Graphs for Statistical Learning and Modeling

Zeyu (Jerry) Wei

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

November 3, 2022

Abstract

Graph, consisting of a set of vertices and a set of edges, is a geometric object that can not only visualize but also mathematical characterize the geometric structures in data. Graphs also model relations or connections between different units and have applications in various fields such as epidemiology, sociology, biology, and chemistry. We first take advantage of graphs from a geometric perspective. We propose a data analysis framework that constructs weighted graphs, called skeletons, to encode the geometric structures in data and utilizes the learned geometric information to assist the downstream analysis tasks such as clustering and regression.

For 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.

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 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.

Graph, as a structure to represent connections, is a helpful tool in modeling contact networks, which is incorporated in various epidemic models. However, missing links in the observed contact network are inevitable, which raises concern over the robustness of epidemic models in this regard. To address this concern, we assess epidemic models under missingness and present some preliminary results from this ongoing project.

Chapter 1

Introduction

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, 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 description 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 lower 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 also 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 a 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 represent geometric information, graph is a structure of connections, which make it natural to represent various networks, with contact network being one example. Due to the advancement in mobile communication technology, collection of contact network data, 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 call 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 flows data are also used as proxy to contact network for epidemic modeling (Fajgelbaum et al., 2021; Alsing et al., 2020). Facing the challenge of the global pandemic, the Google COVID-19 Aggregated Mobility Research Dataset becomes 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 network has substantial implications 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 mathematical 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 usage of contact network in epidemic modeling. We assess how missingness in the contact network affects the identification of risky sets. We present some preliminary theoretical and simulation results from this ongoing project.

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 to group 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) find 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 represents 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 $O_P(n^{-\frac{2}{4+d}})$ under usual smoothness conditions (Scott, 2015; Wasserman, 2006), 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 the model-based clustering to overcome the limitations on parametric assumptions. In particular, there are several methods for 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 limitation 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 data consisted 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 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.

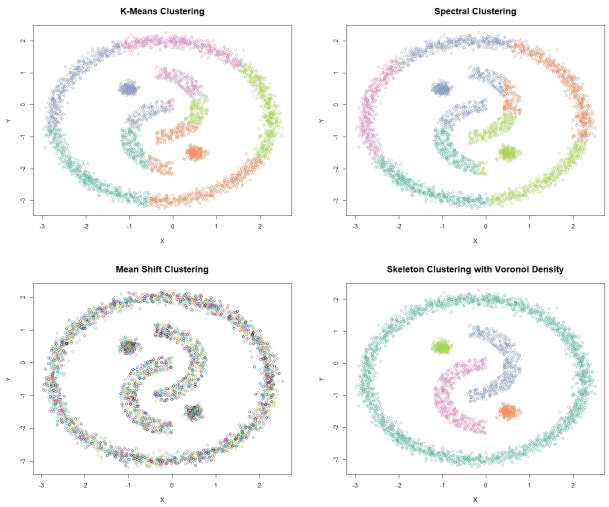


Figure 2.1: Yinyang Data with dimension 200. On the bottom-right is the clustering result of the skeleton clustering with the proposed Voronoi density similarity measure.

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 some directions for future research.

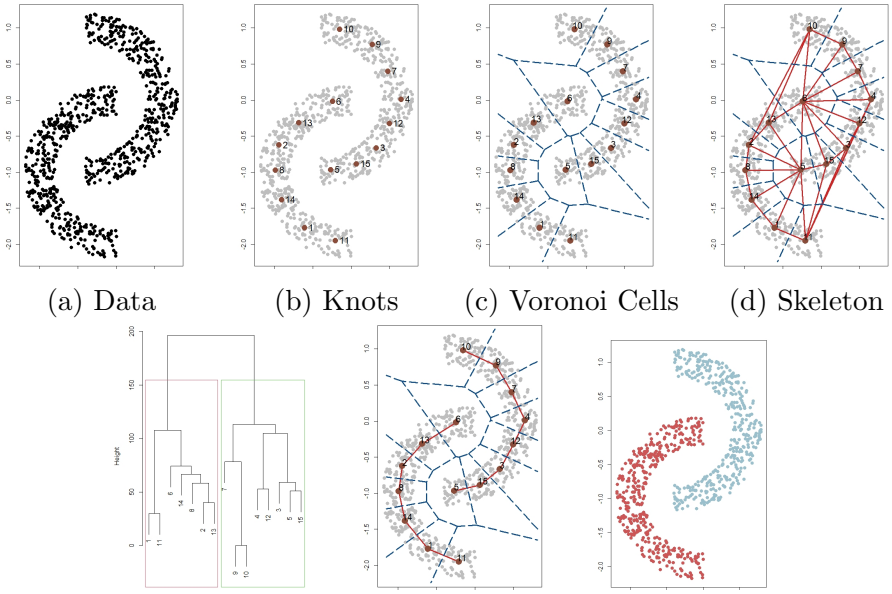
Algorithm 1 Skeleton clustering

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).

2.2 Skeleton Clustering Framework

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. 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 (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 cluster label to each observation according to the knot-cluster that it belongs to (panel (g)).



(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).

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 on 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 clusters, and this procedure behaves like overfitting the k-means. Notably, we do not use k-means procedure to obtain final clustering, but instead we use it as a 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 better representation of the data, but, if we have too many knots, the number of observation 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.

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,

$$\mathbb{C}_j = \{x \in \mathcal{X} : d(x, c_j) \leq d(x, c_\ell) \quad \forall \ell \neq j\}, \tag{2.1}$$

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

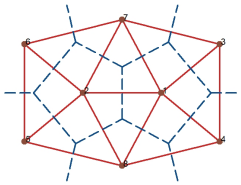


Figure 2.3: Voronoi Tessellation as blue dashed lines and Delaunay Triangulation by red solid lines.

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 choice of linkage should be made base 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 are not exactly the cluster trees, 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 have 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 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

$$A_{j\ell} = \{x \in \mathcal{X} : d(x, c_i) > \max\{d(x, c_j), d(x, c_\ell)\}, \forall i \neq j, \ell\}. \tag{2.2}$$

In this work we take $d(.,.)$ to be 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 belongs 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

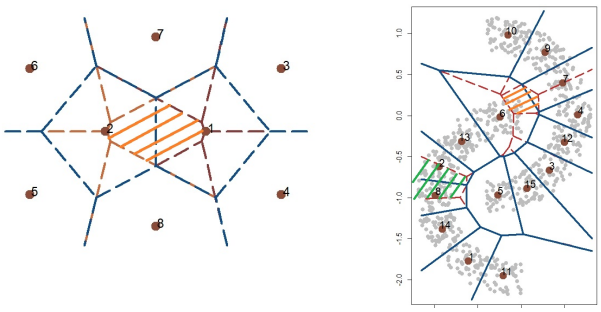


Figure 2.4: **Left:** Orange shaded area illustrates the 2-NN region of knots $1, 2$. **Right:** Shaded areas illustrate the 2-NN region of knots $6, 7$ and knots $2, 8$.

this insight, the Voronoi density measures the edge weight of (c_j, c_ℓ) with

$$S_{j\ell}^{VD} = \frac{\mathbb{P}(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{2.3}$$

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 to 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

$$\hat{S}_{j\ell}^{VD} = \frac{\hat{P}_n(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{2.4}$$

Note that here we are assuming that $c_1, \cdots, c_k$ as given beforehand. In sample version, we replace them by 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 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

$$S_{j\ell}^{FD} = \int_{F_{j\ell}} p(x)\mu_{d-1}(dx) = \int_{F_{j\ell}} d\mathbb{P}(x), \tag{2.5}$$

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 $\hat{S}^{FD}_{j\ell}$ to be

$$\hat{S}^{FD}_{j\ell} = \frac{1}{nh} \sum_{X_i \in \mathbb{C}_j \cup \mathbb{C}_\ell} K\left(\frac{\Pi_{j\ell}(X_i) - (c_\ell + c_j)/2}{h}\right) \tag{2.6}$$

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 the conventional kernel smoothing suffers from the curse of dimensionality (Chen et al., 2017; Chacón et al., 2011; Wasserman, 2006), the kernel estimator in equation (2.6) 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

$$\mathsf{Disk}(x, R, \nu) = \{y : ||x - y|| \leq R, (x - y)^T \nu = 0\} \tag{2.7}$$

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 details, 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 $c_j + t(c_\ell - c_j)$

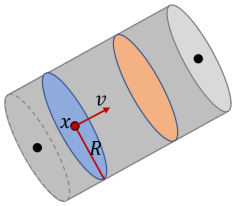


Figure 2.5: The disk area centered at x with a radius R and a direction ν.

for some t. For a point $c_j + t(c_\ell - c_j)$, we define the integrated density in the disk region (called *Disk Density*) as

$$\mathsf{pDisk}_{j\ell,R}(t) = \mathbb{P}\left(\mathsf{Disk}(c_j + t(c_\ell - c_j), R, c_\ell - c_j)\right) = \int_{\mathsf{Disk}(c_j+t(c_\ell-c_j),R,c_\ell-c_j)} p(x)dx. \tag{2.8}$$

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

$$S_{j\ell}^{TD} = \inf_{t\in[0,1]} \mathsf{pDisk}_{j\ell,R}(t) \tag{2.9}$$

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

$$R_j = \sqrt{\frac{1}{|\mathbb{C}_j| - 1} \sum_{X_\ell \in \mathbb{C}_j} \|X_\ell - c_j\|^2} \tag{2.10}$$

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

$$\hat{S}_{j\ell}^{TD} = \inf_{t\in[0,1]} \widehat{\mathsf{pDisk}}_{j\ell,R}(t). \tag{2.11}$$

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 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 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 2.2) 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. We 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

$$\left| \frac{\hat{S}^{VD}_{j\ell}}{S^{VD}_{j\ell}} - 1 \right| = O_p\left(\sqrt{\frac{k}{n}}\right), \tag{2.12}$$

$$\max_{j,\ell} \left| \frac{\hat{S}^{VD}_{j\ell}}{S^{VD}_{j\ell}} - 1 \right| = O_p\left(\sqrt{\frac{k}{n}} \log k\right), \tag{2.13}$$

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 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 case 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 2.13) 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 $\mathcal{L}^* = \{\mathcal{L}_\ell^*\}_{\ell=1,\dots,L}$, 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 value 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 $ARI(\mathcal{L}^*, \hat{\mathcal{L}})$ denotes 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

$$\mathbb{P}\left\{ ARI(\mathcal{L}^*, \hat{\mathcal{L}}) < 1 \right\} \leq k(k-1) \exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1-p_{min}) + \frac{1}{3}\varepsilon} \right) \tag{2.14}$$

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 consists 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

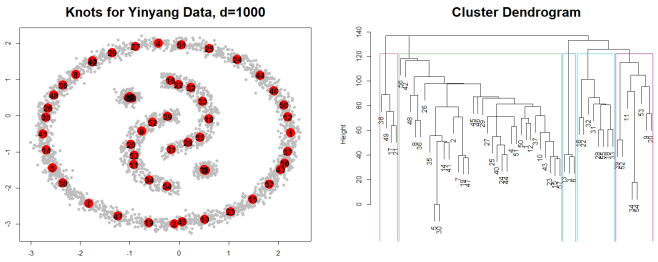


Figure 2.6: Knots chosen by k-means on Yinyang data and the Dendrogram for single linkage hierarchical clustering with similarity measured by Voronoi density.

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 a 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 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

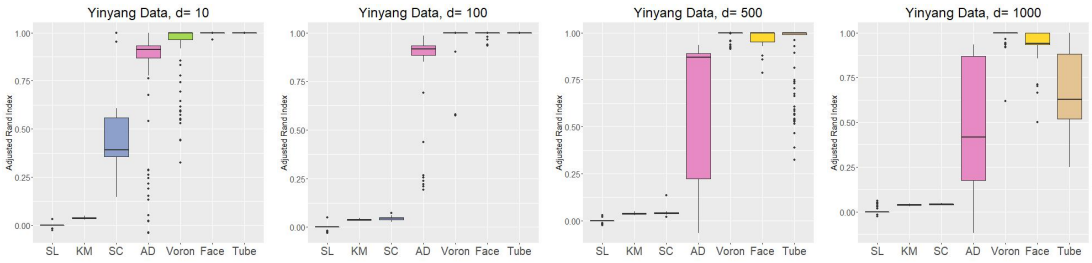


Figure 2.7: Comparison of the final clustering performance in terms of adjusted Rand Index with different clustering methods on Yinyang Data with dimension 10, 100, 500, and 1000.

performance in 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.

Mickey Data

The simulated Mickey data is an intrinsically 2-dimensional data consists 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 $d = 1000$.

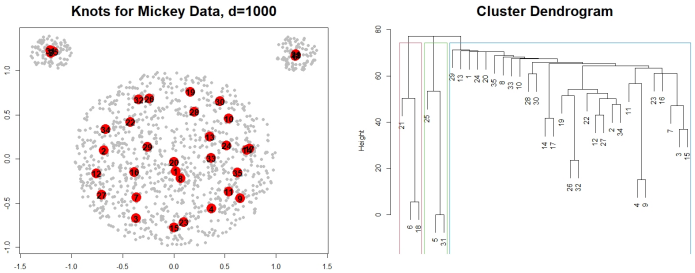


Figure 2.8: An illustration of the analysis of the Mickey data with dimension 100.

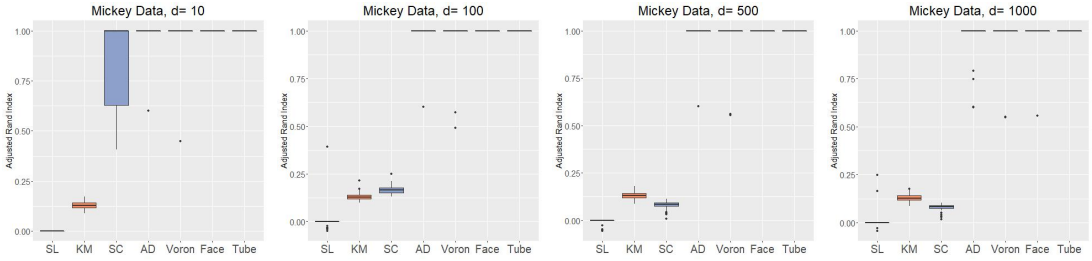


Figure 2.9: Comparison of adjusted Rand index using different similarity measures on Mickey data with dimensions 10, 100, 500, 1000.

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-

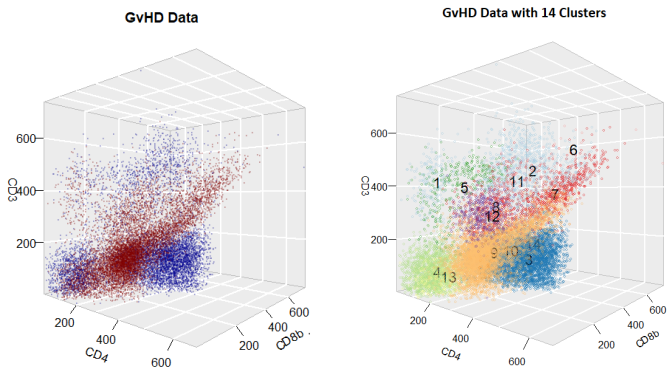


Figure 2.10: **Left:** 3D scatterplot of the positive sample (red) and the control sample (blue). **Right:** Final clustering result of combined GvHD data.

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 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 region with medium to the 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 comparisons for 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 ensure that both sample 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 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 cluster chosen to be $S = 14$ (Baudry et al., 2010).

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 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, 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, CD8β region but

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	$3{\times}10^{-63}$	$4{\times}10^{-8}$	1×10^{-4}	$6{\times}10^{-103}$
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	$2{\times}10^{-13}$	0	$1{\times}10^{-6}$	.09	$6{\times}10^{-30}$	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 0.5 indicates that the two sample has equal proportion in the region. The p-value is the simple proportional test to examine if the two sample has equal proportion in that cluster.

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

has all the observations coming from the control sample. However, the small size (only 17 data points) of Cluster 6 makes unclear if it is a real structure or due to pure randomness. Overall our method succeed 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 have some additional findings. Cluster 9, 12, and 13 also have high proportion of positive samples. These clusters are in 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 are 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 graphical representation of each clusters 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 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 bypass 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 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 on 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 relate 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 exploits the local manifold structures in an exact sense, they are not robust to variations in the covariates that perturbs them away from the true manifold structure.

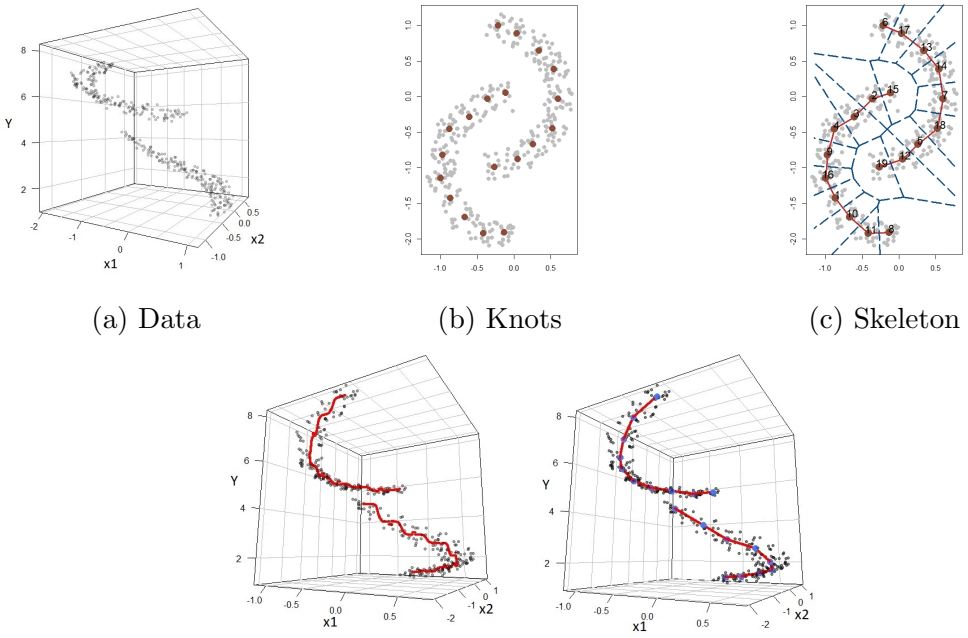
Many other manifold estimation approaches exist in the statistical literature. Guhaniyogi and Dunson (2016) utilize random compression of the feature vector in combination with Gaussian process regression. Zhang et al. (2013) follow a divide-and-conquer approach that computes an independent kernel Ridge regression estimator for each randomly partitioned subsets. 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. 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. Green et al. (2021) proposes the Principal Components Regression with Laplacian-Eigenmaps (PCR-LE) that projects responses onto the eigenvectors output by Laplacian Eigenmaps. 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 scalar responses on manifold-structured covariates in a way that bypasses the curse of dimensionality, and we achieve this by proposing a new framework that utilizes 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 (2021) and project the covariates onto the skeleton. Then we apply different nonparametric regression methods to the skeleton-projected data. We give brief descriptions about 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 Nadaraya (1964) and Watson (1964) with the famous Nadaraya–Watson estimator, 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 but is powerful and widely used in many applications. We incorporate these techniques mentioned above in our regression framework.

In recent years, many nonparametric regressors were shown to be adaptive to the manifold structure that they converge at rates that depend only on the intrinsic dimensions of data space. Particularly, the kNN regressor and the kernel regressor are both proved to be manifold adaptive with the proper parameter tuning procedures (Kpotufe, 2009a,b, 2011;



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

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

Figure 3.1: Skeleton Regression illustrated by Two Moon Data (d=2).

Kpotufe and Garg, 2013; Kpotufe and Verma, 2017). The regression framework proposed in this work also adapts to the manifold that the nonparametric regression models, fitted on a graph, are dimension-independent. Our framework has additional advantages that predictors from distinct manifolds can be accounted for and is robust to additive noise and noisy observations.

Outline. We start with section 3.2 by presenting the brief procedures of the skeleton regression framework. In section 3.3, we describe the construction of the skeleton. In section 3.4, we apply kernel regression with the geodesic distances on the skeleton. In section 3.5, we fit linear spline on the skeleton structure. In section 3.6, we present some simulation results for skeleton regression and demonstrate the effectiveness of our method on real datasets in Section 3.7. In section 3.8, we conclude the paper and points some directions for future

research.

3.2 Skeleton Regression Framework

We introduce the skeleton regression framework in this section. Given random independent samples $(X_1, Y_1), \ldots, (X_n, Y_n)$, with $X_j \in \mathcal{X} \subseteq \mathbb{R}^d$ the covariates and $Y_j \in \mathbb{R}$ the response for $j = 1, \ldots, n$, a 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 have a low-dimensional manifold structure, $\mathcal{X}$ 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 accommodate for such manifold structured data, we approach the regression task by first representing the sample covariate space with a graph, which we call the skeleton, that summarizes the manifold structures. We then focus instead on the regression function over the skeleton graph which incorporates the covariates geometry in a dimension-independent way.

In this work, we use the methods in Wei and Chen (2021) to construct the skeleton, but it has the potential to be constructed with other approaches and tuned with subject matter knowledge. We illustrate our regression framework on simulated TwoMoon data in Figure 3.1. The covariates of the TwoMoon 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 regression (Sec-

tion 3.4, illustrated in Figure 3.1 d), and linear spline (Section 3.5, illustrated in Figure 3.1 e). The estimated regression function can predict new projected data points. The procedure is summarized in Algorithm 2.

Algorithm 2 Skeleton Regression

Input: Observations $(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_n, Y_n)$.
1. **Skeleton Construction.** Construct a skeleton representation of the covariates using method in Section 3.3. Knots and edges can be tuned with subject knowledge.
2. **Data Projection.** Project the covariates onto the skeleton structure.
3. **Skeleton Regression Function Estimation.** Fitting regression function on the skeleton using kernel regression (Section 3.4) and linear spline (Section 3.5).
4. **Prediction.** Project the new data puts onto the learnt skeleton structure and use the estimated regression function for prediction.

3.3 Skeleton Construction

3.3.1 Knots and Edges

For given covariate space $\mathcal{X} \subseteq \mathbb{R}^d$, we construct a skeleton graph with k knots $\mathcal{V} = \{V_j \in \mathbb{R}^d : j = 1, \ldots k\}$ and the set of connected edges $\mathcal{E} = \{tV_j + (1-t)V_\ell : t \in (0,1), V_j \text{ connected to } V_\ell\}$. For the parametrization, we have $t \in (0,1)$ to exclude the knots. Notably, the knots and edges in our framework, different from the usual graphs, have physical locations in the ambient space. We denote the skeleton as $\mathcal{S} = \mathcal{V} \cup \mathcal{E}$.

The skeleton graph is constructed to give an approximate representation of the data structure and many existing prototype-based methods can be used for this purpose. In this work, we follow an approach in Wei and Chen (2021) to construct the skeleton and give a brief description below for comprehensiveness. The method constructs knots as the centers

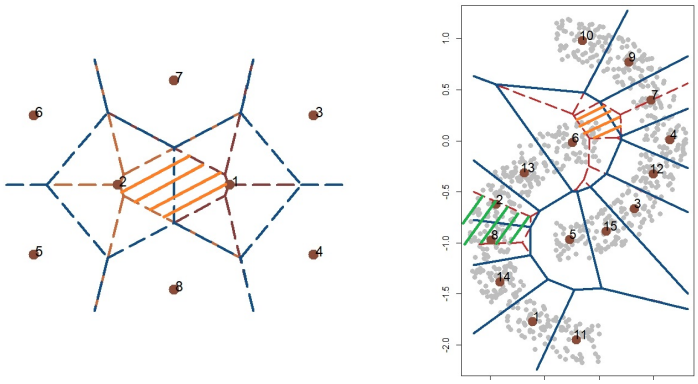


Figure 3.2: **Left:** Orange shaded area illustrates the 2-NN region of knots $1, 2$. **Right:** Shaded areas illustrate the 2-NN region of knots $6, 7$ and knots $2, 8$.

from the k-means clustering with a large number of k [1]. The edges are connected according to the sample 2-Nearest-Neighbor (2-NN) region of a pair of knots (V_j, V_ℓ) (see Figure 3.2)

$$B_{j\ell} = \{X_m, m = 1, \ldots, n : \|x - V_i\| > \max\{\|x - V_j\|, \|x - V_\ell\|\}, \forall i \neq j, \ell\}. \tag{3.1}$$

where $||.||$ denotes the Euclidean norm, and an edge between V_j and V_ℓ is added if $B_{j\ell}$ is non-empty. Provided the desired number of disconnected components, the method can further segment the skeleton by using hierarchical clustering with respect to the Voronoi Density weights defined as $S^{VD}_{j\ell} = \frac{\frac{1}{n}|B_{j\ell}|}{\|V_j - V_\ell\|}$.

Remark 5. The idea of using the k-means algorithm to divide data into cells for fast computation has been applied in many machine learning realms. 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 centers from sub-dimensions. k-means algorithm can be slow for large-scale data, but Johnson et al. (2019) has imple-

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

mented the k-means algorithm efficiently on the GPU base, which dramatically improves the calculation speed of the algorithm.

3.3.2 Skeleton-Based Distance

A feature of the physically-located skeleton is that we can easily define a 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 usually 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 (See Figure 3.3 for an example):

- If $\boldsymbol{S}_j, \boldsymbol{s}_\ell$ are disconnected that they belong to two disjoint components of $\mathcal{S}$, we define $d(\boldsymbol{S}_j, \boldsymbol{s}_\ell) = \infty$.
- If $\boldsymbol{S}_j$ and $\boldsymbol{s}_\ell$ are on the same edge, we define the skeleton distance as their Euclidean distance that

$$d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell) = ||\boldsymbol{S}_j - \boldsymbol{s}_\ell|| \tag{3.2}$$

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

$$d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell) = ||\boldsymbol{S}_j - V_0|| + ||\boldsymbol{s}_\ell - V_0|| \tag{3.3}$$

- Otherwise, let knots $V_{i(1)}, \ldots, V_{i(m)}$ be the vertices on the shortest 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

$$d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell) = ||\boldsymbol{S}_j - V_{i(1)}|| + ||\boldsymbol{s}_\ell - V_{i(m)}|| + \sum_{p=1}^{m-1} \left\|V_{i(p)}, V_{i(p+1)}\right\| \tag{3.4}$$

Remark 6. We may view the geodesic distance on the skeleton as an estimate of the geodesic

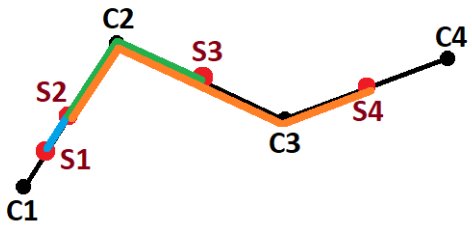


Figure 3.3: Illustration of skeleton-based distance. Let C_1, C_2, C_3, C_4 be the knots, and let S_2, S_3, S_4 be the mid-point on the edges E_{12}, E_{23}, E_{34} respectively. Let S_1 bet the midpoint between C_1 and S_2 on the edge. Let $d_{ij} = \|C_i - C_j\|$ denotes the length of the edge E_{ij}. $d_{\mathcal{S}}(S_1, S_2) = \frac{1}{4}d_{12}$ illustrated by the blue path ($m = 0$ case). $d_{\mathcal{S}}(S_2, S_3) = \frac{1}{2}d_{12} + \frac{1}{2}d_{23}$ illustrated by the green path ($m = 1$ case). $d_{\mathcal{S}}(S_2, S_4) = \frac{1}{2}d_{12} + d_{23} + \frac{1}{2}d_{34}$ illustrated by the orange path ($m = 2$ case).

distance on the underlying data manifold. Moreover, to make a stronger connection to the manifold structure, it is possible to define edge distances by local manifold learning techniques that have better approximations to local manifold structure. However, using more complex local edge weights can pose issues for the data projection step described in the next section and we leave this as a future direction.

3.3.3 Data Projection

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

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

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

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

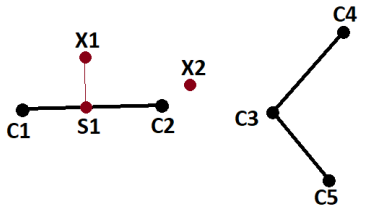


Figure 3.4: Illustration of projection to the skeleton. The skeleton structure is given by the black dots and lines. Data point X_1 is projected to S_1 on the edge between C_1 and C_2. Data point X_2 is projected to knot C_2.

3.4 Skeleton Kernel Regression

In this section, we apply kernel smoothing to the skeleton-projected covariates based on the skeleton-based distances. Instead of estimating the regression function defined on $\mathcal{X}$, we estimate the *projected* regression function

$$m_{\mathcal{S}}(\boldsymbol{s}) = \mathbb{E}(y|\pi(\boldsymbol{x}) = \boldsymbol{s}), \boldsymbol{s} \in \mathcal{S} \tag{3.5}$$

on the skeleton domain $\mathcal{S}$. Let $K_h(.) = K(./h)$ be a non-negative kernel function with bandwidth $h > 0$ and let $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ denote the skeleton-projected covariates that $\boldsymbol{s}_i = \Pi(\boldsymbol{x}_i)$ for $i = 1, \ldots, n$, the corresponding skeleton-based kernel (S-kernel) regressor for a point

$\boldsymbol{s} \in \mathcal{S}$ is

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{j=1}^{N} K_h(d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}))Y_j}{\sum_{j=1}^{N} K_h(d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}))} \tag{3.6}$$

An example kernel function is the Gaussian kernel that

$$K_h(d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell)) = \exp\left(- \frac{d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell)^2}{h^2} \right) \tag{3.7}$$

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. The smoothing bandwidth h can be chosen by cross-validation.

3.4.1 Consistency of S-Kernel Regressor

In this section, we present the convergence result of the presented S-kernel regressor. We assume the skeleton is fixed and given and focus on the regression function estimation step. To assess the estimation error, we need to analyze the distribution on the skeleton. However, due to the covariate projection procedure, the probability measures on the knots and edges are different, and hence we treat them separately. On an edge, the domain of the projected regression function varies 1-dimensionally and the estimation becomes a classical univariate problem. In particular, we impose the one-dimensional Lebesgue measure with respect to the parametrization t as in the definition of $\mathcal{E}$, and the true projected regression model for a point $\boldsymbol{s}$ on edge (V_j, V_ℓ) is

$$m_{\mathcal{S}}(\boldsymbol{s}) = m_{j\ell}(\boldsymbol{s}) = m_{j\ell}(t)$$

where t is the parametrization for $\boldsymbol{s}$. Differently, a whole region of the covariate space can be projected onto a knot, leading to nontrivial probability mass at the point. Hence, we assign discrete counting measure on each knot, and the true projected regression model at $\boldsymbol{s} \in \mathcal{V}$

is a constant function

$$m_{\mathcal{S}}(\boldsymbol{s}) = M_j, \boldsymbol{s} = V_j.$$

For simplicity, we write $K_h(\boldsymbol{S}_j, \boldsymbol{s}_\ell) \equiv K_h(d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell))$ for $\boldsymbol{S}_j, \boldsymbol{s}_\ell \in \mathcal{S}$. Let $\mathcal{B}(\boldsymbol{s}, h_n) \subset \mathcal{S}$ be the support for the kernel function $K_h(.)$ at point $\boldsymbol{s} \in \mathcal{S}$ with bandwidth h_n. We can decompose the kernel regression estimator into edge parts and knot parts as

$$\begin{aligned}
\hat{m}(\boldsymbol{s}) &= \frac{\sum_{j=1}^n Y_j K_h(\boldsymbol{S}_j, \boldsymbol{s})}{\sum_{j=1}^n K_h(\boldsymbol{S}_j, \boldsymbol{s})} \\
&= \frac{\frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E}) + \frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V})}{\frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E}) + \frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V})} \\
&= \frac{\frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^n Y_j K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h_n))}{\frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^n K_h(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h_n))}
\end{aligned} \tag{3.8}$$

In the last line, we stress that the knots and edges in the kernel estimator only make meaningful contribution 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 $Y_j = U_j + m_{\mathcal{S}}(\boldsymbol{S}_j), \boldsymbol{S}_j \in \mathcal{S}$, and $\mathbb{E}(U_j|X_j) = 0$ almost surely. Let $\sigma^2(\boldsymbol{s}) = \mathbb{E}(|U_j|^2|\boldsymbol{S}_j = \boldsymbol{s})$. We assume

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

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

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

A4 The kernel function has compact support and satisfies $\int K(x)dx = 1$, $\int |K(x)|\, dx < \infty$, $\int xK(x)dx = 0$, $\int |x|K(x)dx < \infty$, $\int K^2(x)dx < \infty$, and $\int x^2K(x)dx < \infty$

Conditions A1 and A4 are general and are commonly assumed for kernel regression. For the smoothness conditions A2 and A3, instead of having the second-order smoothness condition that is usually assumed for kernel regression analysis, we here only have Lipschitz continuity.

We do not assume higher-order derivative smoothness because odd-degree derivatives require specifying directions on the graph, which can lead to model formulation issues. We leave discussions on higher-order splines as future work.

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 $n \to \infty, h_n \to 0$, for sufficiently large n, we have $\mathcal{B}(\boldsymbol{s}, h_n) \subset E_{j\ell}$, and the skeleton distance is the 1-dimensional Euclidean distance for any point within the support. Therefore, we have the convergence rate similar to the 1-dimensional kernel regression estimator (Bierens, 1983; Wasserman, 2006; Chen et al., 2017).

Theorem 3 (Consistency on Edge Points). For $\boldsymbol{s} \in \mathcal{E}$ an edge point, assume conditions (A1-4) hold for all points in $\mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)$, as $n \to \infty$, $h_n \to 0$, $nh_n \to \infty$,

$$|\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})| = O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right) \tag{3.9}$$

We leave the proof in Appendix J.

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 degree 1 knots on the skeleton graph where a non-trivial region of points are projected onto such knots. For example see knot C2 in Figure 3.4.

Theorem 4 (Consistency on Knots with Nonzero Mass). For $\boldsymbol{s} \in \mathcal{V}$ a knot point, assume conditions (A1-4) hold for all points in $\mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)$ and let the discrete probability mass at $\boldsymbol{s}$ be $p(\boldsymbol{s}) > 0$. We have, as $n \to \infty$, $h_n \to 0$, and $nh_n \to \infty$,

$$|\hat{m}(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})| = O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right) \tag{3.10}$$

Note that for the stochastic variation part, instead of having the usual $O_p\big(\sqrt{\frac{1}{nh_n}}\big)$ rate, we have $O_p\big(\sqrt{\frac{1}{n}}\big)$ rate which comes from the observations projected onto the knots. The detailed proof is provided in Appendix J.

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 is the case for knots 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 $O(h_n) + O_p\big(\sqrt{\frac{1}{nh_n}}\big)$ 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. For $\boldsymbol{s} \in \mathcal{V}$ a knot point, assume conditions (A1-4) hold for all points in $\mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)$ and let the discrete probability mass at $\boldsymbol{s}$ be $p(\boldsymbol{s}) = 0$. 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

$m_\ell(0) = \lim_{t \downarrow 0} m_\ell(t)$. We have, as $n \to \infty$, $h_n \to 0$, and $nh_n \to \infty$,

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{\ell \in \mathcal{I}} m_\ell(0) g_\ell(0)}{\sum_{\ell \in \mathcal{I}} g_\ell(0)} + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)$$

Remark 7. The domain of the regression function $\mathcal{S}$ can be seen as bounded, and hence the boundary bias issue can arise. However, the boundary of the skeleton is the set of degree 1 knots, and, under our formulation, knots have discrete measures so we don't need to consider boundary bias. Also, the boundary of the true manifold structure can be different from the boundary of the skeleton graph, which makes the boundary consideration more complicated.

3.5 Linear Spline Regression on Graph

In this section, we propose to fit a skeleton-based linear spline model (S-Lspline) for regression estimation. We construct a linear model on each edge of the graph while requiring the predicted values to agree on shared vertices and consequently getting a continuous model on the graph. For the fitting process, notably, two points can determine a line, and hence the linear model on each edge is determined by the values on the two connected vertices. Specifically, a linear function $f(t) = \alpha + \beta t$ parametrized by (α, β) is equivalent to $f(t) = f(0) + (f(1) - f(0))t$ parametrized by $f(0) = \alpha$ and $f(1) = \alpha + \beta$. Also, fitting exactly one value to each knot ensures the continuity at the knots as required by the linear spline model. As a result, the linear spline model is parameterized by the values on each knot.

With the values-on-knots parametrization, we can fit the S-Lspline model through ordinary least squares with a graph-transformed $n \times v$ covariate matrix $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T$ where $v = |\mathcal{V}|$ is the number of knots and $\boldsymbol{z}_j$ is the length v transformed data vector for $\boldsymbol{x}_j \in \mathcal{X}$. The covariates are transformed in the following way:

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

$$\boldsymbol{z}_{ij} = 1, \qquad \boldsymbol{z}_{ij'} = 0$$

for $j' \neq j$.

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

$$\boldsymbol{z}_{ij} = \frac{||\boldsymbol{s}_i - V_j||}{||V_j - V_\ell||}, \qquad \boldsymbol{z}_{i\ell} = \frac{||\boldsymbol{s}_i - V_\ell||}{||V_j - V_\ell||}, \qquad \boldsymbol{z}_{ij'} = 0$$

for $j' \neq j, \ell$.

Let $\hat{\boldsymbol{y}}$ be the length v vector of predicted values on all the knots. 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 $\boldsymbol{z}_i^T \hat{\boldsymbol{y}} = \hat{y}_j$, the predicted value on vertex V_j. If $\boldsymbol{x}_i$ is projected onto an edge between knots V_j and V_ℓ, let $\hat{y}_j$ and $\hat{y}_\ell$ be the corresponding predicted values at V_j and V_ℓ, and the linear interpolation between $\hat{y}_\ell$ and $\hat{y}_j$ at $\boldsymbol{s}_i$ can be written as

$$\hat{y}_j + \frac{||\boldsymbol{s}_i - V_j||}{||V_j - V_\ell||} \cdot (\hat{y}_\ell - \hat{y}_j) = \frac{||\boldsymbol{s}_i - V_\ell||}{||V_j - V_\ell||} \cdot \hat{y}_j + \frac{||\boldsymbol{s}_i - V_j||}{||V_j - V_\ell||} \cdot \hat{y}_\ell = \boldsymbol{z}_i^T \hat{\boldsymbol{y}} \tag{3.11}$$

Consequently, the S-Lspline model in matrix form can be written as

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{Z}) = \boldsymbol{\beta}^T \boldsymbol{Z} \tag{3.12}$$

for $\boldsymbol{\beta}$ the $v \times 1$ column vector of coefficients with each coefficient $\beta_j = \hat{y}_j$ representing the predicted value on the corresponding knot. To estimate the parameter $\boldsymbol{\beta}$, we use the least square method, which leads to

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T \boldsymbol{Z})^{-1} \boldsymbol{Z} \boldsymbol{y} \tag{3.13}$$

Note that the S-Lspline model with the graph-transformed covariates does not include an intercept, which is different from the usual ordinary linear regression model.

Remark 8. An alternative way to validate the value-on-knots parameterization is via the

calculation of the degree of freedom. On each graph, the sum of the vertex degrees is twice the number of edges since each of the edges 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 have 2 degrees of freedom to fit a linear on each edge, and hence, without the constraints, the total number of degrees of freedom to spare is $2e$. Then for each vertex V_i with degree r_i, the continuity constraint imposes $r_i - 1$ equations, and consequently, the continuity constraints consume a total of $\sum_{i=1}^{v} r_i - 1 = r - v$ degrees of freedom. To put it together, we have $2e - (r - v) = v$ degrees of freedom, which agrees with the degrees of freedom given by the parametrization of values on the knots.

Remark 9. One natural idea is to extend the linear spline model to a higher-order spline on the skeleton. However, higher-order spline models may run into several issues. First, to ensure the desired degree of smoothness on the skeleton graph, we may need polynomials with degrees higher than that used in Euclidean case. Secondly, the odd-degree derivatives are directional and hence are dependent on the directions of the edges, and different edge directions can lead to different models on a graph. The discussions on higher-order splines on graphs is beyond the scope of this paper and we leave it as future work.

3.6 Simulations

In this section, we use simulated data to study the empirical performance of the proposed skeleton regression framework. We first show an example with the intrinsic domain composed of several disconnected components that we call the Yinyang data (Section 3.6.1). Then we add noisy observations to the Yinyang data (Section 3.6.2) to demonstrate the effectiveness of

our method in coping with noisy points. Moreover, we present an example where the domain is a continuous manifold with a Swiss roll shape (Section 3.6.3). For all the simulations included in this section, there are random perturbations in the intrinsic dimensions, and we add random Gaussian variables as covariates to make the ambient dimension large.

3.6.1 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 pertubations the generated covariates do not lay exactly on the corresponding manifold structures. The responses are generated from a trigonometric function on the ring and 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

$$Y = \epsilon + \begin{cases} \sin(\theta * 4) + 1.5 & \text{for points on the outer ring} \\ 0 & \text{for points on the bottom-right Gaussian cluster} \\ 1 & \text{for points on the right clump} \\ 2 & \text{for points on the left clump} \\ 3 & \text{for points on the upper-left Gaussian cluster} \end{cases} \tag{3.14}$$

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.

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. For each fold, there are 2560 training samples. We use the skeleton construction method described in Section 3.3.1 to construct skeletons with varying number of knots on each training set. The construction procedure also cuts each skeleton into 5 disjoint components according to the Voronoi Density weights (Section 3.3.1). 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 K).

We evaluate the skeleton-based kernel regression (S-kernel) as proposed in Section 3.4 and the skeleton spline model(S-Lspline) with the proposed parametrization in Section 3.5. For comparisons, we apply the classical k-Nearest-Neighbors regression in two ways: one vanilla version based on Euclidean distances (kNN) and one skeleton-related version using the skeleton-based distances (S-kNN). For penalization regression methods, we test Lasso and Ridge regression. Among the recent manifold and local regression methods, we compare with the Spectral Series approach with radial kernel (SpecSeries) for its superior performance and readily available R implementation [2]. We take the medium, 5 percentile, and 95 percentile

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

of the 5-fold cross-validation SSEs across each parameter setting for each method on the 100 datasets, and report the smallest medium SSE for each method along with the corresponding best parameter setting in Table 3.6.

We observed that all the skeleton-based methods (S-Kernel, S-kNN, and S-Lspline) all demonstrate performance superior than the usual kNN in this setting. SpecSeries approach performs worse than the classical kNN in this example while is only slightly better than the Lasso regression. Ridge and Lasso regression, although with the penalization effect, give relatively high SSE. Therefore, skeleton regression framework has advantage dealing with covariates lying around manifold structures with noisy features.

In Figure 3.7, we present the medium SSE of the S-Lspline, S-Kernel, and S-kNN methods on skeletons with different number of knots, with the vertical dashed line indicating $[\sqrt{n}] = 51$ knots as suggested by the empirical rule, where n is the training sample size. We see that the empirical rule leads to satisfactory results in this simulation study, approximately identifying the "elbow" position, but carrying out cross-validation for fine-tuning is advised in practice.

3.6.2 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 3.6.1 (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

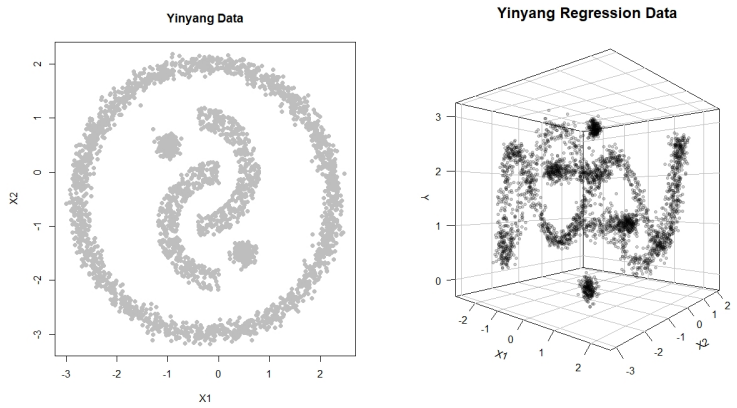


Figure 3.6: Yinyang Regression Data

Method	Medium 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
S-Kernel	112.8 (102.0, 121.7)	38	bandwidth = 6 r_{hns}
S-kNN	139.6 (129.6,148.7)	38	neighbor = 36
S-Lspline	95.8 (88.6, 102.6)	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 5 percentile and 95 percentile of the SSEs from the given parameter settings are reported in bracket.

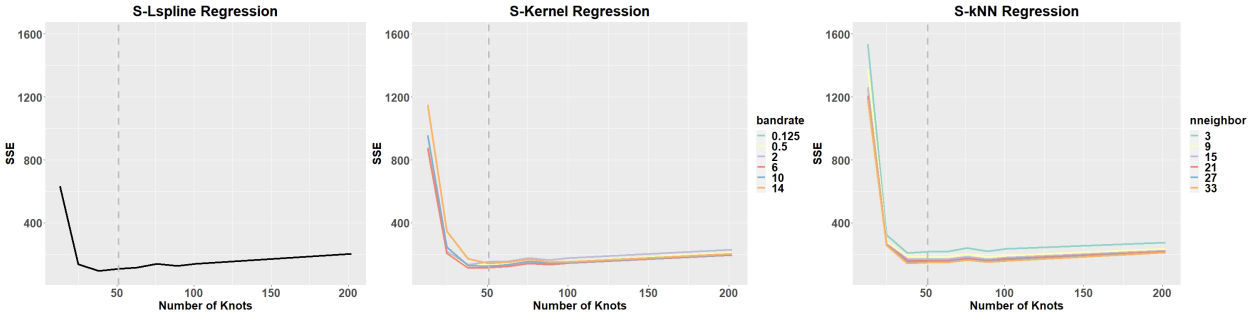


Figure 3.7: Yinyang $d = 1000$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

the same as in Equation 3.14. 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.

We randomly generate the Noisy Yinyang data 100 times and follow the same analysis procedure as in Section 3.6.1 except that we leave the skeleton that we fit our regression estimators on to be a fully connected graph. Similarly, we take the medium, 5 percentile, and 95 percentile of the 5-fold cross-validation SSEs across each parameter setting for each method on the 100 datasets, and report the smallest medium SSE for each method along with the corresponding best parameter setting in Table 3.2. We see that all the skeleton-based regression methods outperform the usual kNN and the SpecSeries approach. Ridge and Lasso regressions again fail to give good performance on this simulated dataset.

In Figure 3.7, we plot the medium SSE of the skeleton-based methods on skeletons with different number of knots. Using the empirical rule to construct skeleton with $[\sqrt{3200}] = 57$ knots leads to good regression results and approximately identifies the "elbow" position in Figure 3.7. However, using a number of knots larger than that given by the empirical rule leads to better regression results for some skeleton-based methods. This improvement relates to the phenomenon observed in Wei and Chen (2021) that, when noisy observations are included, we need a skeleton with more knots and cut the skeleton into more disjoint components to give a clean representation of the key manifold structures. Therefore, in practice, when facing data with noisy feature vectors, empirically tuning the number of knots favoring larger than $[\sqrt{n}]$ values is advised.

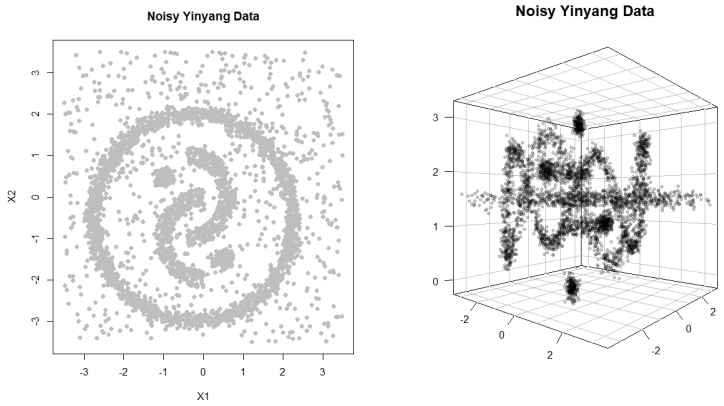


Figure 3.9: Noisy Yinyang Regression Data

Method	Medium 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
S-Kernel	385.7 (365.2, 406.0)	57	bandwidth = 6 r_{hns}
S-kNN	417.6 (396.1, 440.6)	71	neighbor = 36
S-Lspline	377.7 (358.1, 398.9)	71	-

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 bracket.

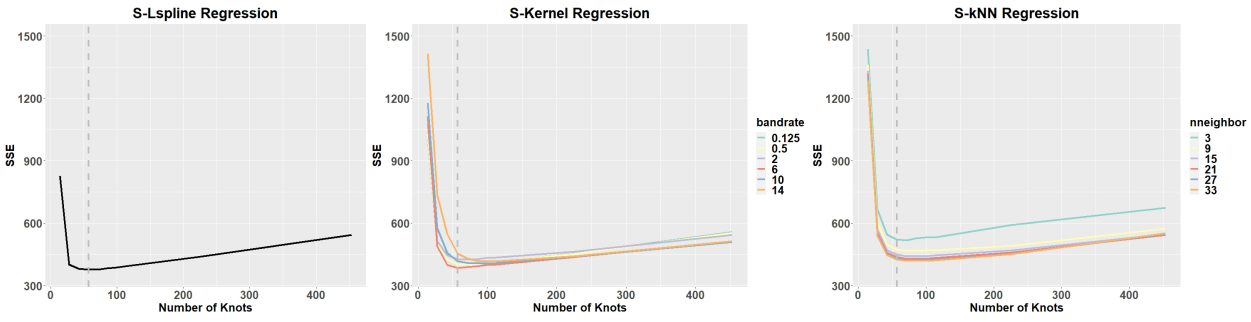


Figure 3.10: Noisy Yinyang $d = 1000$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

3.6.3 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 randomly generate the data 100 times and use the same analysis procedures as in Section 3.6.1. Similarly, we take the medium, 5 percentile, and 95 percentile of the 5-fold

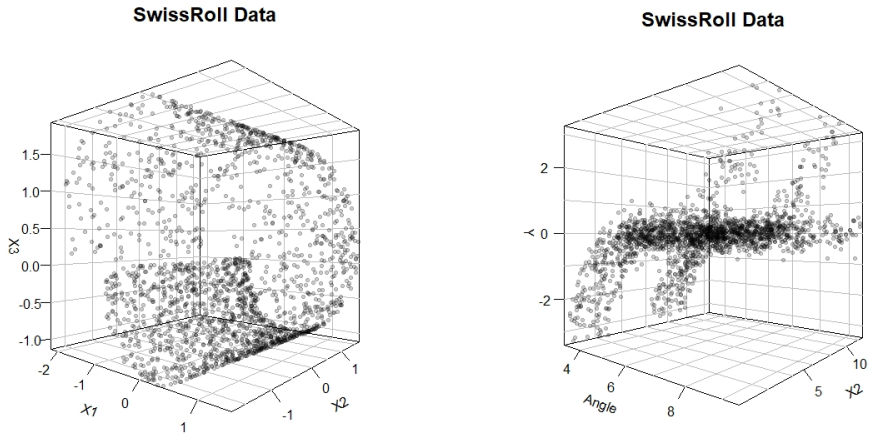


Figure 3.12: SwissRoll Regression Data

Method	Medium 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
S-Kernel	588.7 (527.0, 653.7)	70	bandwidth = 4 r_{hns}
S-kNN	614.7 (561.2, 692.6)	70	neighbor = 27
S-Lspline	578.6 (508.0, 629.6)	60	-

Table 3.3: Regression results on SwissRoll $d = 1000$ 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 bracket.

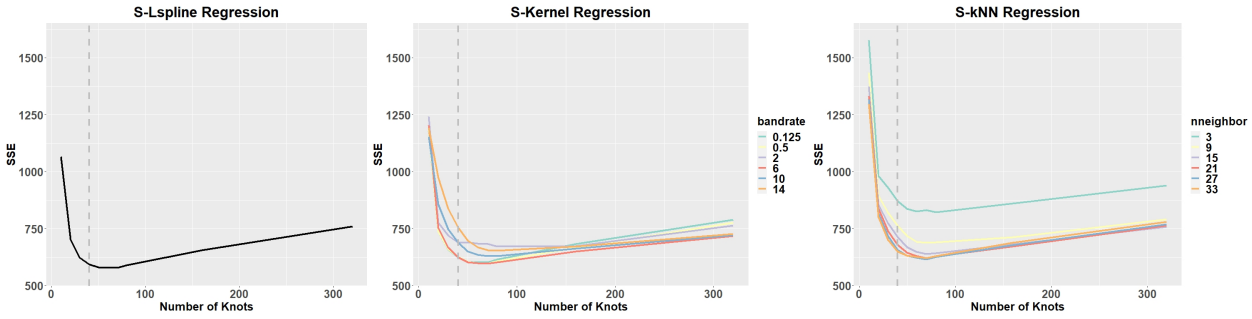


Figure 3.13: SwissRoll $d = 1000$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

cross-validation SSEs across each parameter setting for each method on the 100 datasets, and report the smallest medium SSE for each method along with the corresponding best parameter setting in Table 3.3. All the proposed skeleton-based methods have performance better than the usual kNN regressor, while the S-Lspline method gives the best performance in terms of SSE. The SpecSeries approach in this setting has performance similar to the Lasso regression and fails to improve much of the regression result utilizing information about the underlying manifold structure, potentially due to the large number of noisy dimensions. Therefore, the proposed skeleton regression framework can also be powerful for data on connected, multidimensional manifolds.

By plotting the medium SSE under skeletons with a varying number of knots, we note that the best performances for all the skeleton-based methods are achieved with the number of knots larger than $[\sqrt{1600}] = 40$ knots. Considering the intrinsic structure of the Swiss Roll input space to be a 2D plane, more knots on the plane can give a better representation of the data structure and hence leading to better prediction accuracy. We conjecture that the best number of knots should depend on the intrinsic dimension of the covariates and we leave the detailed discussion on this as future work, while empirically deploying cross-validation to choose the number of knots is recommended.

3.7 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 3.7.1). For the second example, we study the galaxy sample from the Sloan Digital Sky

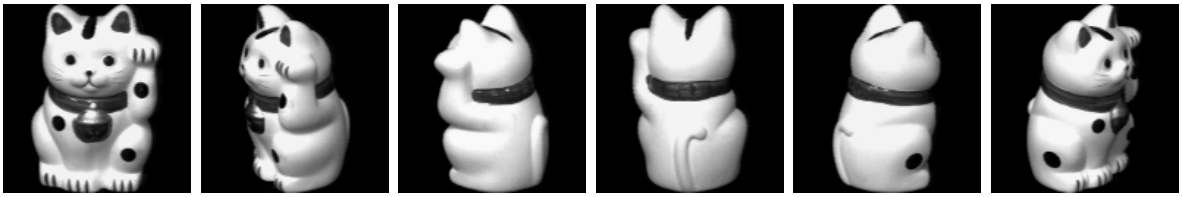


Figure 3.15: A part of the lucky cat images from the COIL-20 processed dataset. Each image is of size 128 pixels.

Method	SSE	Parameter
kNN	888.9	neighbor=9
Ridge	-	-
Lasso	-	-
SpecSeries	-	-
S-Kernel	1205.9	bandwidth $= 4r_{hns}$
S-kNN	2604.2	enighbor $= 6$
S-Lspline	338.1	-

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

Survey (SDSS) to predict the spectroscopic redshift (Section 3.7.2), a measure of distance from a galaxy to earth.

3.7.1 Lucky Cat 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 lucky cat obtained through rotating the object by 72 equispaced angles on a single axis. Several examples of the images are given in Figure 3.15. The response is the angle of rotation. However, this response has the circular response issue that degree 0 is the same as degree 360. To avoid this issue, we remove the last 8 images from the sequence and only use the first 64 images. Hence, our dataset consists of 64 samples of a 1-dimensional manifold embedded in $\mathbb{R}^{16384}$

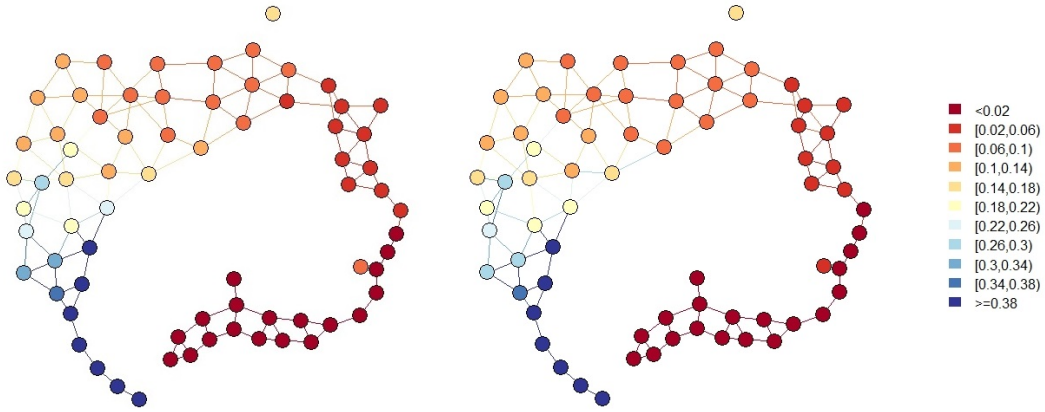
along with scalar values representing the angle of rotation.

To assess the performance of each method, we use leave-one-out cross-validation. Similar to the simulation studies, we use the skeleton construction method with Voronoi weights in Wei and Chen (2021) to construct the skeleton on the training set. In practice, we observe that a small number of knots can still lead to loops in the constructed skeleton structure, and with some tuning, we fit $2[\sqrt{n}] = 16$ knots to each training set. Also, with the knowledge that the underlying manifold should be one connected structure, we do not cut the constructed skeleton structure in this experiment. Ridge regression, Lasso regression, and Spectral Series approach fail to run on this high-dimensional data with the implementations in R. The best SSE from each method is listed in Table 3.4 along with the corresponding parameters. We see that the S-Lspline method gives outstanding performance on this real data, significantly outperforming the kNN regressor.

3.7.2 SDSS Data

In this section, we apply the proposed methods to a size 5000 galaxy sample, which is a random subsample from the Sloan Digital Sky Survey (SDSS). This data has 5 covariates measuring apparent magnitudes of stars from images taken using 5 photometric filters. These 5 covariates can be understood as the color of a galaxy and they are inexpensive measurements. The response is the spectroscopic redshift, which is an expansive measurement of the actual distance from a galaxy to the earth.

We construct the skeleton with the method in Wei and Chen (2021) and fit the S-Lspline model which gives the predicted values on each knot. We color the knots by their predicted



(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	Parameter
kNN	67.8	neighbor=12
Ridge	870.3	$\lambda = 0.001$
Lasso	882.7	$\lambda = 0.001$
SpecSeries	66.6	bandwidth = 2
S-Kernel	90.6	bandwidth = $10r_{hns}$
S-kNN	95.8	neighbor = 39
S-Lspline	89.6	-

Table 3.5: Regression results on SDSS data. The best SSE from each method is listed with the corresponding parameters used.

redshift values and color the edges by the average predicted values of the two connected knots. The resulting skeleton graph is shown in Figure 3.17 left. To compare to the true values, we also color the knots by the average responses within the corresponding Voronoi cells and color the edges by the average responses within the 2-Nearest-Neighbor regions (Figure 3.17 right).

The predictions given by S-Lspline are very close to the true values. Additionally, our

method provides the advantage to see the underlying nearly 1-dimensional structural trend of the real data, which can provide more intuition for practitioners.

We also carry out the same analysis procedure as the simulation studies in Section 3.6 comparing the 5-fold cross-validation SSE of different regression methods on this dataset. Although the proposed skeleton-based methods do not have superior performance on this dataset, the skeleton representation illustrates the hidden manifold structure in the covariate space and the corresponding regression function. The 1-dimensional structure recovered from the skeleton explains reveals that the data lay on a 1-dimensional manifold and the regression function gradually increases along the manifold. Therefore, nonparametric methods such as kNN and SpecSeries that can adapt to the intrinsic manifold will work well in this case.

3.8 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 k-Nearest-Neighbors 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 Simplicial Sets**

  In geometry perspective, the skeleton graph constructed in this work only concerns about 0-simplices (points) and 1-simplices (line segments), while additional geometric

information can be encoded in higher-dimensional simplices. Some interesting works in deep learning have incorporated simplicial complexes for clustering and segmentation tasks (Bronstein et al., 2017; Bodnar et al., 2021). For future direction, it can be interesting to generalize the framework in this work to use simplicial sets, which naturally are higher-dimensional generalizations of directed graphs.

- **Other nonparametric smoothers on graphs**

  There are some existing works that apply nonparametric regression estimators to graphs. Particularly, Wang et al. (2016) has generalized the idea of trend filtering Kim et al. (2009); Tibshirani (2014) to graphs and compared it to Laplacian smoothing and Wavelet smoothing. Compared to our work, those regression estimators on graphs apply to data whose inputs and responses are all on the vertices of a given graph. Therefore, those graph smoothers, with different regularizations, only fit values on the vertices without modeling the regression function on the edges (although in Wang et al. (2016) linear interpolation is used to draw some plots).

  Although the problem setup is different, one may construct responses on the knots in the skeleton graph as the mean values of the corresponding Voronoi cell or the k-Nearest-Neighbors average, and then use graph smoothers to fit the values on the knots. Linear interpolation can again be used to predict the response at points on the edge, and this can make a direct comparison to the skeleton-based linear spline method introduced in this work.

- **Time-varying covariates and responses**

  Another possible direction is to apply the skeleton regression framework to time-varying covariates and responses. Particularly, covariates across time can be utilized together

to construct knots in a skeleton representing data across time. The edges in the skeleton are allowed to change according to the covariate distribution at different times, which illustrate how the covariate distributions have changed. Also, we can represent the regression function on the skeleton, so it is easy to illustrate how the regression function changes.

- **Streaming data and online skeleton update.** Steaming data problem is becoming more and more prevalent these days. So one future direction is to study how to update the skeleton structure and its regression function in an online fashion. Reconstructing the entire skeleton could be computationally expansive but performing a local update on edge/knot editing is cheap. We will study how to design a simple and reliable skeleton update method in the future.

Chapter 4

Assessing Epidemic Models under Missingness in Contact Network

4.1 Introduction

graph is a structure of connections, which make it natural to represent various networks, with contact network being one example. Due to the advancement in mobile communication technology, collection of contact network data, 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 call 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 flows data are also used as proxy to contact network for epidemic modeling (Fajgelbaum et al., 2021; Alsing et al., 2020). Facing the challenge of the global pandemic, the Google COVID-19 Aggregated Mobility Research

Dataset becomes 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 network has substantial implications 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 mathematical models, in terms of policy decisions, to missingness about the underlying contact graph.

4.2 Non-Robustness to Missingness

In this section, we formulate the contact network missingness problem mathematically and present some preliminary results.

4.2.1 Problem Setup

Real-world contact patterns tend to have high clustering, geographic locality, some measure of sparsity, and some shortcuts or idiosyncratic long-range links (Banerjee et al., 2013; Harris et al., 2019). Previously used statistical models in the study of social networks such as latent space models, geography with idiosyncratic links, small-worlds networks such as lattices with idiosyncratic rewiring all have these features (Watts and Strogatz, 1998;

Hoff et al., 2002; Penrose, 2003; Jackson, 2008). To capture this, we study a sequence of undirected, unweighted contact networks G_n on vertex set V_n indexed by population $|V_n| = n$, constructed as follows. Let L_n be a sequence of graphs modeling local neighborhood connections. Let E_n be a sequence of Erdos-Renyi random graphs with link probability β_n. Then the overall contact network can be set as $G_n := L_n \cup E_n$.

We look at a disease process that originates with seed i_0 at time $t = 0$ and propagates through G_n as follows. In every period $t \in \mathbb{N}$, every node that is presently infected infects each of its neighbors independently with identical probability p. That is, for an infected node i, there is a probability p that i infects a neighbor j, independently for all neighbors of i. An infected node recovers the following period. The process is SIS so a recovered node is susceptible once again. The infection status of node i at time t is given by $y_{it} \in \{0, 1\}$.

The researcher's goal is to estimate the α-risk set of individuals who have infection probability larger than a threshold α that, for $\alpha \in (0, 1)$,

$$Q_\alpha(T; G_n) = \{j : \Pr(y_{jT} = 1 \mid i_0, G_n) \geq \alpha\}.$$

However, the researcher observes $\hat{G}_n$ rather than the true graph G_n. We assume $\hat{G} = L$ and use them interchangeably depending on context. And the researcher can only identify

$$Q_\alpha\left(T; \hat{G}_n\right) = \left\{j : \Pr\left(y_{jT} = 1 \mid i_0, \hat{G}_n\right) \geq \alpha\right\}.$$

where for the estimation we assume the researcher knows about the true p as, for this work, we focus on the impact of missingness in the contact network. However, to separate out the effect of missingness alone and to focus on the impact of missingness on transmission process, we assume a diminishing number of missing links. Let $d_{H,j}$ denote the degree of node j in some graph H.

Assumption 1. The share of links in G_n coming from E_n tends to zero that

$$n\beta_n = o\left(\mathbb{E}\left(d_{L,j}\right)\right).$$

Since the epidemic spreads through the graphs, the structure of the walk-count is a relevant feature of the network topology that influences the ability to track spread. Let $x_i\left(j,t;H_n\right) = \left[H_n^t\right]_{ij}$ be the number of walks from i to j of length t through graph H_n. Consider any two walks on H_n from i_0 to j of length T. The walk k is a sequence of edges of length T where the first edge begins with i_0 and the last edge ends at j. We can denote the set of walks between i_0 to j of length T as $\mathcal{P}_{i_0,j}(T)$ and often suppress the i_0, j and T notation where it is obvious. Note that $|\mathcal{P}_{i_0,j}(T)| = x_j(i_0, T; H_n)$.

We say that walks s and s' overlap on $r(s,s')$ edges if r edges in s and s' are used at the same times. For example, the walks $\{i_0a, ab, bc, cd, de, ef, fg, gj\}$ and $\{i_0b, bc, cd, de, eh, hf, fg, gj\}$ are each of length 8 and $r = 2$ as they share the edges fg, gj as being traversed by the disease at the same time (here periods 7 and 8). Although many other edges are used by both walks, they are not traversed at the same time and hence not the "same" walk. Note that this implies that two edges are not in common if they are the same edge, but in different parts of the sequence. It is useful to let K denote the set of all unordered pairs of walks in $\mathcal{P}$.

Assumption 2. The virulence of the disease on network G_n satisfies (as $n \to \infty$)

1. $p_n \cdot \mathbb{E}d_G > 1$.
2. $x_j\left(i_0;T,G_n\right) \cdot p_n^T = o\left(\min\{1,\ 1/\mathbb{E}_r\left[p_n^{-r}\right]\}\right)$ where r is the number of edges in common between any two random walks of length T picked between i_0 and j, such that $\mathbb{E}_r\left[p_n^{-r}\right] = \frac{1}{\binom{x_j(i_0,T;H_n)}{2}} \sum_{(s,s')\in K} p_n^{-r(s,s')}$.

This assumption captures three natural features of the world. Condition (1) requires that the reproductive number is at least one—the disease spreads. Condition (2) has two components. First, it says that the probability of infection relative to the number of walks is sufficiently low that j is far from guaranteed to become infected (notice at minimum we require $x_j(T) \cdot p^T < 1$). Second, we require that if the graph has a large amount of correlation between the various walks between nodes i_0 and j, then the probability of node j getting infected declines. This accounts for correlation in the graph, and, combined with Condition (1), requires that the various walks are not too correlated – a feature we will see holds for most reasonable graphs used to model data. Note that in the extreme if no walks have overlap, then $x_j(T) \cdot p^T < 1$ is the binding condition.

4.2.2 Infection Probability Approximation

We begin by calculating the probability that a given node is infected in a given period. We establish an upper and lower bound for this likelihood which will be useful in calculating the statistician's risk set.

Proposition 6. Let Assumption 2 hold. Then for a given j node,

$$x_j(T) \cdot p^T \cdot (1 + o(1)) \leq \mathbb{P}(y_{jT} = 1 \mid G, p, T, i_0) \leq 1 - \left(1 - p^T\right)^{x_j(T)}. \tag{4.1}$$

In fact, under the maintained assumption since the binomial approximation applies it follows that the upper and lower bound are of the same order since $1 - (1 - p^T)^{x_j(T)} = x_j(T) \cdot p^T \cdot (1 + o(1))$. Therefore the infection probability for node j at time T can be closely approximated by $x_j(T) \cdot p^T$. This is useful because it allows us to focus on the number of walks as well as its distribution.

PROOF PROPOSITION 6. For the sake of more compact notation, we write $x_j(i_0, T; H_n) = x_j(T)$ for the duration of this proof. Let $z_s = 1$ if node j is infected by at time T along this walk s and 0 otherwise. First for upper bound, we note that

$$\begin{aligned}
\mathbb{P}(y_{jT} = 1 \mid G, p, T, i_0) &= \mathbb{P}\left(\cup_s z_s = 1, s = 1, \ldots, x_j(T)\right) \\
&= 1 - \mathbb{P}\left(\cap_s z_s = 0, s = 1, \ldots, x_j(T)\right) \\
&\leq 1 - \Pi_{s=1,\ldots,x_j(T)} \mathbb{P}\left(z_s = 0\right) \\
&= 1 - \left(1 - p^T\right)^{x_j(T)} \\
&= x_j(T) p^T (1 + o(1))
\end{aligned}$$

Where the approximation step in the last line follows from the binomial approximation. For the lower bound, we note that

$$\begin{aligned}
\mathbb{P}\left(\cup_s z_s = 1, s = 1, \ldots, x_j(T)\right) &\geq \sum_{s=1,\ldots,x_j(T)} \mathbb{P}(z_s = 1) - \sum_{s'<s, s=1,\ldots,x_j(T)} \mathbb{P}(z_{s'} = 1 \text{ and } z_s = 1) \\
&= \underbrace{x_j(T) \cdot p^T}_{(A)} - \underbrace{\sum_{s'<s, s=1,\ldots,x_j(T)} \mathbb{P}(z_s = 1 \text{ and } z_{s'} = 1)}_{(B)}
\end{aligned}$$

where we get the inequality by ignoring the further correlation terms from inclusion-exclusion formula.

Clearly (A) is on the same order as the upper bound, so we need to only show that (B) is of lesser order than (A) to complete the proof. Let $r(s, s')$ be a quantifier for the number of edges shared by s and s'. It follows that:

$$\mathbb{P}(z_s = 1 \text{ and } z_{s'} = 1) = p^{2T - r(s,s')}$$

We can therefore write

$$\begin{aligned}\sum_{s'<s;s,s'\in S} \mathbb{P}(z_s = 1 \text{ and } z_{s'} = 1) &= \sum_{s'<s;s,s'\in S} p^{2T-r(s,s')} \\ &= \binom{x_j(T)}{2} \mathbb{E}_r p^{2T-r} \\ &= \frac{x_j(T)^2 - x_j(T)}{2} p^{2T} \mathbb{E}_r p^{-r}.\end{aligned}$$

To show that (B) is of lesser order than (A), note that:

$$\frac{\frac{x_j(T)^2 - x_j(T)}{2} p^{2T} \mathbb{E}_r p^{-r}}{x_j(T) \cdot p^T} = \frac{x_j(T) - 1}{2} p^T \mathbb{E}_r p^{-r}$$

We can then show that this term goes to 0 as $n \to \infty$. First, note that:

$$x_j(T) p^T \mathbb{E}_r p^{-r} \to 0$$

by the second part of Assumption 2. Then, note that:

$$\begin{aligned}p^T \mathbb{E}_r p^{-r} &= p^T \frac{\sum_{(s,s')} p^{-r(s,s')}}{\binom{x_j(T)}{2}} \\ &= \frac{2}{x_j(T)(x_j(T)-1)} \sum_{(s,s')} p^{T-r(s,s')} \\ &\leq \frac{2}{x_j(T)(x_j(T)-1)} x_j(T)(x_j(T)-1) p^T \\ &= 2p^T \to 0\end{aligned}$$

where the first equality follows from the definition of $\mathbb{E}_r p^{-r}$. Thus, we know that:

$$\frac{x_j(T) - 1}{2} p^T \mathbb{E}_r p^{-r} \to 0$$

which completes the proof. $\square$

Next, we will prove a result that relates path counts in a graph to its spectral properties. Let H_n be an arbitrary adjacency matrix and we define a condition on the eigenvectors of H_n:

Definition 7 (Bounded Eigenvectors)**.** A graph H_n has bounded eigenvectors if, given its eigenvalues $\lambda_1(n) > \lambda_2(n) > ... > \lambda_n(n)$, the corresponding eigenvectors satisfying:

$$H_n u_i = \lambda_i(n) u_i$$

where $||u_i||_\infty$ exists for all i.

This condition makes entries of each eigenvector to be bounded – this ensures that we can properly normalize the eigenvectors. We define the following notation for the normalized eigenvectors

$$q_i = \frac{1}{||u_i||_\infty} u_i.$$

We note a few key implications of bounded eigenvectors. First, entries of each q_i have constant order as n becomes large. Second, if we focus on q_1, bounded eigenvectors has a natural interpretation in terms of centralities. As long as H_n is connected, by Perron-Frobenius, we know that entries of u_1 (and thus q_1) will be positive. This positive quantity is typically referred to as eigenvector centrality. By assuming that there is a maximal entry of u_1 as n becomes large, we assume that there is an upper bound on how central any given node can be in H_n.

We can then prove a proposition that relates the probability of infection on a graph H_n to its spectral properties.

Proposition 8. Let H_n be an undirected graph with bounded eigenvectors and have eigenvalues $\lambda_1(n) > \lambda_2(n) > ... > \lambda_n(n)$. Let q_1 be the first (normalized) eigenvector of H_n, with entry j as $q_1(j)$. Then:

$$\mathbb{P}(y_{jT} = 1 \mid T, i_0, H_n) \sim \lambda_1(n)^T q_1(i) q_1(j)$$

As $n \to \infty$.

This result directly relates the probability of infection in some graph H_n to the spectral properties of the graph itself. We do so by first proving a result on the walk counts between nodes i and j in H_n, and then applying Proposition 6.

Proof Proposition 8. We can first note that because H_n is a real symmetric matrix, we can diagonalize it as follows:

$$H_n = Q\Lambda Q' \Rightarrow H_n^T = Q\Lambda^T Q'$$

Where Λ is a diagonal matrix of the eigenvalues, and Q is a matrix with the eigenvectors as columns. We denote q_k as the kth column of Q. Then, we can compute:

$$\begin{aligned}
\left[H_n^T\right]_{ij} &= \left[Q\Lambda^T Q'\right]_{ij} \\
&= \left[\sum_{k=1}^{n} \lambda_k(n)^T q_k q_k'\right]_{ij} \\
&= \left[\lambda_1(n)^T q_1 q_1' + \sum_{k=2}^{n} \lambda_k(n)^T q_k q_k'\right]_{ij} \\
&= \lambda_1(n)^T \left[q_1 q_1'\right]_{ij} + \left[\sum_{k=2}^{n} \frac{\lambda_1(n)^T}{\lambda_1(n)^T} \lambda_k(n)^T q_k q_k'\right]_{ij} \\
&= \lambda_1(n)^T \left[q_1 q_1'\right]_{ij} + \lambda_1^T \sum_{k=2}^{n} \left(\frac{\lambda_k(n)}{\lambda_1(n)}\right)^T \left[q_k q_k'\right]_{ij} \\
&= \lambda_1(n)^T \left(\left[q_1 q_1'\right]_{ij} + \sum_{k=2}^{n} \left(\frac{\lambda_k(n)}{\lambda_1(n)}\right)^T \left[q_k q_k'\right]_{ij}\right)
\end{aligned}$$

We can then note that each term in the summation portion of the expression will be $o\left(\lambda_1(n)^T\right)$. We know that by definition $\lambda_1(n) > |\lambda_k(n)|$ for $k > 1$, by the Perron-Frobenius Theorem. In addition, by bounded eigenvectors, we know that entries of q_k will be of constant

order. Therefore, we have that:

$$\left[H_n^T\right]_{ij} = \lambda_1(n)^T \left(q_1(i)q_1(j)' + \sum_{k=1}^{n} \left(\frac{\lambda_k(n)}{\lambda_1(n)} \right)^T q_k(i)q_k(j) \right)$$

$$\sim \lambda_1^T q_1(i)q_1(j)$$

As $n \to \infty$. Note that our computations hold for generic nodes i and j. An application of Proposition 6 then yields:

$$\mathbb{P}(y_{jT} = 1 \mid T, i_0, H_n) \sim p^T \lambda_1(n)^T q_1(i_0)q_1(j)$$

Completing the proof. $\square$

4.2.3 Main Theorem

With the previous results, we now turn to the scenario of a policy maker studying an infection process on a graph. As previously introduced, the policy maker observes some base graph, $\hat{G}_n$. In reality, the disease proceeds on $G_n = \hat{G}_n \cup E_n$, where E_n is an Erdos-Renyi random graph on the same n nodes, with link probability β_n. We can examine the following quantity:

$$S(T) := \frac{\sum_j \mathbb{P}(y_{jT} = 1 \mid \hat{G}_n, p_n, T, i_0)}{\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n, p_n, T, i_0)}$$

This quantity captures the ratio of the expected number of infected nodes that the policy maker observes, compared to the true number of expected infections. This simplifies from the α-risk set $Q_\alpha(T; G_n)$ calculations by taking expectations that

$$\frac{|Q_\alpha(T; \hat{G}_n)|}{|Q_\alpha(T; G_n)|} = \frac{\sum_j 1\left\{\mathbb{P}(y_{jT} = 1 \mid \hat{G}_n, p_n, T, i_0) \geq \alpha\right\}}{\sum_j 1\left\{\mathbb{P}(y_{jT} = 1 \mid G_n, p_n, T, i_0) \geq \alpha\right\}}$$

but the calculation of the defined $S(T)$ can serve as a starting point.

We keep track of a few spectral properties of $\hat{G}_n$. As before, we let $\lambda_1(n)$ denote the

largest eigenvalue of $\hat{G}_n$; we generally write $\lambda_1(n) := \lambda_1$ from this point forward. Then, let u_1 be the eigenvector that exactly satisfies $\hat{G}_n u_1 = \lambda_1 u_1$, with maximal element ν. Following definition 7 as before, we denote $q_1 = u_1/\nu$. We can note that while u_1 itself is bounded, we know that w, a function of the ℓ_1-norm of u_1 will not be bounded as u_1 is a positive vector. As a final preliminary, we make a two part assumption: first, we make a further assumption on the structure of the graph, and second, we make an assumption on β_n, the rate at which idiosyncratic, unobserved links form in the graph.

Assumption 3. The following condition holds for $\hat{G}_n$, which establishes a condition for β_n. For some function $f_n = ||q_1||_1^2$, the lower bound on the idiosyncratic links is:

$$\beta_n = \omega\left(\frac{\lambda_1}{f_n}\right)$$

We can further characterize f_n in the case where ν is the maximal element of u_1. We can note that $f_n = ||q_1||_1^2$ implies:

$$\nu(\sqrt{f_n} - 1) = ||u_1||_1 - \nu$$

We can consider the implication of Assumption 3. We can first consider f_n. The equivalent characterization of f_n places a restriction on the relative centrality of the maximally central node. We can interpret $||u_1||_1 - \nu$ as the cumulative long run influence of nodes that are not maximally central – those with long run influence less than ν. We know that as $n \to \infty$, $||u_1||_1$ will grow without bound, so thus the right hand side of the the first condition grows without bound. We then use $\sqrt{f_n} - 1$ as the rate of growth, in terms of ν. The condition on β_n places a minimum on the rate at which idiosyncratic links form, in terms of this growth rate.

Then, we define the following quantities:

$$v := \sum_j q_1(i_0)q_1(j)$$

$$w := \sum_j \sum_k q_1(j)q_1(k) = ||q_1||_1^2 = \left(\frac{||u_1||_1}{\nu}\right)^2 = f_n$$

Where the final equality follows from Assumption 3. We can then state our first result.

Theorem 9. Let assumption 1, 2, and 3 hold and the observed graph $\hat{G}_n$ satisfies condition 7. For $S(T)$ defined as follows:

$$S(T) := \frac{\sum_j \mathbb{P}(y_{jT} = 1 \mid \hat{G}_n, p_n, T, i_0)}{\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n, p_n, T, i_0)}$$

For any $\varepsilon, \delta > 0$, the following holds.

1. There exists T_ε such that for all $T \geq T_\varepsilon$

$$S(T) < \varepsilon$$

2. There exists T_δ such that for all $T < T_\delta$

$$S(T) > 1 - \delta$$

As $n \to \infty$.

Proof. We can first re-write the numerator of $S(T)$ using Propositions 6 and 8.

$$\begin{aligned}\sum_j \mathbb{P}(y_{jT} = 1 \mid \hat{G}_n, p_n, T, i_0) &\sim \sum_j p^T \lambda_1^T q_1(i_0)q_1(j) \\ &= p^T \lambda_1^T q_1(i_0) \sum_j q_1(j) \\ &= p^T \lambda_1^T v\end{aligned}$$

Then, we can work with the denominator, $\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n, p_n, T, i_0)$. The goal here will be to re-write the expression in terms of $\hat{G}_n$ and β_n.

We will begin by computing the number of expected infections based on a single link in

E_n. We can think through the necessary summations. First, we sum from $s_1 = 0$ to $T-1$, where s_1 denotes the number of links in the walk before the link that goes through E_n. We sum from 0, as when $s_1 = 0$, this implies there is a link in E_n that connects directly to i_0. Then, we need to sum over the possible locations for the "start" of the extra link, accounting for the number of paths from i_0 to this node. This will be multiplied by a term that sums over both the locations over the end points of the link, and also the end point of the walk.

$$\underbrace{\beta_n}_{\text{Link}} \underbrace{p^T}_{\text{Disease}} \sum_{s_1=0}^{T-1} \left[\underbrace{\left(\sum_j x_j(i_0; s_1, \hat{G}_n) \right)}_{\text{Possible locations for link start}} \underbrace{\left(\sum_z \sum_k x_k(z; T-1-s_1, \hat{G}_n) \right)}_{\text{Possible Locations for Link End, Path End}} \right]$$

We can then compute the same expression for when there are two links used in E_n. The expression will have a similar format, following the same logic.

$$\beta_n^2 p^T \sum_{s_1=0}^{T-2} \sum_{s_2=s_1+1}^{T-1} \left[\left(\sum_j x_j(i_0; s_1, \hat{G}_n) \right) \left(\sum_z \sum_k x_k(z; s_2 - s_1 - 1, \hat{G}_n) \right) \left(\sum_v \sum_\ell x_\ell(v; T-1-s_2, \hat{G}_n) \right) \right]$$

The first time summation is over 0 to $T-2$, to leave space for the second link in E_n. The second summation index for time ensures that the second link must be "later" in the walk than the first one. The path count terms go from i_0 to the start of the first link in s_1 periods. The second term goes from the end of the first link to the start of the second link in $s_2 - s_1 - 1$ periods in $\hat{G}_n$. Then, the last term goes from the end of the second link to the end of the walk in the remaining $T - 1 - s_2$ periods.

For the sake of space, we now suppress dependence on $\hat{G}_n$. We can then extend this

computation to when L of the T links are in E_n.

$$\beta_n^L p^T \underbrace{\sum_{s_1=0}^{T-L} \sum_{s_2=s_1+1}^{T-L+1} \dots \sum_{s_L=s_{L-1}+1}^{T-1}}_{L \text{ sums}} \underbrace{\left[\left(\sum_j x_j(i_0; s_1) \right) \left(\sum_z \sum_k x_z(k; s_2 - s_1 - 1) \right) \dots \left(\sum_v \sum_t x_v(t; T - 1 - s_L) \right) \right]}_{L+1 \text{ Terms}}$$

For each link in E_n, we sum over the possible positions of the link in the walk, accounting for the fact there must be L links and the links have order. Then, we do a similar walk count computation for each section of the walk in $\hat{G}_n$. There will be $L+1$ of these sections, broken up by the L links in E_n. We can then finish this computation by summing over L, from 0 to T.

$$p^T \sum_{L=0}^{T} \beta_n^L \left[\sum_{s_1=0}^{T-L} \sum_{s_2=s_1+1}^{T-L+1} \dots \sum_{s_L=s_{L-1}+1}^{T-1} \left(\sum_j x_j(i_0; s_1) \right) \left(\sum_z \sum_k x_z(k; s_2 - s_1 - 1) \right) \dots \left(\sum_v \sum_t x_v(t; T - s_L - 1) \right) \right]$$

Then recall that by Proposition 8:

$$x_j(i; T) \sim \lambda_1^T q_1(i) q_1(j)$$

As $n \to \infty$. We can then plug this results into the expression derived above, and simplify. We will first work with the expression for when L of the T links in the walk are in E_n.

$$\beta_n^L p^T \sum_{s_1=0}^{T-L} \sum_{s_2=s_1+1}^{T-L+1} \dots \sum_{s_L=s_{L-1}+1}^{T-1} \Bigg[\left(\sum_j \lambda_1^{s_1} q_1(i_0) q_1(j) \right) \left(\sum_z \sum_k \lambda_1^{s_2-s_1-1} q_1(i) q_1(j) \right)$$
$$\dots \left(\sum_v \sum_t \lambda_1^{T-s_L-1} q_1(i) q_1(j) \right) \Bigg]$$
$$= \beta_n^L p^T \lambda^{T-L} v w^L \sum_{s_1=0}^{T-L} \sum_{s_2=s_1+1}^{T-L+1} \dots \sum_{s_L=s_{L-1}+1}^{T-1} 1$$
$$= \beta_n^L p^T \lambda^{T-L} v w^L \binom{T}{L}$$

Where $\binom{T}{L}$ is the standard binomial coefficient, which comes from selecting where in the T steps the L links will go. We can then sum over L:

$$\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n, p_n, T, i_0) = p^T v \sum_{L=0}^{T} \beta_n^L \lambda_1^{T-L} w^L \binom{T}{L}$$
$$= p^T v (\lambda_1 + \beta_n w)^T$$

Then, we can analyze the behavior of $S(T)$. By the above computations, we get that:

$$S(T) = \frac{\sum_j \mathbb{P}(y_{jT} = 1 \mid \hat{G}_n, p_n, T, i_0)}{\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n, p_n, T, i_0)}$$
$$\sim \frac{\lambda_1^T p^T v}{p^T v (\lambda_1 + \beta_n w)^T}$$
$$= \left(\frac{\lambda_1}{\lambda_1 + \beta_n w} \right)^T$$

Where the asymptotic is with respect to $n \to \infty$. Then, by Assumption 3, we know that:

$$\frac{\lambda_1}{\lambda_1 + \beta_n w} = \frac{\lambda_1}{\lambda_1 + \beta_n f_n} < 1$$

And thus there exists some T' such that for all $T \geq T'$, $S(T) < \varepsilon$ for $\varepsilon > 0$. Conversely, we also know that for some T'', for all $T < T''$, $S(T) > 1 - \delta$ for all $\delta > 0$. This completes the proof. $\square$

Remark 10. We can work with a simplified islands model. We assume that there are K

islands, each of which is a homogeneous, directed tree with degree d. We assume that the trees are homogeneous no matter the starting point (they go on for more than T levels no matter where in the tree you start). We can first compute that:

$$\sum_j \mathbb{P}(y_{jT} = 1 \mid \hat{G}_n) = p^T d^T$$

Where the only nodes that can be infected are those that are at in the tree that contains i_0. Note that this comes from there being d^T nodes at distance T from i_0, but that we ignore other nodes within distance T from i_0 due to the directed nature of the tree.

We can compute the expected number of infections in G_n, where trees can be linked together via E_n. We can compute:

$$\begin{aligned}\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n) &= p^T \sum_{L=0}^{T} \beta_n^L d^{T-L} K^L \binom{T}{L} \\ &= p^T \left(\beta_n K + d\right)^T\end{aligned}$$

We can walk through the first step, and then note the simplification follows from the binomial theorem. Fix some number $L \leq T$, such that L is the number of links in E_n on a given walk. We can note that the probability of there being L links is β_n^L, and that within $\hat{G}_n$, this will access d^{T-L} nodes. It only reaches d^{T-L} nodes because L of the links are in E_n, and thus only serve as "connectors" on the walks. Then, we can link to K possible islands at each stage, which happens L times. In doing so, we assume that links between sections of tree happen sufficiently "far" from each other – this avoids cases where we have paths that loop (starting down a tree, and then linking to a node further up the tree). Finally, there are $\binom{T}{L}$ places to choose the L links. Finally, the disease travels with probability p^T.

Then, we can work with $S(T)$:

$$\begin{aligned} S(T) &= \frac{\sum_j \mathbb{P}(y_{jT} = 1 \mid \hat{G}_n)}{\sum_j \mathbb{P}(y_{jT} = 1 \mid G_n)} \\ &= \frac{p^T d^T}{p^T (\beta_n K + d)^T} \\ &= \left(\frac{d}{d + \beta_n K} \right)^T \end{aligned}$$

So the condition for $S(T)$ to be arbitrarily close to 1 for small T, and arbitrarily close to 0 for large T will be:

$$\beta_n = \omega \left(\frac{d}{K} \right)$$

If $K = \Theta(n)$, then we have $\beta_n > (cd)/n$ for some constant c. The key assumption here will be that links in E_n are sufficiently far apart in the trees. This makes intuitive sense, as there will be exponetially more nodes "farther down" than "up" each tree.

Remark 11. Consider the following model for $\hat{G}_n$. There are K communities in space, each with a lattice like structure within them denoted K_i $i \in \{1, ..., K\}$. Each K_i contains n_i nodes, with $\sum_i n_i = n$. For simplicity, we can let all islands have the same population $n_K = n/K$. Let $i_0 \in K_1$, and all nodes in K_1 are at most T steps away. The policy maker is unaware of links that are added with iid probability β_n. Note that even if $\beta_n < \log n/n$, the islands may still be connected. We can compute that for some $\varepsilon > 0$, the following ensures that the islands are connected with high probability:

$$\beta_n = (1 + \varepsilon) \frac{K^2 \log n}{n^3}$$

By noting that:

$$\begin{aligned}\mathbb{P}(K_i \text{ connected to } K_j) &= 1 - (1-\beta_n)^{n_K^2} \\ &= 1 - (1-\beta_n)^{n^2/K^2} \\ &= \beta_n \frac{n^2}{K^2}\end{aligned}$$

Where the last line assumes that the standard binomial approximation is reasonable.

We can build up to the same computations as before, but now taking advantage of the island structure. We suppress the dependence of $x_j(i; T, \hat{G}_n)$ on $\hat{G}_n$ for the sake of brevity – in all cases, we consider paths within $\hat{G}_n$. We can first compute:

$$\begin{aligned}\sum_{j=1}^{n} \mathbb{P}(y_j T = 1 \mid \hat{G}_n, p_n, T, i_0) &= p^T \sum_{j \in K_1} x_j(i_0; T) \\ &\sim p^T \lambda_1^T q_1(i_0) \sum_{j \in K_1} q_1(j)\end{aligned}$$

Where the only change from the more general case is that we now have a smaller eigenvector weight, as the values in q_1 are non-negative. As before, we can write the denominator of $S(T)$ as a function walks in $\hat{G}_n$. We can begin building up as before.

We will build up to the sum by assuming that there are L links in a given walk that are idiosyncratic – these can be across islands. When $L = 0$, we have the same expression as above for when we only consider $\hat{G}_n$: the disease cannot escape K_1 without the idiosyncratic links. We can then begin with $L = 1$.

$$\underbrace{\beta_n}_{\text{Link}} \underbrace{p^T}_{\text{Disease}} \sum_{s_1=0}^{T-1} \left[\underbrace{\left(\sum_{j \in K_1} x_j(i_0; s_1) \right)}_{\text{Possible locations for link start}} \underbrace{\left(\sum_{z \in G_n} \sum_{k \in G_n} x_k(z; T-1-s_1) \right)}_{\text{Possible Locations for Link End, Path End}} \right]$$

Note that the first summation only considers $j \in K_1$, as those are the only nodes reachable from i_0 before using idiosyncratic links. The double summation sums over all nodes in the

graph, as after the idiosyncratic link, the walk could go anywhere. We know that if $z \in K_i$ and $k \in K_j$, then $x_k(z;t) = 0$ for all t, as this only considers walks through the disconnected islands of $\hat{G}_n$. Thus, all of those terms will be 0. We can note that if we apply the spectral results, we find that this expression reduces to:

$$\begin{aligned}
&\beta_n p^T \sum_{s_1=0}^{T-1} \left[\left(\sum_{j \in K_1} x_j(i_0; s_1) \right) \left(\sum_{z \in G_n} \sum_{k \in G_n} x_k(z; T-1-s_1) \right) \right] \\
\sim\ &\beta_n p^T \sum_{s_1=0}^{T-1} \left[\left(\sum_{j \in K_1} \lambda_1^{s_1} q_1(i_0) q_1(j) \right) \left(\sum_{K_i} \sum_{z \in K_i} \sum_{k \in K_i} \lambda_1^{T-s_1-1} q_1(z) q_1(k) \right) \right] \\
=\ &\beta_n p^T \lambda_1^{T-1} \left[\left(q_1(i_0) \sum_{j \in K_1} q_1(j) \right) \left(\sum_{K_i} \sum_{z \in K_i} \sum_{k \in K_i} q_1(z) q_1(k) \right) \right]
\end{aligned}$$

In an analogous expression to before, though with slightly different eigenvector weights. We can potentially refine the second term here, noting the cases where z and k are not in the same island, as thus the eigenvalue weights should be ignored. We can note that the $v_I := q_1(i_0) \sum_{j \in K_1} q_1(j)$ weights will cancel in the final expression for $S(T)$, as before. Thus, the only change in the overall expression will be in w. We write this as w_I:

$$w_I := \sum_{j \in G_n} \sum_{k \in G_n} \sum_{K_i} 1\{j, k \in K_i\} q_1(j) q_1(k) = \sum_{K_i} \sum_{j \in K_i} \sum_{k \in K_i} q_1(j) q_1(k)$$

We can note that $w_I \leq w$, as defined before, as we only add a subset of the non-negative terms. The overall computation for $S(T)$ will be the same as before, only now with v_I and w_I in place of v and w. Thus we will find that for $S(T)$ to have the desired properties,

$$\beta_n = \omega \left(\frac{\lambda_1}{w_I} \right)$$

Where we have a more extreme condition than before.

4.3 Simulation Result

In this section, we present preliminary simulation results to demonstrate our theorem. Fix $n = 35000$ and we generate L as a random geographic network with specified expected degree Penrose (2003). Each node in L is assigned a position on $[0, 1]^2$ uniformly at random, and forms links to all nodes within some radius r. To obtain the desired expected degree $d(L) = 75$, we define $r = \sqrt{\frac{d(L)}{\pi n}} = \sqrt{\frac{75}{\pi n}}$. We add links independently to L with probability $\beta = (1+\varepsilon)\frac{\log n}{n}$, with $\varepsilon = 0.0001$. This computation gives $\beta = 0.0003$, for an expected degree of 10.46 in E. We set a basic reproductive number $r_0 = 3$. We set instances where multiple links exist between two nodes to count as only a single link. To select p, we first set a basic reproductive number $r_0 = 3$. Then, we compute $p = \frac{r_0}{\bar{d}(G)}$, where $\bar{d}(G)$ is the average degree for the realized G. This computation gives $p = 0.035$.

We track two quantities: $\frac{|Q(\alpha;\hat{G},T)|}{n}$ and $\frac{|Q(\alpha;G,T)|}{n}$. These quantities represent the coverage of the confidence sets under the observed and true graphs. Figure 4.1 shows the ratios over time for $\alpha = 0.95$. Results are similar with lower values of α. For the first few periods, the ratios are extremely similar. However, they sharply diverge as the true confidence set covers almost the entire graph, while the observed confidence set expands slowly. The result demonstrates that missingness is truly a robustness concern for epidemic modeling.

4.4 Discussion

In this ongoing project, we have preliminary modeling results demonstrating that missingness in contact network has non-trivial effects on modeling epidemic behaviors. Therefore, further studies on network missingness in epidemic modeling are necessary. For future work,

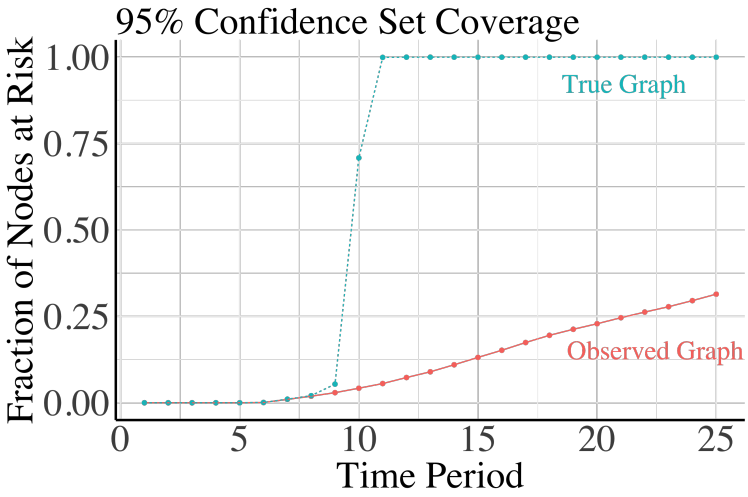


Figure 4.1: We track the number of nodes included within the $\alpha = 0.95$ level confidence set, compared to the total number of nodes in the graph under the observed and true graphs. For other values of α, the results are extremely similar. In the first few periods, the coverage of the confidence sets are similar under the observed and true graphs; however, they sharply diverge as the true confidence set rapidly envelopes the entire graph.

we first aim to theoretically characterize the conditions when missingness ruins epidemic model predictions. Moreover, as network geometry has impact on epidemic modeling, we want to study how network geometry interacts with missingness in the contact network, and how they jointly influence epidemic modeling. Furthermore, we want to assess the uncertainty caused by missingness, and how such uncertainty compare to the variation in the disease transmission process.

Bibliography

E. Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

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.

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.

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.

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.

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.

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.

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.

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.

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. 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, 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 2021. doi: 10.1257/aeri.20200401. URL https://www.aeaweb.org/articles?id=10.1257/aeri.20200401.

J. Fan and J. Fan. Design-adaptive Nonparametric Regression. 87(420):998–1004, 1992.

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.

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.

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.

K. M. Harris, C. T. Halpern, E. A. Whitsel, J. M. Hussey, L. A. Killeya-Jones, J. Tabor, and S. C. Dean. Cohort profile: The national longitudinal study of adolescent to adult health (add health). *International Journal of Epidemiology*, 48(5):1415–1415k, 2019.

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.

M. O. Jackson. *Social and economic networks*. Princeton: Princeton University Press, 2008.

J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

M. C. Jones. Simple boundary correction for kernel density estimation. *Statistics and computing*, 3(3):135–146, 1993.

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, 2009a.

S. Kpotufe. Escaping the curse of dimensionality with a tree-based regressor. 2 2009b. URL https://arxiv.org/abs/0902.3453v1.

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.

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.

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.

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.

M. Penrose. *Random geometric graphs*, volume 5. Oxford university press, 2003.

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.

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.

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.

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.

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.

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.

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.

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, 2006. doi: 10.1007/0-387-30623-4.

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.

Z. Wei and Y.-C. Chen. Skeleton clustering: Dimension-free density-based clustering. 2021. URL https://arxiv.org/abs/2104.10770.

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.

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.

Chapter 5

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 chooses 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 $O(ndk^{(1-1/d)})$. 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 weights 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 is 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 satisfies 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 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 $\int K^2(x)dx < \infty$, $\int |x|K(x)dx < \infty$, $\int x^2K(x)dx < \infty$.

Assumption (D1) is a commonly assumed for the density estimation problem, but usually with higher-order smoothness conditions. Notably, for consistency of 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 10 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, 2006; Scott, 2015) satisfied by many common kernel functions, including Gaussian kernel.

Theorem 10 (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

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

Theorem 10 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 10 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 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 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 on to 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 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 for this bias.

From Theorem 10, 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 to 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 10 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 poor representation of the data an 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 on 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 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 11 (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

$$\mathsf{pDisk}_{j\ell,R}(t) = O(R^{d-1}), \tag{2}$$

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

Theorem 11 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 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 10 and 11). 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 11 gives $O_p\left(\sqrt{\frac{1}{nhR^{d-1}}}\right)$ while the second term in Theorem 10 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 are 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 the 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 2.2.

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 $S_{j\ell}^{VD}$ 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 $\hat{S}_{j\ell}^{VD}$ 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 $\sigma_{j\ell}^2 = \frac{p_{j\ell}(1-p_{j\ell})}{n}$. 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 2.12.

To get the uniform bound, we first start with the concentration bound. Note that $\big(I(X_i \in A_{j\ell}) - p_{j\ell}\big)$ has zero mean and $|I(X_i \in A_{j\ell}) - p_{j\ell}| \leq 1$. Hence by Bernstein 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 2.12. 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 $\int K^2(x)dx < \infty, \quad \int |x|K(x)dx < \infty, \quad \int x^2 K(x)dx < \infty$.

Proof of Theorem 10.

Our analysis starts with the usual bias-variance decomposition that

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

We analyze the two term 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

$$\mathbb{E}[\hat{S}_{j\ell}^{FD}] = \mathbb{E}\bigg(\frac{1}{nh}\sum_{i=1}^{n} K\bigg(\frac{\Pi_{j\ell}(X_i) - c_*}{h}\bigg) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\bigg) \tag{4}$$

$$= \frac{1}{h}\int_{x\in\mathcal{X}} K\bigg(\frac{\Pi_{j\ell}(x) - c_*}{h}\bigg) I(x \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell) p(x)\mu_d(dx) \tag{5}$$

$$= \frac{1}{h}\int_{\mathbb{L}_{j\ell}} K\bigg(\frac{c_* + t\frac{c_\ell - c_j}{||c_\ell - c_j||} - c_*}{h}\bigg)\bigg(\int_{(\overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\cap T_{j\ell,t}} p(y)\mu_{d-1}(dy)\bigg) d\bigg(c_j + t\frac{c_\ell - c_j}{||c_\ell - c_j||}\bigg) \tag{6}$$

$$= \frac{1}{h}\int_{\mathbb{R}} K\bigg(\frac{\big\|t\frac{c_\ell - c_j}{||c_\ell - c_j||}\big\|}{h}\bigg) q_{j\ell}(t)dt \tag{7}$$

$$= \frac{1}{h}\int_{\mathbb{R}} K\bigg(\frac{t}{h}\bigg) q_{j\ell}(t)dt \tag{8}$$

$$= \int_{\mathbb{R}} K(u) q_{j\ell}(hu)du, \tag{9}$$

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 forth 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

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

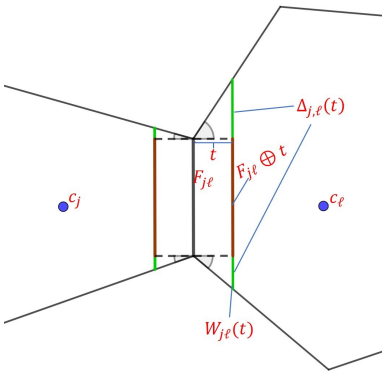


Figure 1: Decomposition of $W_{j\ell}(t)$. The dark red segment is $F_{j\ell} \oplus t$, which has the same shape with $F_{j\ell}$. The green segments consist $\Delta_{j,\ell}(t)$, the part leading to geometric bias.

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 1 for an example.

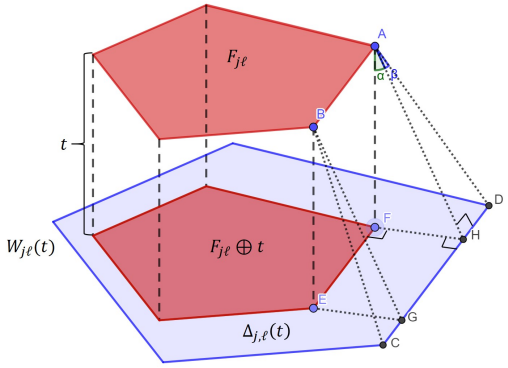


Figure 2: Decomposition of $W_{j\ell}(t)$. The red regions are $F_{j\ell}$ and the projected $F_{j\ell} \oplus t$, while the blue band region denotes $\Delta_{j,\ell}(t)$. All the α angles such as $\angle FAH$ and all the β angles such as $\angle HAD$ are bounded by θ_0 from assumption (B4).

Thus, the difference

$$\begin{aligned} q_{j\ell}(hu) - q_{j\ell}(0) &= \int_{W_{j\ell}(hu)} p(y)\mu_{d-1}(dy) - \int_{W_{j\ell}(0)} p(y)\mu_{d-1}(dy) \\ &= \underbrace{\int_{F_{j\ell}\oplus hu} p(y)\mu_{d-1}(dy) - \int_{F_{j\ell}} p(y)\mu_{d-1}(dy)}_{(I)} + \underbrace{\int_{\Delta_{j\ell}(hu)} p(y)\mu_{d-1}(dy)}_{(II)}. \end{aligned}$$

(I) is the usual bias caused by the change of 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 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

$$\big|q_{j\ell}(hu) - q_{j\ell}(0)\big| \leq \mu_{d-1}(F_{j\ell})C_g h|u| + p_{max}h|u| \cdot O\big(\mu_{d-2}(\partial F_{j\ell})\tan(\theta_0)\big) \tag{10}$$

This, together with equation (9), implies that

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

As a result,

$$|\mathbb{E}[\hat{S}_{j\ell}^{FD}] - S_{j\ell}^{FD}| = O\left(\frac{h}{k^{1-1/d}}\right) + O\left(\frac{h}{k^{1-2/d}}\right) \tag{11}$$

Moreover, note that

$$\frac{h}{k^{1-1/d}} \times \frac{k^{1-2/d}}{h} = \frac{1}{k^{1/d}} \to 0 \tag{12}$$

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

$$|\mathbb{E}[\hat{S}_{j\ell}^{FD}] - S_{j\ell}^{FD}| = O\left(\frac{h}{k^{1-2/d}}\right) \tag{13}$$

Stochastic variation: For the stochastic variation part, we have

$$\begin{aligned} Var(\hat{S}_{j\ell}^{FD}) &= Var\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i)-c_*)}{h}\right) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\right) \\ &\leq \frac{1}{nh^2}\mathbb{E}\left[K^2\left(\frac{\Pi_{j\ell}(X_i)-c_*}{h}\right) I(X_i \in \overline{\mathbb{C}}_j \cup \overline{\mathbb{C}}_\ell)\right] \\ &\leq \frac{1}{nh}\int K^2(u)\left(q_{j\ell}(0) + \mu_{d-1}(F_{j\ell})C_g + p_{max}h|u|\mu_{d-2}(\partial F_{j\ell})\tan(\theta_0)\right)du \\ &\leq \frac{1}{nh}\int K^2(u)\left(q_{j\ell}(0) + O\left(\frac{h}{k^{1-1/d}}\right) + O\left(\frac{h}{k^{1-2/d}}\right)\right)du \end{aligned} \tag{14}$$

by the same decomposition in (9) and the bound in (10) and the assumptions (K1). Note that similar to(12), the second term in (14) is at a slower rate than the third term, so we can simplify it as

$$Var(\hat{S}_{j\ell}^{FD}) = O\left(\frac{q_{j\ell}(0)}{nh}\right) + O\left(\frac{1}{nk^{1-2/d}}\right). \tag{15}$$

Combining (11) and (14), we conclude that for $\forall j, \ell$,

$$|\hat{S}_{j\ell}^{FD} - S_{j\ell}^{FD}| = O\left(\frac{h}{k^{1-2/d}}\right) + O_p\left(\sqrt{\frac{q_{j\ell}(0)}{nh}}\right) + O_p\left(\sqrt{\frac{1}{nk^{1-2/d}}}\right) \tag{16}$$

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

$$q_{j\ell}(0) = S_{j\ell}^{FD} \geq p_{\min} \min_{(j,\ell)\in E} \mu_{d-1}(F_{j\ell}) = p_{\min}\frac{c_0}{k^{1-\frac{1}{d}}}. \tag{17}$$

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 is decreasing with number of knots, and we have that This implies that

$$\left|\frac{\hat{S}_{j\ell}^{FD}}{S_{j\ell}^{FD}} - 1\right| = O\left(hk^{1/d}\right) + O_p\left(\sqrt{\frac{k^{1-\frac{1}{d}}}{nh}}\right) + O_p\left(\sqrt{\frac{k}{n}}\right) \tag{18}$$

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

$$\frac{k^{1-\frac{1}{d}}}{nh} \times \frac{n}{k} = \frac{1}{hk^{1/d}} \to \infty, \tag{19}$$

so the second term dominates the third term in (18) and the rate reduces to

$$\left|\frac{\hat{S}_{j\ell}^{FD}}{S_{j\ell}^{FD}} - 1\right| = O\big(hk^{1/d}\big) + O_p\left(\sqrt{\frac{k^{1-\frac{1}{d}}}{nh}}\right), \tag{20}$$

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 11.

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 $\hat{S}^{TD}_{j\ell} - S^{TD}_{j\ell}$.

The difference admits the following decomposition:

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

It is easier to start with term (III) and 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

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

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

$$2p_{\min}C_{d-1}R^{d-1} \le \mathsf{pDisk}_{j\ell,R}(t) \le 2p_{\max}C_{d-1}R^{d-1} = O(R^{d-1}) \tag{21}$$

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 is 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,

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

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

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

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

(II): Stochastic variation.

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

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

$$|\widehat{\mathsf{pDisk}}_{j\ell,R}(t) - \mathsf{pDisk}_{j\ell,R}(t)| = O(h^2 R^{d-1}) + O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right). \tag{22}$$

(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 term (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 the 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

$$\begin{aligned}|\hat{S}^{TD}_{j\ell} - S^{TD}_{j\ell}| &= (I) + (II) + (III) \\ &= O(h^4 R^{2d-2}) + O_p\left(\frac{R^{d-1}}{nh^3}\right) + O(h^2 R^{d-1}) + O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right) \\ &= O(h^2 R^{d-1}) + O_p\left(\sqrt{\frac{R^{d-1}}{nh}}\right) + O_p\left(\frac{R^{d-1}}{nh^3}\right).\end{aligned}$$

Using the fact that $S_{j\ell}^{TD} \geq 2p_{\min}C_{d-1}R^{d-1}$ from equation (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 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. Best linkages for each data scenario are in bold.

	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 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 cluster 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 number of clusters and include the number of cluster at which the max is achieved in the bracket.

From Table 1, either average linkage or single linkage achieve the best and 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

in Figure 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 lighter color band. The true number of clusters $S = 5$ is shown as the red dotted vertical line.

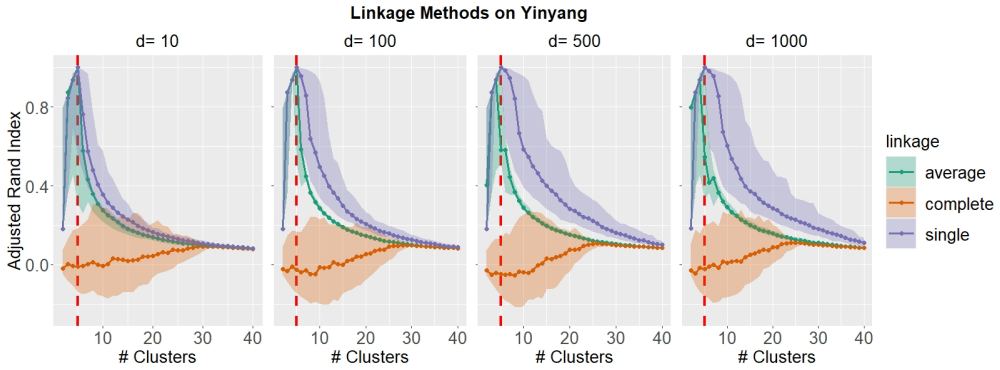


Figure 3: Clustering results with different linkage methods across different numbers of final clusters on Yinyang data. Line for medium and band from 5th percentile to 95th percentile. The vertical red dashed line indicates the true number of 5 clusters.

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 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 4.

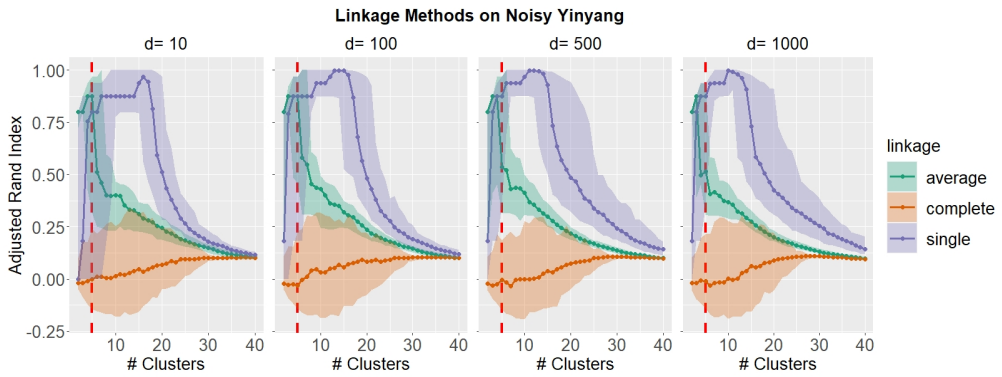


Figure 4: Clustering results with different linkage methods across different numbers of final clusters on Yinyang data with noisy points. The vertical red dashed line indicates the true number of 5 clusters.

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 get higher ($d = 500, 1000$). The performance of single linkage improves with S being slightly larger than the actual number 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 improves the clustering performance; see Figure 5. This suggests that our framework may be used for anomaly detection.

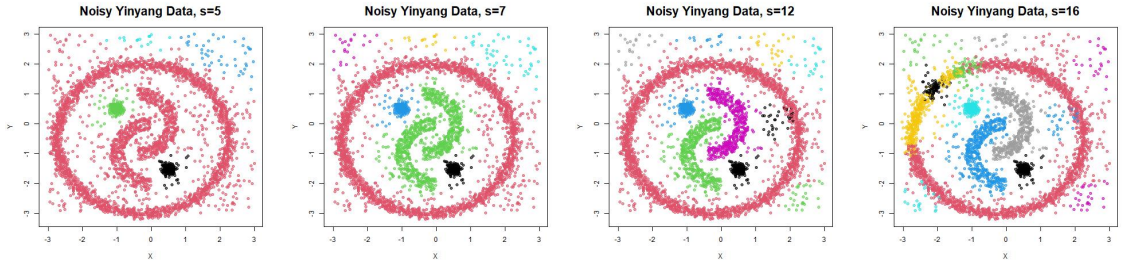


Figure 5: The clustering results with single linkage in skeleton clustering with different number of final clusters S for Noisy Yinyang data, $d = 1000$.

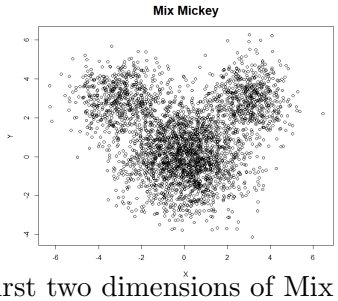


Figure 6: First two dimensions of Mix Mickey data.

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 2D case that we call it the Mix Mickey data. The large cluster is centered at $(0, 0)$ with the covariance matrix being a diagonal matrix of 2 and 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 via the same way we generate the Yinyang data. Figure 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 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.

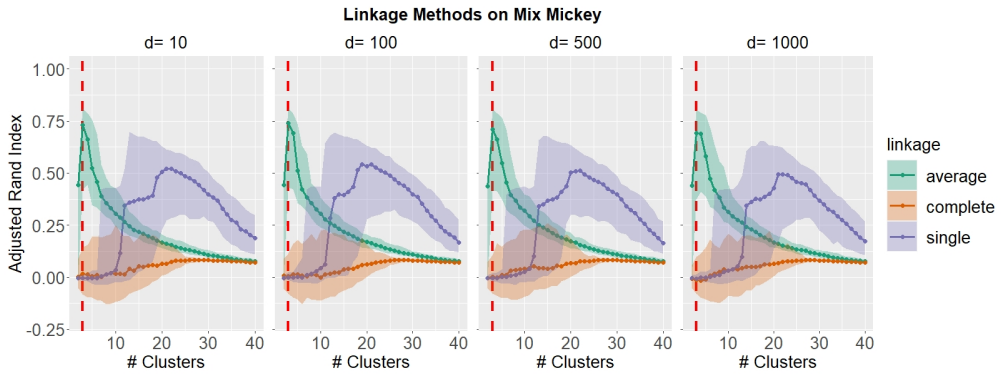


Figure 7: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data. The vertical red dashed line indicates the true number of 3 clusters.

We observe that average linkage gives good performance at $S = 3$ (the true number of 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 8.

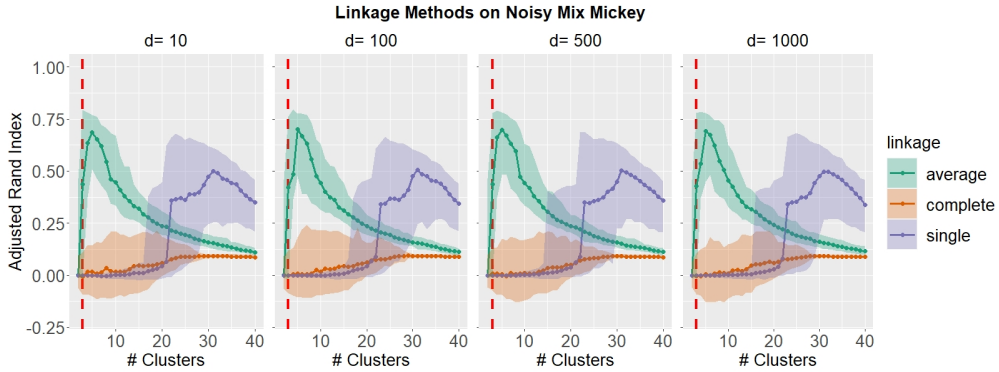


Figure 8: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data with Noise. The vertical red dashed line indicates the true number of 3 clusters.

Average linkage still give good performance and is superior than the single linkage, which 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 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 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 10 display the median clustering performance. Again, we see that average linkage has the best performance.

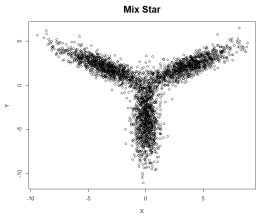


Figure 9: First two dimensions of the Mix Star data.

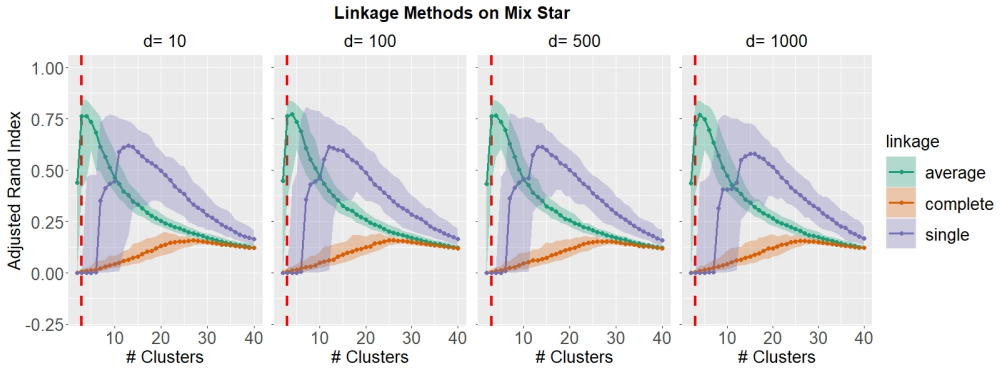


Figure 10: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data. The vertical red dashed line indicates the true number of 3 clusters.

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 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 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 criterion for skeleton segmentation. Broadly speaking, the appropriate choice of linkage method depends on the intrinsic

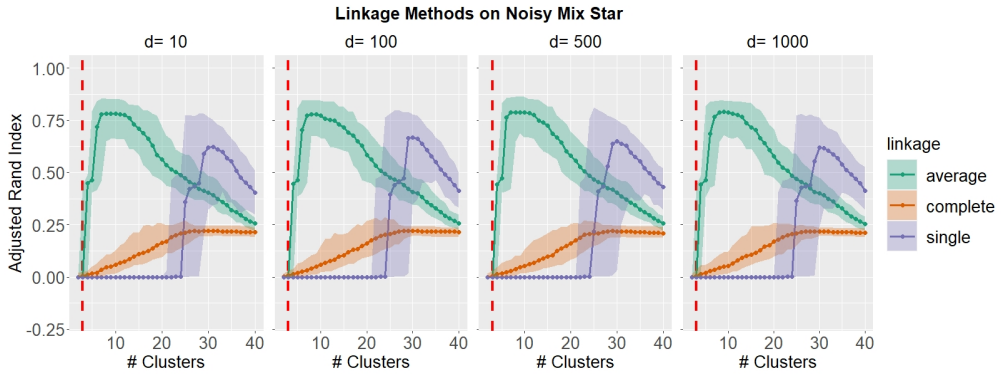


Figure 11: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data with Noise.

geometric structure of the data and may require subject matter knowledge or exploratory analysis. Specifically, if the intrinsic clusters are well-separated, single linkage is preferred as it gives clear cuts for disjoint components. But if the clusters are believed to have some degree of overlapping with each cluster approximately spherically shaped, average linkage criterion can lead to better performance.

All Linkage Comparisons

Figures 12 and 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.

Figures 14 and 15 present the median clustering performance under different number 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 how the clusters will be like in high dimensions, we present 2D

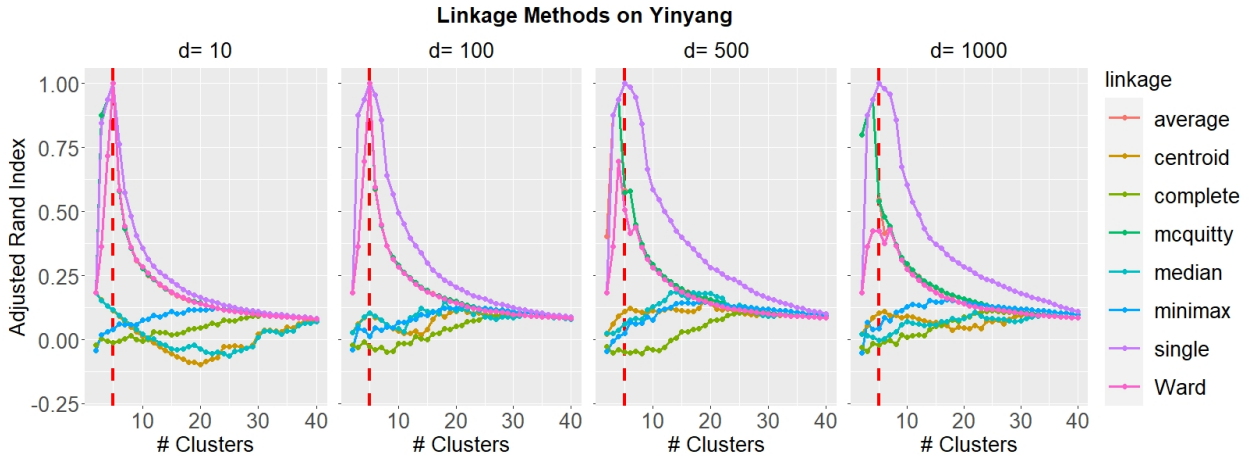


Figure 12: Clustering results with different linkage methods across different numbers of final clusters on Yinyang Data.

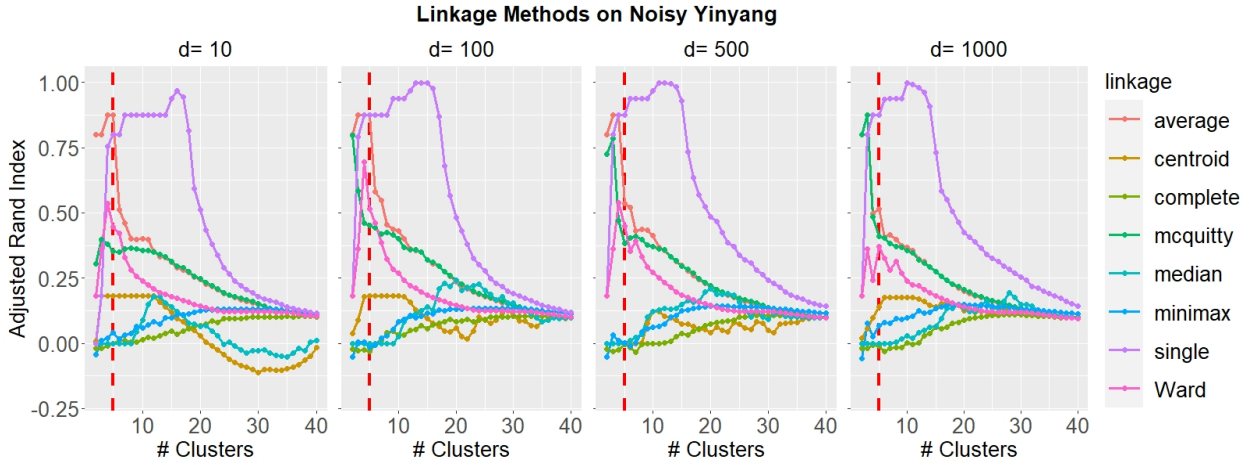


Figure 13: Clustering results with different linkage methods across different numbers of final clusters on Noisy Yinyang Data.

scatterplot of clustering results under $S = 3$ (final number of clusters is 3) of the first two coordinates in Figure 16. We use the data with $d = 1000$ and color the clusters using red, green, and blue. Clearly, average linkage successfully recover the actual clusters while other methods fail to recover. Note that single linkage does not perform well because clusters are overlapping with each other.

Figures 17 and 18 present the median clustering performance under different number of clusters for the Mix Star and noisy Mix Star data. We observe that average linkage and

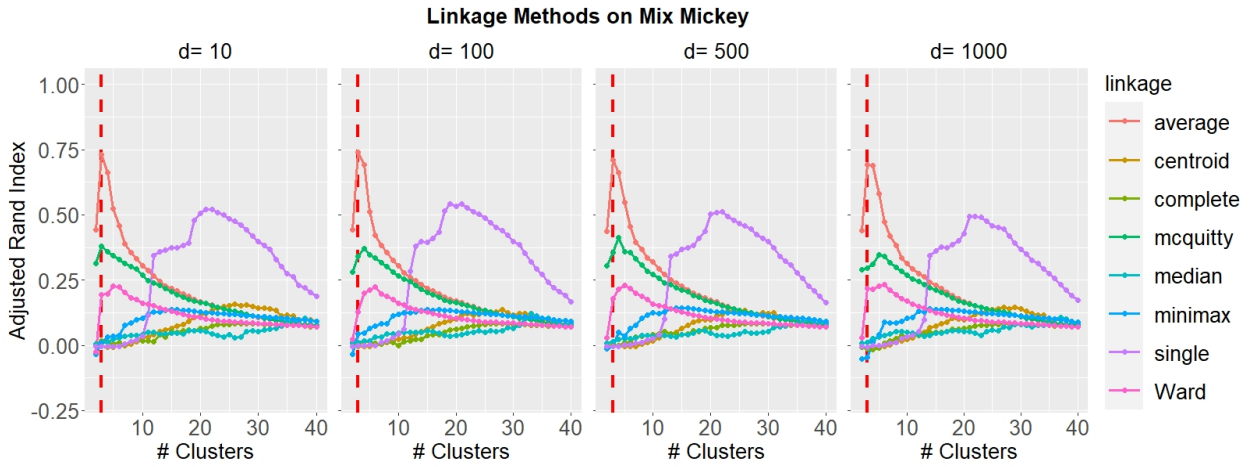


Figure 14: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data.

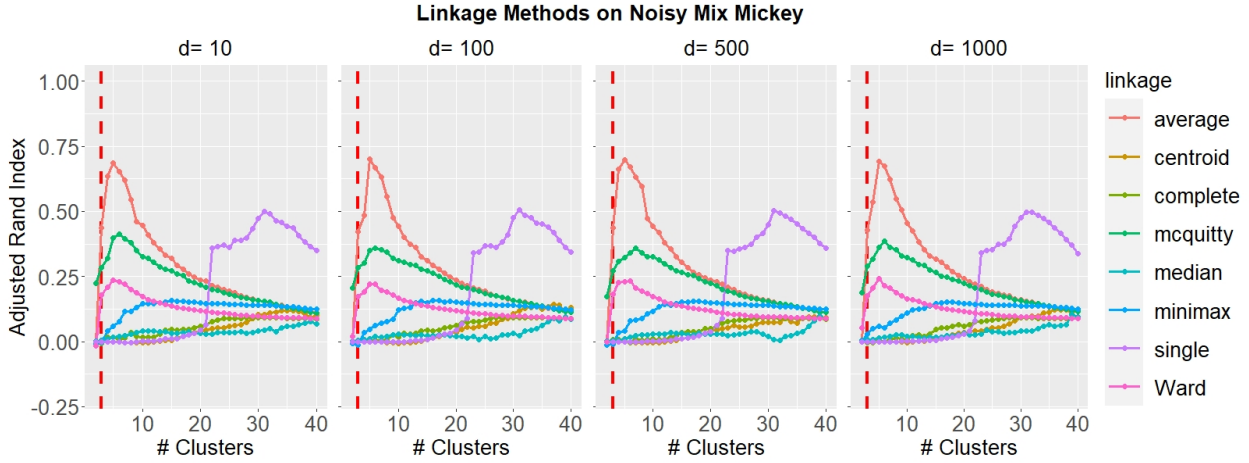


Figure 15: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data with Noise.

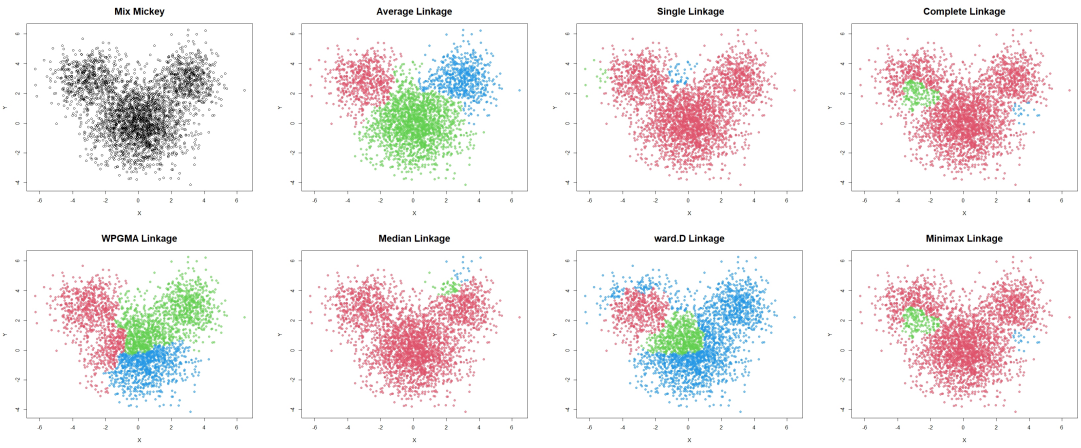


Figure 16: Comparing linkage criteria in segmentation on the Mix Mickey data, $d = 1000$.

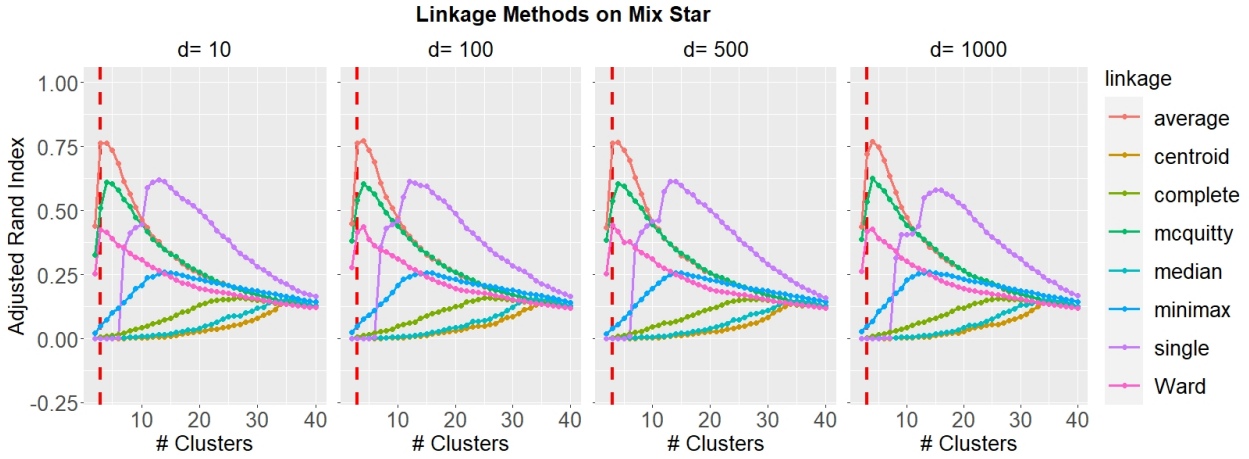


Figure 17: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data.

single linkage dominate all other methods.

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,

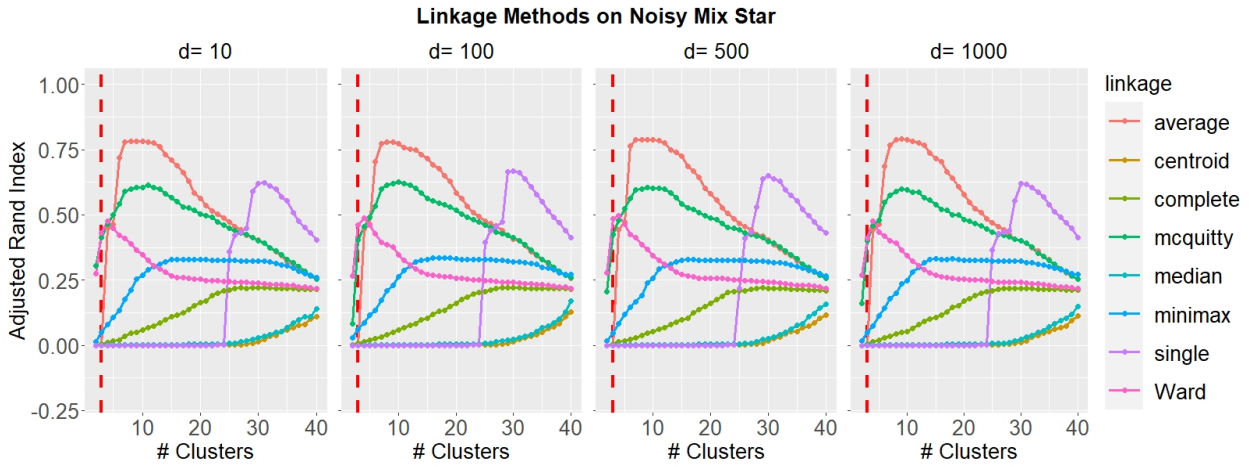


Figure 18: Clustering results with different linkage methods across different numbers of final clusters on Mix Star data with Noise.

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 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.

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

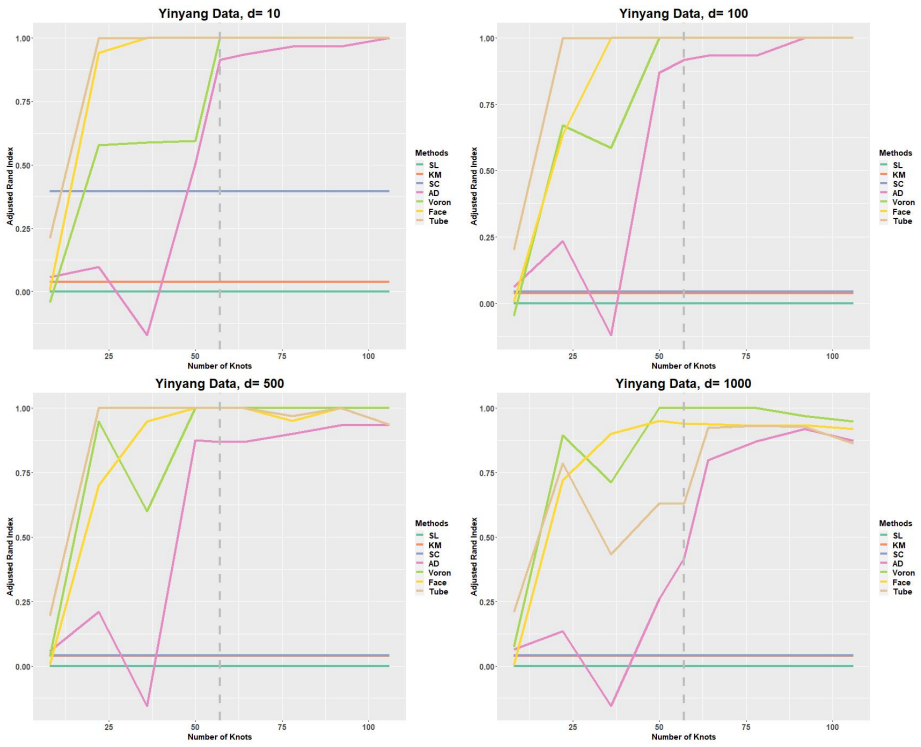


Figure 19: Adjusted Rand indexes of different clustering methods against different number of knots on 100 simulated Yinyang data.

SOM may return knots with very tiny sample size, 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 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

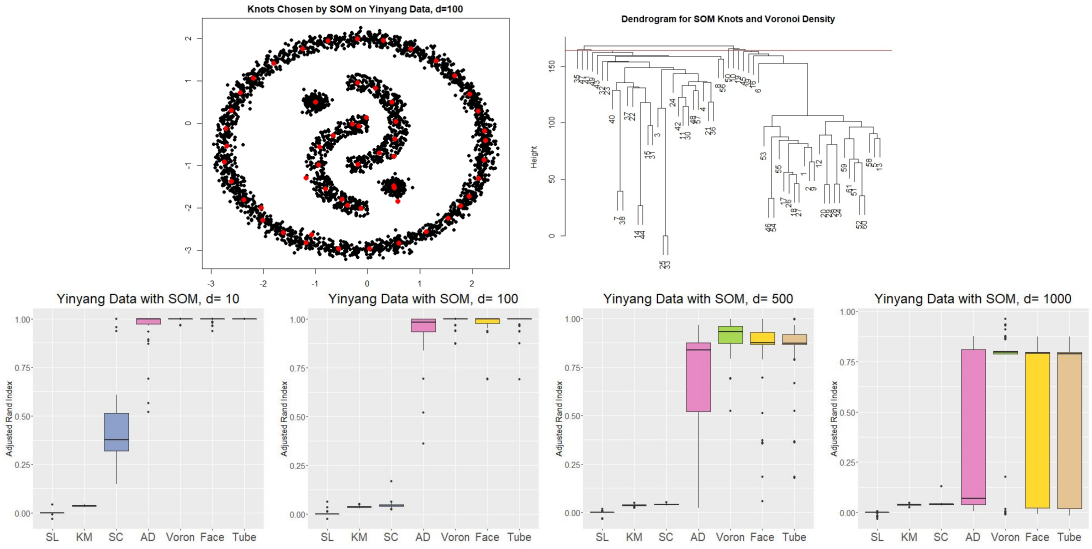


Figure 20: Adjusted Rand indexes using SOM for knots selection on Yinyang data.

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 that we need to perform some post-processing to remove tiny knots; in the case of k-means, we do not need such 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

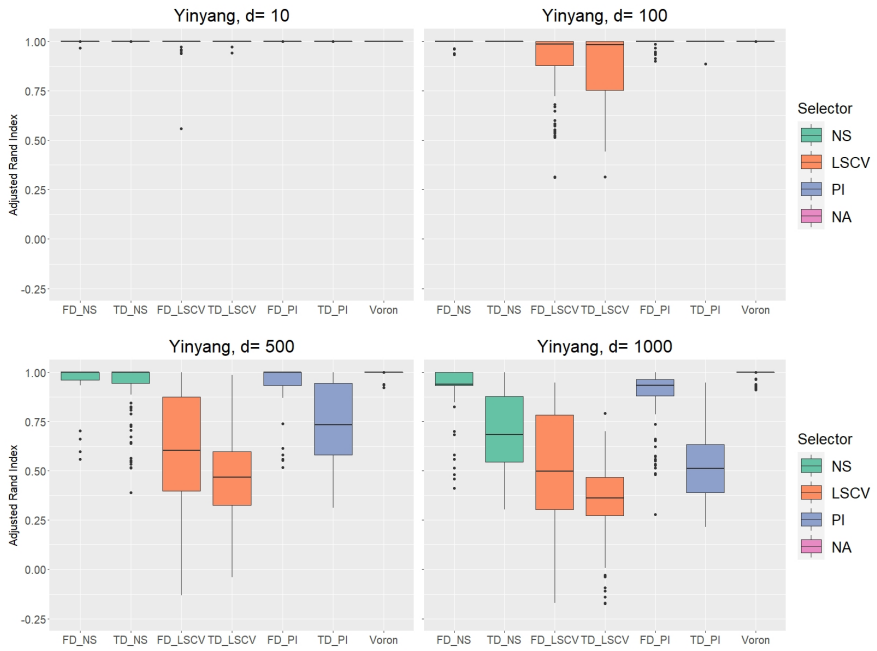


Figure 21: Performance of skeleton clustering on Yinyang data $d = 10, 100, 500, 1000$ with Face and Tube density by different bandwidth selectors. Voronoi density result is included for comparison.

estimation result of a kernel density estimator. In Figure 21, we conduct a simulation using 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 purpose. 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 $O(n_{loc}^{-1/5})$) 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 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 $d = 10, 100, 500, 1000$. We compare the Face and Tube density where the bandwidth is selected

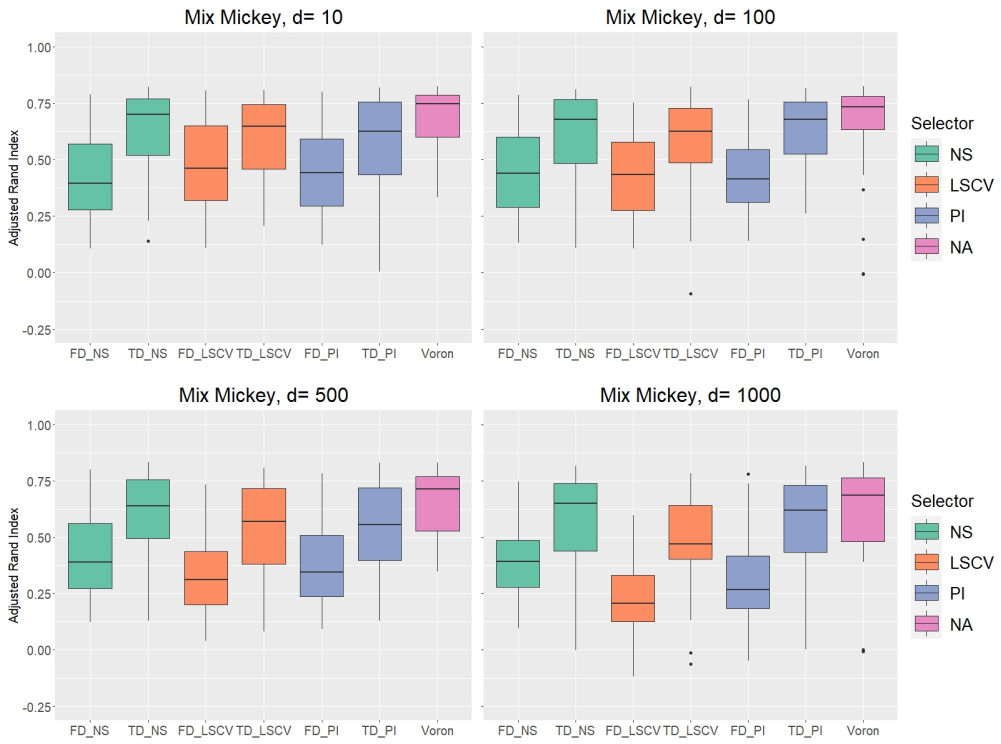


Figure 22: Performance of skeleton clustering on Mix Mickey data $d = 10, 100, 500, 1000$ with Face and Tube density by different bandwidth selectors. Voronoi density result is included for comparison.

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 adjust Rand index in Figure 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 result across different rates of bandwidth. Note that for the TD, the theory (Theorem 11) 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 a the optimal clustering performance. Figure 23 bottom-right panel suggests that the choice $h \asymp n_{loc}^{-1/10}$ may have a better clustering performance in this case.

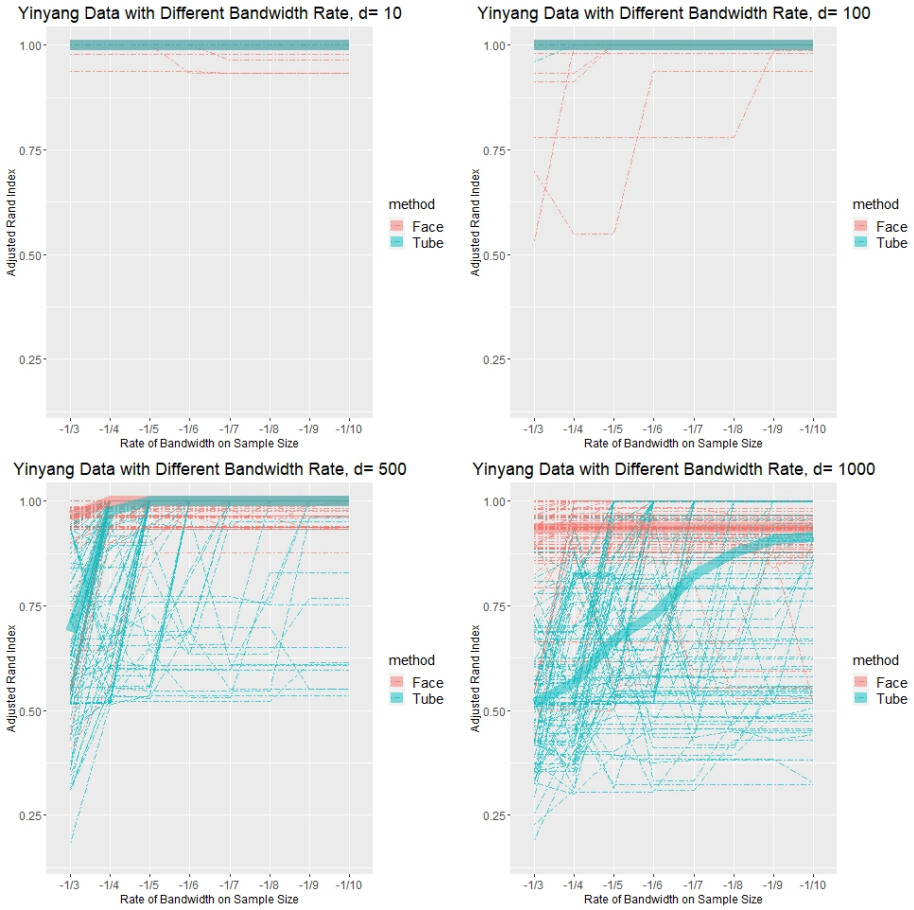


Figure 23: Adjusted Rand indexes of skeleton clustering with Face and Tube density under different bandwidth rate on 100 simulated Yinyang datasets. The thick lines indicate the median adjusted Rand index of a given method.

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 24. The two approaches (adaptive and fixed radius) have a similar performance.

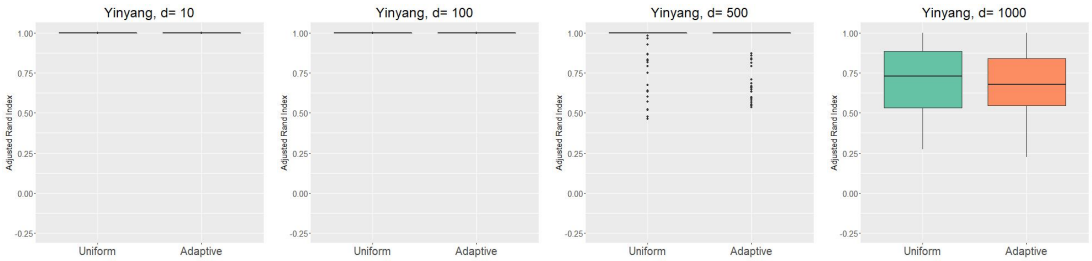


Figure 24: Comparison of radius choices on Yinyang data with dimensions 10, 100, 500, 1000.

Higher Standard Deviations for Noisy Dimensions

We investigate how does changing the noise level of the added noisy dimensions of our simulation examples change 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 methods on 100 simulated datasets with under setting are presented in Figure 25.

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

much 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 a 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 27. 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 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 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.

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 approached are applied as well as the classical approaches, with the final number of clusters chosen to be 2. The result is displayed in Figure 31. Again, the density-based skeleton clustering methods work well even when the dimension is large.

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 the four skeleton clustering approaches and compare them to three traditional methods: the direct single linkage hierarchical clustering (SL), the direct k-means clustering (KM), and spectral clustering (SC).

The result is shown in the left panel of Figure 32 with the adjusted Rand index plotted against different number 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 small number of clusters.

The right panel of Figure 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 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 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.

Same comparison procedure as in Section H is employed. The performance of different similarity measures is presented in Figure 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 Computational Complexity

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 (2021) 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 much time-consuming. Finding the edge to project depends on identifying the two nearest knots, which is provided in the skeleton construction step. Projection is taking inner product computations and takes $O(nd)$ for all the feature vectors.

The next major computational burden is to calculate the skeleton-based distance between points on the skeleton. The general version of Dijkstra's algorithm (Dijkstra, 1959) takes $\Theta(|\mathcal{E}|+|\mathcal{V}|^2) = \Theta(k^2)$ for each run. However, ideally, we want the pairwise distances between all the inputs, but finding the shortest path for $\frac{n(n-1)}{2}$ times can be time-consuming. In practice, we can speed up the calculation by constraining the distance calculation to local neighborhoods.

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.

J Proofs

Kernel Regression: Convergence on Edge Point

PROOFTHEOREM 3. Let $\mathcal{B}(\boldsymbol{s}, h_n) \subset \mathcal{S}$ be the support for the kernel function $K_h(.)$ at point $\boldsymbol{s} \in \mathcal{S}$ with bandwidth h_n. 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 $n \to \infty, h_n \to 0$, for sufficiently large n, by the property of an open set, we have

$$\mathcal{B}(\boldsymbol{s}, h_n) \subset E_{j\ell}$$

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 $K_h(\boldsymbol{S}_j, \boldsymbol{s}_\ell) \equiv K_h(d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell)) = K_h(\|\boldsymbol{S}_j - \boldsymbol{s}_\ell\|) = K\left(\frac{\boldsymbol{S}_j - \boldsymbol{s}_\ell}{h}\right)$.

Consequently, the skeleton-based kernel regression estimator reduces to

$$\hat{m}_n(\boldsymbol{s}) = \frac{\frac{1}{nh_n}\sum_{j=1}^n Y_j K(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n})}{\frac{1}{nh_n}\sum_{j=1}^n K(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n})} \tag{23}$$

and we can use the classical asymptotic results for kernel regression in continuous case with (Bierens, 1983; Wasserman, 2006; Chen et al., 2017).

Let $\hat{g}_n(\boldsymbol{s}) = \frac{1}{nh_n}\sum_{j=1}^n K\left(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n}\right)$. We express the difference as

$$\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s}) = \frac{[\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})]\hat{g}_n(\boldsymbol{s})}{\hat{g}_n(\boldsymbol{s})} = \frac{\frac{1}{nh_n}\sum_{j=1}^n [Y_j - m_{\mathcal{S}}(\boldsymbol{s})]K(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n})}{\frac{1}{nh_n}\sum_{j=1}^n K(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n})} \tag{24}$$

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:

$$\begin{aligned}
|\mathbb{E}\hat{g}_n(\boldsymbol{s}) - g(\boldsymbol{s})| &= \left| \frac{1}{h_n} \int K\left(\frac{\boldsymbol{s}-y}{h_n}\right) g(y)dy - g(x) \int K(y)dy \right| \\
&= \left| \int K(z)[g(\boldsymbol{s} - h_n z) - g(\boldsymbol{s})]dz \right| \\
&\leq \int K(z) C_1 \left| h_n z \right| dz = C_1 h_n \int K(z) \left| z \right| dz = O(h_n),
\end{aligned}$$

where C_1 is the Lipschitz constant of the density function. For the variance, we have

$$\begin{aligned}
\mathrm{V}ar\left(\hat{g}_n(\boldsymbol{s})\right) &\leq \frac{1}{nh_n^2} \int K^2\left(\frac{\boldsymbol{s}-y}{h_n}\right) g(y)dy \\
&= \frac{1}{nh_n} \int K^2(z) g(\boldsymbol{s} - h_n z)dz \\
&\leq \frac{1}{nh_n} \int K^2(z)[g(\boldsymbol{s}) + C_1 \left| h_n z \right|]dz \\
&= \frac{1}{nh_n} \left[g(\boldsymbol{s}) \int K^2(z)dz + C_1 h_n \int K^2(z) \left| z \right| dz \right] \\
&= \frac{1}{nh_n} g(\boldsymbol{s}) \int K^2(z)dz + o\left(\frac{1}{nh_n}\right).
\end{aligned}$$

Putting it altogether, we have

$$|\hat{g}_n(\boldsymbol{s}) - g(\boldsymbol{s})| = O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right).$$

Note that we only assume Lipschitz continuity and hence has the bias of rate $O(h_n)$ rather than the usual $O(h_n^2)$ 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 (24). We start with the decomposition

$$
\begin{aligned}
[\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})]\hat{g}(\boldsymbol{s}) = & \underbrace{\frac{1}{nh_n}\sum_{j=1}^{n} U_j K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)}_{q_1(\boldsymbol{s})} \\
& + \underbrace{\frac{1}{n}\sum_{j=1}^{n}\left\{[m_{\mathcal{S}}(\boldsymbol{S}_j) - m(\boldsymbol{s})]K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)\frac{1}{h_n} - \mathbb{E}\left[[m_{\mathcal{S}}(\boldsymbol{S}_j) - m_{\mathcal{S}}(\boldsymbol{s})]K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)\frac{1}{h_n}\right]\right\}}_{q_2(\boldsymbol{s})} \\
& + \underbrace{\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[[m_{\mathcal{S}}(\boldsymbol{S}_j) - m_{\mathcal{S}}(\boldsymbol{s})]K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)\frac{1}{h_n}\right]}_{q_3(\boldsymbol{s})}.
\end{aligned}
$$

First, we show that

$$
q_1(\boldsymbol{s}) = O_p\left(\sqrt{\frac{1}{nh_n}}\right).
$$

Let

$$
v_{n,j}(\boldsymbol{s}) = U_j K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)\frac{1}{\sqrt{h_n}}
$$

and we have

$$
\sqrt{nh_n}q_1(\boldsymbol{s}) = \frac{1}{\sqrt{n}}\sum_{j=1}^{n} v_{n,j}(\boldsymbol{s}).
$$

Thus, its mean is

$$
\mathbb{E}v_{n,j}(\boldsymbol{s}) = \mathbb{E}\left\{U_j K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)\frac{1}{\sqrt{h_n}}\right\} = 0
$$

and the variance is

$$
\begin{aligned}
\mathbb{E}[v_{n,j}(\boldsymbol{s})^2] = \mathbb{E}U_j^2 K\left(\frac{\boldsymbol{s}-\boldsymbol{S}_j}{h_n}\right)^2\frac{1}{h_n} &= \int \sigma_u^2(\boldsymbol{s}-h_n z)g(\boldsymbol{s}-h_n z)K(z)^2 dz \\
&\to \sigma_u^2(\boldsymbol{s})g(\boldsymbol{s})\int K(z)^2 dz = O(1),
\end{aligned}
$$

where for the second equality we use the change of variable and by assumption we have

$\int K(z)^2 dz < \infty$. Therefore,

$$q_1(\boldsymbol{s}) = O_p\left(\sqrt{\frac{1}{nh_n}}\right).$$

For the second term, note that $\mathbb{E}(q_2(\boldsymbol{s})) = 0$ and the variance is

$$\begin{aligned}\mathbb{E}\left[\sqrt{nh_n}q_2(\boldsymbol{s})\right]^2 &= \int [m_{\mathcal{S}}(\boldsymbol{s}-h_n z) - m_{\mathcal{S}}(\boldsymbol{s})]^2 g(\boldsymbol{s}-h_n z)K(z)^2 dz \\ &\quad - h_n\left\{\int [m_{\mathcal{S}}(\boldsymbol{s}-h_n z) - m_{\mathcal{S}}(\boldsymbol{s})] g(\boldsymbol{s}-h_n z)K(z)dz\right\}^2 \\ &\to 0\end{aligned}$$

when $h_n \to 0$, and hence,

$$q_2(\boldsymbol{s}) = o_p\left(\sqrt{\frac{1}{nh_n}}\right).$$

For the last term, note that we have

$$\begin{aligned}q_3(\boldsymbol{s}) &= \int [m_{\mathcal{S}}(\boldsymbol{s}-h_n z) - m_{\mathcal{S}}(\boldsymbol{s})] g(\boldsymbol{s}-h_n z)K(z)dz \\ &= \int [m_{\mathcal{S}}(\boldsymbol{s}-h_n z)g(\boldsymbol{s}-h_n z) - m_{\mathcal{S}}(\boldsymbol{s})g(\boldsymbol{s})]K(z)dz - m_{\mathcal{S}}(\boldsymbol{s})\int [g(\boldsymbol{s}-h_n z) - g(\boldsymbol{s})]K(z)dz \\ &\leq C_1 h_n \int |z|K(z)dz + C_2 h_n \int |z|K(z)dz\end{aligned}$$

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,

$$q_3(\boldsymbol{s}) = O(h_n)$$

Putting all three terms together, $[\hat{m}(\boldsymbol{s}) - m(\boldsymbol{s})]\hat{g}(\boldsymbol{s}) = O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)$. As a result, equation (24) becomes

$$\begin{aligned}\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s}) = \frac{[\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})]\hat{g}(\boldsymbol{s})}{\hat{g}(\boldsymbol{s})} &= \frac{O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)}{g(\boldsymbol{s}) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)} \\ &= O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)\end{aligned}$$

by Taylor expansion of the fraction.

□

Kernel Regression: Convergence on Knot with Zero Mass

For the ease of proof, we first prove Proposition 5 and then prove Theorem 4.

PROOFPROPOSITION 5. 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

$$\begin{aligned}
\hat{m}(\boldsymbol{s}) &= \frac{\frac{1}{n}\sum_{j=1}^{n} Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^{n} Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h_n))}{\frac{1}{n}\sum_{j=1}^{n} K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^{n} K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h_n))} \\
&= \frac{\frac{1}{n}\sum_{j=1}^{n} Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^{n} Y_j I(\boldsymbol{S}_j = \boldsymbol{s})}{\frac{1}{n}\sum_{j=1}^{n} K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^{n} I(\boldsymbol{S}_j = \boldsymbol{s})} \\
&= \frac{\varepsilon_{1,n}(\boldsymbol{s}) + \nu_{1,n}(\boldsymbol{s})}{\varepsilon_{2,n}(\boldsymbol{s}) + \nu_{2,n}(\boldsymbol{s})}.
\end{aligned}$$

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

$$\hat{m}(\boldsymbol{s}) = \frac{\frac{1}{nh_n}\sum_{j=1}^{n} Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n))}{\frac{1}{nh_n}\sum_{j=1}^{n} K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n))}.$$

However, different from the case on edges, the support of the kernel intersects with multiple edges even when $h_n \to 0$, so we study the contribution of each edge individually. Note that when $h_n \to 0$, the only knot that exists in the intersection $\mathcal{B}(\boldsymbol{s}, h_n) \cap \mathcal{E}$ 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 $I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) = \sum_{\ell \in \mathcal{I}} I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n))$. With this, we can rewrite $\hat{m}(\boldsymbol{s})$

as

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{\ell\in\mathcal{I}} \frac{1}{nh_n} \sum_{j=1}^{n} Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n))}{\sum_{\ell\in\mathcal{I}} \frac{1}{nh_n} \sum_{j=1}^{n} K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n))}$$
$$= \frac{\sum_{\ell\in\mathcal{I}} \hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s})}{\sum_{\ell\in\mathcal{I}} \hat{g}_{n,\ell}(\boldsymbol{s})}.$$

where

$$\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh_n} \sum_{j=1}^{n} K\left(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n}\right) I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n)),$$
$$\hat{m}_{n,\ell}(\boldsymbol{s}) \cdot \hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh_n} \sum_{j=1}^{n} Y_j K\left(\frac{\boldsymbol{S}_j - \boldsymbol{s}}{h_n}\right) I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n)).$$

Thus, we will analyze $g_{n,\ell}(\boldsymbol{s})$ 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 $t = 0$ and any $\boldsymbol{S}_j \in E_\ell$ will be mapped to $T_j > 0$. With this reparameterization, we can write

$$\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh_n} \sum_{j=1}^{n} K\left(\frac{T_j}{h_n}(\boldsymbol{v}_\ell - \boldsymbol{s})\right) I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n)),$$
$$\hat{m}_{n,\ell}(\boldsymbol{s}) \cdot \hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{nh_n} \sum_{j=1}^{n} Y_j K\left(\frac{T_j}{h_n}(\boldsymbol{v}_\ell - \boldsymbol{s})\right) I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n)).$$

To study the limiting behavior when $h_n \to 0$, 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 $\sigma^2_\ell(t) = \mathbb{E}(|U_j|^2|\boldsymbol{S}_j = (1 - t)\boldsymbol{s} + t\boldsymbol{v}_\ell)$, $\sigma^2_\ell(0) = \lim_{t\downarrow 0} \sigma^2_\ell(t)$.

Using the fact that $T_j(\boldsymbol{v}_\ell - \boldsymbol{s}) = \boldsymbol{S}_j - \boldsymbol{s}$,

$$\mathbb{E}(f(T_j(\boldsymbol{v}_\ell - \boldsymbol{s}))I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n))) = \mathbb{E}(f(\boldsymbol{S}_j - s)I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n)))$$
$$= \int_{t>0} f(t)g_\ell(t)dt$$

for any integrable function f. The bias of the denominator can be written as

$$\begin{aligned}\left|\mathbb{E}\hat{g}_{n,\ell}(\boldsymbol{s}) - \frac{1}{2}g_\ell(0)\right| &= \left|\frac{1}{h_n}\int_{t>0} K\left(\frac{t}{h_n}\right)g_\ell(t)dt - g_\ell(0)\int_{z>0}K(z)\right| \\ &= \left|\int_{z>0} K(z)[g_\ell(h_n z) - g_\ell(0)]dz\right| \\ &\leq \int_{z>0} K(z)C_1 h_n z dz \\ &= C_1 h_n \int_{z>0} K(z)z dz = O(h_n).\end{aligned}$$

For stochastic variation, we have

$$\begin{aligned}\mathrm{V}ar\left(\hat{g}_{n,\ell}(\boldsymbol{s})\right) &\leq \frac{1}{nh_n^2}\int_{t>0} K^2\left(\frac{t}{h_n}\right)g_\ell(t)dt \\ &= \frac{1}{nh_n}\int_{z>0} K^2(z)g(h_n z)dz \\ &\leq \frac{1}{nh_n}\int_{z>0} K^2(z)[g(0) + C_1\left|h_n z\right|]dz \\ &= \frac{1}{nh_n}\left[g(0)\int_{z>0}K^2(z)dz + C_1 h_n \int_{z>0} K^2(z)\left|z\right|dz\right] \\ &= O\left(\frac{1}{nh_n}\right).\end{aligned}$$

Thus,

$$\hat{g}_n(\boldsymbol{s}) = \sum_{\ell\in\mathcal{I}}\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{2}\sum_{\ell\in\mathcal{I}} g_\ell(0) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)$$

For the numerator,

$$\begin{aligned}\hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s}) = &\underbrace{\frac{1}{nh_n}\sum_{j=1}^{n} U_j K\left(\frac{t_j}{h_n}\right)I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n))}_{Q_1} \\ &+ \underbrace{\frac{1}{nh_n}\sum_{j=1}^{n} m_{\mathcal{S}}(\boldsymbol{S}_j)K\left(\frac{t_j}{h_n}\right)I(\boldsymbol{S}_j \in E_\ell \cap \mathcal{B}(\boldsymbol{s}, h_n))}_{Q_2},\end{aligned}$$

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

$$\begin{aligned}
\mathrm{V}ar(Q_1) &\leq \frac{1}{nh_n^2}\int_{t>0}\sigma_\ell^2(t)K^2\left(\frac{t}{h_n}\right)g_\ell(t)dt \\
&= \frac{1}{nh_n}\int_{z>0}\sigma_\ell^2(h_nz)K^2(z)g_\ell(h_nz)dz \\
&= \frac{1}{nh_n}\int_{z>0}\sigma_\ell^2(0)K^2(z)g_\ell(0)dz + O\left(\frac{1}{nh_n}\right) = O\left(\frac{1}{nh_n}\right).
\end{aligned}$$

For Q_2, we have

$$\begin{aligned}
\left|\mathbb{E}(Q_2) - \frac{m_\ell(0)g_\ell(0)}{2}\right| &= \left|\frac{1}{h_n}\int_{t>0} m_\ell(t)K(t/h_n)g(t)dt - m_\ell(0)g_\ell(0)\int_{z>0}K(z)dz\right| \\
&= \left|\int_{z>0} m_\ell(h_nz)K(z)g_\ell(h_nz)dz - m_\ell(0)g_\ell(0)\int_{z>0}K(z)dz\right| \\
&\leq \int_{z>0}\left\{\left[m_\ell(0) + C_2h_nz\right]\left[g_\ell(0) + C_1h_nz\right] - m_\ell(0)g_\ell(0)\right\}K(z)dz \\
&\leq [C_1m_\ell(0) + C_2g_\ell(0)]h_n\int_{z>0}K(z)zdz + o(h_n) = O(h_n).
\end{aligned}$$

The varaince of Q_2 is bounded via

$$\begin{aligned}
\mathrm{V}ar(q_2) &\leq \frac{1}{nh_n^2}\int_{t>0} m_\ell^2(t)K^2\left(\frac{t}{h_n}\right)g_\ell(t)dt \\
&= \frac{1}{nh_n}\int_{z>0} m_\ell^2(h_nz)K^2(z)g_\ell(h_nz)dz \\
&\leq \frac{1}{nh_n}\int_{z>0}\left\{m_\ell(0) + C_2\left|h_nz\right|\right\}^2K^2(z)\left\{g_\ell(0) + C_1\left|h_nz\right|\right\}dz \\
&= \frac{1}{nh_n}\left\{m_\ell^2(0)g_\ell(0)\int_{z>0}zK^2(z)dz + O(h_n)\right\} \\
&= O\left(\frac{1}{nh_n}\right)
\end{aligned}$$

Putting the terms Q_1 and Q_2 together, we have

$$\hat{m}_{n,\ell}(\boldsymbol{s})\hat{g}_{n,\ell}(\boldsymbol{s}) = \frac{1}{2}m_\ell(0)g_\ell(0) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right).$$

As a result, we conclude that

$$\begin{aligned}
\hat{m}(\boldsymbol{s}) &= \frac{\sum_{\ell \in \mathcal{I}} \hat{m}_{n,\ell}(\boldsymbol{s}) \hat{g}_{n,\ell}(\boldsymbol{s})}{\sum_{\ell \in \mathcal{I}} \hat{g}_{n,\ell}(\boldsymbol{s})} \\
&= \frac{\frac{1}{2}\sum_{\ell \in \mathcal{I}} m_\ell(0) g_\ell(0) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)}{\frac{1}{2}\sum_{\ell \in \mathcal{I}} g_\ell(0) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right)} \\
&= \frac{\frac{1}{2}\sum_{\ell \in \mathcal{I}} m_\ell(0) g_\ell(0)}{\frac{1}{2}\sum_{\ell \in \mathcal{I}} g_\ell(0)} + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right) \\
&= \frac{\sum_{\ell \in \mathcal{I}} m_\ell(0) g_\ell(0)}{\sum_{\ell \in \mathcal{I}} g_\ell(0)} + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right),
\end{aligned}$$

which completes the proof. □

Kernel Regression: Convergence on Knot with Nonzero Mass

ProofTheorem 4.

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

$$\begin{aligned}
\hat{m}(\boldsymbol{s}) &= \frac{\frac{1}{n}\sum_{j=1}^n Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^n Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h_n))}{\frac{1}{n}\sum_{j=1}^n K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^n K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{V} \cap \mathcal{B}(\boldsymbol{s}, h_n))} \\
&= \frac{\frac{1}{n}\sum_{j=1}^n Y_j K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^n Y_j I(\boldsymbol{S}_j = \boldsymbol{s})}{\frac{1}{n}\sum_{j=1}^n K_{h_n}(\boldsymbol{S}_j, \boldsymbol{s}) I(\boldsymbol{S}_j \in \mathcal{E} \cap \mathcal{B}(\boldsymbol{s}, h_n)) + \frac{1}{n}\sum_{j=1}^n I(\boldsymbol{S}_j = \boldsymbol{s})} \\
&= \frac{\varepsilon_{1,n}(\boldsymbol{s}) + \nu_{1,n}(\boldsymbol{s})}{\varepsilon_{2,n}(\boldsymbol{s}) + \nu_{2,n}(\boldsymbol{s})}.
\end{aligned}$$

We look at each term individually and note that we have the edge components terms identical to the proof of Proposition 5, so

$$\begin{aligned}
\varepsilon_{1,n}(\boldsymbol{s}) &= h_n \left\{ \sum_{\ell \in \mathcal{I}} m_\ell(0) g_\ell(0) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right) \right\} = O(h_n) + O_p\left(\sqrt{\frac{h_n}{n}}\right), \\
\varepsilon_{2,n}(\boldsymbol{s}) &= h_n \left\{ \sum_{\ell \in \mathcal{I}} g_\ell(0) + O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right) \right\} = O(h_n) + O_p\left(\sqrt{\frac{h_n}{n}}\right).
\end{aligned}$$

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 $O_p\left(\sqrt{\frac{h_n}{n}}\right)$, we conclude

$$\begin{aligned}\hat{m}(\boldsymbol{s}) &= \frac{O(h_n) + O_p\left(\sqrt{\frac{h_n}{n}}\right) + m_{\mathcal{S}}(\boldsymbol{s})p(\boldsymbol{s}) + O_p\left(\sqrt{\frac{1}{n}}\right)}{O(h_n) + O_p\left(\sqrt{\frac{h_n}{n}}\right) + p(\boldsymbol{s}) + O_p\left(\sqrt{\frac{1}{n}}\right)} \\ &= \frac{O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right)}{O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right) + p(\boldsymbol{s})} + \frac{m_{\mathcal{S}}(\boldsymbol{s})p(\boldsymbol{s})}{O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right) + p(\boldsymbol{s})} \\ &= \frac{O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right)}{p(\boldsymbol{s})} + O\left[\left(\frac{O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right)}{p(\boldsymbol{s})}\right)^2\right] \\ &\quad + m_{\mathcal{S}}(\boldsymbol{s})p(\boldsymbol{s})\left\{\frac{1}{p(\boldsymbol{s})} + \frac{O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right)}{p(\boldsymbol{s})^2}\right\} \\ &= m_{\mathcal{S}}(\boldsymbol{s}) + O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right),\end{aligned}$$

which completes the proof.

□

K Additional Simulation Results

In this section, we focus on the impact on the final regression performance when cutting the skeleton into different numbers of disjoint components. We use the same simulated datasets

as in Section 3.6: Yinyang data, Noisy Yinyang data, and SwissRoll data. We mainly follow the same analysis procedure, use 5-fold cross-validation SSE to assess the regression result on each dataset, and, for each simulated dataset, repeat the procedure for 100 times as the dataset is randomly generated. The only 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 described in Section 3.3.1. we then fit and assess the skeleton-based regression methods on the differently-cut skeletons.

Vary Skeleton Cuts for Yinyang Data

In this section, we use the Yinyang data (Section 3.6.1) to study the influence of cutting the skeleton into different numbers of disjoint components on the performance of skeleton-based methods. We randomly generate the Yinyang 1000-dimensional data for 100 times and use 5-fold cross-validation to calculate the sum of squared errors (SSE) on each dataset. We fit the skeleton-based methods in the same way as in Section 3.6 except that we fix the number of knots to be 38 and cut the initial graph into different numbers of disjoint components (1 to 25) when constructing the skeleton. The medium 5-fold cross-validation SEEs across the 100 datasets with different numbers of disjoint components is plotted in Figure 34.

We see that the S-Lspline method is sensitive to the change in skeleton structure, and, in the case of Yinyang data, since there are 5 true disjoint structures in the covariate space, a cut of 5 leads to the best regression result. By the construction of the S-Lspline regressor, an edge between two intrinsically different structures will let the estimation on one structure to take unrelated information from the other structure and hence deteriorate the regression

performance. For future research, incorporating edge weights into the S-Lspline regressor may alleviate the interference between different structures. The S-Kernel regressor also achieves the best performance when the skeleton is segmented into 5 disjoint components. The skeleton-based kernel regression method demonstrates larger changes in performance corresponding to the changes in skeleton segmentation when the bandwidth is large. This is explicable as larger bandwidth allows more information from large distances, which are more likely to be modified become non-informative as the segmentation changes. The S-kNN regressor, differently, has 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 approach that the regression result is accurate as long as the local neighborhood is accurately identified.

Vary Skeleton Cuts for Noisy Yinyang Data

We then test the skeleton-based regression methods on the Noisy Yinyang data (Section 3.6.2) when the skeletons are constructed with different numbers of disjoint components. Similarly, we randomly generate the Noisy Yinyang 1000-dimensional data for 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 medium 5-fold cross-validation SEEs across the 100 datasets with different numbers of disjoint components is plotted in Figure 35.

Vary Skeleton Cuts for SwissRoll data

In this section, we test the skeleton-based methods on SwissRoll data (Section 3.6.3) with skeletons cut into different numbers of disjoint components. Similarly, we randomly generate the SwissRoll 1000-dimensional data for 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 70 and construct skeletons with different numbers of disjoint components. The medium 5-fold cross-validation SEEs across the 100 datasets with different numbers of disjoint components is plotted in Figure 36.

We observe that the S-Lspline regressor is sensitive to the change in skeleton structure, and the skeleton graph that is connected leads to the best regression result. This makes senses as intrinsically the covariates lay around one connected manifold. The S-Kernel regressor also has the best performance when the skeleton is constructed as one connected component. After the initial increase in SSE with respect to the increase in the number of disjoint components, the SSE by S-Kernel regressor stays relatively stable as more components are separated out. The S-kNN regressor also has best regression performance when the skeleton is left as a fully connected graph. Generally, the SSE by the S-kNN regressor increases with the number of disjoint components, but for small number of neighbors, there can be a decrease in SSE when the skeleton is cut into more disjoint components. One explanation is that, as the response function has discontinuous changes, segmenting the covariate space to be more fragmented can help with estimation in the region with discontinuous changes in response.

L 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 3.7.1. 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.

Cup Images Data

We start with a sequence of images of a cup, with some examples in Figure 38. The best SSE from each method is listed in Table 2 along with the corresponding parameters. We see that the S-Lspline method gives the best performance in terms of SSE, while the usual kNN regressor also performs well on this data.

Sauce Box Image Data

We look at another sequence of images taken around a sauce box, with some examples in Figure 40. The best SSE from each method is listed in Table 3 along with the corresponding parameters. We see that in this case the usual kNN regressor give the best performance in terms of SSE, while the S-Lspline method gives satisfactory results. The good performance of kNN regressor in this case 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 proposed skeleton regression method shows stable performance across differ-

ent object images and, by explicitly modeling the latent manifold structure, can give more structured model on the response.

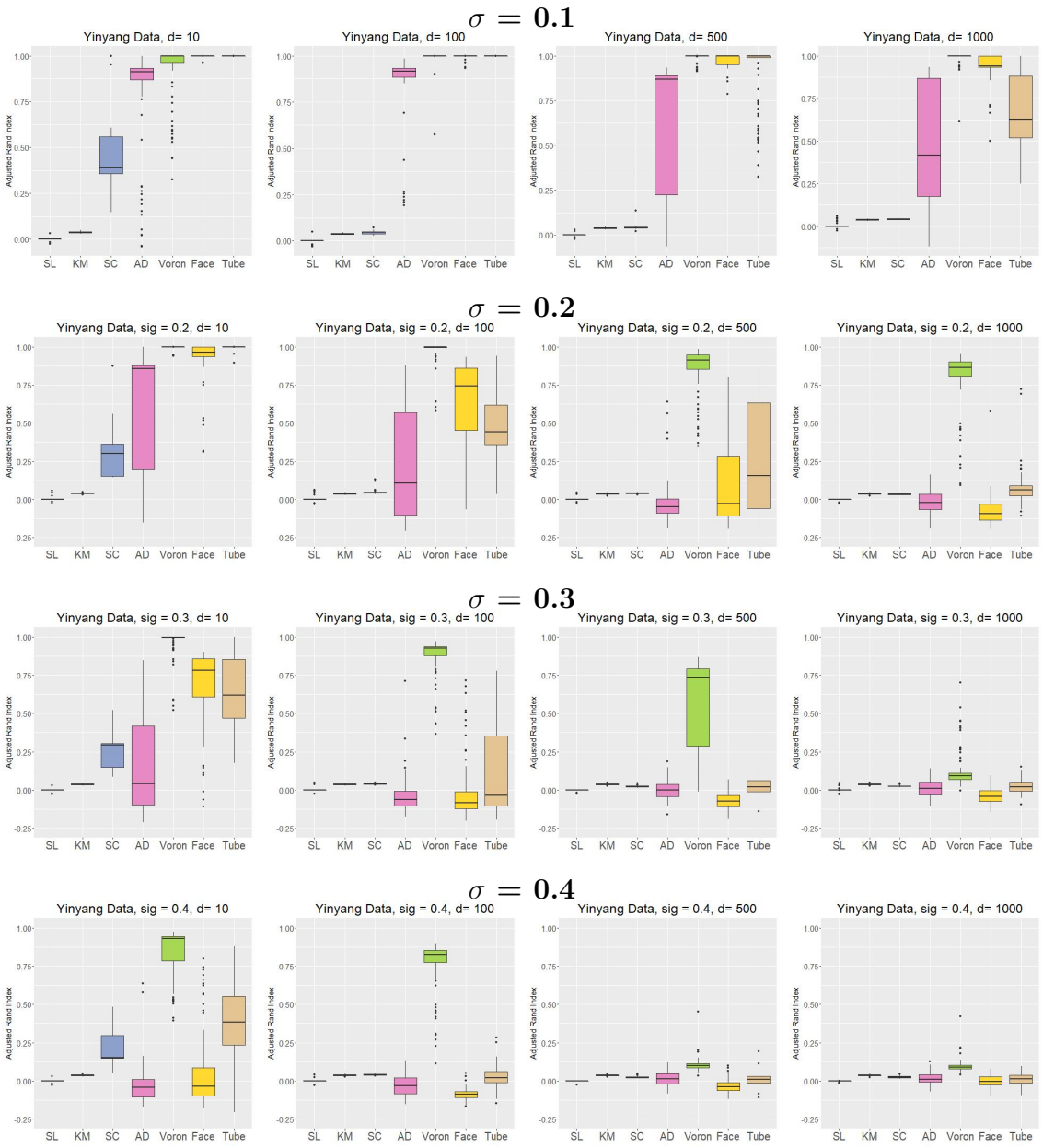


Figure 25: Adjusted Rand index performance of clustering methods on Yinyang data with different standard deviation for added dimensions.

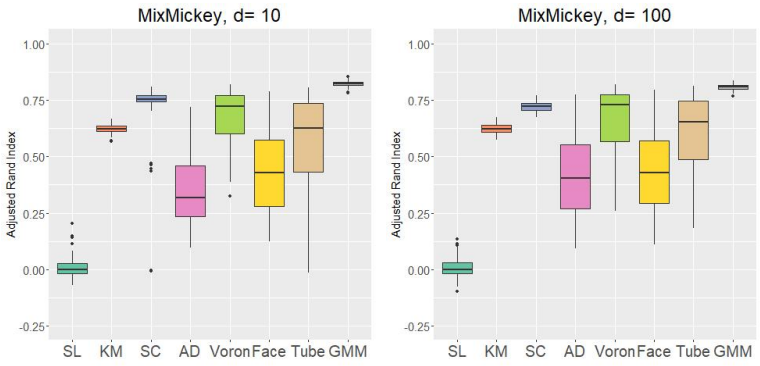
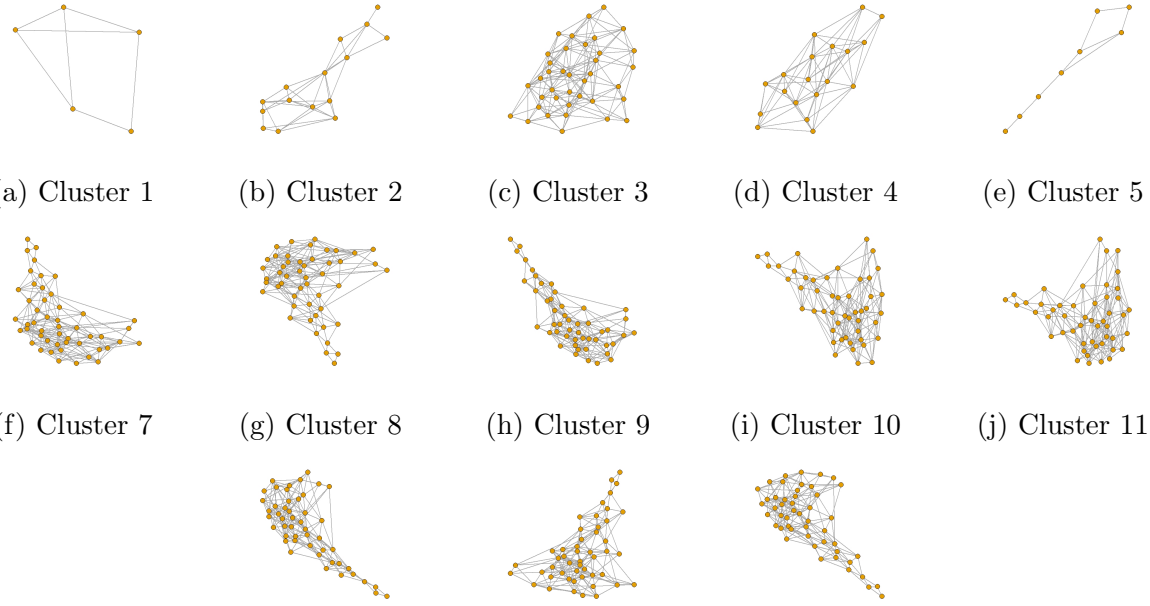


Figure 26: Comparison of clustering methods on Mix Mickey data $d = 10, 100$ with GMM included.



(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 27: Skeleton structures of the clusters identified for the GvHD dataset in Section 2.6

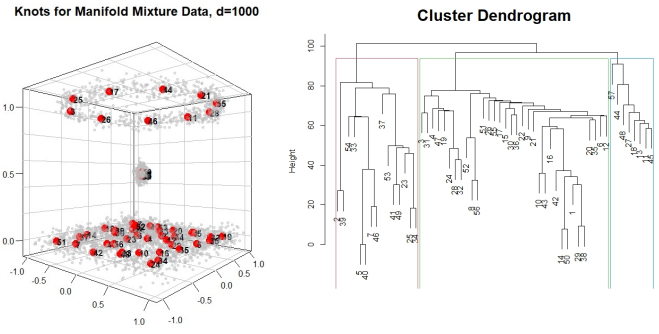


Figure 28: Results on Manifold Mixture data with dimension 100.

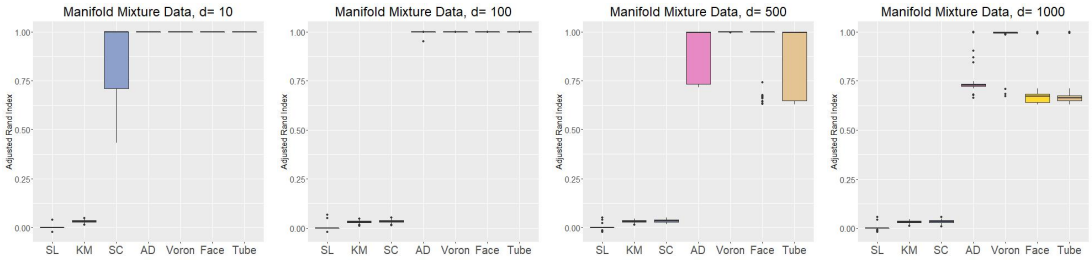


Figure 29: Comparison of adjusted Rand index using different similarity measures on Manifold Mixture data with dimensions 10, 100, 500, 1000.

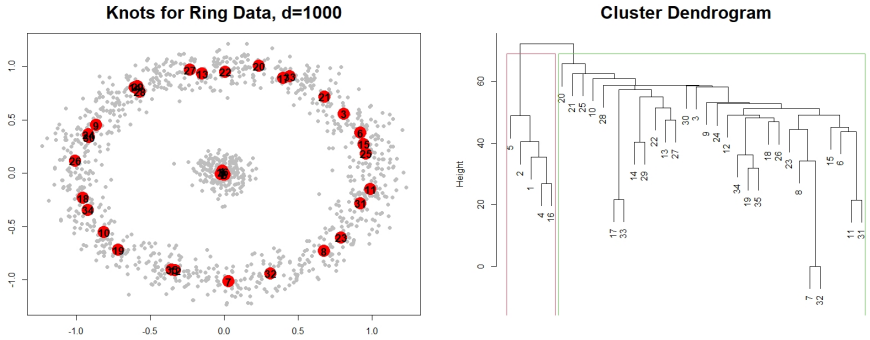


Figure 30: Results on Ring data with dimension 1000.

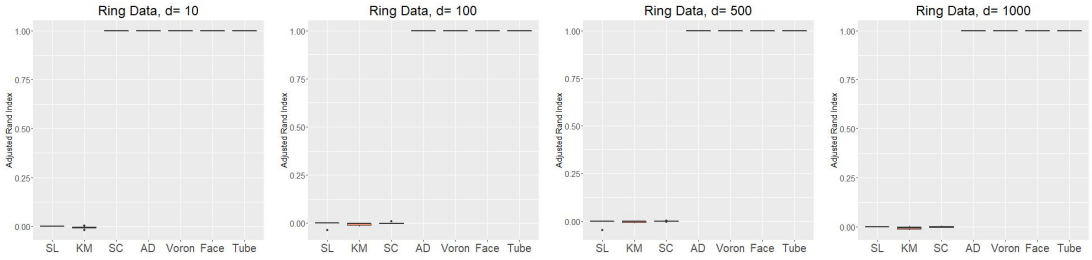


Figure 31: Comparison of the rand index using different similarity measures on Ring data with dimensions 10, 100, 500, 1000. Medium of 100 repetitions.

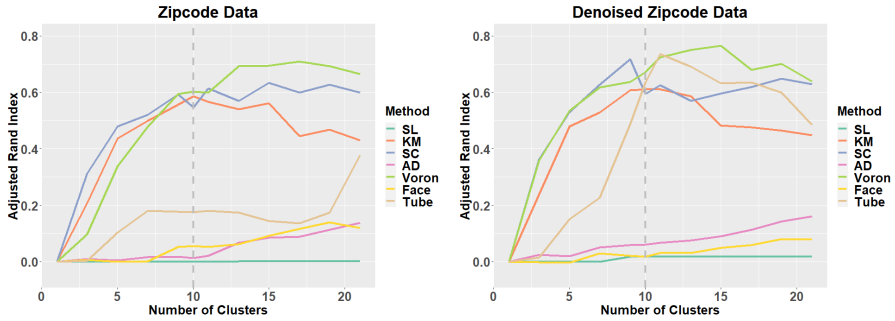


Figure 32: Comparison of different similarity measures on all Zipcode Data.

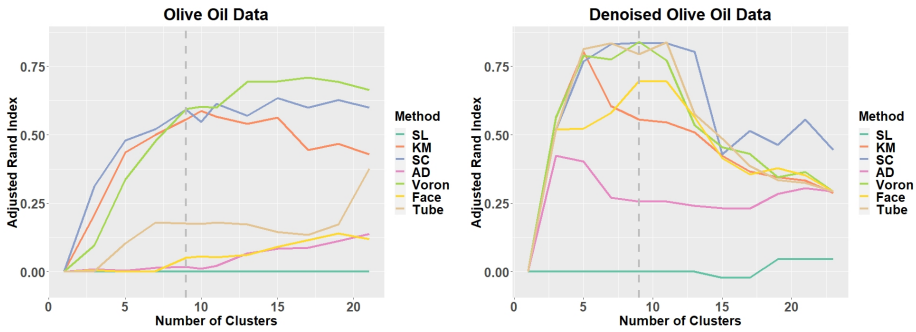


Figure 33: The clustering performance under different number of final clusters of the Olive oil data.

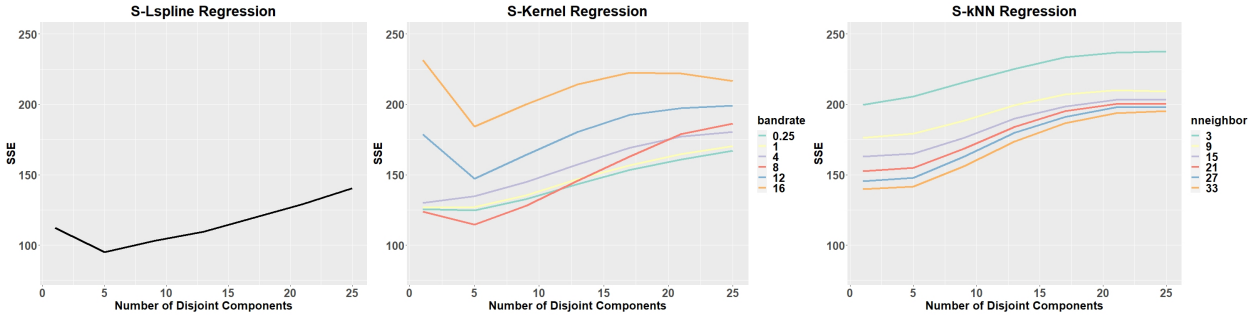
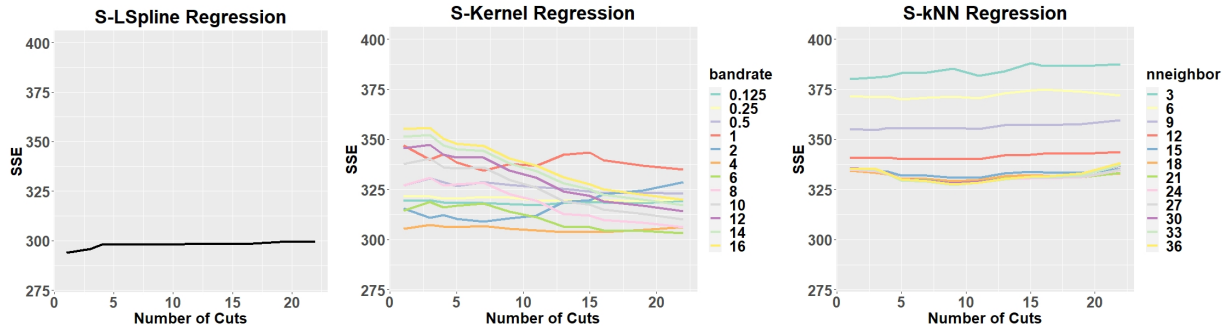


Figure 34: Yinyang $d = 1000$ data skeleton regression results with the number of knots fixed as 38 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.



(a) S-Lspline (b) S-Kernel (c) S-kNN

Figure 35: Noisy Yinyang Regression fitted points $d = 1000$ with varying number of cuts results, with number of knots fixed as 99.

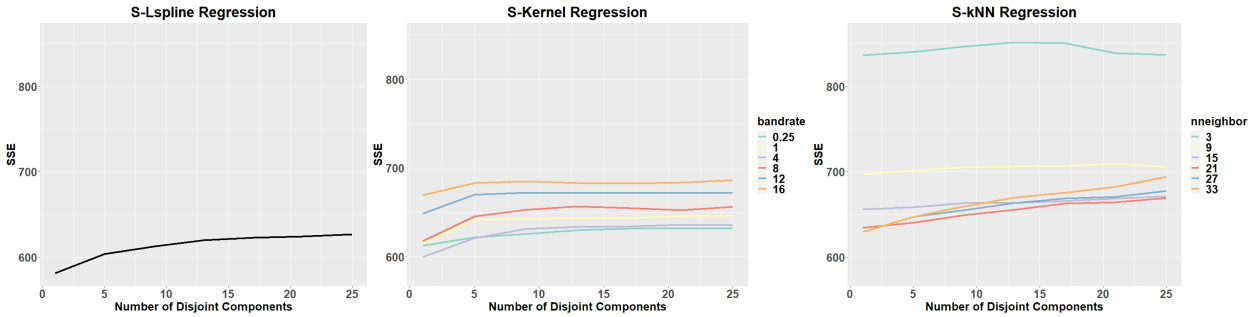


Figure 36: SwissRoll $d = 1000$ data skeleton regression results with the number of knots fixed as 70 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

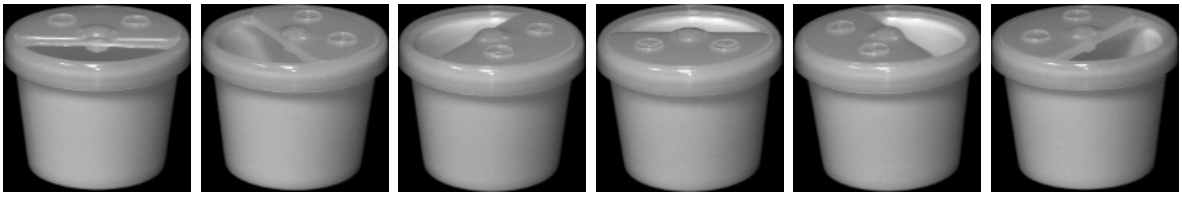


Figure 38: A part of the cup images from the COIL-20 processed dataset. Each image is of size 128 pixels.

Method	SSE	Parameter
kNN	1147.2	neighbor=3
Ridge	-	-
Lasso	-	-
SpecSeries	-	-
S-Kernel	2561.5	bandwidth = $4r_{hns}$
S-kNN	4730.6	nenighbor = 3
S-Lspline	1073.4	-

Table 2: Regression results on cup images data from COIL-20. The best SSE from each method is listed with the corresponding parameters used.

Figure 40: A part of the sauce images from the COIL-20 processed dataset. Each image is of size 128 pixels.

Method	SSE	Parameter
kNN	955.6	neighbor=3
Ridge	-	-
Lasso	-	-
SpecSeries	-	-
S-Kernel	2998.8	bandwidth = $4r_{hns}$
S-kNN	5285.4	enighbor = 6
S-Lspline	1220.1	-

Table 3: Regression results on sauce images data from COIL-20. The best SSE from each method is listed with the corresponding parameters used.