equal to 1, or 0), we can bound the measure of the set of points projected onto one of these knots by the measure of the corresponding submanifold, with rate $O\left(k^{-1}\right)$. In this case, we have the number of end knots to be 2, which is $O(1)$. Together, we have the total measure of all the knot-projecting points to be bounded by

$$O(k) \times O(k^{-3}) + O(1) \times O\left(k^{-1}\right) = O\left(k^{-1}\right)$$

which diminishes as k increases.

□

That is, in this 1D circular manifold case, the total measure of knot-projecting point is only a small proportion of the total measure of the data space and decreases with the number of knots k.

M Additional Simulation Results

Vary Skeleton Cuts

In this section, we examine the effect of cutting the skeleton into various numbers of disjoint components on the final regression performance. We use the same simulated datasets from Section 3.4, including Yinyang data, Noisy Yinyang data, and SwissRoll data. The analysis procedure is mainly the same, where we use 5-fold cross-validation SSE to evaluate the regression results for each dataset and repeat the process 100 times with randomly generated datasets. The main difference is that, during the skeleton construction step, we segment the skeleton graph into different disjoint components using single-linkage hierarchical clustering

with respect to the Voronoi Density weights, as outlined in Section 3.2.1. We then fit and evaluate the skeleton-based regression methods on the skeletons that have been differently cut.

Vary Skeleton Cuts for Yinyang Data

In this section, we investigate how cutting the skeleton into different numbers of disjoint components affects the performance of skeleton-based methods using the Yinyang data (from Section 3.4.2). We randomly generate 1000-dimensional Yinyang data 100 times and use 5-fold cross-validation to calculate the SSE on each dataset. We fit the skeleton-based methods in the same manner as in Section 3.4, with the exception that the number of knots is fixed at 38 and we cut the initial graph into various numbers of disjoint components (ranging from 1 to 25) when constructing the skeleton. The median 5-fold cross-validation SSEs across the 100 datasets for different numbers of disjoint components are plotted in Figure A.36.

Our results show that the S-Lspline method is sensitive to changes in the skeleton structure. In the case of Yinyang data, since there are 5 true disjoint structures in the covariate space, a cut of 5 results in the best regression performance. By design, S-Lspline regressors may incorporate unrelated information from one structure to another when an edge connects two structurally different areas, thus leading to a decline in the regression performance. For future research, incorporating edge weights into the S-Lspline regressor may help to mitigate the interference between different structures. The S-Kernel regressor also achieves optimal performance when the skeleton is segmented into 5 disjoint components. Skeleton-based kernel regression methods exhibit large changes in performance as the skeleton

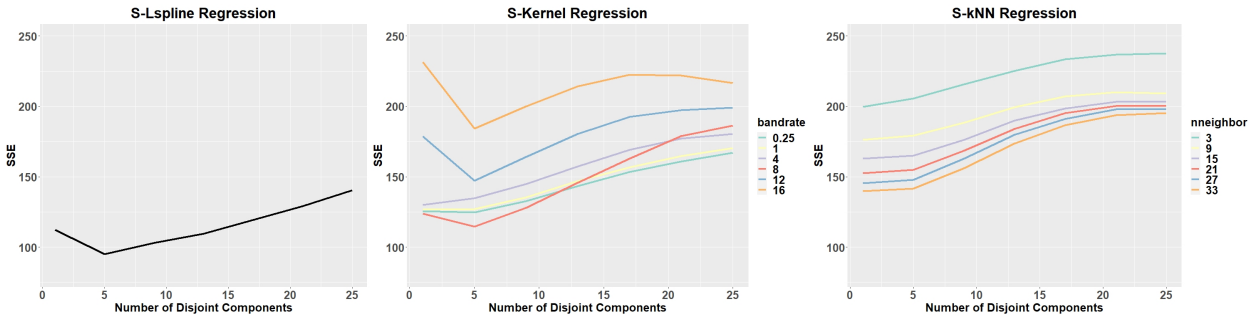


Figure A.36: Yinyang $d = 1000$ data skeleton regression results with the number of knots fixed as 38 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

segmentation changes when the bandwidth is large. This is understandable as larger bandwidths allow more information from large distances, which are more likely to be non-informative as the segmentation changes. On the other hand, the S-kNN regressor has the best regression performance when the skeleton is left as a fully connected graph. This may be due to the locally adaptive nature of the k-nearest-neighbor method that ensures regression results are accurate as long as local neighborhoods are identified accurately.

Vary Skeleton Cuts for Noisy Yinyang Data

We then evaluate the performance of the skeleton-based regression methods on the Noisy Yinyang data (from Section 3.4.3) when the skeletons are constructed with different numbers of disjoint components. Similarly, we randomly generate 1000-dimensional Noisy Yinyang data 100 times and use 5-fold cross-validation to calculate the sum of squared errors (SSE) on each dataset. We fix the number of knots to be 71 and construct skeletons with different numbers of disjoint components. The median 5-fold cross-validation SSEs across the 100 datasets for different numbers of disjoint components are plotted in Figure A.37.

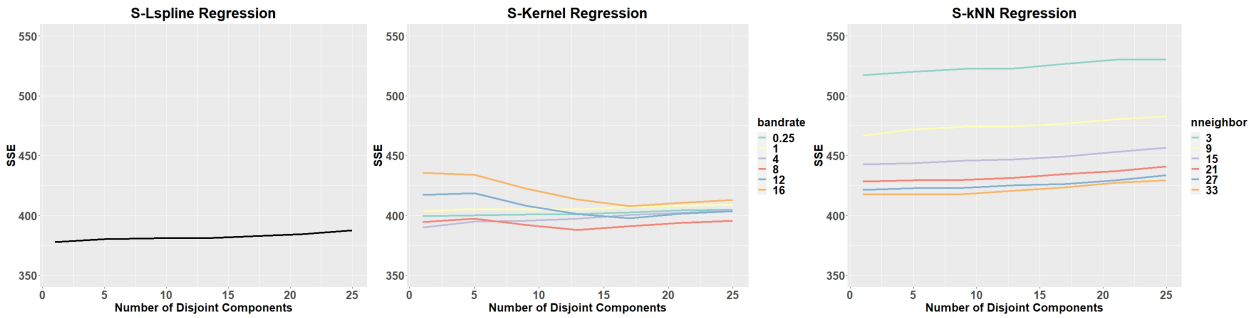


Figure A.37: Noise Yinyang $d = 1000$ data skeleton regression results with the number of knots fixed as 38 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

With the presence of noise, the S-Lspline method does not show significant variations in performance when the skeleton graph is cut into different disjoint components. The best regression result is obtained when the graph is left as a fully connected graph. In contrast, the performance of the S-kernel method varies with the number of disjoint components. The best results, regardless of the bandwidth, are obtained when the skeleton is segmented into around 13 components, which is larger than the true number of 5 components in the data. Lastly, the S-kNN method demonstrates an increase in SSE with an increase in the number of disjoint components.

Vary Skeleton Cuts for SwissRoll data

In this section, we evaluate the performance of skeleton-based methods on SwissRoll data (from Section 3.4.4) with skeletons cut into different numbers of disjoint components. Similarly, we randomly generate 1000-dimensional SwissRoll data 100 times and use 5-fold cross-validation to calculate the sum of squared errors (SSE) on each dataset. We fix the number of knots to 70 and construct skeletons with different numbers of disjoint components.

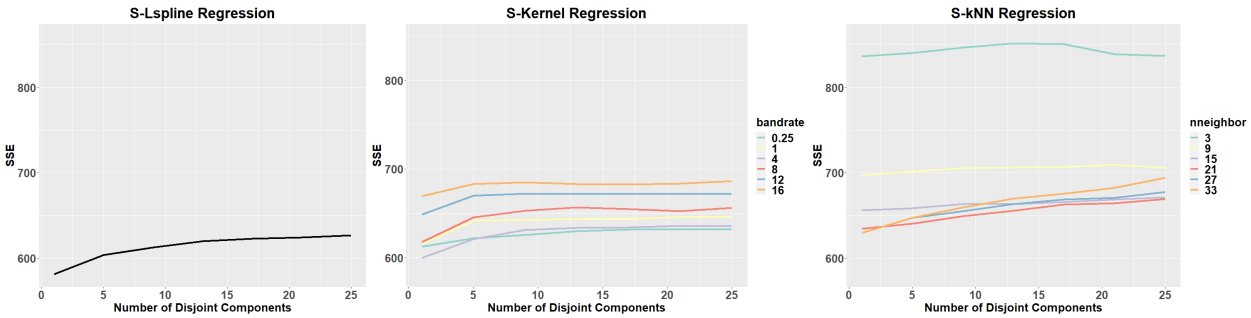


Figure A.38: SwissRoll $d = 1000$ data skeleton regression results with the number of knots fixed as 70 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

The median 5-fold cross-validation SSEs across the 100 datasets for different numbers of disjoint components are plotted in Figure A.38.

We find that the S-Lspline regressor is sensitive to changes in the skeleton structure, with the best regression results obtained when the skeleton is constructed as one connected graph. This makes sense as the covariates lie on one connected manifold. The S-Kernel regressor also performs best on the fully connected skeleton. After an initial increase in SSE as the number of disjoint components increases, the SSE of the S-kernel regressor remains relatively stable. The S-kNN regressor also achieves the best regression performance when the skeleton is left as a fully connected graph. Overall, the SSE of the S-kNN regressor increases with the number of disjoint components, but for a small number of neighbors, there can be a decrease in SSE when the skeleton is cut into more disjoint components. One possible explanation is that, as the response function has discontinuous changes, segmenting the covariate space into more fragments can improve estimation in regions where the response changes abruptly.

Penalized S-Lspline Empirical Results

In this section, we present the results of the S-Lspline regression with penalizations as introduced in Section 3.3.3. We use the same datasets as the simulations in Section 3.4 and follow the analysis procedure but with the regression methods to be the S-Lspline method with different types of penalties and varying penalization parameters λ.

The results of the penalized S-Lspline methods on the Yinyang $d = 1000$ data are summarized in Table A.2 with the plots illustrating the effect of a varying number of knots and varying penalty parameters shown in Figure A.39.

The results of the penalized S-Lspline methods on the Noisy Yinyang $d = 1000$ data are summarized in Table A.3 with plots in Figure A.40. We observe that adding penalization does not improve the regression results.

The results of the penalized S-Lspline methods on the Swill Roll $d = 1000$ data are summarized in Table A.4 with plots in Figure A.41. Including penalization terms for the linear spline model does not improve the regression results in this setting.

Overall, we observe that adding penalization terms to the linear spline model based on the skeleton graph has minimal effect on the regression performance, and having no penalization actually gives the best results although by a very tiny margin. We also test the penalized versions of the S-Lspline method on the real data examples (COIL-20 and SDSS datasets) and similarly observe that having penalization terms only leads to minimal effects on the regression performance much and the vanilla version of the S-Lspline can give the best result for most of the times.

Method	Medium SSE (5%, 95%)	lambda	nknots
No Penalization	110.4 (105.6, 118.5)	-	38
Laplacian Smoothing Order 0	110.4 (105.6, 118.5)	0.001	38
Laplacian Smoothing Order 1	111.2 (104.4, 117.8)	0.001	38
Laplacian Smoothing Order 2	110.2 (104.7, 118.9)	0.001	38
Trend Filtering Order 0	110.6 (105.6, 118.9)	0.001	38
Trend Filtering Order 1	111.3 (104.4, 118.0)	0.001	38
Trend Filtering Order 2	110.3 (104.7, 120.5)	0.001	38

Table A.2: S-Lspline regression results on Yinyang $d = 1000$ data with varying penalties and parameters. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5th percentile and 95th percentile of the SSEs from the given parameter settings are reported in brackets.

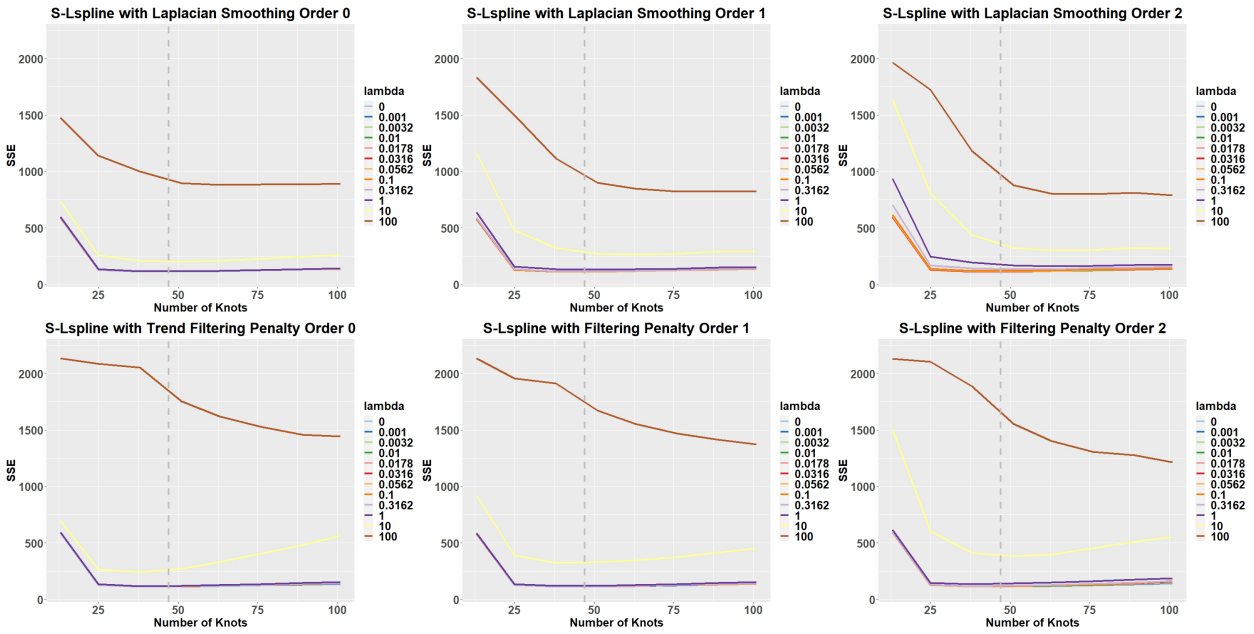


Figure A.39: S-Lspline regression results on Yinyang $d = 1000$ data with varying penalties and parameters. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

Method	Medium SSE (5%, 95%)	lambda	nknots
No penalization	375.0 (354.7, 396.3)	-	57
Laplacian Smoothing Order 0	377.0 (355.1,394.4)	0.001	42
Laplacian Smoothing Order 1	375.0 (354.7, 396.3)	0.001	57
Laplacian Smoothing Order 2	377.2 (355.9, 403.3)	0.001	71
Trend Filtering Order 0	377.0 (355.1, 394.4)	0.003	42
Trend Filtering Order 1	375.0 (354.7, 398.0)	0.001	57
Trend Filtering Order 2	378.0, (355.9, 399.0)	0.001	42

Table A.3: S-Lspline regression results on Noisy Yinyang $d = 1000$ data with varying penalties and parameters. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5th percentile and 95th percentile of the SSEs from the given parameter settings are reported in brackets.

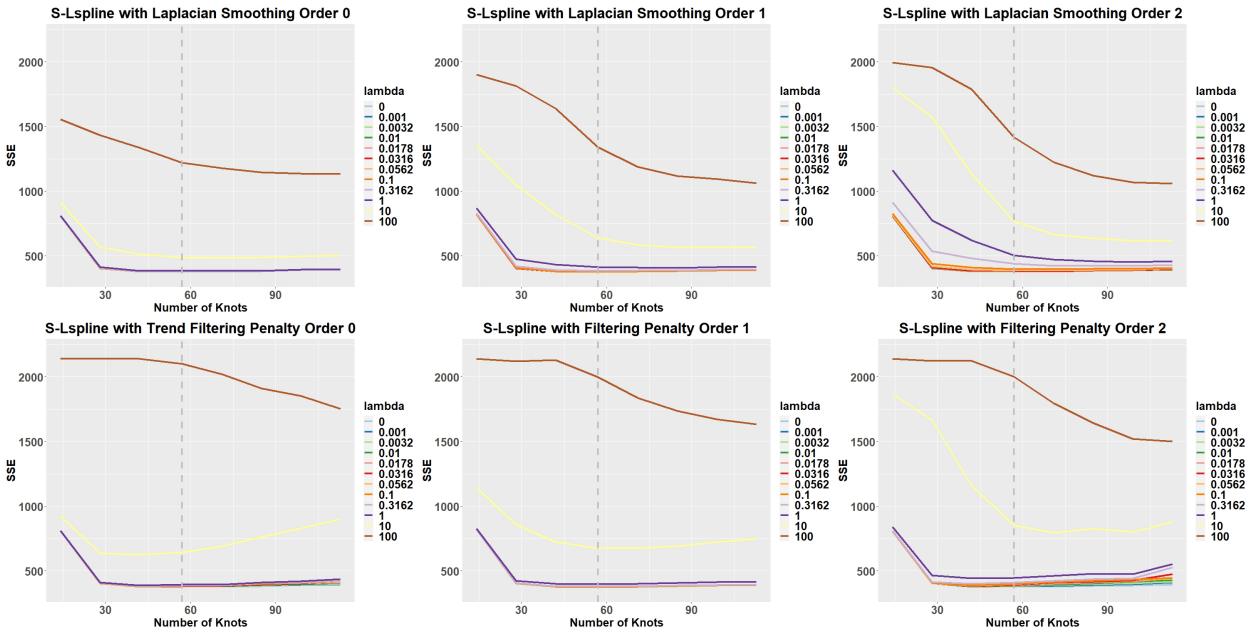


Figure A.40: S-Lspline regression results on Noisy Yinyang $d = 1000$ data with varying penalties and parameters. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

Method	Medium SSE (5%, 95%)	lambda	nknots
No penalization	572.7 (521.0, 640.9)	-	60
Laplacian Smoothing Order 0	573.4 (521.4, 632.6)	0.001	60
Laplacian Smoothing Order 1	583.7 (524.7, 633.1)	0.001	60
Laplacian Smoothing Order 2	573.1 (521.5, 641.3)	0.001	60
Trend Filtering Order 0	580.2 (524.6, 685.7)	0.01	60
Trend Filtering Order 1	583.7, (524.6, 633.0)	0.001	60
Trend Filtering Order 2	574.4, (522.0, 657.1)	0.001	60

Table A.4: S-Lspline regression results on Swiss Roll $d = 1000$ data with varying penalties and parameters. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5th percentile and 95th percentile of the SSEs from the given parameter settings are reported in brackets.

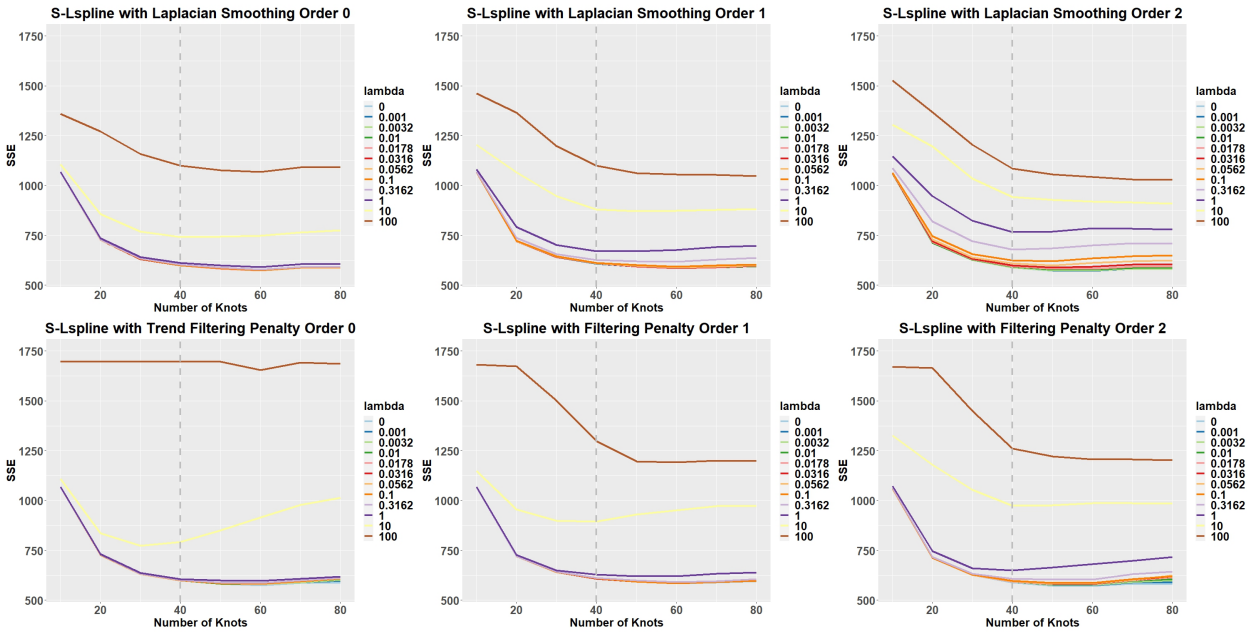


Figure A.41: S-Lspline regression results on Swill Roll $d = 1000$ data with varying penalties and parameters. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

Low Dimensional Simulation Data Examples

For the simulations in Section 3.4, we add random Gaussian variables to make the datasets have a large dimension, and illustrate that the skeleton regression framework can have advantages in dealing with such large-dimensional datasets. In this section, we look instead at the low-dimensional settings where the datasets do not have noisy variables. We show that the skeleton-based regression methods can also have competitive performance in such settings compared to existing regression approaches.

Yinyang Data d = 2

We use the same data generation mechanism as in Section 3.4.2 but without additional noisy variables (having $d = 2$). We follow the same analysis procedure as in Section 3.4.2 and take the median, 5th percentile, and 95th percentile of the 5-fold cross-validation Sum of Squared Errors (SSEs) for each parameter setting of each method over the 100 simulated datasets. We present the smallest median SSE for each method in Table A.5 along with the corresponding best parameter setting. The plots illustrating the effect of varying the numbers of knots are included in Figure A.42.

We observe that all the skeleton-based methods (S-Kernel, S-kNN, and S-Lspline) have performance comparable to the standard kNN in this setting. Ridge and Lasso regression, despite the regularization effect, resulted in relatively high SSEs. The SpecSeries method as a spectral approach and the Fast-KRR method as a kernel machine learning approach have improved performance compared to the classical Ridge and Lasso penalization regression

Method	Medium SSE (5%, 95%)	nknots	Parameter
kNN	60.1 (57.1, 63.0)	-	neighbor=36
Ridge	1355.3 (1312.0, 1392.5)		$\lambda = 0.001$
Lasso	1354.8 (1311.4, 1391.9)		$\lambda = 0.001$
SpecSeries	71.2 (67.2, 74.1)	-	bandwidth = 0.1
Fast-KRR	115.9 (108.4, 124.2)	-	$\sigma = 1$
S-Kernel	60.4 (57.3, 64.7)	63	bandwidth = 4 r_{hns}
S-kNN	60.9 (58.0, 63.9)	76	neighbor = 36
S-Lspline	60.0 (57.0, 63.7)	63	-

Table A.5: Regression results on Yinyang $d = 2$ data. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5th percentile and 95th percentile of the SSEs from the given parameter settings are reported in brackets.

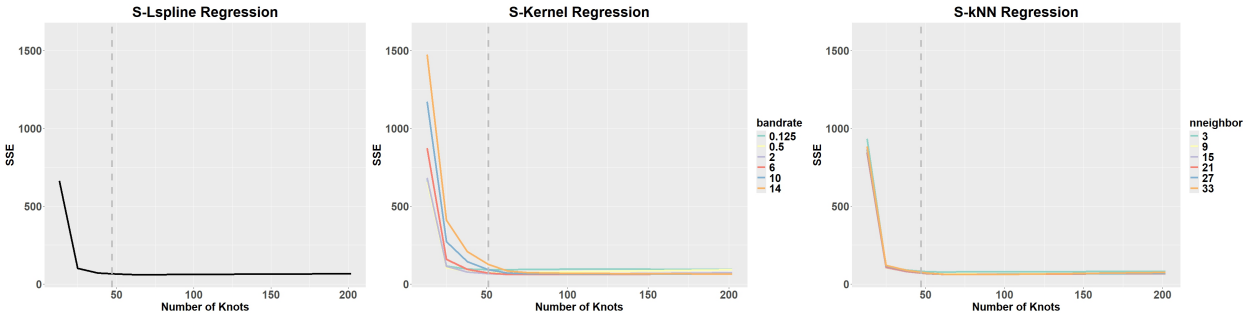


Figure A.42: Yinyang $d = 2$ data regression results with varying number of knots. The medium SSE across the 2 simulated datasets with each given parameter setting is plotted.

methods, but do not give the top performances. This can be due to the underlying data structure being comprised of multiple disconnected components which diminishes the power of such manifold learning methods. Therefore, the skeleton regression framework also gives competitive performance in datasets without noisy variables, but the advantage of skeleton-based methods is manifested more if the number of noisy variables in the input vector gets larger (see Appendix 3.4.2).

Noisy Yinyang Data d = 2

We follow the same data generation mechanism as in Section 3.4.3 to get Noisy Yinyang data without the additional variable dimensions and only have $d = 2$. We follow the same analysis procedure as in Section 3.4.3 and take the median, 5th percentile, and 95th percentile of the 5-fold cross-validation Sum of Squared Errors (SSEs) for each parameter setting of each method over the 100 simulated datasets. We present the smallest median SSE for each method in Table A.6 along with the corresponding best parameter setting, with the plots illustrating the effect of varying the numbers of knots included in Figure 3.7.

We observe similar patterns as shown in the Yinyang data that all the skeleton-based methods (S-Kernel, S-kNN, and S-Lspline) have performance comparable to the standard kNN in this setting. Ridge and Lasso regression methods result in relatively high SSEs, while the SpecSeries method and the Fast-KRR method have improved performance compared to the classical Ridge and Lasso penalization regression methods. Notably, the Spectral Series regression gives a top-level performance in this setting, and this can be due to the noisy observations adding a uniform density to the data space and making the structures in the

Method	Medium SSE (5%, 95%)	Number of knots	Parameter
kNN	228.5 (213.0, 244.3)	-	neighbor=6
Ridge	1938.5 (1906.0, 1973.0)	-	$\lambda = 0.005$
Lasso	1938.6 (1905.8, 1972.9)	-	$\lambda = 0.0016$
SpecSeries	243.9 (229.0, 259.0)	-	bandwidth = 0.1
Fast-KRR	497.8 (475.3, 520.6)	-	$sigma = 1$
S-Kernel	239.0 (223.2, 254.4)	226	bandwidth = 4 r_{hns}
S-kNN	250.7 (234.4, 264.7)	226	neighbor = 9
S-Lspline	234.3 (218.5 249.8)	226	-

Table A.6: Regression results on Noisy Yinyang $d = 2$ data.The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5 percentile and 95 percentile of the SSEs from the given parameter settings are reported in brackets.

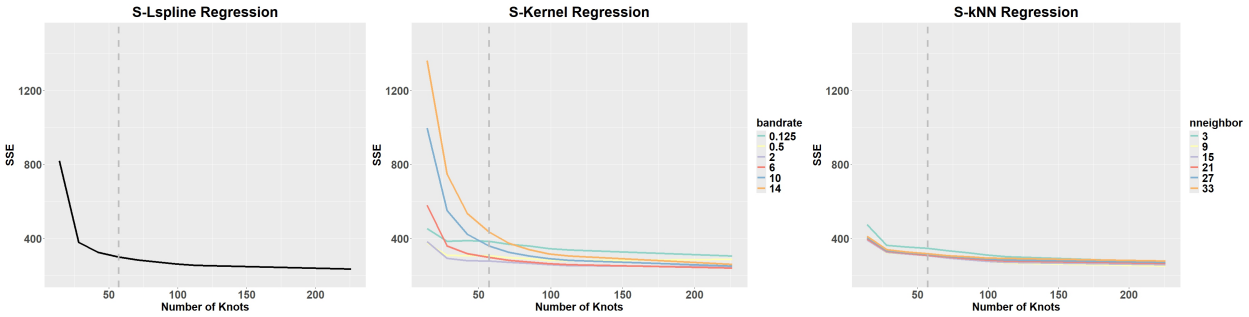


Figure A.43: Noisy Yinyang $d = 1000$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

Method	Medium SSE (5%, 95%)	nknots	Parameter
kNN	309.1 (281.5, 342.1)	-	neighbor=6
Ridge	1123.8 (1054.6, 1202.4)	-	$\lambda = 0.00126$
Lasso	1123.3 (1053.9, 1201.3)	-	$\lambda = 0.0025$
SpecSeries	331.8 (307.3, 351.8)	-	bandwidth $= 0.1$
Fast-KRR	563.8 (533.0, 598.6)	-	$\sigma = 1$
S-Kernel	348.1 (345.5, 365.3)	320	bandwidth $= 8\ r_{hns}$
S-kNN	363.5 (361.6, 406.7)	320	neighbor $= 9$
S-Lspline	368.9 (329.4, 409.1)	160	-

Table A.7: Regression results on SwissRoll $d = 3$ data. The smallest medium 5-fold cross-validation SSE from each method is listed with the corresponding parameters used. The 5 percentile and 95 percentile of the SSEs from the given parameter settings are reported in brackets.

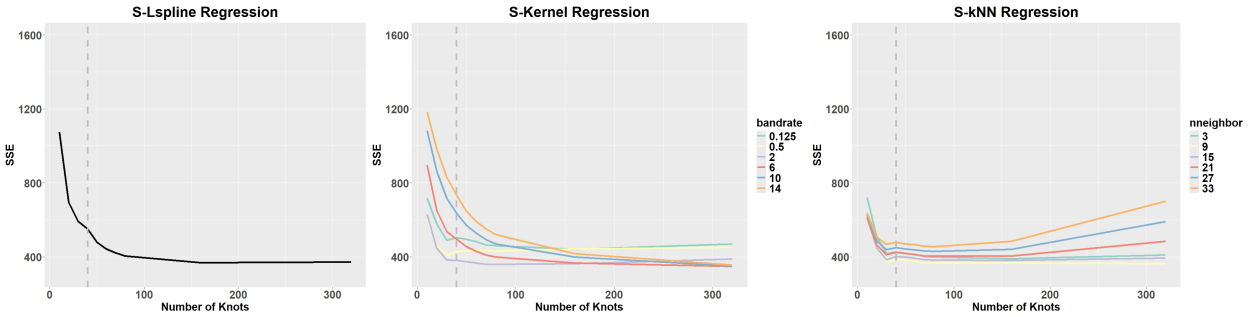


Figure A.44: SwissRoll $d = 3$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

data connected.

SwissRoll Data d = 3

We follow the same data generation mechanism as in Section 3.4.4 but with $d = 3$ so that no random normal variables are added as additional features. We present the smallest median SSE for each method in Table A.7 along with the corresponding best parameter setting, and the plots illustrating the effect of varying the numbers of knots are included in Figure 3.7.

The kNN method has the best performance in this setting, and the skeleton-based methods have comparable performance. Note that the Spectral Series approach has performance slightly better than the skeleton-based methods in this case, which corroborates its effectiveness in dealing with connected and smooth manifolds.

N Additional Real Data Examples

In this section, we present results on some additional examples from the COIL-20 dataset Nene et al. (1996), following the same procedure as in Section 3.5.1. Each dataset consists of 72 gray-scale images of size 128×128 pixels as 2D projections of a 3D object obtained through rotating the object by 72 equispaced angles on a single axis. The response is the angle of rotation, and to avoid the circular response issue, we remove the last 8 images from the sequence and only use the first 64 images from each dataset. We use leave-one-out cross-validation to assess the performance of each method.

Lucky Cat Data

This dataset consists of gray-scale images from rotating a lucky cat by equispaced angles on a single axis. Several examples of the images are given in Figure A.46. Following the same analysis procedure from Section 3.5.1, we use leave-one-out cross-validation, and the best SSE from each method is listed in Table A.8 along with the corresponding parameters.

We observe that the kNN method gives outstanding performance on this real-world data. The salient regions of black regions of the lucky cat make Euclidean distance between the pixel vectors reliable for this series of images, and hence the kNN method can always correctly

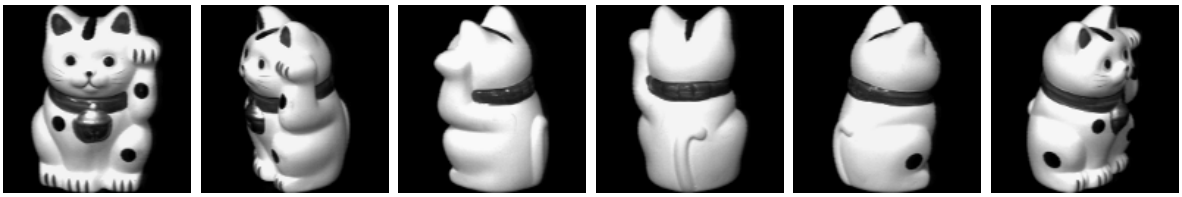


Figure A.46: A part of the lucky cat images from the COIL-20 processed dataset. Each image is of size 128 pixels.

Method	SSE	Parameter
kNN	175.0	neighbor=2
Ridge	-	-
Lasso	-	-
SpecSeries	-	-
Fast-KRR	-	-
S-Kernel	551.4	bandwidth = $0.125r_{hns}$
S-kNN	456.3	neighbor = 2
S-Lspline	338.1	-

Table A.8: Regression results on Lucky Cat data from COIL-20. The best SSE from each method is listed with the corresponding parameters used.

identify the two closest neighboring images and has the correct predictions and only give out errors at the beginning and ending parts of the sequence of images. Among the skeleton-based methods, the S-Lspline method gives the best SSE result, while the S-Kernel method using kernel smoothing on the skeleton-based distance gives a result worse than the other methods, which is similarly observed for the cup data in Section 3.5.1.

Sauce Box Image Data

We look at another sequence of images taken around a sauce box, with some example images in Figure A.48. The best SSE from each method is listed in Table A.9 along with the corresponding parameters. In this case, the usual kNN regressor gives the best performance in terms of SSE, while the S-Lspline method gives satisfactory results. The good performance of the kNN regressor can be due to the distinctive marks on the box, which makes neighbor search through Euclidean distance on the vectorized image inputs effective. However, the S-Lspline method explicitly models the latent structure of the data and can give a more structured model of the response. Among the skeleton-based methods, the S-Kernel and S-kNN methods using the skeleton-based distance give worse results, which is similarly observed for the cup data in Section 3.5.1.

Figure A.48: A part of the sauce images from the COIL-20 processed dataset. Each image is of size 128 pixels.

Method	SSE	Parameter
kNN	931.3	neighbor=2
Ridge	-	-
Lasso	-	-
SpecSeries	-	-
Fast-KRR	-	-
S-Kernel	1996.3	bandwidth = $0.5r_{hns}$
S-kNN	2450.0	neighbor = 1
S-Lspline	1220.1	-

Table A.9: Regression results on sauce images data from COIL-20. The best SSE from each method is listed with the corresponding parameters used.

Chapter 4 Appendices

O Simulation Details

To illustrate and expand on the results from the main text, we run a number of simulations. Here, we describe the simulations in detail.

Graph Generation

Graph geometry plays a key role in our results. We build a network as follows, to generate an empirical analogue to the L_n that we study theoretically. L_n is generated as a graph of n nodes in the following manner.

1. The base construction of the graph is a q-dimensional lattice, to mimic the properties of Assumption 1. We place n_{side} nodes evenly spaced on $[0,1]^q$, meaning that there are n_{side}^q nodes in the lattice portion of the graph.
2. The remainder of n nodes are placed uniformly at random throughout $[0,1]^q$.
3. All nodes, regardless of whether they are in the lattice or placed randomly, link to all nodes within distance r. We set r as:

$$r = \max\left\{\frac{1}{n_{side}-1}, \frac{\sqrt{q}}{2}\frac{1}{n_{side}-1}\right\}$$

This ensures that the graph is connected, even when q is large and thus nodes can be far apart.

We use the following parameters to generate L_n in the graphs used in the main texts. In the first specification, we set $n = 4,000$, $q = 4$ and $n_{side} = 7$. In the second specification, we set

$n = 4,000$, $q = 2$, and $n_{side} = 50$. To generate G_n, we add links with i.i.d. probability β_n. As a base rate, we use $\beta_n = \frac{1}{10n}$ – in one variant of parameters, we set $\beta_n = \frac{1}{100n}$. Summary statistics are shown in Table A.10 in the main text, and for additional simulations in Table A.11.

Table A.10: Graph statistics for L_n with $n = 4,000$ nodes

Statistic	L_n	G_n	L_n	G_n
Dimension	4.0	4.0	2.0	2.0
Diameter	19.0	11.609	93.0	20.439
Mean Degree	10.164	10.263	5.826	5.926
Min Degree	3.0	3.095	2.0	2.0
Max Degree	24.0	24.103	16.0	16.13
Mean Clustering Coefficient	0.265	0.258	0.379	0.37
Average Path Length	7.548	6.018	31.807	10.312

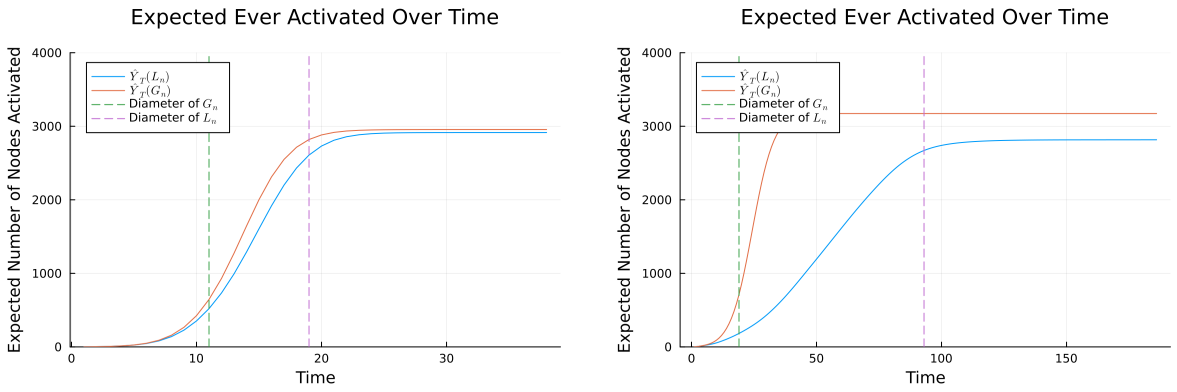
For $q = 4$, 60 percent of nodes are in the lattice, while with $q = 2$ 62.5 percent are. Statistics for G_n are the expectation over 2,500 draws of E_n, which is drawn Erdos-Renyi with $n = 4,000$ and $\beta_n = \frac{1}{10n} = \frac{1}{40000}$.

Diffusion Process

We use a Susceptible-Infected-Removed (SIR) diffusion process. Each node is infected (activated)for a single period and has the opportunity to transmit the process with i.i.d. probability p_n to each of its neighbors. After nodes are activated, they are removed and cannot be re-activated. We set the basic reproductive number to be $\mathcal{R}_0 = 2.5$, and set $p_n = \mathcal{R}_0/\bar{d}$, where $\bar{d}$ is the mean degree in L_n.

Simulation of Theorem 3

To investigate the content of Theorem 3, we directly simulate the sample analog. For 2,500 simulations, we do the following. We choose the initial seed i_0 uniformly at random and fix



(a) $q = 4$ (b) $q = 2$

Figure A.49: This figure plots the same information as Figure 4.2, but separated by graph for both $q = 4$ and $q = 2$. The trajectory of $\hat{Y}_T(L_n)$ initially lags behind that of $\hat{Y}_T(G_n)$, leading to the decrease in the ratio shown in Figure 4.2. As $\hat{Y}_T(L_n)$ catches up, the ratio increases.

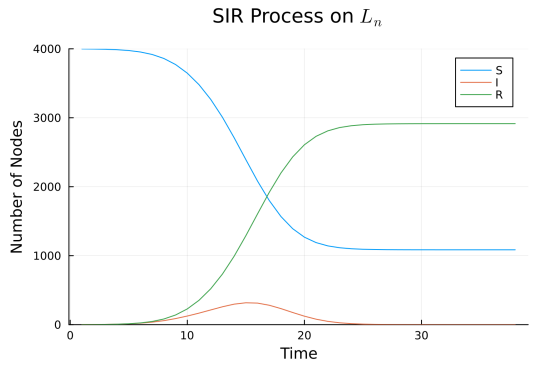
it throughout the process. The SIR process is simulated for T periods, where we set T to be twice the diameter of L_n.

1. Simulate the SIR process on L_n.
2. Generate a draw of E_n, with links i.i.d. with probability β_n.
3. We define $G_n := L_n \cup E_n$, and simulate the SIR process on G_n.

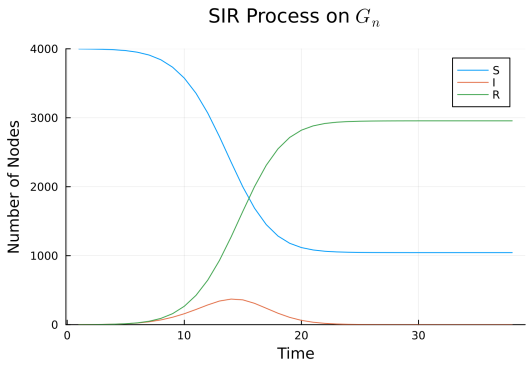
We track the number of ever-activated nodes in each simulation at each time step. We then take the average over simulations at each time step. In the main text, results are shown in Figure 4.2. Additional results are shown in Figures A.49 and A.50.

Simulation of Theorem 2

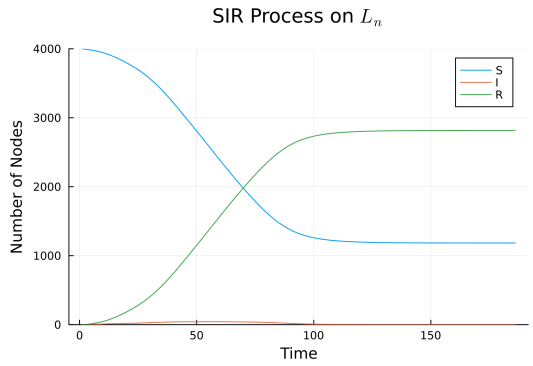
As an analog to Theorem 2, we simulate SIR processes on a fixed G_n with slightly perturbed starting points. We choose i_0 to be in the center of the lattice of L_n, which forms the backbone of G_n. Then, we build a set of alternative seeds J_{i_0}. First, we find the second



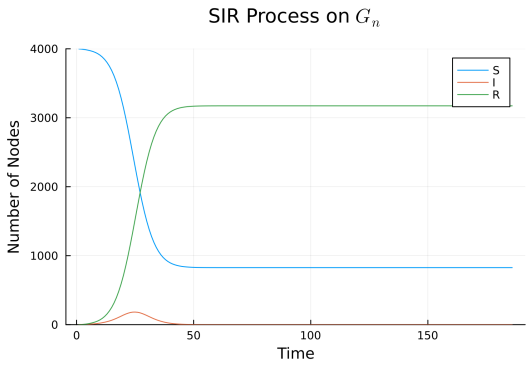
(a) $q = 4$, L_n



(b) $q = 4$, G_n



(c) $q = 2$, L_n



(d) $q = 2$, G_n

Figure A.50: Simulations meant to emulate Theorem 3, disaggregated into the standard SIR framework. The figure is a result of averaging over simulation draws. Note that we see a larger spike in activations under G_n, which makes intuitive sense – the additional links allow for more infections to occur. We show results for both $q = 4$ and $q = 2$, both with $\beta_n = \frac{1}{10n}$. Note that the gap between total activations with $q = 2$ is larger, as the additional links have a larger effect.

distance of the closest link in E_n – denote this $d(e_2)$. Then, all nodes at $d(e_2)+1$ are included in J_{i_0}. We then choose a $j_0 \in J_{i_0}$ uniformly at random.

The SIR process is then run, starting at both i_0 and j_0. We record which nodes are ever activated at each step of the process, under each simulation. To follow Theorem 2, we fix the percolation across the simulation starting at i_0 and j_0. To do so, we use the fact that for a one-period SIR model, each link can transmit the disease at most one time. Therefore, we can simulate ex-ante which links will be able to transmit, which occurs with probability p_n, and intersect this with G_n to get the realized percolation.

We then compute a standard Jaccard index to track the intersection of the two epidemics. Let $I_P(i_0)$ be the set of ever-infected nodes under the epidemic from i_0, and $I_P(j_0)$ be the corresponding set from j_0. Then, we compute:

$$\mathcal{J} := \mathbb{E}\left[\frac{|I_P(i_0) \cap I_P(j_0)|}{|I_P(i_0) \cup I_P(j_0)|} \mid G_n, P\right]$$

We define the Jaccard index $\mathcal{J}$ in a slightly different fashion than to Δ_n, the Jaccard index in Theorem 2. Note that $\mathcal{J}$ is a re-arrangement of Δ_n: the union of $I_P(i_0)$ and $I_P(j_0)$ contains the intersection and the disjoint set, which are the key pieces of Δ_n. We use this index because we do not condition on the event that the epidemics have *some* overlap (denoted as Γ_n and that there exists a path from j_0 to the second closes link to i_0 in E_n. Empirically, instances in which there is no overlap during some time periods are common, meaning that the empirical analog of Δ_n will not be well defined (as the denominator will be zero). Thus, we re-arrange terms and use the standard Jaccard index $\mathcal{J}$. Note that due to the re-arrangement, small values of $\mathcal{J}$ indicate very little overlap between epidemics, similar

to high values of Δ_n.

Aggregate Patterns Are Well-Approximated by Compartmental Models

As an additional exercise, we study the approximation of the diffusion process by a standard differential equations SIR compartmental model. In practice, diffusion occurs between a discrete set of n agents and transmission occurs over discrete time. The compartmental differential equations model simplifies matters by using a mean field approximation, but there is a potential for error which we now explore.

Instead of the network-based SIR model, we assume the policymaker estimates the parameters of a version of the standard differential equation SIR model. Changes in the number of susceptible ($S(t)$), infected ($I(t)$), and removed ($R(t)$) at time t are given by:

$$\dot{S}(t) := -\frac{s}{n} S(t-1) I(t-1)$$
$$\dot{I}(t) := \frac{s}{n} S(t-1) I(t-1) - rI(t-1)$$
$$\dot{R}(t) := rI(t-1)$$

Where s and r are parameters that govern the disease process. Note that $\mathcal{R}_0 = s/r$. This model is exactly a discrete-time analog of the standard SIR model.

We assume that the policymaker estimates $\hat{s}$ and $\hat{r}$ from observed data via a set of moment conditions, matching both the number of infected and removed people at each time step. It will be useful to define some additional notation. Let N be the number of simulations. Let $I_n^s(t)$ be the number of infected people at time t in simulation n. Let $R_n^s(t)$ be defined analogously for recovered. Let $I(t; s, r)$ be the number of infected at time t with parameters

r and s. Let $R(t; s, r)$ be defined analogously. Then, the policymaker solves the following problem for each simulation run, given T periods of data. We then collect the moment conditions in the following vector:

$$M_n(t) = \begin{pmatrix} I_n^s(t) - I(t; s, r) \\ R_n^s(t) - R(t; s, r) \end{pmatrix}$$

Then the policymaker solves:

$$\{\hat{s}_n, \hat{r}_n\} := \text{argmin}_{s,r} \frac{1}{T} \sum_{t=1}^{T} M_n(t)' M_n(t)$$

For each simulation. Then, we compute the following quantities, getting the average trajectory from the fitted SIR models.

$$\bar{I}(t) = \frac{1}{N} \sum_{n=1}^{N} I(t; s_n, r_n), \quad \bar{R}(t) = \frac{1}{N} \sum_{n=1}^{N} R(t; s_n, r_n), \quad \mathcal{R}_0 = \frac{1}{N} \sum_{n=1}^{N} \frac{s_n}{r_n}$$

We can also compare directly to the metric of average ever activated, our policy object of interest for much of the main text, by computing $\bar{I}(t) + \bar{R}(t)$ at each time period.

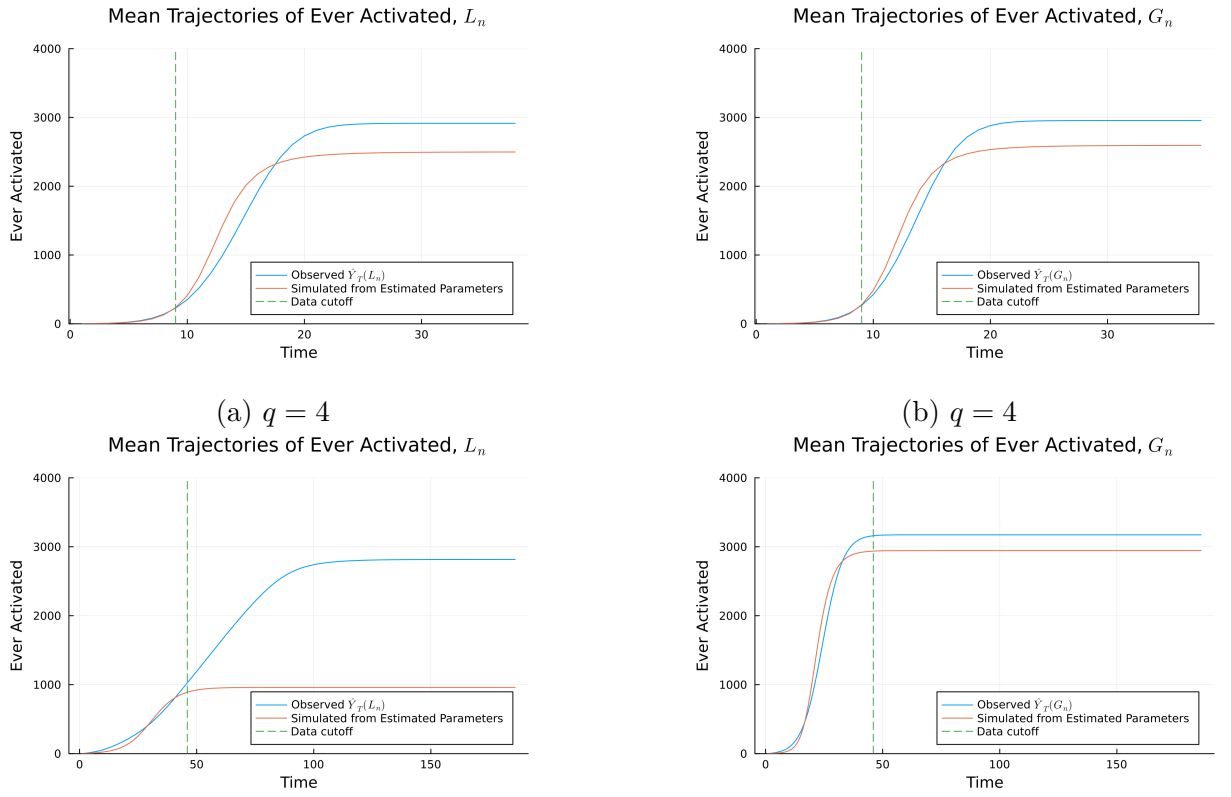
We conduct two exercises. In the first exercise, we simulate a diffusion process on G_n for T periods. We then estimate the parameters of interest, $(\hat{r}, \hat{s})$ at $\hat{t} = T/4$ and we generate forecasts from the compartmental model. We compare this to the actual diffusion trajectory. The second exercise replicates the first, with the only change being that we simulate the diffusion process on L_n instead. Note that this is not what generates the diffusion process in the "real world"—that is diffusion on G_n. However, together the two simulations capture two features: (a) the deviation of the mean-field model from the underlying discrete process and (b) how the deviation depends on the relative structure of G_n to $L_n = G_n - E_n$. We repeat both sets of simulations for both $q = 4$ and $q = 2$.

Figure A.51 presents the results. We begin with $q = 4$ and it is helpful to look at the diffusion on L_n first in Panel A.51a. Recall that this shows how well the mean-field approach captures the dynamics of a hypothetical network structure ignoring links in E_n. In the periods where the SIR process is fit to the simulated data, the fit is very good. The estimated $\hat{\mathcal{R}}_0$, derived by taking the average across simulations of $\hat{s}/\hat{r}$, is 1.46 under $\hat{Y}_T(L_n)$, well below the true $\mathcal{R}_0$ of 2.5. Note that while Lemma 11 implies that there exists a consistent estimator of $\mathcal{R}_0$, the estimator we propose in theory uses activation-level data. Here, we base our estimate of $\hat{\mathcal{R}}_0$ using the aggregate diffusion pattern. The estimated forecasts (in orange) diverge quickly from the true diffusion, $\hat{Y}(L_n)$. Because of the initially exponential growth structure of the compartmental model, early in the medium run, it overshoots, though the diffusion saturates much earlier, and in fact, the overall diffusion count, in the long run, is underestimated.

That is, in a sample, the compartmental model can be made to fit well, but with a lower growth rate for the number of ever-infected nodes. However, because of the lower implied $\mathcal{R}_0$, the compartmental model dramatically underestimates the total number of expected activations out of the sample. Ex-post, a policymaker could fit this type of model and do extremely well, but it would not be helpful for predicting the future trajectory.

In Panel A.51b, we turn to diffusion on G_n. The estimated $\hat{\mathcal{R}}_0$, derived by taking the average across simulations of $\hat{s}/\hat{r}$, is 1.52 under $\hat{Y}_T(G_n)$, still below the true $\mathcal{R}_0$ of 2.5. We find very similar results as the case with L_n. The principle difference is that the idiosyncratic links, E_n, generate a slightly closer forecast curve to the true trajectory. Also of note, the

implied diffusions from the compartmental model on L_n and G_n are also similar. While the estimates are such that the historical fit is quite good, the exponential structure makes the process run too fast and then fade too early as well, relative to a slower more persistent polynomial process (which of course could be historically fit by looking backwards).



(a) $q = 4$ (b) $q = 4$

(c) $q = 2$ (d) $q = 2$

Figure A.51: A comparison of the mean ever activated under the true network SIR model and the estimated trajectory from the differential equations model. Panel (A) and (B) use $q = 4$, while (C) and (D) use $q = 2$. Panel (A) shows simulations when $\hat{Y}_T(L_n)$ is used as the data generating process, while Panel (B) shows when $\hat{Y}_T(G_n)$ is used. The data cutoff is at $T/4$. Before this point, the compartmental SIR model is fit to the data via the generalized method of moments. After this point, we run the differential equation model forward to see how it performs out of sample.

For $q = 2$, there is a shift between L_n and G_n. With $q = 2$ and L_n, as seen in Panel A.51c, the process cannot be well approximated by the model. The fitted compartmental SIR looks almost nothing like the true trajectory: while fitting to data the SIR model makes

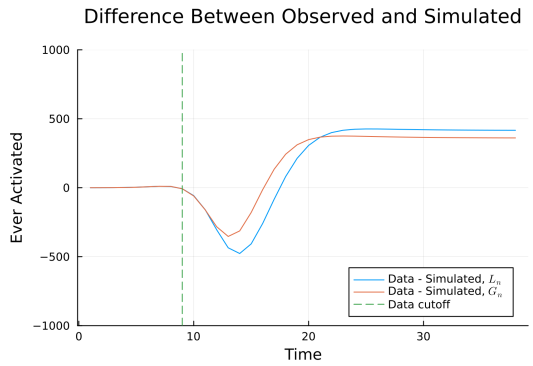
a complete "S" curve shape, it dramatically underestimates the total activations. Turning to the G_n case, as seen in Panel A.51d, the compartmental SIR model is able to match the data more closely, because the diffusion moves much more quickly.

For $\hat{Y}_T(L_n)$, the average (across simulations) root mean squared error (RMSE) is 11.43, while with $\hat{Y}_T(G_n)$ it is 11.89. Unsurprisingly, the RMSE under $\hat{Y}_T(G_n)$ is larger, as the data is inherently noisier. The simulated trajectories quickly diverge from the data out of sample. In the next $T/4$ periods, the average RMSE with $\hat{Y}_T(L_n)$ is 429.08, while with $\hat{Y}_T(G_n)$ it is 354.21. This divergence is shown in Figure A.51.

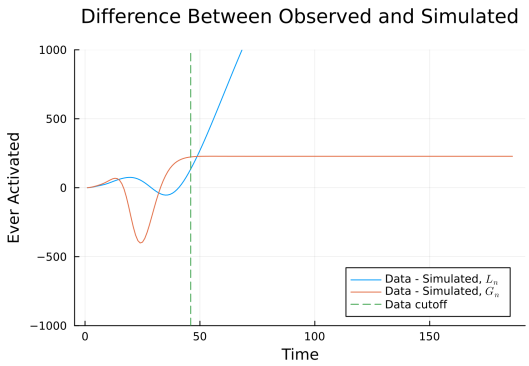
As an additional exercise, we plot the difference between the simulated forward and "true" trajectories under each data-generating process. Results are shown in Figure A.52. We can note that under the true data-generating process of $\hat{Y}_T(G_n)$, the maximum under and over-estimation by the SIR differential equation model is smaller than under $\hat{Y}_T(L_n)$. The additional i.i.d. links increase the degree of the polynomial, meaning that an exponential SIR model can more closely approximate the process. This effect is much larger with $q = 2$ than with $q = 4$, as this is when the SIR model approximates the process more poorly.

As discussed above, Figure A.53 demonstrates that the fitted value of $\hat{\mathcal{R}}_0$ is typically below the true value of $\mathcal{R}_0 = 2.5$. In particular, with $q = 2$ and L_n, the estimation procedure dramatically underestimates the true value of $\mathcal{R}_0$. As discussed in the main text, this is because the estimation procedure does not use the micro-data of exactly which nodes are activated and when, as suggested in Proposition 11.

In sum, a compartmental SIR model can, in many cases, fit well looking backward to a



(a) $q = 4$



(b) $q = 2$

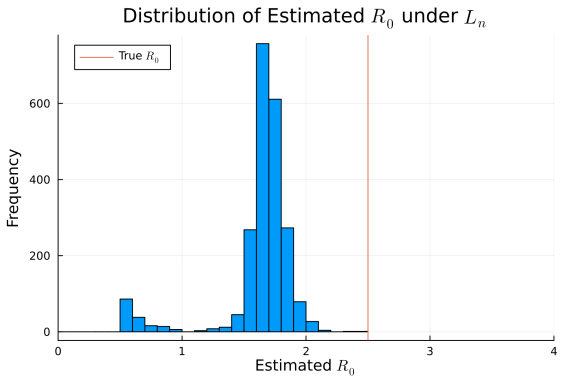
Figure A.52: Differences between $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$ and the fitted values from the differential equation SIR model, for both $q = 4$ and $q = 2$.

polynomial diffusion process. This fit is even better the higher the dimension of L_n, as it admits more expansive balls. But in all cases, the compartmental SIR estimates too rapid a diffusion that saturates and stabilizes too quickly: historical aggregate fits may be excellent and at the same time may serve as poor forecast tools.

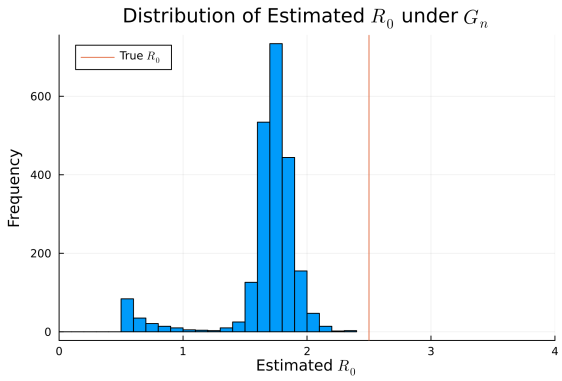
Extreme Sensitivity with $q = 2$

We explore an additional set of simulations in the case of $q = 2$, this time using a much smaller value of $\beta_n = \frac{1}{100n}$. We show average graph statistics in Table A.11. Results are shown in Figures A.54.

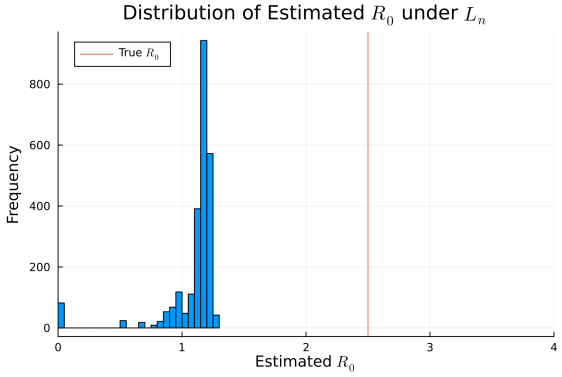
As shown in Figure A.54, despite a much smaller value of β_n forecasting issues persist. The minimum value of $\hat{Y}_T(L_n)/\hat{Y}_T(G_n)$ is achieved at $T = 46$, taking a value of 0.649. This value is still lower than the case with $q = 4$ and $\beta_n = \frac{1}{10n}$ (which had a minimum of 0.780), showing the extreme sensitivity in the lower dimension. Note that over very short time ranges, the value of the ratio is slightly above 1 – this is a result of finite sample noise,



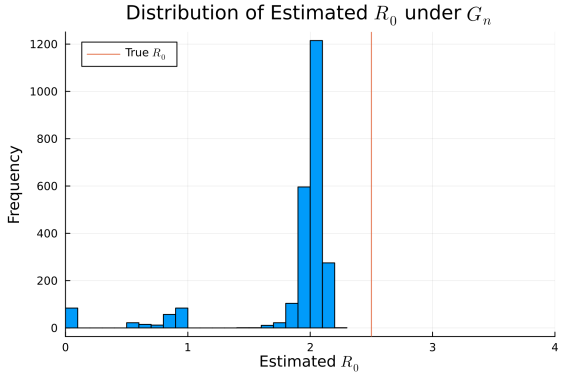
(a) $q = 4$



(b) $q = 4$



(c) $q = 2$



(d) $q = 2$

Figure A.53: Distribution of estimated $\hat{\mathcal{R}}_0$ across simulations when L_n is based on $q = 4$. Note that the distribution of values sits below the true value of $\mathcal{R}_0 = 2.5$. Values very close to zero come from data where the epidemic stops after a very small number of activations.

with several diffusion processes on L_n infecting a large number of nodes quickly, and a few processes on G_n infecting very few nodes. For sensitive dependence, j_0 is at a distance 16 from i_0: this much larger distance comes from both the clustered nature of the graph, and the lack of i.i.d. links to connect disparate locations (due to the low value of β_n). Because there are so few links in E_n, due to the small value of β_n, the local neighborhood containing all j_0 is 7.13 percent of the graph, and only 10.90 percent of the neighborhood is candidate j_0. With this in mind, it is not surprising to see the process exhibit severe sensitive dependence on the seed location: at half of the diameter of G_n ($T = 22$), the value of $\mathcal{J} = 0.09$ on

Table A.11: Graph statistics for L_n generated with $q = 2$ and G_n generated with $\beta_n = \frac{1}{100n}$

Statistic	L_n	G_n
Dimension	2.0	2.0
Diameter	93.0	45.059
Mean Degree	5.826	5.836
Min Degree	2.0	2.0
Max Degree	16.0	16.007
Mean Clustering Coefficient	0.379	0.38
Average Path Length	31.774	18.802

Statistics for G_n are taken as an average over 2,500 draws.

average, indicating almost totally disjoint diffusion processes.

The third and fourth panels of Figure A.54 show the compartmental SIR fitting exercise. Here, the introduction of E_n has less of an impact, as shown by the relative similarity between the results for L_n and G_n. This result is not surprising, given the very small value of β_n. Recall that we fit the SIR model to $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$, over the first 46 time steps (corresponding to $T/4$, equivalent to half of the diameter of L_n). In the fitting period, using $\hat{Y}_T(L_n)$, the average RMSE is 62.069, while in the next $T/4$ periods it is 1235.168 – a very similar set of values to the $q = 2$ case in the main text. With $\hat{Y}_T(G_n)$, the within-sample average RMSE is 101.128, while in the next $T/4$ periods it is 1242.687. These values are much more similar to the L_n case than the corresponding values for $q = 2$ in the main text – this is because there are many fewer additional links in G_n. Therefore, while the additional links increase the dimensionality of the diffusion process, the compartmental SIR model still gives a poor approximation. As further evidence, in both cases, the compartmental model dramatically underestimates the true value of $\mathcal{R}_0 = 2.5$: under L_n it is estimated as 1.10, and under G_n it is estimated as 1.21.

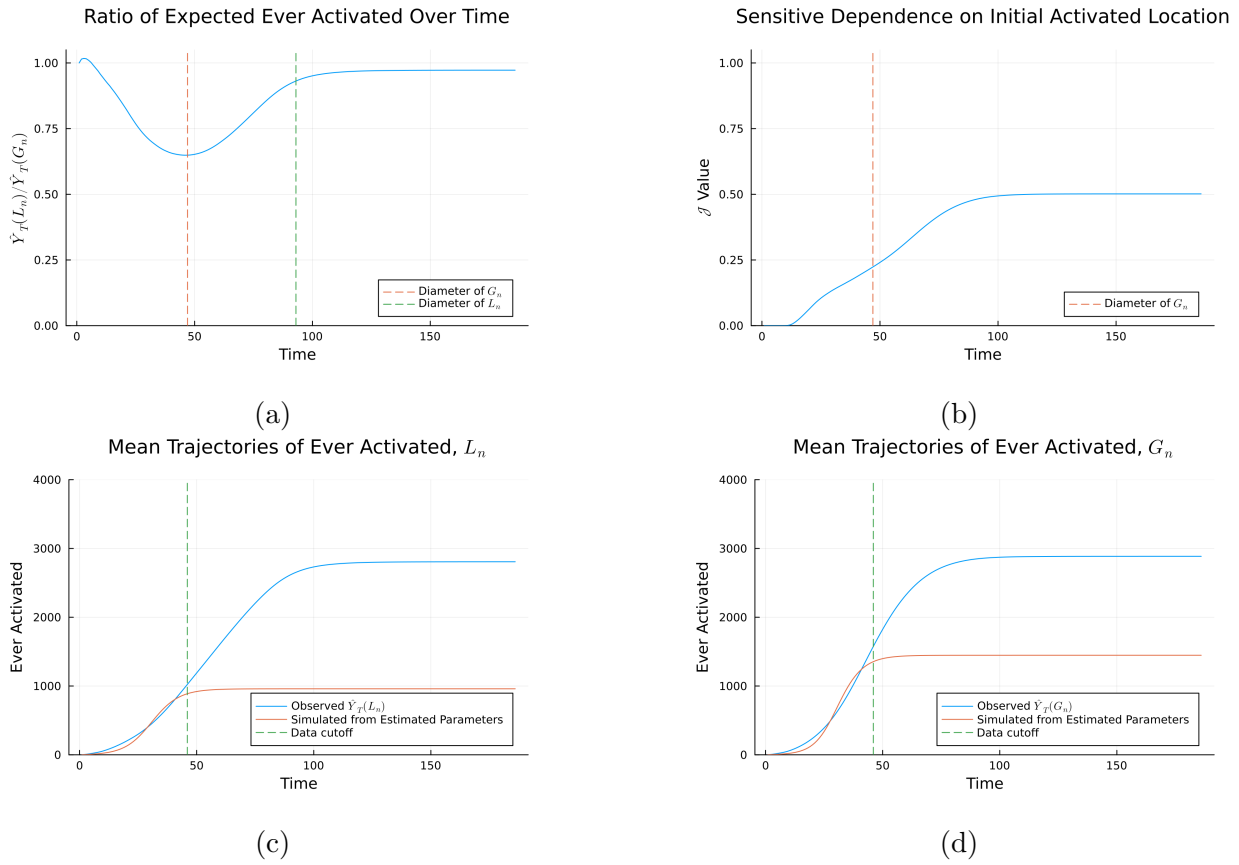


Figure A.54: Results with $q = 2$ and $\beta_n = \frac{1}{100n}$. Panel (A) shows the ratio $\hat{Y}_T(L_n)/\hat{Y}_T(G_n)$, while Panel (B) shows the Jaccard index $\mathcal{J}$. Panels (C) and (D) show the fitted values of a SIR differential equation model, fit to $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$. Averages are taken over 2,500 Monte Carlo simulations.

P Empirical Example: Location Data from the COVID-19 Epidemic

We give a detailed description of the data processing procedures, along with additional results using a graph constructed from location data. We build a network using visitor flows based on cell phone location data, provided by SafeGraph (Kang et al., 2020). Our primary analysis studies the entirety of California and Nevada, with a small portion of Arizona included. Note that we only include areas in the United States. The region includes major cities including San Francisco, Los Angeles, and Las Vegas. We work with Census tracts as the unit of

observation, which each contain approximately 4,000 people. Given privacy concerns, we focus on movement between tracts, rather than tracking individual people. We use tract-to-tract flows on March 1st, 2020. This date was before the WHO declared COVID-19 a pandemic, and before the United States government declared a national state of emergency.

We construct graphs in the following manner. Fix a cutoff c. Then we take the following steps.

1. For each pair of Census tracts a and b, we construct the average flow between tracts by taking the average of the flow from a to b and the flow from b to a. Call this value f_{ab}.
2. Tracts a and b will be linked in the graph only if $f_{ab} > c$.

We choose c based on the empirical distribution of f_{ab}, the flows between tracts. We refer to this procedure as "pruning." If the process results in a disconnected graph, we choose the largest connected subgraph. As before, we set T as twice the diameter of L_n.

Disease Process

As with the simulated graphs, we fix $\mathcal{R}_0 = 2.5$. We then compute $p_n = \mathcal{R}_0/\bar{d}$, where $\bar{d}$ is the average degree in L_n. Note that in this case, the meaning of $\mathcal{R}_0$ is substantively different – because nodes now refer to Census tracts, infecting 2.5 nodes in the disease-free state on average means infected 2.5 tracts on average.

Errors Induced by Cutoff Choice

We first study errors induced by choosing different cutoffs for pruning the graph. We construct G_n by setting $c = 5$, which is at the 91st percentile of the empirical distribution of tract-to-tract flows. Then, we generate L_n by choosing $c = 6$. Note that every link in L_n will be in G_n, meaning that we can construct the implied error graph E_n.

We conduct the same three analyses that we did with the simulated graph. First, we study a version of Theorem 3, comparing the expected number of infections on each graph. Second, we study a version of Theorem 2, comparing the overlap between epidemics after perturbing the starting point. Finally, we consider the exercise of fitting a SIR differential equation model.

For the sake of brevity, we only note differences unique to this section when compared to the procedures discussed in Section O. When considering the simulation of Theorem 3, the key change is that we hold G_n fixed: it is generated once from the data. When we take expectations, they are taken only over the disease process only. Otherwise, the process is identical. When considering the simulation of Theorem 2, the only change is how i_0 is selected – we set i_0 to be the node with the highest degree in G_n. The process of fitting a differential equation SIR model is exactly as before. In addition, we conduct simulations with E_n taken to be an Erdos-Renyi random graph, rather than via the pruning procedure. In the main text, we set β_n so that the i.i.d. errors generate the same expected volume of links as the pruning procedure. As an additional set of results, we set $\beta_n = \frac{1}{10n}$, to compare with the Monte Carlo simulations. Summary statistics of the resulting graphs are shown in

Table A.12.

Table A.12: Graph statistics for L_n and both hypothetical G_ns constructed from California, Nevada, and Arizona Census tract flow data

Statistic	L_n	G_n^{92}	G_n^{β}
Error Type	—	Pruned	IID
Diameter	21.0	15.0	7.687
Mean Degree	12.962	15.486	16.172
Min Degree	1.0	1.0	1.839
Max Degree	298.0	329.0	301.148
Mean Clustering Coefficient	0.389	0.393	0.234
Average Path Length	7.253	5.866	4.03

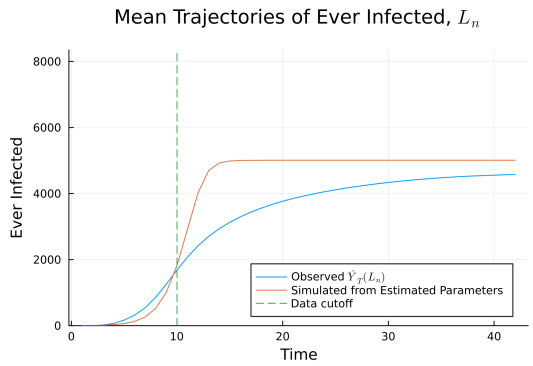
Statistics for G_n^{β} with i.i.d. errors are averaged over 2,500 draws.

Additional Results

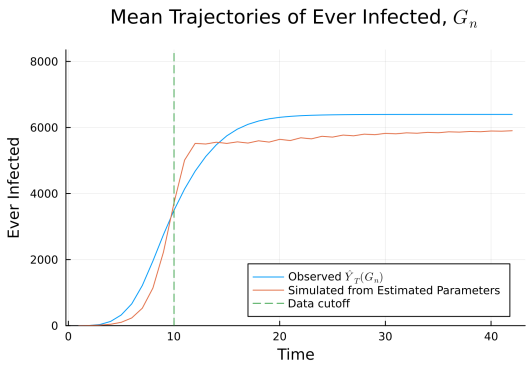
We again estimate the compartmental SIR model using the simulated epidemics above. This process is identical to the procedure conducted in Section 4.7. One pattern of note is that the model fit to $\hat{Y}_T(G_n)$ generated from the pruning procedure underestimates the average number of infections, while the model fit to $\hat{Y}_T(L_n)$ overestimates.

In the estimation period before $T/4$, the RMSE for $\hat{Y}_T(L_n)$ is 202.98, while in the next $T/4$ periods it is 1953.41. When fit to $\hat{Y}_T(G_n^{93})$, the RMSE in the first $T/4$ periods is 452.09, while in the next $T/4$ periods it is 1320.60. Notably, the model has a much better fit out of sample for G_n^{93}. For the i.i.d. errors on G_n^{β}, the results are similar. In the estimation period, the RMSE fitted to $\hat{Y}_T(L_n)$ is 200.541, while in the next $T/4$ periods it is 1944.63. When fit to $\hat{Y}_T(G_n)$, the RMSE in the first $T/4$ periods is 700.93, while in the next $T/4$ periods it is 1095.14.

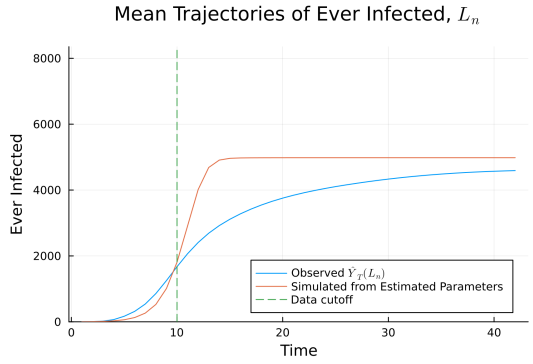
We then show a set of additional figures, corresponding to the simulations from the



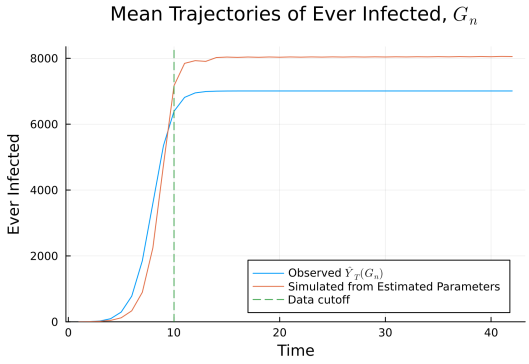
(a) Pruned



(b) Pruned



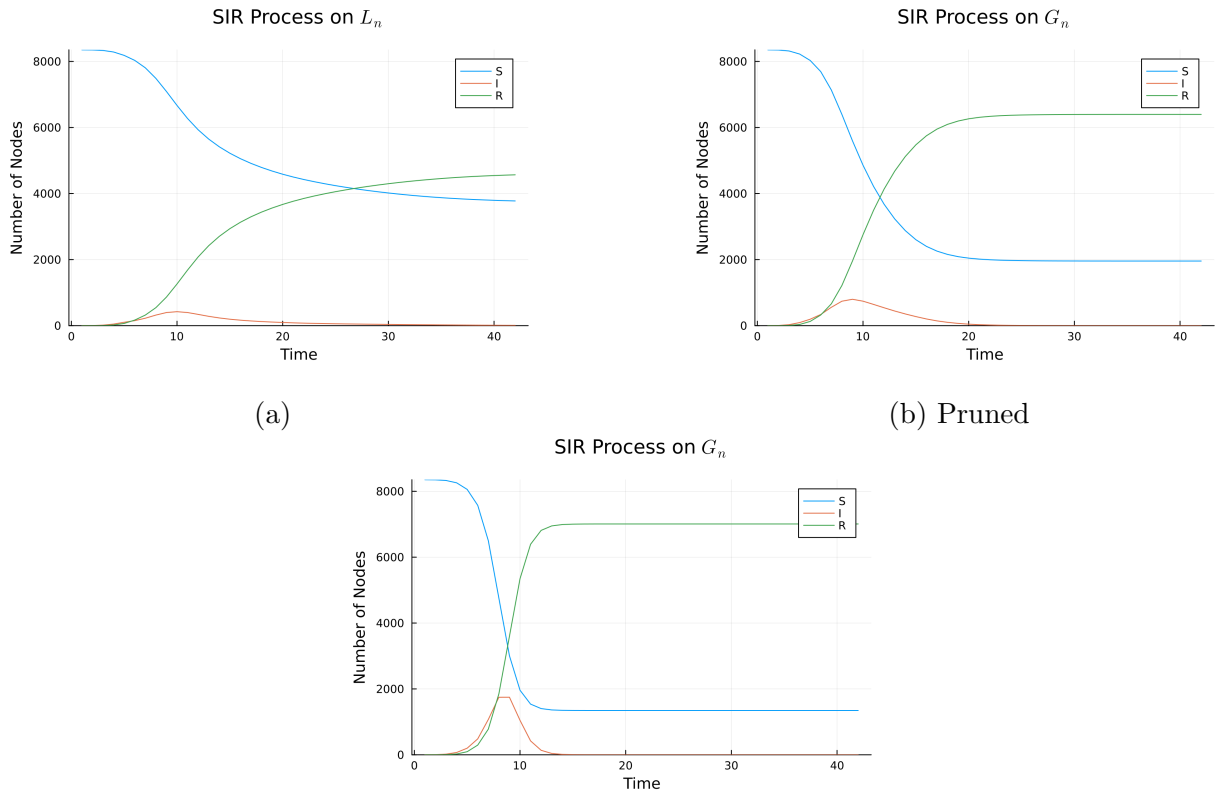
(c) I.I.D.



(d) I.I.D.

Figure A.55: A comparison of the mean ever infected under the true network SIR model and the estimated trajectory from the differential equations model. Here, L_n is generated from location flow data in California, Nevada, and a portion of Arizona. Panel (A) and (B) use the pruning procedure, while (C) and (D) have i.i.d. links. Panel (A) shows simulations when $\hat{Y}_T(L_n)$ is used as the data generating process, while Panel (B) shows when $\hat{Y}_T(G_n)$ is used. The data cutoff is at $T/4$. Before this point, the SIR model is fit to the data via the generalized method of moments. After this point, we run the differential equation model forward to see how it performs out of sample.

main text. We first disaggregate the simulated diffusion processes into a standard SIR framework, as shown in Figure A.56. Second, we show the distribution of estimated $\hat{\mathcal{R}}_0$ across simulations in Figure A.57. Figure A.56 demonstrates that with i.i.d. errors, the infection profile is relatively sharp, as the epidemic quickly expands to cover the whole graph during the intermediate range of T.



(a)

(b) Pruned

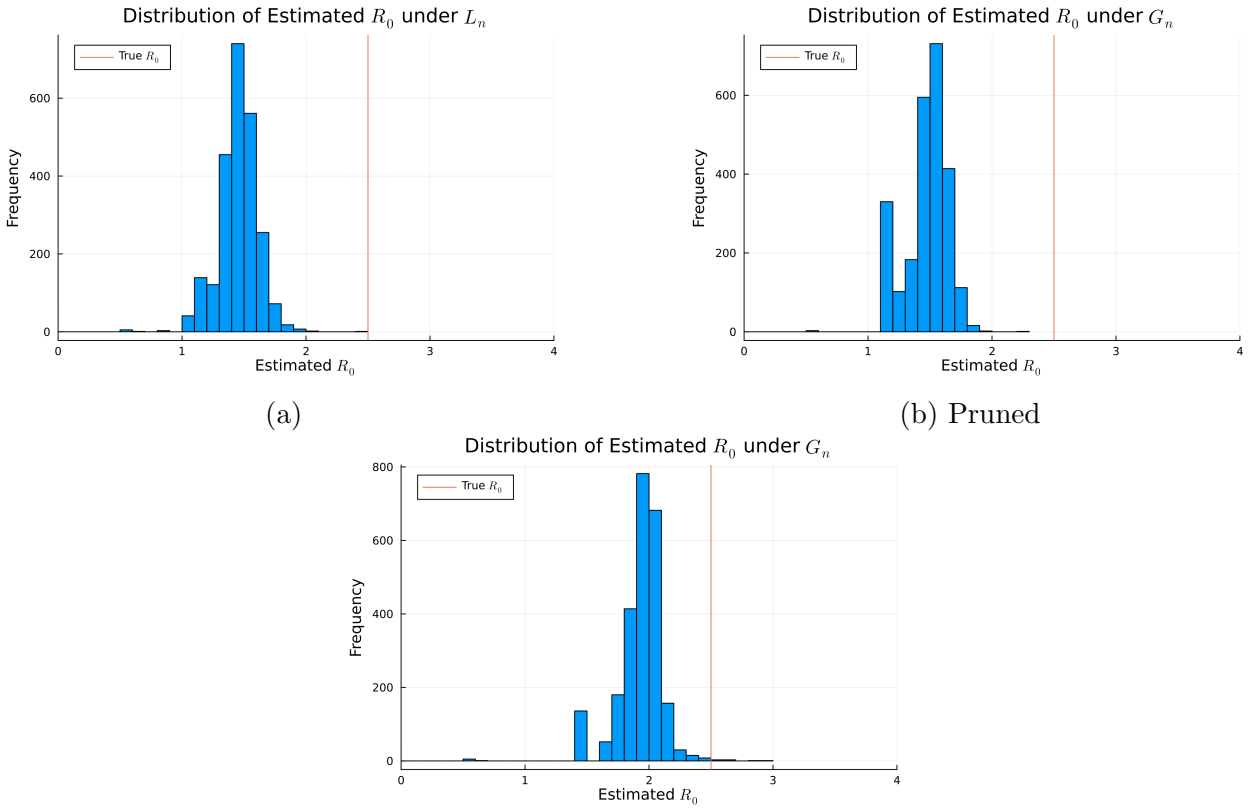
(c) I.I.D.

Figure A.56: Trajectories of $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$ disaggregated into the standard SIR curves for L_n and G_n for each scenario. Note that the L_n specifications are identical, as it is exactly the same graph.

Lower Rates of I.I.D. Errors

To make a more direct comparison to the Monte Carlo simulations, we repeat the simulation exercises using E_n generated i.i.d. with $\beta_n = \frac{1}{10n}$. Graph statistics are shown in Table A.13, again for L_n, and the average statistics for G_n over 2,500 draws of E_n. Compared to G_n in the main text (in Table A.12), note that the change in degree, clustering, and average path length are all much smaller, as E_n is much more sparse in this case.

Results are shown in Figure A.58. We take averages over 2,500 simulations. The first panel shows the simulation of Theorem 3. Note that in this case, the minimum ratio of



(a) (b) Pruned

(c) I.I.D.

Figure A.57: The distribution of values of $\hat{\mathcal{R}}_0$ estimated when fitting the compartmental SIR model to the COVID-19 travel data.

$\hat{Y}_T(L_n)/\hat{Y}_T(G_n)$ is achieved at $T = 18$ and takes the value 0.686. This value is much larger than the values from the main text with either the pruned or i.i.d. errors and comparable to the values with the same level of β_n and graph dimension $q = 4$ in the Monte Carlo simulations. The second panel shows the simulation of Theorem 2. As in the main text, we choose the local neighborhood containing all j_0 conservatively: we chose the set to be all nodes within distance 2 of i_0. The distance from i_0 to j_0 is therefore 2, and the neighborhood that contains all possible j_0 contains 0.80 percent of the graph. Of the neighborhood, 89.55 percent of the nodes are candidates for j_0. Halfway to the diameter of G_n, the value of the

Table A.13: Average graph statistics with i.i.d. errors in the travel data for California, Nevada, and a small portion of Arizona

Statistic	L_n	G_n
Diameter	21.0	16.874
Mean Degree	12.962	13.062
Min Degree	1.0	1.0
Max Degree	298.0	298.106
Mean Clustering Coefficient	0.388	0.38
Average Path Length	7.295	6.116

G_n is generated from L_n using i.i.d. additional links, which occur with $\beta_n = \frac{1}{10n}$.

average Jaccard index is 0.24, indicating largely distinct epidemics.

The third and fourth panels of Figure A.58 show the fitted compartmental SIR models, relative to $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$. As before, the compartmental model underestimates the true $\mathcal{R}_0 = 2.5$: under $\hat{Y}_T(L_n)$, it estimates a value of 1.40, and under $\hat{Y}_T(G_n)$ estimates a value of 1.49. In the first $T/4$ periods, in sample, the average RMSE under $\hat{Y}_T(L_n)$ is 198.96. In the next $T/4$ periods, it is 1,966.58. Under $\hat{Y}_T(G_n)$, in sample, the average RMSE is 222.11, whereas in the next $T/4$ periods it is 1389.65. Similar to the Monte Carlo exercise, we see that the additional links in E_n help increase the dimensionality of the epidemic, leading to a better fit with the exponential compartmental model.

Q Empirical Example: Diffusion in Mobile Phone Marketing

We use data from Banerjee et al. (2019) as an additional empirical example of our diffusion results. We build 69 separate village graphs, by composing networks based on survey data from Karnataka, India. We have a number of directed networks:

1. Relative

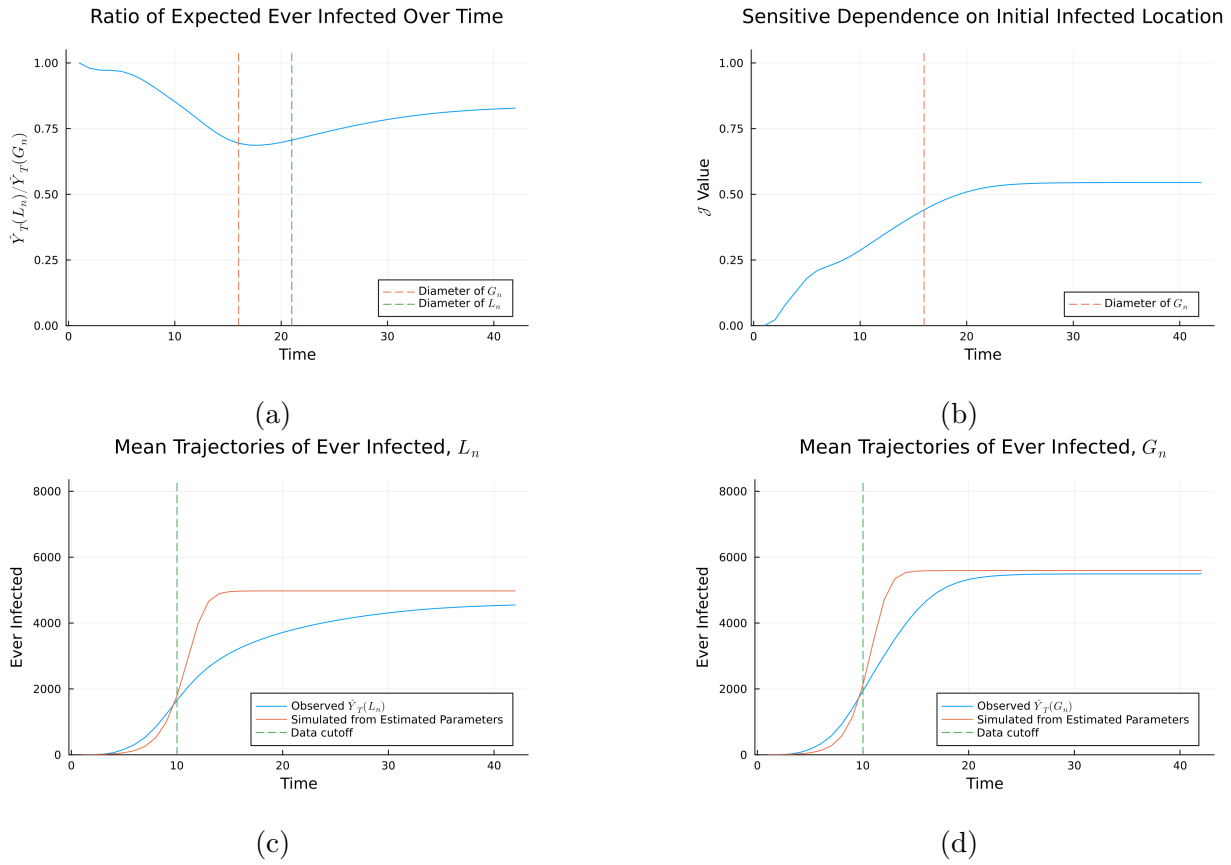


Figure A.58: Results using the COVID-19 travel data, with G_n using E_n generated i.i.d. with $\beta_n = \frac{1}{10n}$. Panel (A) shows the ratio $\hat{Y}_T(L_n)/\hat{Y}_T(G_n)$, while Panel (B) shows the Jaccard index $\mathcal{J}$. Panels (C) and (D) show the fitted values of a SIR differential equation model, fit to $\hat{Y}_T(L_n)$ and $\hat{Y}_T(G_n)$. Averages are taken over 2,500 Monte Carlo simulations.

2. Give advice: does the household i give advice to household j
3. Seek advice: does household i get advice from household j
4. Go to visit: does household i visit household j in free time
5. Come to visit: does household i come visit household j in free time
6. Borrow: does i borrow kerosene or rice from household j
7. Lend: does i lend kerosene or rice to household j

To construct a set of undirected networks for each village, we take the union of these seven

networks. Links are assumed to be undirected, and the network is made symmetric. This network data comes from a sequence of studies conducted in Karnataka, India. We use the 2012 data in our setting, the more recent of two waves of data collection. Graph statistics are shown in Table A.14.

Table A.14: Average village graph information from Banerjee et al. (2019).

Statistic	L_n	G_n
Nodes	196.072	196.072
Diameter	7.087	6.787
Mean Degree	6.541	6.849
Min Degree	1.0	1.004
Max Degree	25.71	26.219
Mean Clustering Coefficient	0.228	0.199
Average Path Length	3.303	3.168

For L_n, averages are taken across the 69 villages in our sample. For G_n, averages are taken across the 69 villages and 2,500 draws of E_n, where E_n is generated with $\beta_n = \frac{1}{2n_v}$ in each village separately, where n_v is the number of households in the village.

R Empirical Example: Peer Effects in Insurance

We use data from Cai et al. (2015) to investigate an example with peer effects in a diffusion setting. In order to encourage weather insurance, a valuable product with low takeup in rural China, the researchers conducted two waves of information sessions.

To construct network data, we use the list of directed links given in their data along with additional survey data. We drop some households who are listed in the network data but not in the additional survey data – we assume that this is a result of attrition between the surveys. We then transform the directed network in each village into an undirected network: if household i lists household j as a friend, or vice versa, we link i to j.

We use the same definition of treatment as in Cai et al. (2015). A household is considered

to be treated if they participate in an intensive information session in the first wave of the experiment. We then compute diffusion exposure using these households as seeds. When we estimate the effect of diffusion exposure, we include only households that did not participate in the first wave of the information sessions. This procedure is consistent with the prior research.

In addition, we include a number of controls to be in line with the original paper. We control for the head of household gender, age, education, and area of rice production. In addition, following the approach in Cai et al. (2015), we control for the degree to address potential concerns about selection on household sociability. Finally, we include village fixed effects. Tables A.15 and A.16 report graph summary statistics for all values of k for the Monte Carlo simulations conducted in the main text.

Table A.15: Average graph statistics from Cai et al. (2015)

Graph Statistic	Value
Nodes	104.30
Min Degree	0.40
Max Degree	15.79
Mean Degree	6.51
Components	5.60
Average Path Length	3.59
Diameter	8.06
Local Clustering	0.30
Exposure	0.99

Averages are taken over the 47 villages in the data. When there are multiple components, paths of infinite length (when nodes are disconnected from one another) are ignored. Mean exposure is computed before standardizing to have mean zero and standard deviation one, as we do in the regressions.

Table A.16: Graph statistics for the average graph L_n generated by dropping links with i.i.d. probability $\beta_n = \frac{1}{kn_v}$ in each village

k	MinDeg	MaxDeg	MeanDeg	Comp.	PathLen.	Diam.	Clus.	Exposure
-	0.38	13.32	5.60	5.60	3.59	8.06	0.30	1.10
15.00	0.37	13.19	5.54	5.63	3.61	8.10	0.29	1.10
14.00	0.37	13.19	5.53	5.64	3.61	8.10	0.29	1.10
13.00	0.37	13.17	5.53	5.64	3.61	8.10	0.29	1.10
12.00	0.37	13.16	5.52	5.64	3.61	8.11	0.29	1.10
11.00	0.37	13.15	5.51	5.65	3.61	8.11	0.29	1.10
10.00	0.37	13.13	5.51	5.65	3.62	8.12	0.29	1.10
9.00	0.37	13.11	5.49	5.66	3.62	8.12	0.29	1.10
8.00	0.36	13.09	5.48	5.67	3.62	8.13	0.29	1.10
7.00	0.36	13.05	5.46	5.68	3.62	8.14	0.29	1.10
6.00	0.36	13.01	5.44	5.69	3.63	8.15	0.29	1.09
5.00	0.35	12.95	5.40	5.71	3.64	8.17	0.28	1.09

"Comp." stands for the number of components. "PathLen." stands for path length. "Diam." stands for diameter. "Clus." stands for the clustering coefficient.

S Additional Theoretical Results

Decaying Diffusion in the Polynomial Case

Assumption .1 (Polynomial Diffusion Process). For some constant $q > 1$ and all discrete-time t, $\mathcal{E}_t = \Theta(t^{q+1})$ and $\mathcal{S}_t = \mathcal{E}_t - \mathcal{E}_{t-1} = \Theta(t^q)$. Furthermore, let $p_n(T) = \frac{p_{n,0}}{T^\lambda}$ for some constant $0 < \lambda < q$ and $p_{n,0} \in \left(\left(\frac{1}{\log n} \right)^{\frac{q}{2q+2}}, 1 \right]$.

We assume that diffusion decays at a polynomial rate over time, governed by the constant λ. It follows that we only have expected diffusion through links in E_n if $\lambda < q$. In the case where $p_n(T)$ were to decay exponentially quickly, it would then follow we still would not have expected diffusion through E_n. With fast decay on $p_n(T)$, the missing links do not have a large impact because the diffusion process just dies before hitting regions that cause lots of damage.

We note that the graph classes that are allowed under the homogenous p_n are still valid here, with sufficiently slow decay of p_n. For example, again consider a latent space network where nodes form links locally in a Euclidean space with dimension q. Since volumes in Euclidean space expand at a polynomial rate and for $\lambda < q$, this ensures that Assumption .1 will be satisfied.

Assumption .2 (Forecast Period). We impose that the sequence T_n has for each n, $T_n \in [\underline{T}_n, \overline{T}_n]$ where the following holds:

1. $\overline{T}_n = \min\left\{n^{\frac{1}{q+1}}, \left(\frac{n}{p_{n,0}}\right)^{\frac{1}{q-\lambda}}\right\}$
2. $\underline{T}_n = (p_{n,0}\log n)^{\frac{1}{q+\lambda+2}}$.

Again, the assumption is very close to that of the main text, adapting the constants to deal with the decaying diffusion rates. We can note that the time frame considered will generally start earlier, but also potentially end earlier.

Assumption .3. $\beta_n \in \left(\frac{1}{np_{n,0}T^{q-\lambda}}, \frac{1}{n}\right)$.

Compared to the homogeneous diffusion rate case where we assume $\beta_n \in \left(\frac{1}{np_{n,0}T^{q}}, \frac{1}{n}\right)$, with decaying $p_n(T)$ we impose a larger missing link rate. Under these conditions, a similar result to Theorem 3 holds.

Theorem .1. Under Assumptions .1, .2, and .3, as $n \to \infty$, $\frac{\hat{Y}_T(L_n)}{\hat{Y}_T(G_n)} \to 0$.

This result forms a direct analog of Theorem 3 with a decaying diffusion rate, and thus the proof is omitted as it proceeds in an identical manner.

Bibliography

E. Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

D. Acemoglu, V. Chernozhukov, I. Werning, and M. D. Whinston. Optimal targeted lockdowns in a multigroup SIR model. *American Economic Review: Insights*, 3(4):487–502, 2021.

A. Advani and B. Malde. Credibly identifying social effects: Accounting for network formation and measurement error. *Journal of Economic Surveys*, 32(4):1016–1044, 2018.

S. Alam, F. D. Albareti, C. A. Prieto, F. Anders, S. F. Anderson, T. Anderton, B. H. Andrews, E. Armengaud, É. Aubourg, S. Bailey, et al. The eleventh and twelfth data releases of the sloan digital sky survey: final data from sdss-iii. *The Astrophysical Journal Supplement Series*, 219(1):12, 2015.

Y. Alimohammadi, C. Borgs, R. van der Hofstad, and A. Saberi. Epidemic forecasting on networks: Bridging local samples with global outcomes. Technical report, Working paper, 2023.

J. Alsing, N. Usher, and P. J. Crowley. Containing covid-19 outbreaks with spatially targeted short-term lockdowns and mass-testing. *medRxiv*, 2020. doi: 10.1101/2020.05.05.20092221. URL https://www.medrxiv.org/content/early/2020/05/28/2020.05.05.20092221.

N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician*, 46:175–185, 1992. ISSN 15372731. doi: 10.1080/00031305.1992.10475879.

N. Amenta, D. Attali, and O. Devillers. Complexity of delaunay triangulation for points on lower-dimensional polyhedra. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1106–1113, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

R. M. Anderson and R. M. May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1991.

B. Aragam, C. Dan, E. P. Xing, and P. Ravikumar. Identifiability of nonparametric mixture models and Bayes optimal clustering. *The Annals of Statistics*, 48(4):2277 – 2302, 2020. doi: 10.1214/19-AOS1887.

P. M. Aronow and C. Samii. Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, pages 1912–1947, 2017.

A. Aswani, P. Bickel, and C. Tomlin. Regression on manifolds: Estimation of the exterior derivative. *The Annals of Statistics*, 39(1):48 – 81, 2011. doi: 10.1214/10-AOS823. URL https://doi.org/10.1214/10-AOS823.

S. Athey, D. Eckles, and G. W. Imbens. Exact p-values for network interference. *Journal of the American Statistical Association*, 113(521):230–240, 2018.

A. Azzalini and N. Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007.

A. Babenko and V. Lempitsky. The inverted multi-index. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3069–3076, 2012. doi: 10.1109/CVPR.2012.6248038.

O. Bachem, M. Lucic, and A. Krause. Practical coreset constructions for machine learning, 2017.

A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson. The diffusion of microfinance. *Science*, 341(6144), 2013.

A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson. Using gossips to spread information: Theory and evidence from two randomized controlled trials. *The Review of Economic Studies*, 86(6):2453–2490, 2019.

M. Barnett, G. Buchak, and C. Yannelis. Epidemic responses under uncertainty. *Proceedings of the National Academy of Sciences*, 120(2):e2208111120, 2023.

P. Battaglia, J. B. C. Hamrick, V. Bapst, A. Sanchez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. Allen, C. Nash, V. J. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv*, 2018. URL https://arxiv.org/pdf/1806.01261.pdf.

J.-P. Baudry, A. E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353, 2010. doi: 10.1198/jcgs.2010.08111.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

M. Belkin and P. Niyogi. Convergence of Laplacian Eigenmaps. Technical report, 2008.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. Technical report, 2006a. URL http://www.cse.msu.edu/.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006b. URL http://jmlr.org/papers/v7/belkin06a.html.

L. Bengtsson, J. Gaudart, X. Lu, S. Moore, E. Wetter, K. Sallah, S. Rebaudet, and R. Piarroux. Using mobile phone data to predict the spatial spread of cholera. *Scientific Reports 2015 5:1*, 5:1–5, 3 2015. ISSN 2045-2322. doi: 10.1038/srep08923. URL https://www.nature.com/articles/srep08923.

J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, Sept. 1975.

T. Berry and J. Harlim. Variable Bandwidth Diffusion Kernels. Technical report, 2014.

T. Berry and T. Sauer. Consistent manifold representation for topological data analysis. *Foundations of Data Science*, 0(0):0–0, 2019. ISSN 2639-8001. doi: 10.3934/fods.2019001.

H. J. Bierens. Uniform consistency of kernel estimators of a regression function under generalized conditions. *Journal of the American Statistical Association*, 78(383):699–707, 1983. ISSN 01621459. URL http://www.jstor.org/stable/2288140.

C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montufar, P. Lió, and M. Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1026–1037. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/bodnar21a.html.

C. Borgs, J. T. Chayes, R. Van der Hofstad, G. Slade, and J. Spencer. Random subgraphs of finite graphs: Iii. the phase transition for the n-cube. *Combinatorica*, 26:395–410, 2006.

G. Bouritsas, F. Frasca, S. P. Zafeiriou, and M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. doi: 10.1109/TPAMI.2022.3154319.

A. W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

L. Breiman. *Classification and regression trees*. Routledge, 2017.

R. R. Brinkman, M. Gasparetto, S. J. J. Lee, A. J. Ribickas, J. Perkins, W. Janssen, R. Smiley, and C. Smith. High-Content Flow Cytometry and Temporal Data Analysis for Defining a Cellular Signature of Graft-Versus-Host Disease. *Biology of Blood and Marrow Transplantation*, 13(6):691–700, jun 2007.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. doi: 10.1109/MSP.2017.2693418.

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

J. Cai, A. D. Janvry, and E. Sadoulet. Social networks and the decision to insure. *American Economic Journal: Applied Economics*, 7(2):81–108, 2015.

R. J. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10(1), jul 2015.

G. Canas and L. Rosasco. Learning probability measures with respect to optimal transport metrics. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

M. A. Carreira-Perpinán. A review of mean-shift algorithms for clustering. *arXiv preprint arXiv:1503.00687*, 2015.

J. E. Chacón. A Population Background for Nonparametric Density-Based Clustering. *Statistical Science*, 30(4):518–532, 2015. doi: 10.1214/15-STS526.

J. E. Chacón. Mixture model modal clustering. *Advances in Data Analysis and Classification*, 13:379–404, 6 2019.

J. E. Chacón and T. Duong. Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting. *Electronic Journal of Statistics*, 7:499–532, 2013.

J. E. Chacón, T. Duong, and M. P. Wand. Asymptotics for General Multivariate Kernel Density Derivative Estimators. *Statistica Sinica*, 21(2):807–840, 2011.

B. Chamberlain, J. Rowbottom, D. Eynard, F. Di Giovanni, X. Dong, and M. Bronstein. Beltrami flow and neural diffusion on graphs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1594–1609. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper/2021/file/0cbed40c0d920b94126eaf5e707be1f5-Paper.pdf.

B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi. Grand: Graph neural diffusion. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1407–1418. PMLR, 18–24 Jul 2021b. URL https://proceedings.mlr.press/v139/chamberlain21a.html.

A. Chandrasekhar. Econometrics of network formation. *The Oxford handbook of the economics of networks*, pages 303–357, 2016.

A. Chandrasekhar and R. Lewis. Econometrics of sampled networks. MIT working paper, 2010.

A. G. Chandrasekhar, P. Goldsmith-Pinkham, M. O. Jackson, and S. Thau. Interacting regional policies in containing a disease. *Proceedings of the National Academy of Sciences*, 118(19), 2021.

K. Chaudhuri and S. Dasgupta. Rates of convergence for the cluster tree. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 343–351, Red Hook, NY, USA, 2010. Curran Associates Inc.

K. Chaudhuri, S. Dasgupta, S. Kpotufe, and U. von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12): 7900–7912, 2014.

B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, (1):377–409, dec 1993. doi: 10.1007/BF02573985.

Y.-C. Chen. Generalized cluster trees and singular measures. *The Annals of Statistics*, 47(4):2174 – 2203, 2019. doi: 10.1214/18-AOS1744. URL https://doi.org/10.1214/18-AOS1744.

Y.-C. Chen and A. Dobra. Measuring human activity spaces from GPS data with density ranking and summary curves. *The Annals of Applied Statistics*, 14(1):409 – 432, 2020. doi: 10.1214/19-AOAS1311. URL https://doi.org/10.1214/19-AOAS1311.

Y. C. Chen, C. R. Genovese, and L. Wasserman. A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241, 2016.

Y. C. Chen, C. R. Genovese, and L. Wasserman. Density Level Sets: Asymptotics, Inference, and Visualization. *Journal of the American Statistical Association*, 112(520):1684–1696, oct 2017. ISSN 1537274X. doi: 10.1080/01621459.2016.1228536.

M.-Y. Cheng and H.-T. Wu. Local linear regression on manifolds and its geometric interpretation. *Journal of the American Statistical Association*, 108(504):1421–1434, 2013. doi: 10.1080/01621459.2013.827984.

Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, jul 2006. ISSN 10635203. doi: 10.1016/j.acha.2006.04.006.

A. Cuevas, M. Febrero, and R. Fraiman. Estimating the number of clusters. *Canadian Journal of Statistics*, 28(2):367–382, 2000.

A. Cuevas, M. Febrero, and R. Fraiman. Cluster analysis: A further approach based on density estimation. *Computational Statistics and Data Analysis*, 36(4):441–459, 2001.

B. Delaunay. Sur la sphère vide. a la mémoire de georges voronoï. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, 6:793–800, 1934.

E. W. Dijkstra. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, 1(1):269–271, 1959.

J. Eldridge, M. Belkin, and Y. Wang. Beyond hartigan consistency: Merge distortion metric for hierarchical clustering. volume 40 of *Proceedings of Machine Learning Research*, pages 588–606, Paris, France, 03–06 Jul 2015. PMLR.

S. Engebretsen, K. Engø-Monsen, M. A. Aleem, E. S. Gurley, A. Frigessi, and B. F. de Blasio. Time-aggregated mobile phone mobility data are sufficient for modelling influenza spread: the case of bangladesh. *Journal of The Royal Society Interface*, 17(167):20190809, 2020. doi: 10.1098/rsif.2019.0809. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2019.0809.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

P. D. Fajgelbaum, A. Khandelwal, W. Kim, C. Mantovani, and E. Schaal. Optimal lockdown in a commuting network. *American Economic Review: Insights*, 3(4):503–22, December 2021a. doi: 10.1257/aeri.20200401. URL https://www.aeaweb.org/articles?id=10.1257/aeri.20200401.

P. D. Fajgelbaum, A. Khandelwal, W. Kim, C. Mantovani, and E. Schaal. Optimal lockdown in a commuting network. *American Economic Review: Insights*, 3(4):503–522, 2021b.

J. Fan and J. Fan. Design-adaptive Nonparametric Regression. 87(420):998–1004, 1992.

J. Fan and I. Gijbels. *Local polynomial modelling and its applications*. Routledge, 2018.

J. Fan, I. Gijbels, T. C. Hu, and L. S. Huang. A study of variable bandwidth selection for local polynomial regression. *Statistica Sinica*, 6(1):113–127, 1996. ISSN 10170405.

M. Farboodi, G. Jarosch, and R. Shimer. Internal and external effects of social distancing in a pandemic. *Journal of Economic Theory*, 196:105293, 2021.

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

A. L. N. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005. doi: 10.1109/TPAMI.2005.113.

J. H. Friedman. Multivariate adaptive regression splines. *https://doi.org/10.1214/aos/1176347963*, 19:1–67, 3 1991. ISSN 0090-5364. doi: 10.1214/AOS/1176347963.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1): 32–40, 1975.

E. Giné and A. Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 38, pages 907–921. Elsevier, 2002.

A. D. Gordon. A Review of Hierarchical Classification. *Journal of the Royal Statistical Society. Series A (General)*, 150(2):119, mar 1987.

S. Graf and H. Luschgy. *Foundations of Quantization for Probability Distributions*, volume 1730 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-67394-1. doi: 10.1007/BFb0103945.

S. Graf and H. Luschgy. Rates of Convergence for The Empirical Quantization Error. 30(2): 874–897, 2002.

A. Green, S. Balakrishnan, and R. J. Tibshirani. Minimax optimal regression over sobolev spaces via laplacian eigenmaps on neighborhood graphs. 11 2021. URL https://arxiv.org/abs/2111.07394v1.

A. Griffith. Name your friends, but only five? the importance of censoring in peer effects estimates using social network data. *Journal of Labor Economics*, 40(4):779–805, 2022.

R. Guhaniyogi and D. B. Dunson. Compressed gaussian process for manifold regression. *Journal of Machine Learning Research*, 17(69):1–26, 2016. URL http://jmlr.org/papers/v17/14-230.html.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer New York, 2002. ISBN 978-0-387-95441-7. doi: 10.1007/B97848. URL http://link.springer.com/10.1007/b97848.

J. A. Hartigan and P. M. Hartigan. The Dip Test of Unimodality. *The Annals of Statistics*, 13(1):70–84, mar 1985.

J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, 28(1):100, 1979.

T. Hastie and C. Loader. [local regression: Automatic kernel carpentry]: Rejoinder. *https://doi.org/10.1214/ss/1177011005*, 8:139–143, 5 1993. ISSN 0883-4237. doi: 10.1214/SS/1177011005.

T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning. 2009. doi: 10.1007/978-0-387-84858-7. URL http://link.springer.com/10.1007/978-0-387-84858-7.

C. Hennig. Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification 2010 4:1*, 4:3–34, 1 2010.

T. Heskes. Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks*, 12(6):1299–1305, 2001.

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.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, dec 1985.

P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

M. O. Jackson. Genetic influences on social network characteristics. *Proceedings of the National Academy of Sciences*, 106(6):1687–1688, 2009.

M. O. Jackson and L. Yariv. Diffusion of behavior and equilibrium properties in network games. *American Economic Review*, 97(2):92–98, 2007.

M. O. Jackson and L. Yariv. Diffusion, strategic interaction, and social structure. In *Handbook of social economics*, volume 1, pages 645–678. Elsevier, 2011.

M. C. Jones. Simple boundary correction for kernel density estimation. *Statistics and computing*, 3(3):135–146, 1993.

Y. Kang, S. Gao, Y. Liang, M. Li, and J. Kruse. Multiscale dynamic human mobility flow dataset in the u.s. during the covid-19 epidemic. *Scientific Data*, pages 1–13, 2020.

A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks, 2020.

A. Kazi, L. Cosmo, S.-A. Ahmadi, N. Navab, and M. Bronstein. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. doi: 10.1109/TPAMI.2022.3170249.

J. Kim, Y.-C. Chen, S. Balakrishnan, A. Rinaldo, and L. Wasserman. Statistical inference for cluster trees. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1839–1847. Curran Associates, Inc., 2016.

S. J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *http://dx.doi.org/10.1137/070690274*, 51:339–360, 5 2009. ISSN 00361445. doi: 10.1137/070690274.

F. Klein. Vergleichende betrachtungen über neuere geometrische forschungen. *Mathematische Annalen*, 43:63–100, 1893. URL http://eudml.org/doc/157672.

S. Kpotufe. Fast, smooth and adaptive regression in metric spaces. *Advances in Neural Information Processing Systems*, 22, 2009.

S. Kpotufe. k-nn regression adapts to local intrinsic dimension. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 10 2011. URL https://arxiv.org/abs/1110.4300v1.

S. Kpotufe and V. K. Garg. Adaptivity to local smoothness and dimension in kernel regression. *Advances in Neural Information Processing Systems*, 26, 2013.

S. Kpotufe and N. Verma. Time-accuracy tradeoffs in kernel prediction: Controlling prediction quality. *Journal of Machine Learning Research*, 18(44):1–29, 2017. URL http://jmlr.org/papers/v18/16-538.html.

G. N. Lance and W. T. Williams. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, 9(4):373–380, feb 1967.

A. B. Lee and R. Izbicki. A spectral series approach to high-dimensional nonparametric regression. *Electronic Journal of Statistics*, 10(1):423 – 463, 2016. doi: 10.1214/16-EJS1112. URL https://doi.org/10.1214/16-EJS1112.

J. Li. Clustering based on a multilayer mixture model. *Journal of Computational and Graphical Statistics*, 14(3):547–568, 2005. doi: 10.1198/106186005X59586.

J. Li, S. Ray, and B. G. Lindsay. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(8), 2007.

Z. Lin and F. Yao. Functional regression on the manifold with contamination. *Biometrika*, 108(1):167–181, 07 2020. ISSN 0006-3444. doi: 10.1093/biomet/asaa041. URL https://doi.org/10.1093/biomet/asaa041.

S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

K. Lo, R. R. Brinkman, and R. Gottardo. Automated gating of flow cytometry data via robust model-based clustering. In *Cytometry Part A*, volume 73, pages 321–332. Cytometry A, apr 2008.

W.-Y. Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014.

S. Lubold, A. G. Chandrasekhar, and T. H. McCormick. Identifying the latent space geometry of network models through analysis of curvature. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(2):240–292, 2023.

R. Maitra. Initializing partition-optimization algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):144–157, 2009. doi: 10.1109/TCBB.2007.70244.

M. D. Marzio, A. Panzera, and C. C. Taylor. Nonparametric regression for spherical data. *Journal of the American Statistical Association*, 109(506):748–763, 2014. doi: 10.1080/01621459.2013.866567. URL https://doi.org/10.1080/01621459.2013.866567.

D. M. Mason, W. Polonik, et al. Asymptotic normality of plug-in level set estimates. *The Annals of Applied Probability*, 19(3):1108–1142, 2009.

W. F. Massy. Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, 60(309):234–256, 1965. doi: 10.1080/01621459.1965.10480787. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1965.10480787.

G. Menardi and A. Azzalini. An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, 24(5):753–767, 2014.

S. Milusheva. Managing the spread of disease with mobile phone data. *Journal of Development Economics*, 147:102559, 2020. ISSN 0304-3878. doi: https://doi.org/10.1016/j.jdeveco.2020.102559. URL https://www.sciencedirect.com/science/article/pii/S0304387820301346.

C. B. Morrison, H. Hildebrandt, S. J. Schmidt, I. K. Baldry, M. Bilicki, A. Choi, T. Erben, and P. Schneider. The-wizz: Clustering redshift estimation for everyone. *Monthly Notices of the Royal Astronomical Society*, 467(3):3576–3589, 2017.

F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms. *The Computer Journal*, 26(4):354–359, nov 1983. doi: 10.1093/comjnl/26.4.354.

E. A. Nadaraya. On estimating regression. *http://dx.doi.org/10.1137/1109020*, 9:141–142, 7 1964. ISSN 0040-585X. doi: 10.1137/1109020.

S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). 2 1996. URL https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

M. E. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60(6):7332, 1999.

R. Nugent and W. Stuetzle. Clustering with confidence: A low-dimensional binning approach. In *Classification as a Tool for Research*, pages 117–125. Springer, 2010.

A. D. Peterson, A. P. Ghosh, and R. Maitra. Merging k-means with hierarchical clustering for identifying general-shaped groups. *Stat*, 7(1):e172, 2018.

D. Pollard. A Central Limit Theorem for k-Means Clustering. *The Annals of Probability*, 10 (4):919–926, 1982.

M. Rahman, B. Ménard, R. Scranton, S. J. Schmidt, and C. B. Morrison. Clustering-based redshift estimation: comparison to spectroscopic redshifts. *Monthly Notices of the Royal Astronomical Society*, 447(4):3500–3511, 2015.

W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846, dec 1971.

S. Ray and B. G. Lindsay. The topography of multivariate normal mixtures. *The Annals of Statistics*, 33(5):2042 – 2065, 2005. doi: 10.1214/009053605000000417.

A. Rinaldo, A. Singh, R. Nugent, and L. Wasserman. Stability of density-based clustering. *Journal of Machine Learning Research*, 13:905, 2012.

M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9:65–78, 1982.

N. W. Ruktanonchai, J. R. Floyd, S. Lai, C. W. Ruktanonchai, A. Sadilek, P. Rente-Lourenco, X. Ben, A. Carioli, J. Gwinn, J. E. Steele, O. Prosper, A. Schneider, A. Oplinger, P. Eastham, and A. J. Tatem. Assessing the impact of coordinated covid-19 exit strategies across europe. *Science*, 369(6510):1465–1470, 2020. doi: 10.1126/science.abc5096. URL https://www.science.org/doi/abs/10.1126/science.abc5096.

E. Sadler. Seeding a simple contagion. *SSRN paper 4032812*, 2023.

B. Schölkopf and A. J. Smola. Learning with kernels: Support vector machines, regularization, optimization, and beyond adaptive computation and machine learning. page 626, 2002.

D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

L. Scrucca. Identifying connected components in gaussian finite mixture models for clustering. *Computational Statistics & Data Analysis*, 93:5–17, 2016.

M. Shapiro and E. Delgado-Eckert. Finding the probability of infection in an sir network is np-hard. *Mathematical Biosciences*, 240(2):77–84, 2012.

J. Shin, A. Rinaldo, and L. Wasserman. Predictive clustering, 2019.

B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, 2003. doi: 10.1109/ICCV.2003.1238663.

S. Smirnov and W. Werner. Critical exponents for two-dimensional percolation. *arXiv preprint math/0109120*, 2001.

A. J. Smola and R. Kondor. Kernels and regularization on graphs. In B. Schölkopf and M. K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 144–158, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

A. Sojourner. Identification of peer effects with missing peer data: Evidence from project star. *The Economic Journal*, 123(569):574–605, 2013.

I. Steinwart and C. Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and rkhss. *Constructive approximation*, 35(3):363–417, 2012. ISSN 0176-4276.

W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010.

R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005. doi: 10.1198/106186005X59243.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 63(2):411–423, jan 2001.

R. J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014. ISSN 00905364. URL http://www.jstor.org/stable/43556281.

R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335 – 1371, 2011. doi: 10.1214/11-AOS878. URL https://doi.org/10.1214/11-AOS878.

C. Torrence and G. P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.

M. Tsimidou, R. Macrae, and I. Wilson. Authentication of virgin olive oils using principal component analysis of triglyceride and fatty acid profiles: Part 1—classification of greek olive oils. *Food Chemistry*, 25(3):227 – 239, 1987.

P. Turner, J. Liu, and P. Rigollet. A statistical perspective on coreset density estimation, 2020.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

S. Venkatramanan, A. Sadilek, A. Fadikar, C. L. Barrett, M. Biggerstaff, J. Chen, X. Dotiwalla, P. Eastham, B. Gipson, D. Higdon, O. Kucuktunc, A. Lieber, B. L. Lewis, Z. Reynolds, A. K. Vullikanti, L. Wang, and M. Marathe. Forecasting influenza activity using machine-learned mobility map. *Nature Communications 2021 12:1*, 12:1–12, 2 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-21018-5. URL https://www.nature.com/articles/s41467-021-21018-5.

G. Voronoi. Recherches sur les paralléloèdres primitives. *J. reine angew. Math*, 134:198–287, 1908.

G. Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 24(5): 383–393, 1975.

M. Walesiak and A. Dudek. The choice of variable normalization method in cluster analysis. In K. S. Soliman, editor, *Education Excellence and Innovation Management: A 2025 Vision to Sustain Economic Development During Global Challenges*, pages 325–340. International Business Information Management Association (IBIMA), 2020. ISBN 978-0-9998551-4-1.

M. P. Wand and M. C. Jones. Multivariate plug-in bandwidth selection. *Computational Statistics*, 9(2):97–116, 1994.

D. J. Wang, X. Shi, D. A. McFarland, and J. Leskovec. Measurement error in network data: A re-classification. *Social Networks*, 34(4):396–409, 2012.

Y. Wang. *Smoothing splines: methods and applications*. CRC press, 2011.

Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), oct 2019. ISSN 0730-0301. doi: 10.1145/3326362. URL https://doi.org/10.1145/3326362.

Y. X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17:1–41, 2016. ISSN 15337928.

L. Wasserman. *All of Nonparametric Statistics*. Springer New York, 2006a. doi: 10.1007/0-387-30623-4.

L. Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006b.

L. Wasserman. Topological data analysis, 2016. URL https://arxiv.org/abs/1609.08227.

G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964. ISSN 0581572X. URL http://www.jstor.org/stable/25049340.

D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393 (6684):440–442, 1998.

R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605665.

J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620 – 2648, 2019. doi: 10.3150/18-BEJ1065. URL https://doi.org/10.3150/18-BEJ1065.

Z. Wei and Y.-C. Chen. Skeleton clustering: Dimension-free density-aided clustering. *Journal of the American Statistical Association*, 0(0):1–12, 2023. doi: 10.1080/01621459.2023.2174122. URL https://doi.org/10.1080/01621459.2023.2174122.

A. Wesolowski, N. Eagle, A. J. Tatem, D. L. Smith, A. M. Noor, R. W. Snow, and C. O. Buckee. Quantifying the impact of human mobility on malaria. *Science (New York, N.Y.)*, 338:267, 10 2012. ISSN 10959203. doi: 10.1126/SCIENCE.1223467. URL /pmc/articles/PMC3675794//pmc/articles/PMC3675794/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC3675794/.

H. Wold. Soft modelling by latent variables: The non-linear iterative partial least squares (nipals) approach. *Journal of Applied Probability*, 12:117–142, 1975. ISSN 0021-9002. doi: 10.1017/S0021900200047604.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

D. G. York, J. Adelman, J. E. Anderson Jr, S. F. Anderson, J. Annis, N. A. Bahcall, J. Bakken, R. Barkhouser, S. Bastian, E. Berman, et al. The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.

X. Zhang, X. Shi, Y. Sun, and L. Cheng. Multivariate regression with gross errors on manifold-valued data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):444–458, 2019. doi: 10.1109/TPAMI.2017.2776260.

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression. In S. Shalev-Shwartz and I. Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 592–617, Princeton, NJ, USA, 12–14 Jun 2013. PMLR. URL https://proceedings.mlr.press/v30/Zhang13.html.

Y. Zhao. A survey on theoretical advances of community detection in networks. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(5):e1403, 2017.