Graphs for Statistical Learning and Modeling

Jerry Wei [1] Yen-Chi Chen [1] Tyler McCormick [1,2]

[1]Department of Statistics, University of Washington

[2]Department of Sociology, University of Washington

Two Perspectives on Graphs

- Graph to Encode Geometric Information
  - Manifold Learning:
    Laplacian Eigenmaps*, Diffusion Maps†
    Discrete graph Laplacian converges to the Laplace-Beltrami operator‡
  - Graph Neural Network:
    EdgeConv§, Differentiable Graph Module (DGM)¶
- Graph to Model Networks
  - social network, financial network, contact network, etc.

*Belkin and Niyogi (2003)
†Coifman and Lafon (2006)
‡Belkin and Niyogi (2008); Berry and Harlim (2014); Berry and Sauer (2019)
§Wang et al. (2019)
¶Kazi et al. (2022)

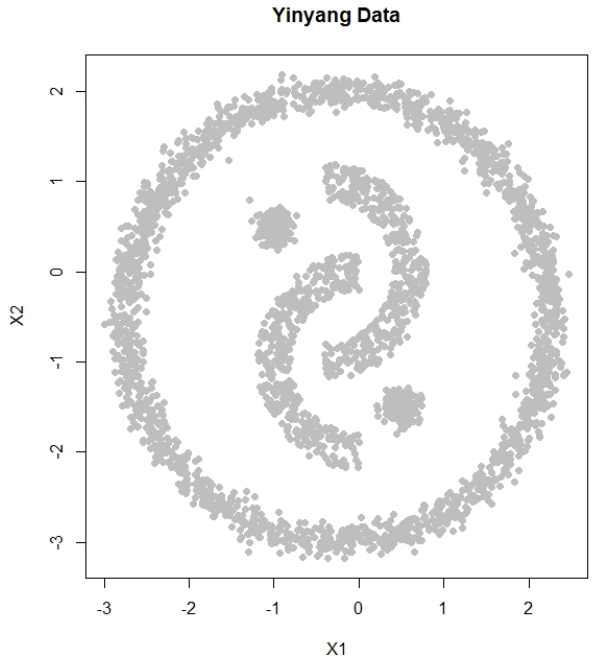
Outline

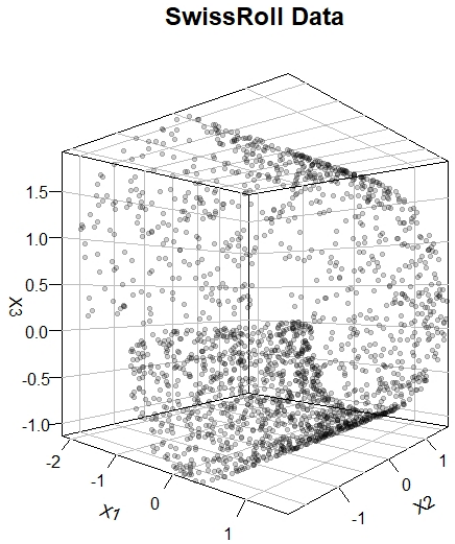
- Graph-Based Approach for Statistical Learning
  - Project One: Skeleton Clustering
  - Project Two: Skeleton Regression
- Network Graph in Epidemic Modeling
  - Project Three: Assessing Epidemic Models under Missingness in Contact Network

Graph-Based Approach for Statistical Learning

Introduction

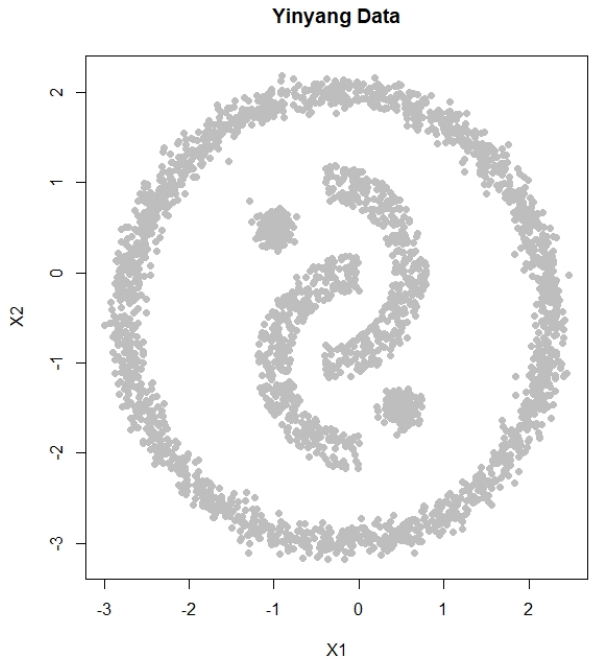
Many data nowadays have low-dimensional geometric structures while the covariates are embedded inside some large-dimensional space.

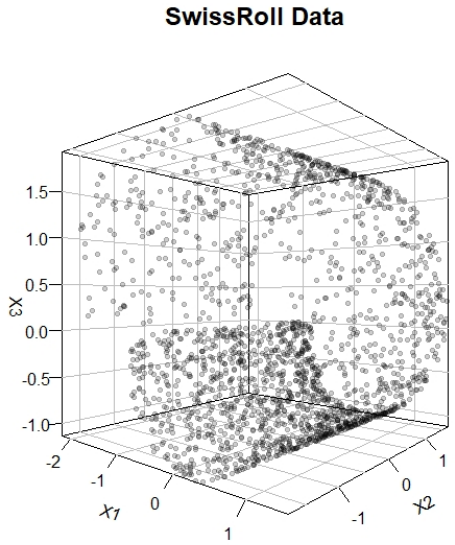




Introduction

Many data nowadays have low-dimensional geometric structures while the covariates are embedded inside some large-dimensional space.





Understanding such geometric structures of the data can facilitate various data analysis tasks.

Our line of work propose to use a graph, called *Skeleton*, to summarize the geometric information and assist various statistical learning tasks.

Project One:
Skeleton Clustering: Dimension-Free Density-Aided Clustering

Jerry Wei and Yen-Chi Chen

Density-Based Clustering

Problem: Cluster large-dimensional data with unbalanced groups and complex cluster shapes.

Density-based Idea: a cluster in a data space is a contiguous region of high point density

Examples: Mode Clustering* , Level-Set Clustering†, DBSCAN‡, Cluster Tree§

*Chen et al. (2016)
†Cuevas et al. (2000, 2001)
‡Ester et al. (1996)
§Stuetzle and Nugent (2010); Chaudhuri and Dasgupta (2010)
¶Scott (2015); Wasserman (2006)

Density-Based Clustering

Problem: Cluster large-dimensional data with unbalanced groups and complex cluster shapes.

Density-based Idea: a cluster in a data space is a contiguous region of high point density

Examples: Mode Clustering* , Level-Set Clustering†, DBSCAN‡, Cluster Tree§

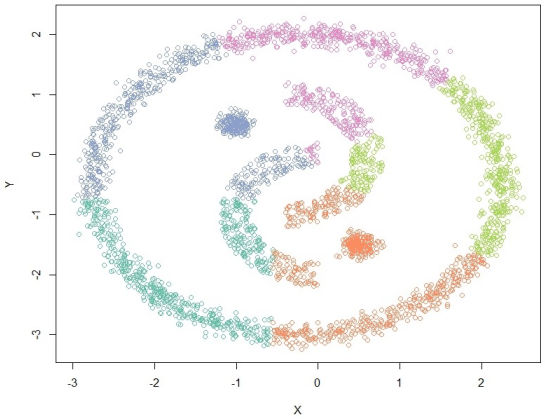
Advantages:

- capable of finding clusters with irregular shapes
- nice interpretation based on the underlying PDF
- can view the clustering problem as an estimation problem

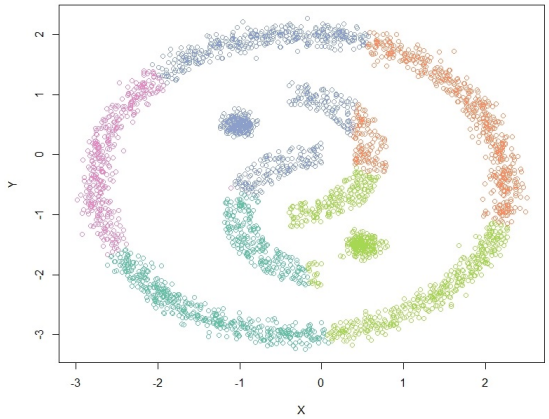
Limitation: the convergence rate of a density estimator is $O_P\left(n^{-\frac{2}{4+d}}\right)$ under usual smoothness conditions¶

*Chen et al. (2016)
†Cuevas et al. (2000, 2001)
‡Ester et al. (1996)
§Stuetzle and Nugent (2010); Chaudhuri and Dasgupta (2010)
¶Scott (2015); Wasserman (2006)

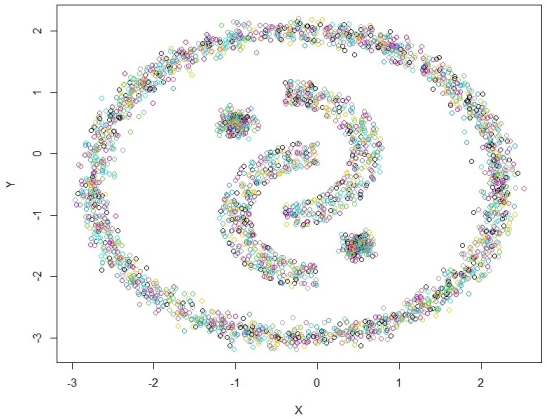
Clustering High-dimensional Data



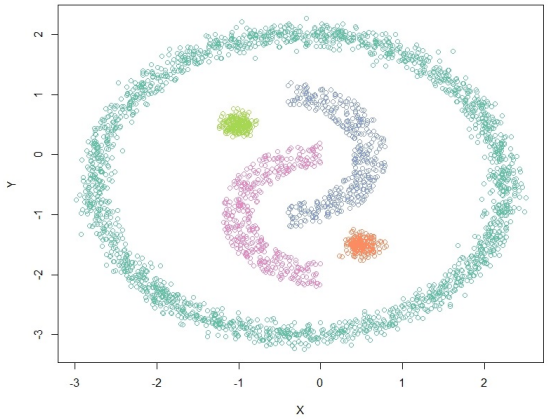
(a) K-means Clustering



(b) Spectral Clustering



(c) Mean Shift Clustering



(d) Skeleton Clustering

Figure: Example Yinyang Data, random $N(0, .01)$ variables included to make a total dimension of 200.

Related Works

- Merging k-means clusters*
- Merging prototypes from model-based clustering to overcome the limitations on parametric assumptions†

Our Contributions

- We introduce a novel clustering framework based on graphs that can deal with complex geometric structures.
- We propose multiple density-based similarity measures that scale well with dimensions.
- We use simulation to show the reliability of our method in agnostic scenarios.
- We show that our method can lead to meaningful clusters in real data.

*Maitra (2009); Peterson et al. (2018)

†Hennig (2010); Baudry et al. (2010); Scrucca (2016); Chacón (2019)

Skeleton Clustering Framework

Let our training data $\mathbb{X} = \{X_1, \ldots, X_n\}$ be IID samples 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 number of clusters.

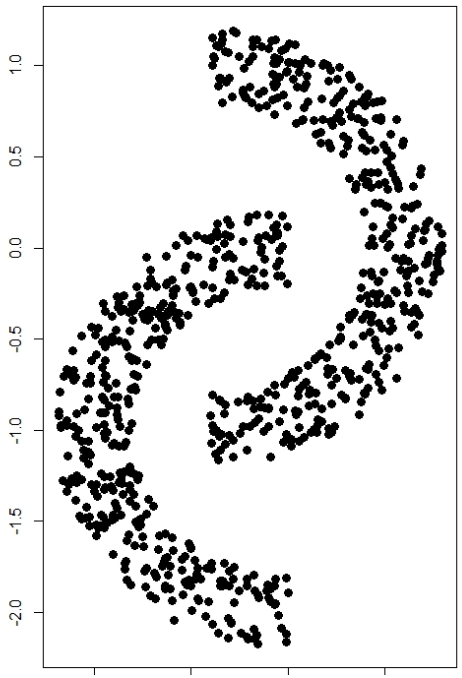
Algorithm Skeleton Clustering

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

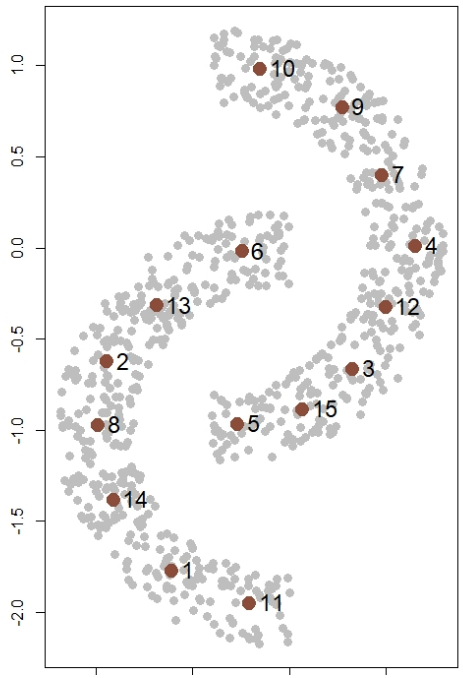
1. Knot construction.
2. Edge construction.
3. Edge weights construction.
4. Knots segmentation.
5. Assignment of labels.

Knots Construction

- Some knots are constructed to give a concise representation of the data structure.
- In practice we use k-Means to choose $k = [\sqrt{n}]$ knots, where n is the number of samples.



(a) Data



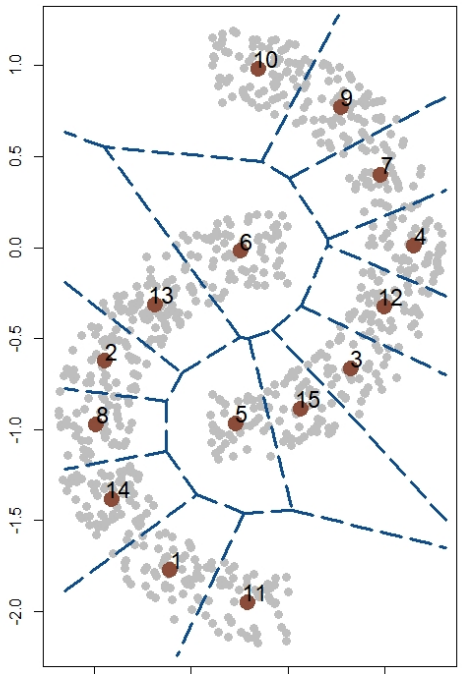
(b) Knots

Edge Construction, Voronoi Cells

Idea: only connect nearby knots.

The Voronoi cell*, $\mathbb{C}_j$, associated with knot c_j is the set of all points in $\mathcal{X}$ whose distance to c_j is the smallest compared to other knots that

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



*Voronoi (1908)

Edge Construction, Delaunay Triangulation

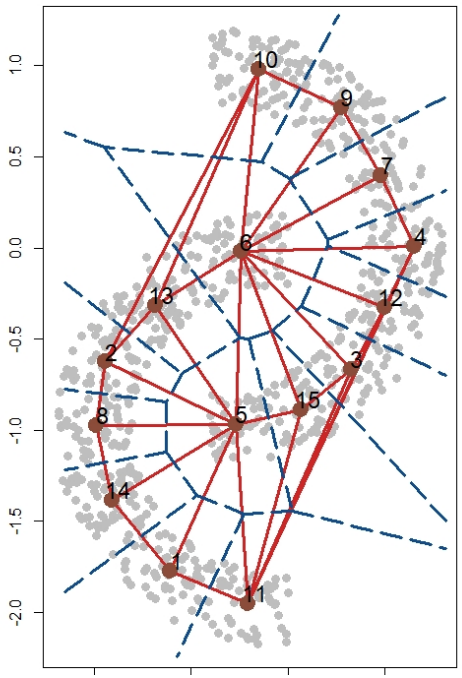
- An edge between (c_i, c_j) is added if $\bar{\mathbb{C}}_i \cap \bar{\mathbb{C}}_j \neq \emptyset$.
- Resulting graph is the Delaunay triangulation $DT(\mathcal{C})^*$ of knots $c_1, \cdots, c_k$
- The computational complexity of the exact Delaunay triangulation algorithm has an exponential dependence on the ambient dimension $d^\dagger$.

*Delaunay (1934)

†Amenta et al. (2007); Chazelle (1993)

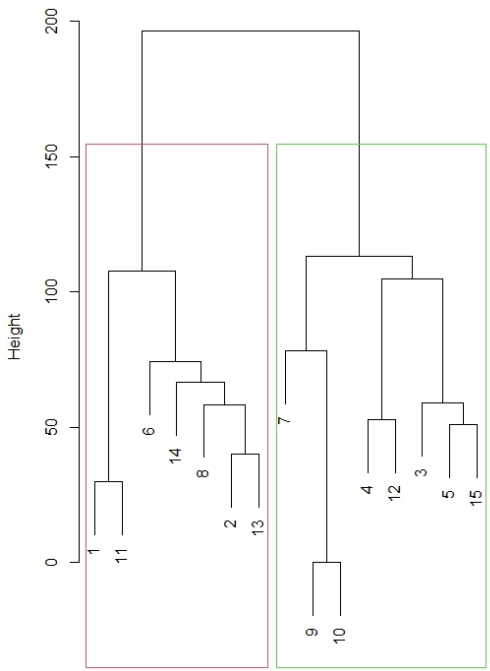
Approximate Delaunay Triangulation

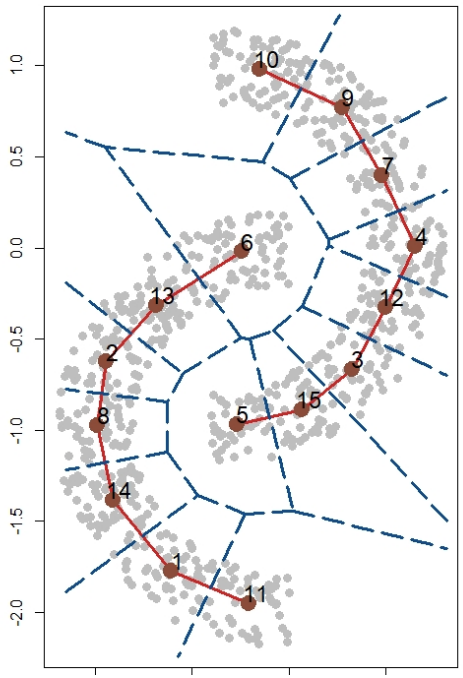
- We approximate the exact Delaunay Triangulation with $\hat{DT}(\mathcal{C})$
- If the Voronoi cells of two knots c_i, c_j share a boundary, there is a non-empty region of points whose 2-nearest knots are c_i, c_j
- Query the two nearest knots for each data point and have an edge between c_i, c_j if there is at least one data point whose two nearest knots are c_i, c_j.



Skeleton Segmentation

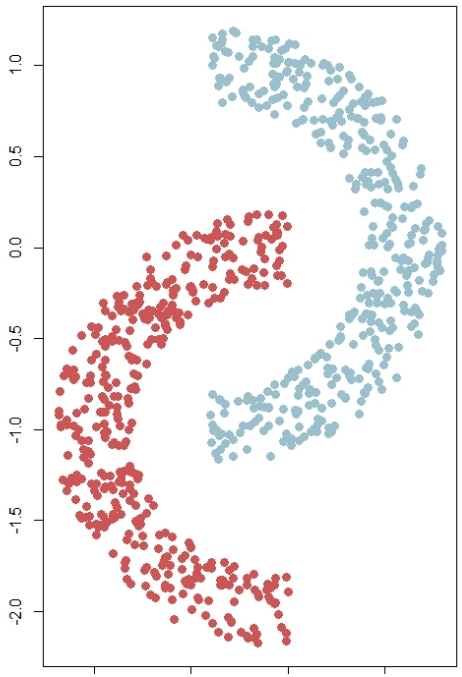
- Density-based weights are assigned to the edges (discussed later).
- Use traditional clustering/segmentation methods such as the hierarchical clustering to segment the learnt skeleton structure.





Label Assignment

- Assign the individual labels according to the segmented skeleton
- In practice we assign the labels the same as the nearest knot.



Skeleton Clustering Framework

Algorithm Skeleton Clustering

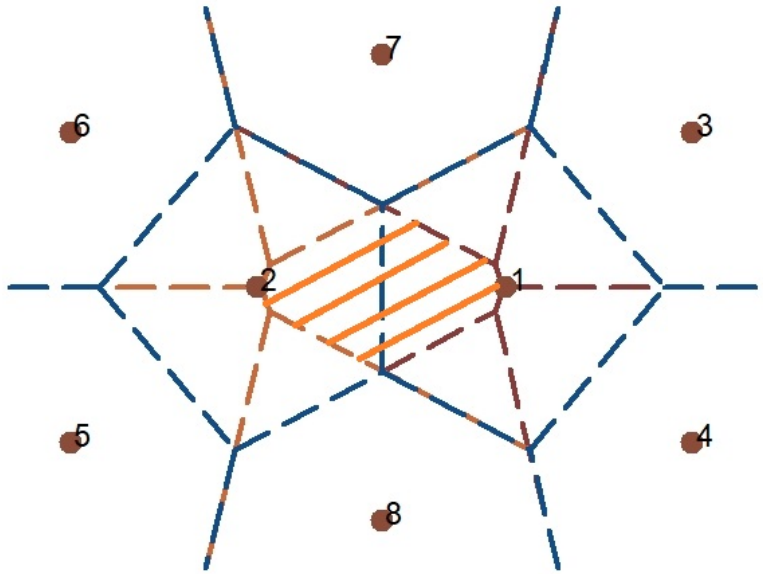
Input: Observations $X_1, \cdots, X_n$, final number of clusters S.

1. **Knot construction.** Perform k-means clustering with a large number of k; the centers are the knots.
2. **Edge construction.** Apply an approximate Delaunay triangulation to the knots.
3. **Edge weights construction (coming next).** Add weights to each edge with density-based similarity measures.
4. **Knots segmentation.** Use linkage criterion to segment knots based on the edge weights into S groups.
5. **Assignment of labels.** Assign cluster labels to each observation based on which knot-group of the nearest knot.

Edge Weight: Voronoi Density

- Define the 2-NN region as
  $A_{j\ell} \equiv \{x \in \mathcal{X} : d(x, c_i) > max\{d(x, c_j), d(x, c_\ell)\}, \forall i \neq j, \ell\}$.
- The *Voronoi density (VD)* is defined as

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

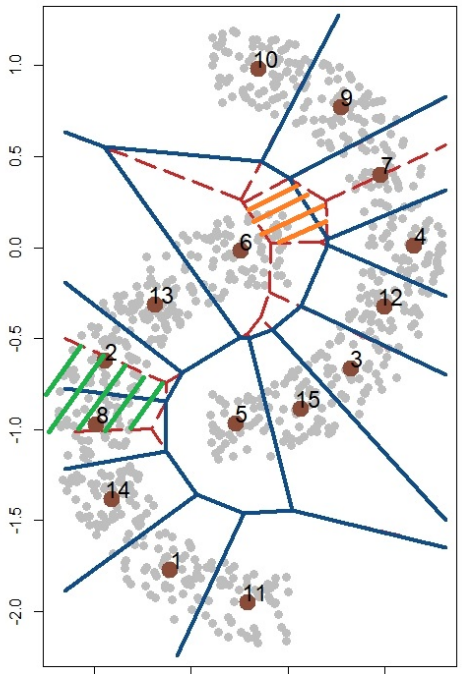


Edge Weight: Voronoi Density Estimation

- Let $\hat{P}_n(A_{j\ell}) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A_{j\ell})$ and our estimator is

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

- Essentially counting points in the 2-NN region, which can be computed fast by k-d tree algorithm*



*Bentley (1975)

Voronoi Density Consistency

(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.

Theorem (Voronoi Density Convergence)

Assume (B1), let n be the sample size and k the number of knots. Then for any pair $j \neq \ell$ that shares an edge, the similarity measure based on the Voronoi density satisfies

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

$$\max_{j,\ell} \left| \frac{\hat{S}_{j\ell}^{VD}}{S_{j\ell}^{VD}} - 1 \right| = O_p\left(\sqrt{\frac{k}{n}} \log k\right), \tag{2}$$

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

Performance Guarantee for Voronoi Density

Let the true partition of the knots be $\mathcal{L}^* = \{\mathcal{L}^*_\ell\}_{\ell=1,\ldots,L}$, with L given. Let the partition based on estimated edge similarities be $\hat{\mathcal{L}} = \{\hat{\mathcal{L}}_\ell\}_{\ell=1,\ldots,L}$. 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 $\varepsilon > 0$.

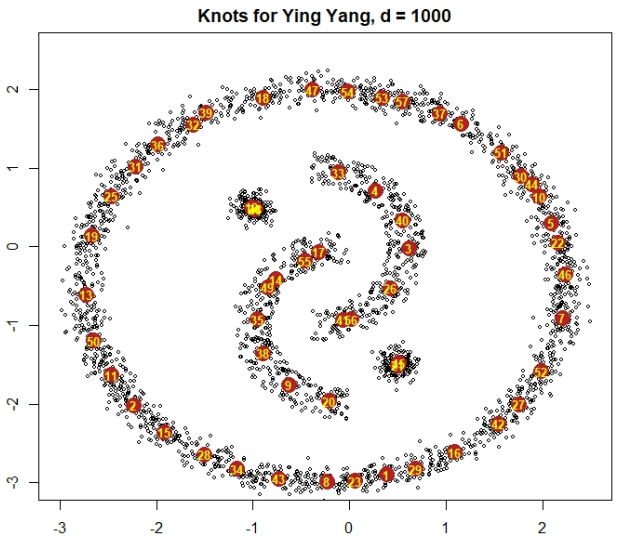
Theorem (Adjusted Rand Index Guarantee)

Assume (B1) and (P1). Let k be the number of knots and let $p_{min} = \min_{j,\ell} \mathbb{P}(A_{j\ell})$, then

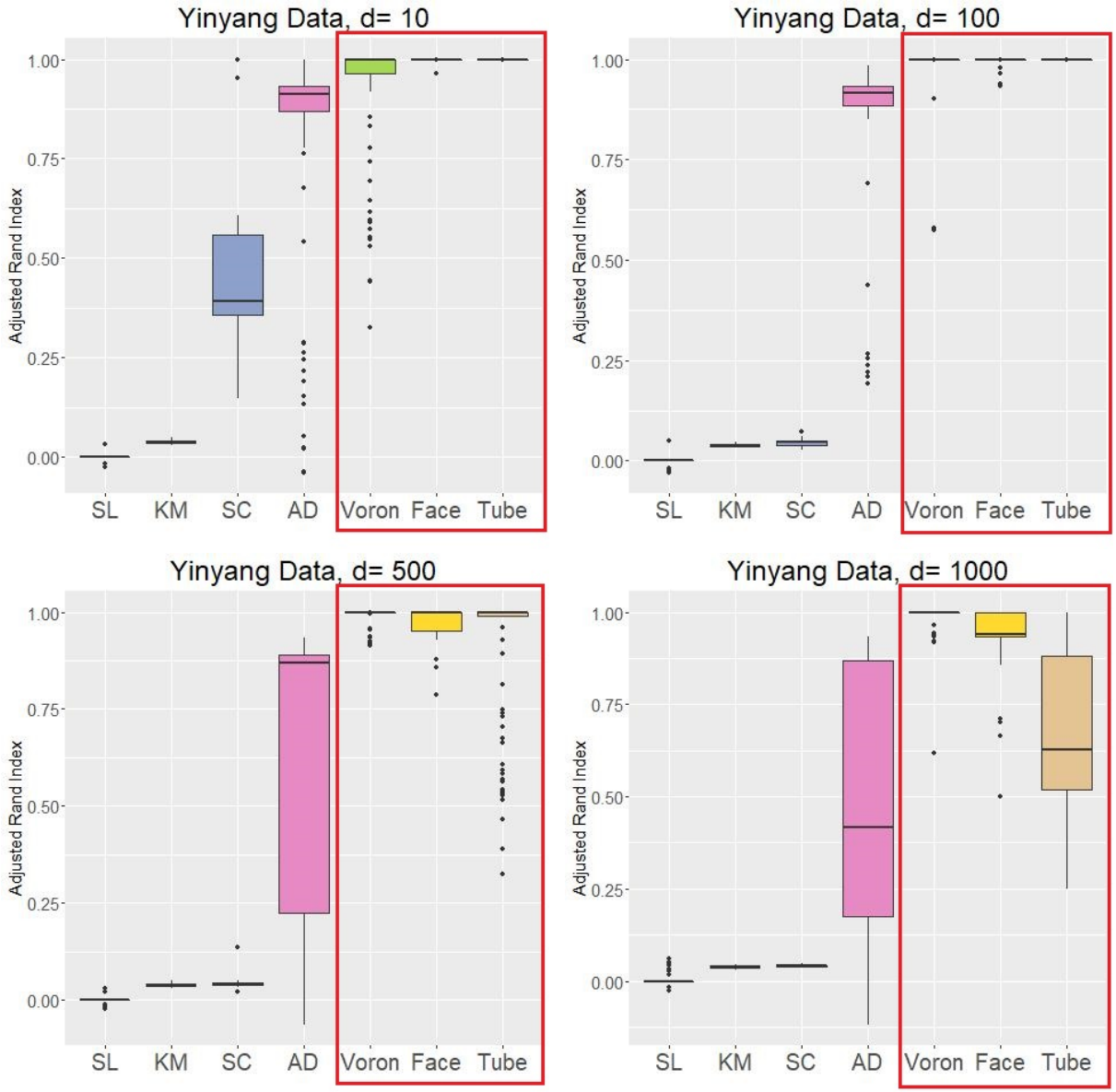
$$\mathbb{P}\left\{ ARI(\mathcal{L}^*, \hat{\mathcal{L}}) < 1 \right\} \leq k(k-1) \exp\left(- \frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1-p_{min}) + \frac{1}{3}\varepsilon} \right) \tag{3}$$

Simulation: Yinyang Data

- Data containing 5 components with different shapes in 2 dimensions (n = 3200).
- Additional variables from Gaussian noise $N(0, .01)$. Increase the dimensions of noise variables so that the total dimensions are $d = 10, 100, 500, 1000$.
- *Classical methods*: direct single-linkage hierarchical clustering (**SL**), direct k-means clustering (**KM**), spectral clustering (**SC**)
- *Skeleton-based Methods*: with Voronoi density (**Voron**), with Face density (**Face**), with Tube density (**Tube**), with average distance density (**AD**).

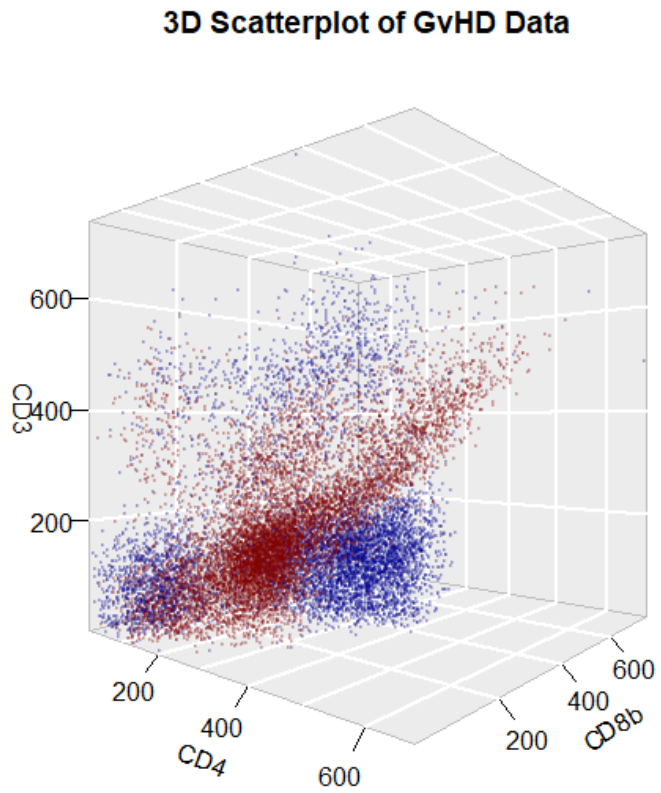


Yinyang Data Clustering Performance



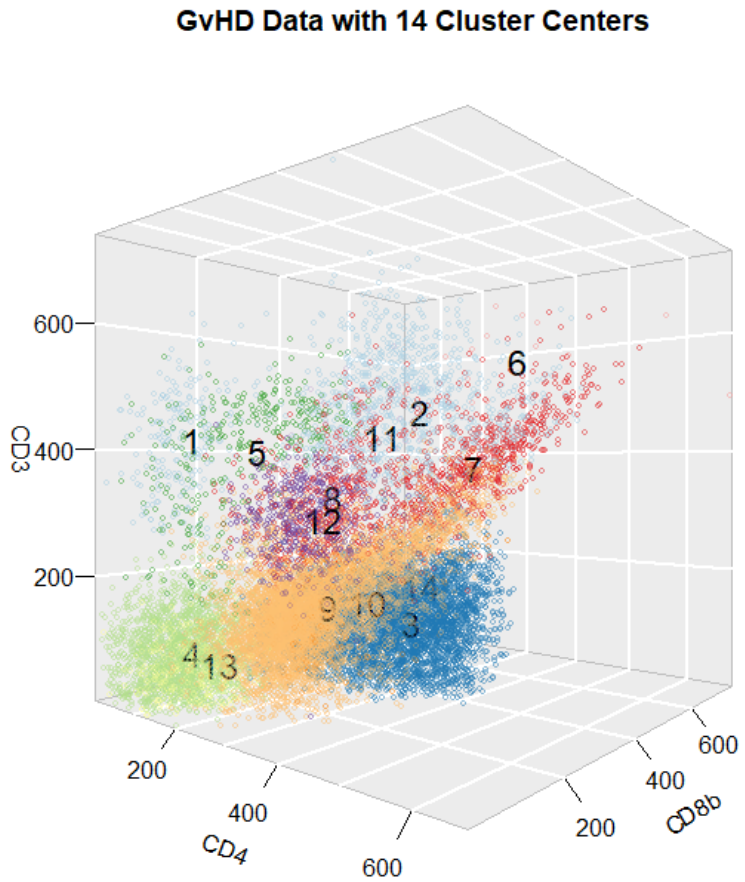
GvHD Data

- Flow cytometry data from Brinkman et al. (2007)
- 9083 observations from a patient with graft-versus-host disease (GvHD) and 6809 observations from a control patient.
- 4 biomarker variables, CD4, CD8β, CD3, and CD8.
- Previous studies* identified high values of CD3, CD4, CD8β cell sub-populations in the GvHD positive sample.

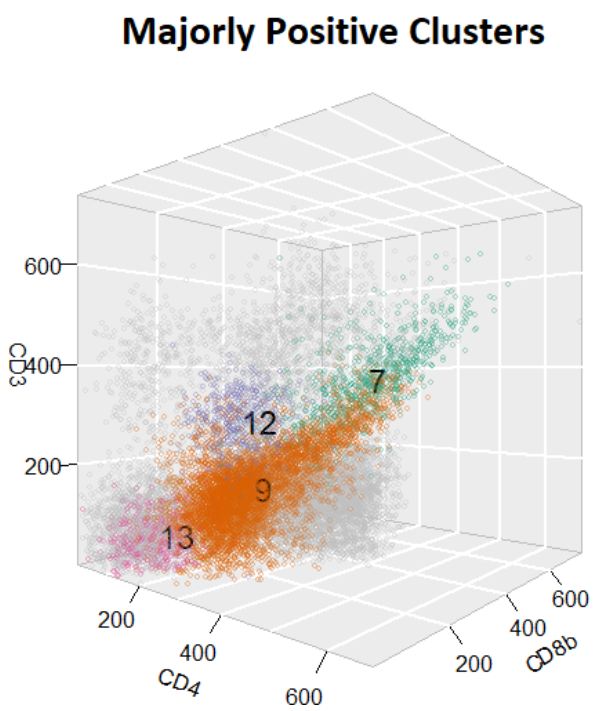


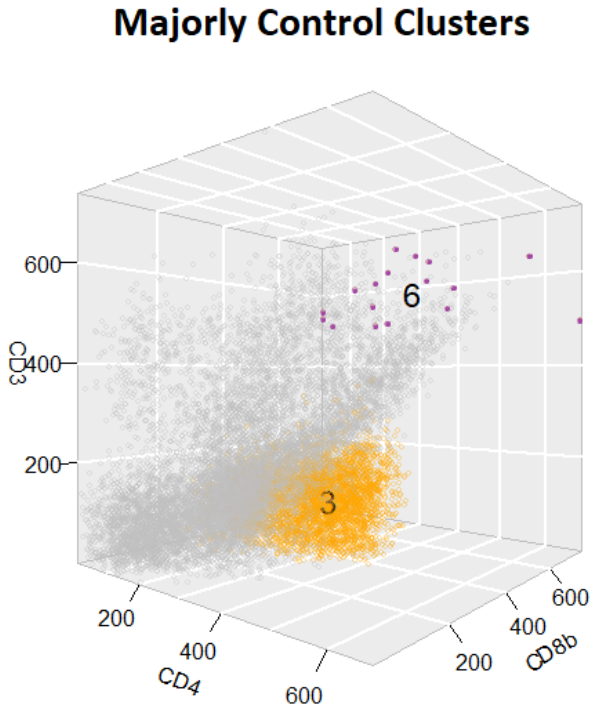
*Brinkman et al. (2007); Baudry et al. (2010)

GvHD Data


GvHD Data with 14 Cluster Centers
600
400
200
CD3
200
400
600
CD4
200
400
600
CD8b
1
5
11
2
6
7
8
12
9
10
3
4
13

GvHD Data

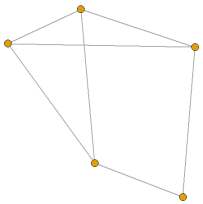




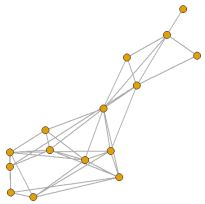
Cluster	1	2	3	4	5	6	7
Size	202	948	3881	1859	338	17	812
Prop	.458	.343	.008	.296	.341	.000	.934
p-value	.30	7×10^{-20}	0	3×10^{-63}	4×10^{-8}	1×10^{-4}	6×10^{-103}

Cluster	8	9	10	11	12	13	14
Size	468	6191	251	37	478	402	8
Prop	.690	.888	.673	.669	.794	.841	.310
p-value	2×10^{-13}	0	1×10^{-6}	.09	6×10^{-30}	3×10^{-33}	.52

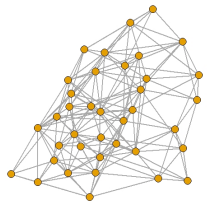
GvHD Data



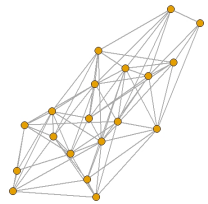
(a) Cluster 1



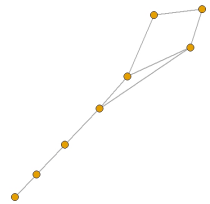
(b) Cluster 2



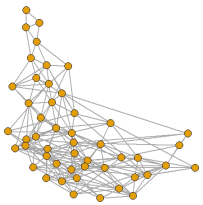
(c) Cluster 3



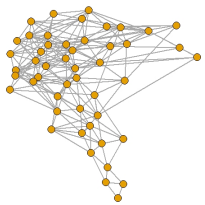
(d) Cluster 4



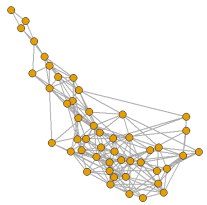
(e) Cluster 5



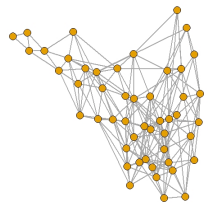
(f) Cluster 7



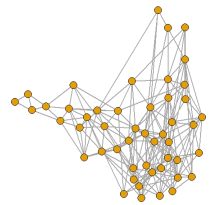
(g) Cluster 8



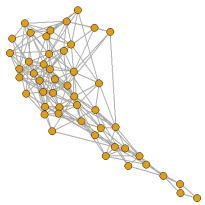
(h) Cluster 9



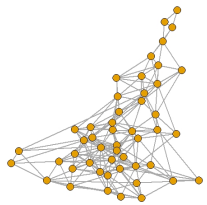
(i) Cluster 10



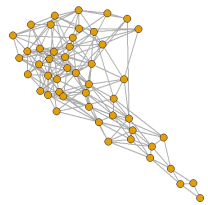
(j) Cluster 11



(k) Cluster 12



(l) Cluster 13



(m) Cluster 14

Figure: Skeleton structures of the clusters identified for the GvHD dataset

Conclusion

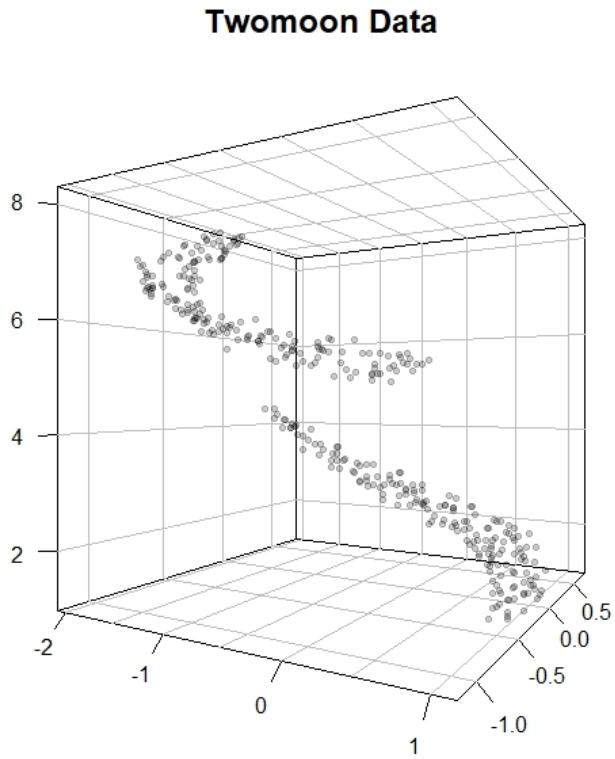
- Use skeleton graphs to cluster large-dimension data with complex cluster shapes.
- Bypass the curse of dimensionality by using surrogate density such as Voronoi density

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

Jerry Wei and Yen-Chi Chen

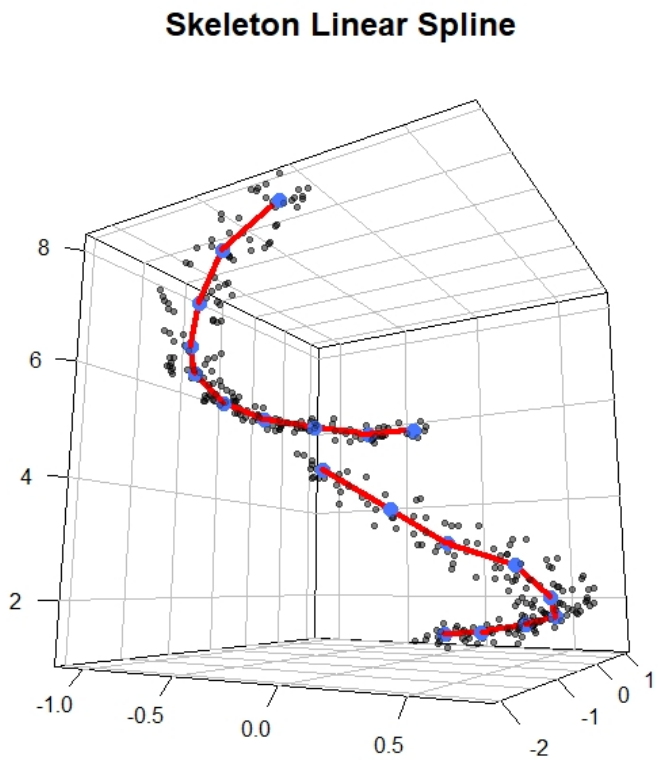
Regression on Covariates with Geometric Structures

Instead of clustering the data points, we have scalar response on covariate space with geometric structures, and we want to do regression.



Regression on Covariates with Geometric Structures

Instead of clustering the data points, we have scalar response on covariate space with geometric structures, and we want to do regression.



Related Literature

Classical two steps approach:
(1) map the data to the tangent space or some embedding space
(2) run usual regression methods with transformed data

- Pioneered by the Principle Component Regression * and the Partial Least Squares†
- Relate the regression coefficients to exterior derivatives‡
- Manifold Adaptive Local Linear Estimator for the Regression (MALLER)§

*Massy (1965)
†Wold (1975)
‡Aswani et al. (2011)
§Cheng and Wu (2013)

Related Literature

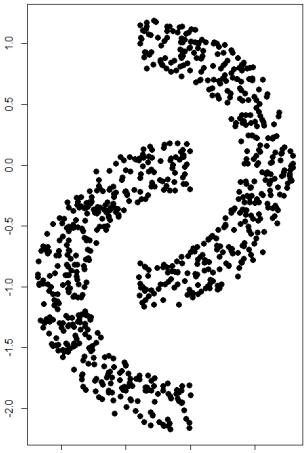
Nonparametric regression/ machine learning approaches:

- kernel machine learning*
- manifold regularization†
- spectral series approach‡

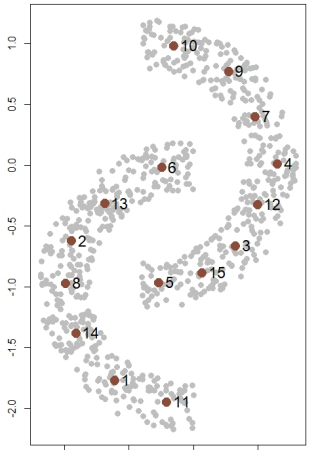
kNN regression and kernel Regression were shown to converge at **rates that depend only on the intrinsic dimensions** of data§.

*Schölkopf and Smola (2002)
†Belkin et al. (2006)
‡Lee and Izbicki (2016)
§Kpotufe (2009a,b, 2011); Kpotufe and Garg (2013); Kpotufe and Verma (2017)

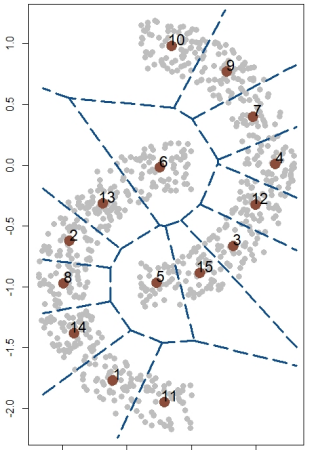
Revisit Skeleton Clustering



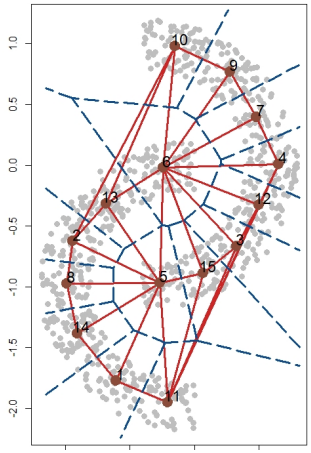
(a) Data



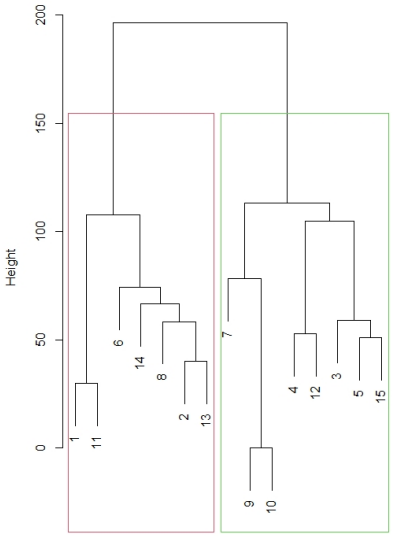
(b) Knots



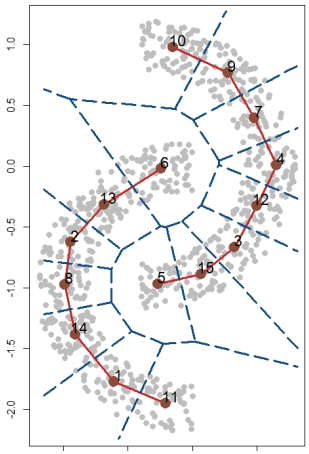
(c) Voronoi Cells



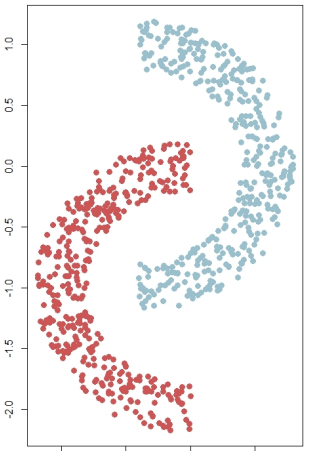
(d) Skeleton



(e) Dendrogram

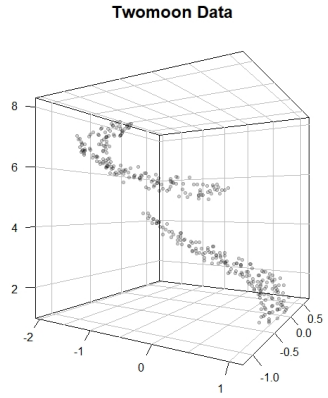


(f) Segmentation

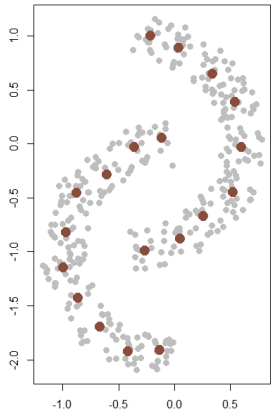


(g) Clustering

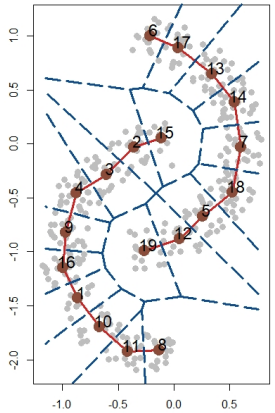
Our Approach: Skeleton Regression Framework



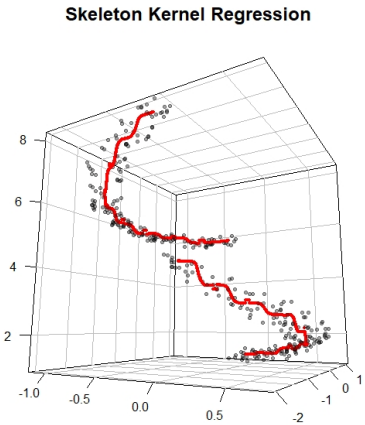
(a) Data



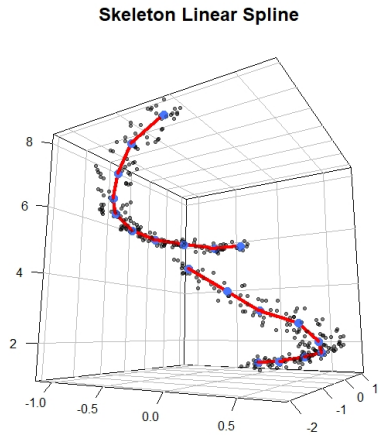
(b) Knots



(c) Skeleton



(d) S-Kernel Regression



(e) Linear Spline

Our Approach: Skeleton Regression Framework

Algorithm Skeleton Regression

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

1. **Skeleton Construction.** Construct a skeleton representation of the input space. Knots and edges can be tuned with subject knowledge.
2. **Data Projection.** Project the input vectors onto the skeleton structure.
3. **Skeleton Regression Function Estimation.** Fitting nonparametric regression functions on the skeleton.
4. **Prediction.** Project the feature vectors of new data onto the learnt skeleton structure and use the estimated regression function for prediction.

Data Projection Illustration

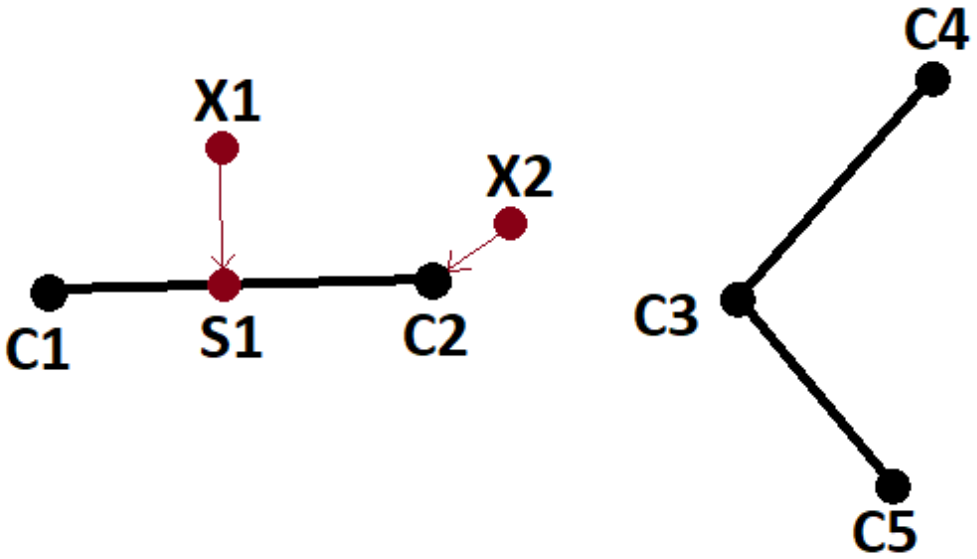


Figure: Illustration of projection to the skeleton. The skeleton structure is given by the black dots and black lines. Data point X_1 is projected to S_1 on the edge between C_1 and C_2. Data point X_2 is projected to knot C_2 as it's two closest neighbors C_2 and C_3 are not connected by an edge in the skeleton.

Data Projection

Construct the skeleton the same way as in the Skeleton Clustering work. Given covariate $\boldsymbol{x}$, let $l_1(\boldsymbol{x}), l_2(\boldsymbol{x}) \in \{1, \ldots, k\}$ be the index of its closest and second closest knots in terms of Euclidean metric on $\mathcal{X}$

- If $V_{l_1(\boldsymbol{x})}$ and $V_{l_2(\boldsymbol{x})}$ are not connected, $\Pi(\boldsymbol{x}) = V_{l_1(\boldsymbol{x})}$.
- If $V_{l_1(\boldsymbol{x})}$ and $V_{l_2(\boldsymbol{x})}$ are connected, let $t = \frac{\left(\boldsymbol{x} - V_{l_1(\boldsymbol{x})}\right)^T \cdot \left(V_{l_2(\boldsymbol{x})} - V_{l_1(\boldsymbol{x})}\right)}{\left\| V_{l_2(\boldsymbol{x})} - V_{l_1(\boldsymbol{x})} \right\|^2}$ be the projection proportion,

$$\Pi(\boldsymbol{x}) = V_{l_1(\boldsymbol{x})} + \left(V_{l_2(\boldsymbol{x})} - V_{l_1(\boldsymbol{x})}\right) \cdot \begin{cases} 0, & \text{if } t < 0 \\ 1, & \text{if } t > 0 \\ t, & \text{otherwise} \end{cases} .$$

Linear Spline Regression on Graph

Construct a linear regression model on each edge while requiring the predicted values to agree on shared vertices.

Linear Spline Regression on Graph

Construct a linear regression model on each edge while requiring the predicted values to agree on shared vertices.

Can be parameterized by the values on all the knots to get graph-transformed data $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T$ of $\boldsymbol{z}_j$ the length v transformed data vector for $\boldsymbol{x}_j$ encoding proportional weights on the corresponding vertices.

Linear Spline Regression on Graph

Construct a linear regression model on each edge while requiring the predicted values to agree on shared vertices.

Can be parameterized by the values on all the knots to get graph-transformed data $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T$ of $\boldsymbol{z}_j$ the length v transformed data vector for $\boldsymbol{x}_j$ encoding proportional weights on the corresponding vertices.
The S-Lspline model in matrix form can be written as

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{Z}) = \boldsymbol{\beta}^T \boldsymbol{Z} \tag{4}$$

for $\boldsymbol{\beta}$ the $v \times 1$ column vector of coefficients, with each coefficient representing the predicted value on the corresponding knot.
For estimation, we can use ordinary least squares regression to get

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T \boldsymbol{Z})^{-1} \boldsymbol{Z} \boldsymbol{y} \tag{5}$$

Yinyang Data

let $\epsilon \sim N(0, 0.01)$ and let θ be the angle of the covariates, then

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

Add IID random $N(0, 0.1)$ variables to make covariates a total of 1000 dimensions.

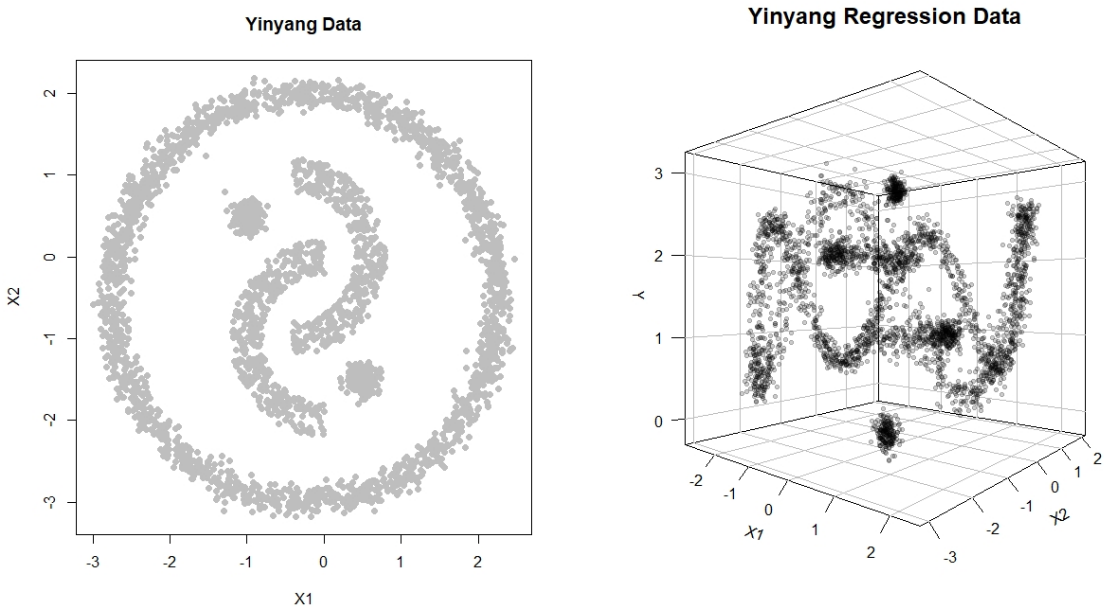


Figure: Yinyang Regression Data

Yinyang Regression Data

Use 5-fold cross-validation to calculate the sum of squared errors (SSE) and repeat for 100 times.

- Skeleton-based linear spline(**S-Lspline**)
- Skeleton-based kernel regression (**S-Kernel**)
- Skeleton-based k-Nearest-Neighbors regressor (**S-kNN**)
- Euclidean-based kNN (**kNN**)
- **Lasso** and **Ridge** regressions
- Spectral Series approach with radial kernel (**SpecSeries**)*.

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	-

*https://projecteuclid.org/journals/supplementalcontent/10.1214/16-EJS1112/supzip_1.zip

SwissRoll Data

let $\epsilon \sim N(0, 0.01)$, 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 $\theta_{13} = \pi 3^{u_1}$.

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

Let $\tilde{\theta}_{13} = \theta_{13} - 2\pi$, and let $\epsilon \sim N(0, 0.3)$. We set

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

Add IID random $N(0, 0.1)$ variables to make covariates a total of 1000 dimensions.

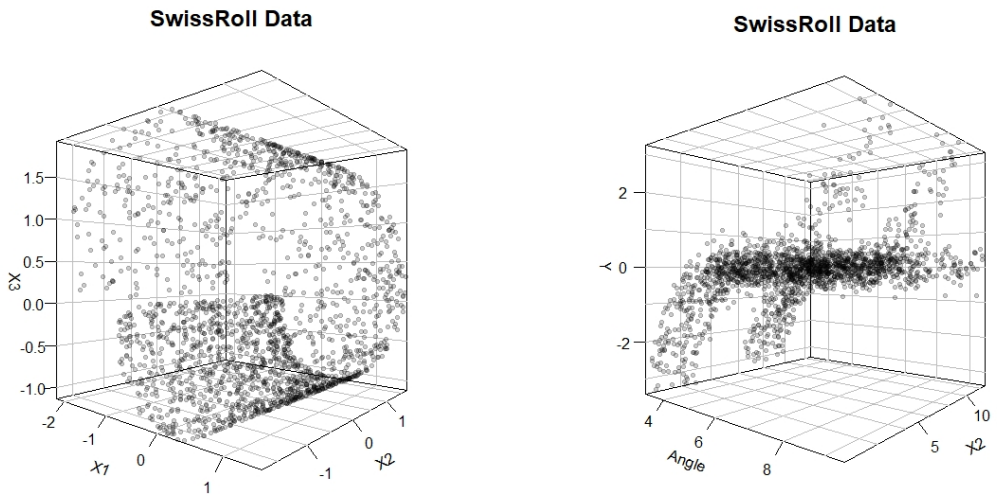


Figure: SwissRoll Regression Data

SwissRoll Data

Use 5-fold cross-validation to calculate the sum of squared errors (SSE) and repeat for 100 times.

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	-

Lucky Cat Data

Set of 72 gray-scale images of size 128×128 pixels, obtained through rotating the object by 72 equispaced angles on a single axis.

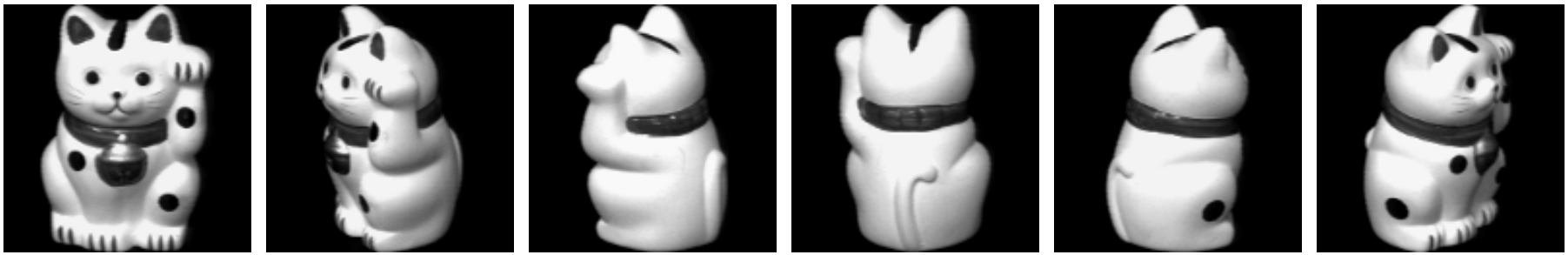


Figure: A part of the lucky cat images from the COIL-20 processed dataset.

- The response for estimation is the angle of rotation.
- To avoid the circular response issue, we remove the last 8 images from the sequence and use the first 64 images.

Lucky Cat Data Performance

Used the leave-one-out cross-validation for assessment

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

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

Conclusion

Contribution:

- Skeleton to represent the geometry of data and assist in regression
- Apply nonparametric regression techniques on graphs (kernel regression, splines, kNN)
- Empirical results demonstrating the usefulness of our framework

Conclusion

Contribution:

- Skeleton to represent the geometry of data and assist in regression
- Apply nonparametric regression techniques on graphs (kernel regression, splines, kNN)
- Empirical results demonstrating the usefulness of our framework

Some future directions:

- Generalizing skeleton graphs to simplicial sets with higher-dimensional geometric information
- Relate skeleton construction to Persistent Homology
- Accounting for the randomness of knots.
- Incorporating explicit dimension reduction.
- Longitudinal data/ online updating of the skeleton representation

Project Three: Assessing Epidemic Models under Missingness in Contact Network

Tyler McCormick, Arun Chandrasekhar, Paul Goldsmith-Pinkham, Jerry Wei, Samuel Thau

Contact Network for Epidemic Modeling

- Mobility data based on phone call and text records*
- Mobility networks derived from commute flows data†
- Google COVID-19 Aggregated Mobility Research Dataset‡

Concerns:

- Proxies to true contact network, mismeasurements inevitable
- Chandrasekhar et al. (2021) illustrate that small misalignment of the model can lead to failure of local targeting policy guided by the epidemiological models.

*Wesolowski et al. (2012); Bengtsson et al. (2015); Engebretsen et al. (2020); Milusheva (2020)

†Fajgelbaum et al. (2021); Alsing et al. (2020)

‡Kapoor et al. (2020); Ruktanonchai et al. (2020); Venkatramanan et al. (2021)

Problem Setup

Model the network as $G = L \cup E$

- L is a graph majorly composed of local connections
- E is a sparse Erdos-Renyi random graph
- we only observe $\hat{G} = L$

Problem Setup

Model the network as $G = L \cup E$

- L is a graph majorly composed of local connections
- E is a sparse Erdos-Renyi random graph
- we only observe $\hat{G} = L$

- Node i_0 is infected at time $t = 0$ node
- The disease pass along each edge in the graph with independent and identically probability $p = p_n$
- Define y_{jt} as an indicator of whether individual j is infected at time t

Define the α-*risk set* as

$$Q(\alpha; G, p, T, i_0) = \{j : \ \mathbb{P}(y_{jT} = 1 \mid G, p, T, i_0) \geq \alpha\}.$$

Problem Setup

Model the network as $G = L \cup E$

- L is a graph majorly composed of local connections
- E is a sparse Erdos-Renyi random graph
- we only observe $\hat{G} = L$

- Node i_0 is infected at time $t = 0$ node
- The disease pass along each edge in the graph with independent and identically probability $p = p_n$
- Define y_{jt} as an indicator of whether individual j is infected at time t

Define the α-*risk set* as

$$Q(\alpha; G, p, T, i_0) = \{j : \mathbb{P}(y_{jT} = 1 \mid G, p, T, i_0) \geq \alpha\}.$$

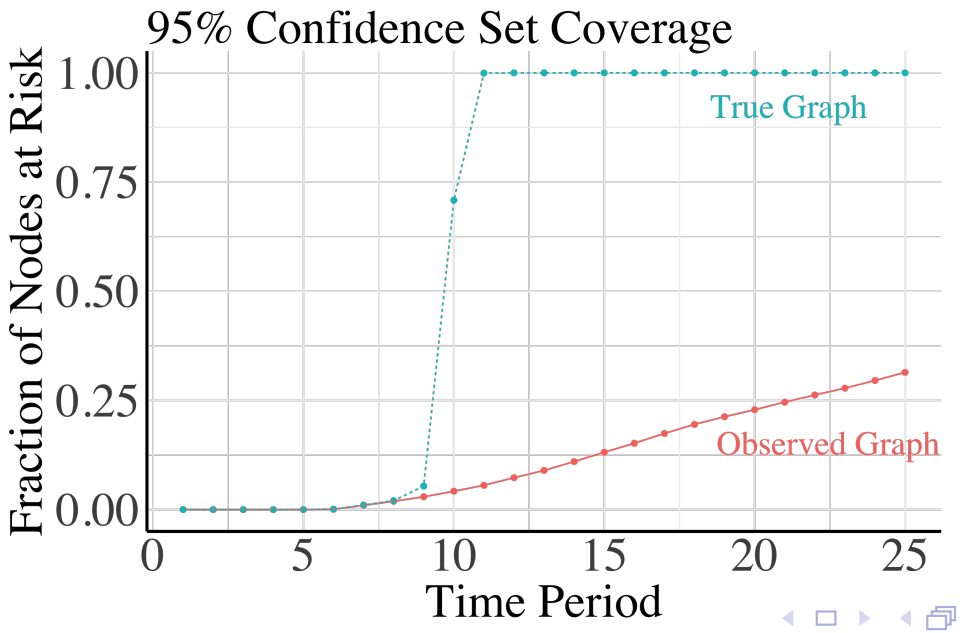
Goal:

For any $\alpha \in (0, 1)$, there exists a T_α independent of n such that for every T fixed in n, with $T_\alpha \leq T \leq C' \log n$ for some C' fixed in n, as $n \to \infty$

$$\frac{|Q(\alpha; \hat{G}, T) \cap Q(\alpha; G, T)|}{|Q(\alpha; G, T)|} \to 0.$$

Preliminary Simulation Result

- Number of nodes $n = 35000$
- Each node in L assigned a position on $[0, 1]^2$ uniformly at random
- Expected degree in L set as 75
- Form links to all nodes within radius $r = \sqrt{\frac{75}{\pi n}}$
- Add links independently with probability $\beta = (1 + \varepsilon)\frac{\log n}{n}$, with $\varepsilon = 0.0001$, giving $\beta = 0.0003$.
- Set a basic reproductive number $r_0 = 3$, giving transmission probability $p = 0.035$
- Track $|Q(\alpha; \hat{G}, T)|/n$ and $|Q(\alpha; G, T)|/n$, with $\alpha = 0.95$



Future Work

- Theoretically characterize the conditions when missingness ruins predictions
- Study how network geometry interacts with the influence of missingness
- Assess the uncertainty caused by missingness, convoluted with the variation in the disease transmission process.

Thanks for listening!

Reference I

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. 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.

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`.

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.

Reference II

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. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006. 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.

Reference III

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.

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.

J. E. Chacón. Mixture model modal clustering. *Advances in Data Analysis and Classification*, 13:379–404, 6 2019.

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.

Reference IV

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.

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.

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.

Reference V

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.

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`.

Reference VI

C. Hennig. Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification 2010 4:1*, 4:3–34, 1 2010.

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.

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`.

Reference VII

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`.

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`.

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.

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`.

Reference VIII

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`.

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.

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.

Reference IX

L. Scrucca. Identifying connected components in gaussian finite mixture models for clustering. *Computational Statistics & Data Analysis*, 93:5–17, 2016.

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.

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.

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.

Reference X

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.

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.

Adjusted Rand Index

For two partitions $X = \{X_1, \ldots, X_r\}$ and $Y = \{Y_1, \ldots, Y_s\}$, let $n_{ij} = |X_i \cap Y_j|$, we have the contingency table

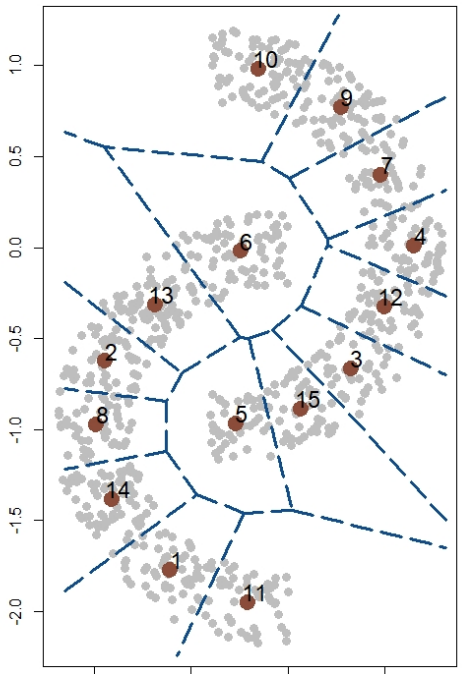
	Y_1	Y_2	$\cdots$	Y_s	*Sums*
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
Sums	b_1	b_2	$\cdots$	b_s	

And the Adjusted Rand Index (ARI) adjusting for permutation chance is

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

Edge Weight: Face Density (FD)

- For connected components we expect to see many observations around their mutual boundary.
- The *Face Density (FD)* as the PDF integrated over the face region.
- let the face region between two knots c_j, c_ℓ be $F_{j\ell} \equiv \mathbb{C}_j \cap \mathbb{C}_\ell$. Then $S_{j\ell}^{FD} = \int_{F_{j\ell}} p(x)dx = \int_{F_{j\ell}} d\mathbb{P}(x)$.

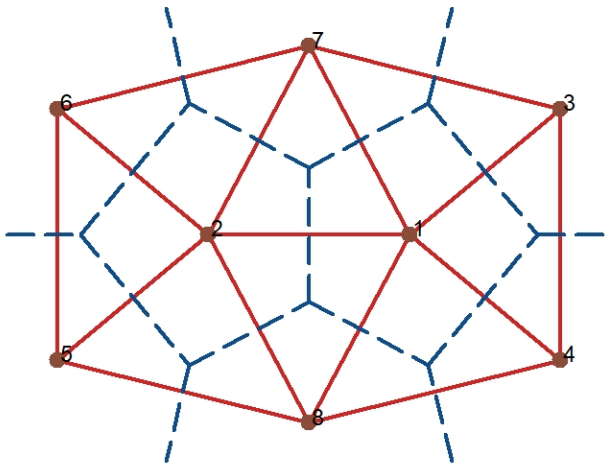


Edge Weight: Face Density Estimation

- The boundary of two Voronoi regions is orthogonal to the line passing through the two corresponding knots and is at the middle point.
- Let $\Pi_{j\ell}(x)$ be the projection of $x \in \mathcal{X}$ onto the line passing through c_j and c_ℓ
- The estimator $\hat{S}_{j\ell}^{FD}$ is defined as

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

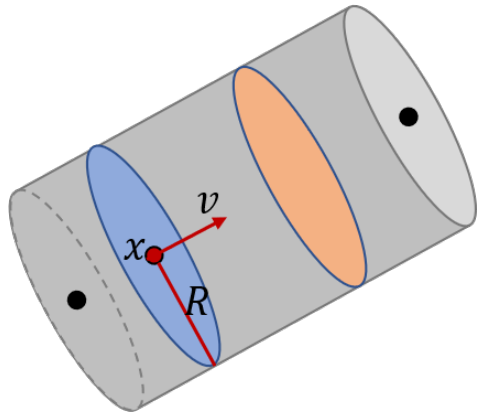
- This is 1-D KDE.



Edge Weight: Tube Density (TD)

- Similar to face density but has a predefined regular shape.
- Define a disk area centered at x with radius R and normal direction ν as

$$\text{Disk}(x, R, \nu) = \{y : ||x - y||_2 \leq R, (x - y)^T \nu = 0\}$$



- Parameterize the central line through c_j, c_ℓ as $\{c_j + t(c_\ell - c_j) : t \in [0, 1]\}$.
- Examine the integrated density within the disks along the central line.

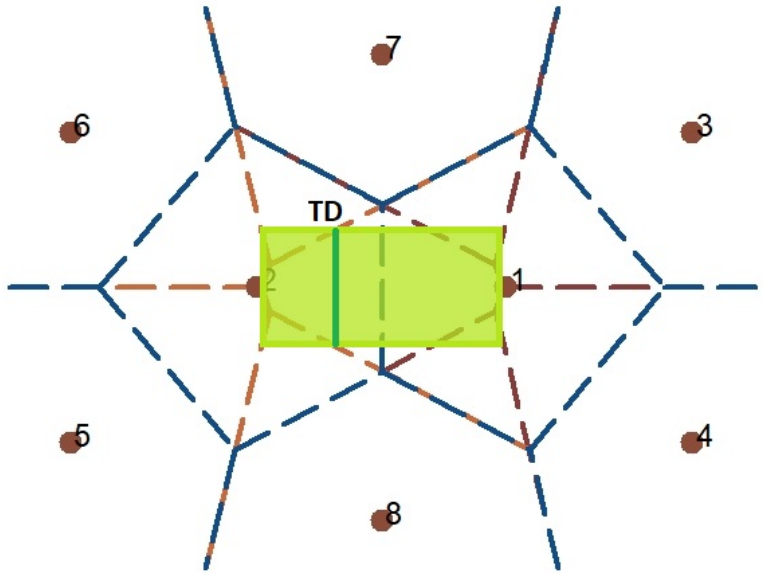
Edge Weight: Tube Density (TD)

Define the integrated density (called disk density) in the disk region as

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

Tube density (TD) is the minimal disk density along the central line, i.e.,

$$S_{j\ell}^{TD} = \inf_{t\in[0,1]} \text{pDisk}_{j\ell,R}(t). \tag{6}$$



Edge Weight: Tube Density Estimation

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

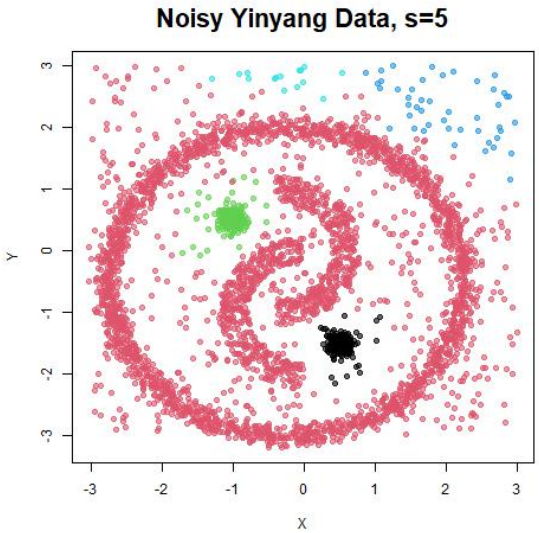
$$\widehat{\text{pDisk}}_{j\ell,R}(t) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - c_j - t(c_\ell - c_j)}{h}\right) I(||X_i - \Pi_{j\ell}(X_i)|| \leq R)$$

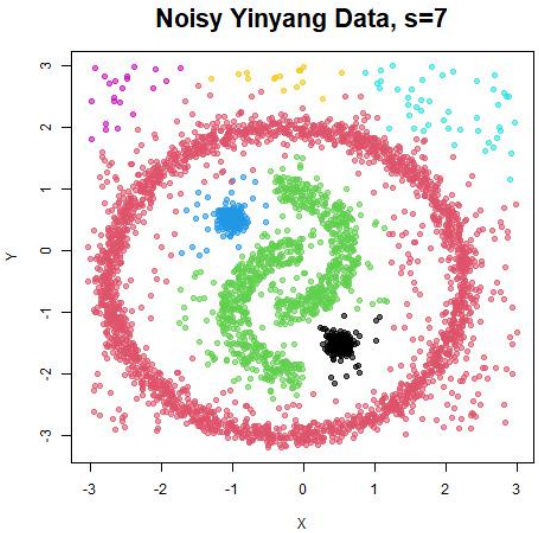
- Estimate the TD as

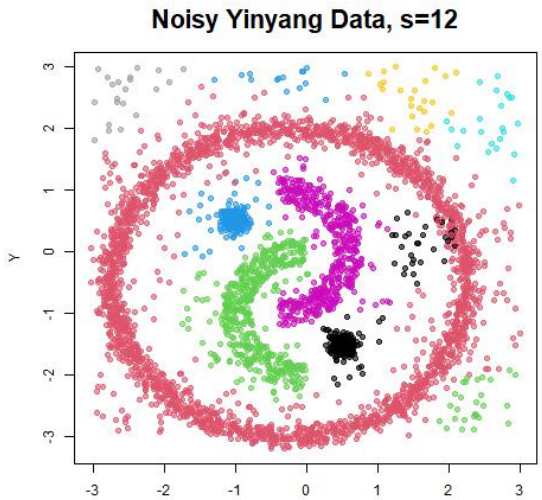
$$\hat{S}_{j\ell}^{TD} = \min_{t\in[0,1]} \widehat{\text{pDisk}}_{j\ell,R}(t). \tag{7}$$

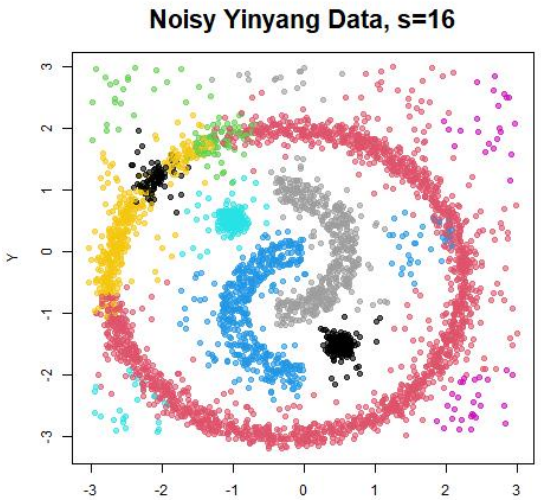
Data with Noise

- Added 640 (20% of the true signals) noisy points to the Yinyang dataset ($d = 1000$)
- Use Voronoi density and apply single linkage for knot segmentation.

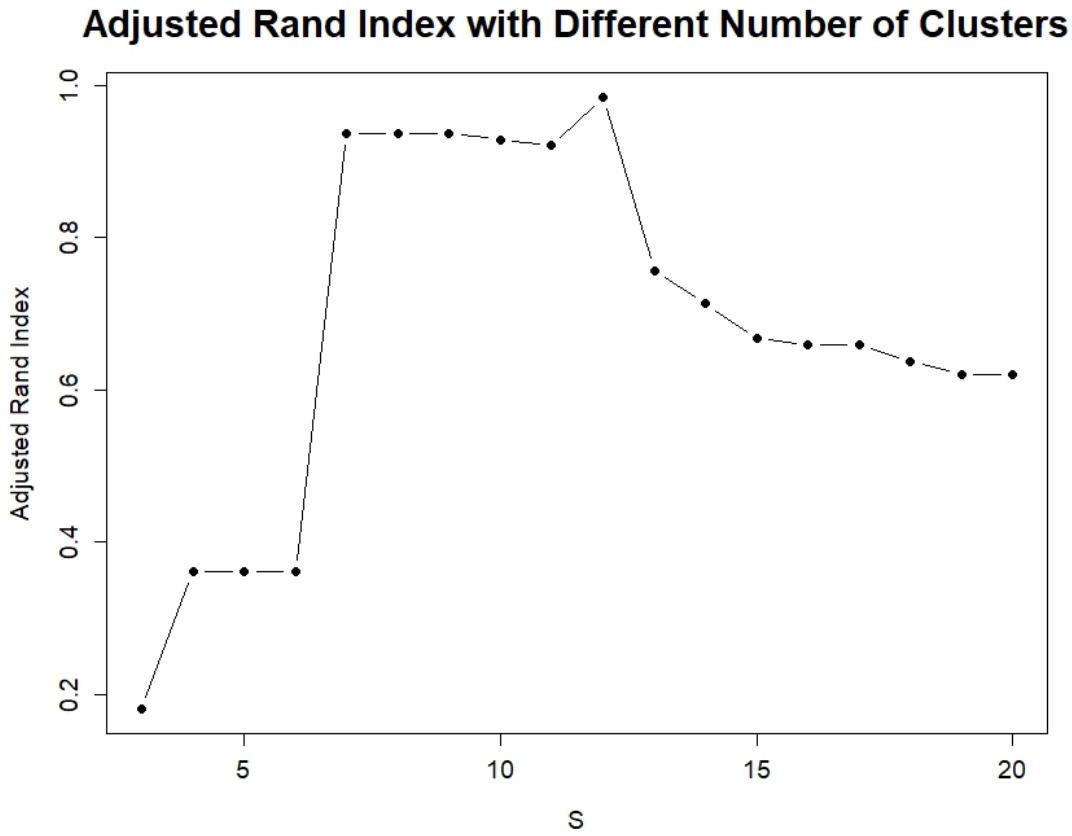






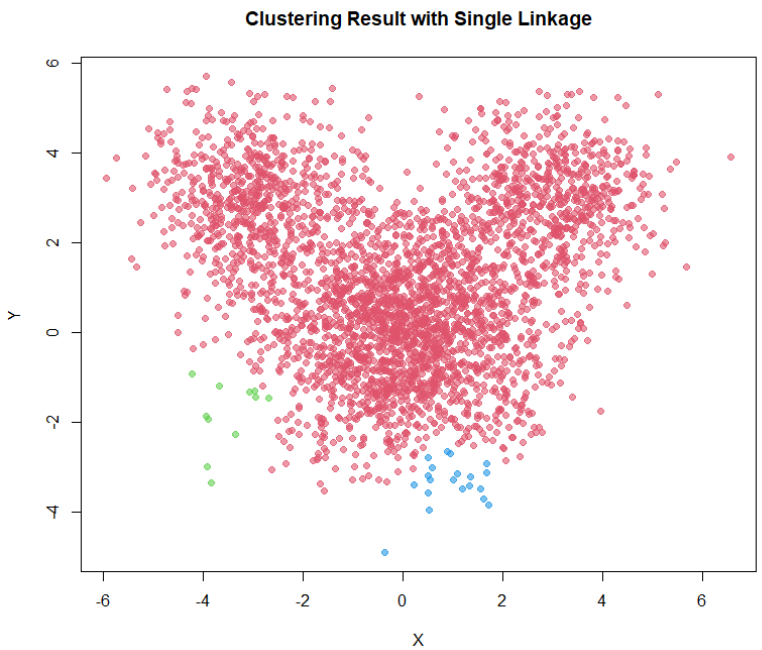


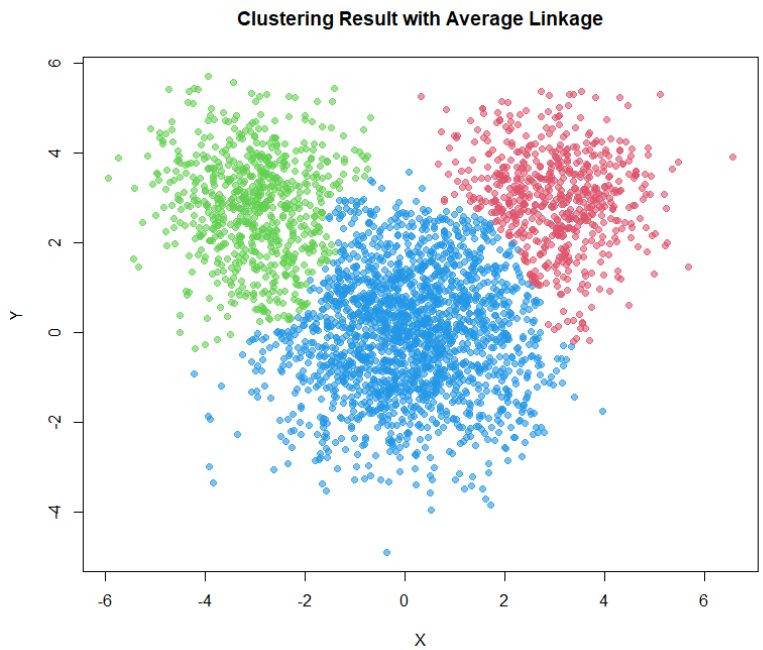
Data with Noise



Overlapping Clusters

- Add additional noises to make the three structures overlap
- Using Single linkage for knots segmentation fails to discover the true structure.
- Using average linkage recovers the underlying three components.





Skeleton-Based Distance

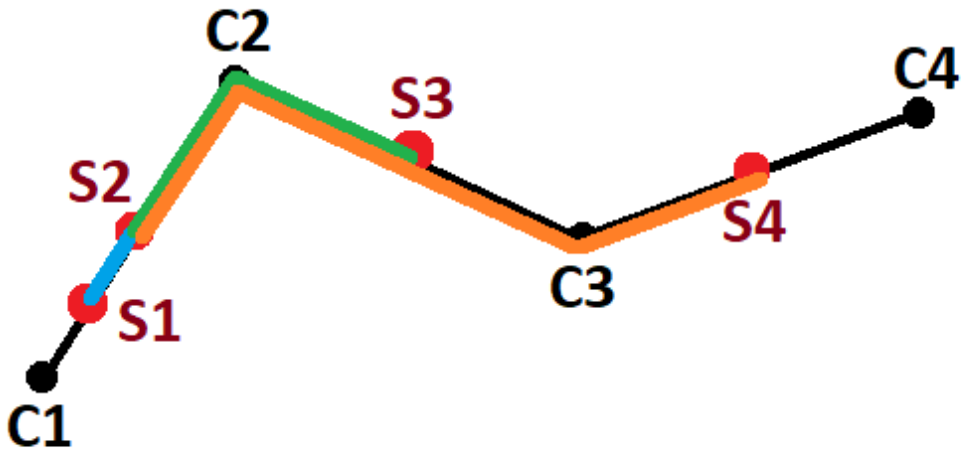


Figure: Illustration of skeleton-based distance.

Let C_1, C_2, C_3, C_4 be the knots, and let S_2, S_3, S_4 be the mid-point on the edges E_{12}, E_{23}, E_{34} respectively. Let S_1 bet the midpoint between C_1 and S_2 on the edge.

Let $d_{ij} = \|C_i - C_j\|$ denotes the length of the edge E_{ij}.

$d_{\mathcal{S}}(S_1, S_2) = \frac{1}{4} d_{12}$ illustrated by the blue path.

$d_{\mathcal{S}}(S_2, S_3) = \frac{1}{2} d_{12} + \frac{1}{2} d_{23}$ illustrated by the green path.

$d_{\mathcal{S}}(S_2, S_4) = \frac{1}{2} d_{12} + d_{23} + \frac{1}{2} d_{34}$ illustrated by the orange path.

Skeleton-Based Distance

Let $\boldsymbol{S}_j, \boldsymbol{s}_\ell \in \mathcal{S}$ be two arbitrary points on the skeleton. We measure the skeleton-based distance as the graph path length:

- If $\boldsymbol{S}_j, \boldsymbol{s}_\ell$ are disconnected, $d_{\mathcal{S}}(\boldsymbol{S}_j, \boldsymbol{s}_\ell) = \infty$.
- If $\boldsymbol{S}_j$ and $\boldsymbol{s}_\ell$ are on the same edge,

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

- For $\boldsymbol{S}_j$ and $\boldsymbol{s}_\ell$ on two different edges that share a knot V_0,

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

- 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 $\boldsymbol{s}_\ell$.

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

Skeleton Kernel Regression

Let $K_h(.) = K(./h)$ be a non-negative kernel function with bandwidth $h > 0$ and $d_{\mathcal{S}}$ the distance on skeleton, the corresponding skeleton-based kernel (S-kernel) regressor for $\boldsymbol{s} \in \mathcal{S}$ is

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{j=1}^{N} K_h(d_{\mathcal{S}}(\Pi(\boldsymbol{x}_j), \boldsymbol{s}))\, Y_j}{\sum_{j=1}^{N} K_h(d_{\mathcal{S}}(\Pi(\boldsymbol{x}_j), \boldsymbol{s}))} \tag{8}$$

A concrete kernel function example is the popular Gaussian kernel that

$$K_h(d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell)) = \exp\left(- \frac{d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell)^2}{h^2} \right) \tag{9}$$

Notably, the kernel function calculation only depends on the skeleton distances and hence is independent of the dimension of the original input or the intrinsic dimension of the manifold structure.

Yinyang Vary Knots

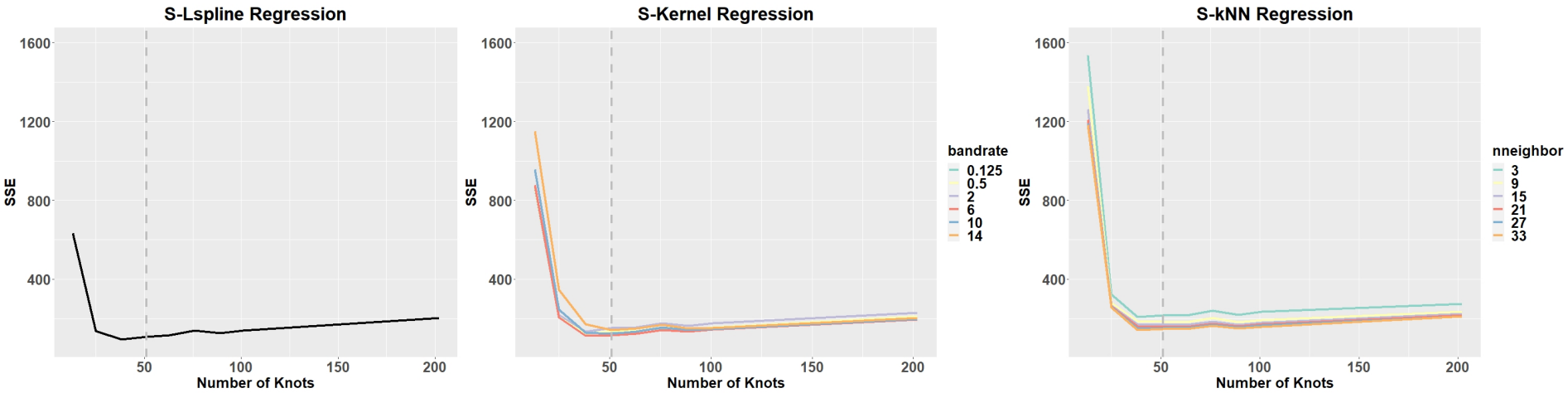


Figure: Yinyang $d = 1000$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

Yinyang Vary Cuts

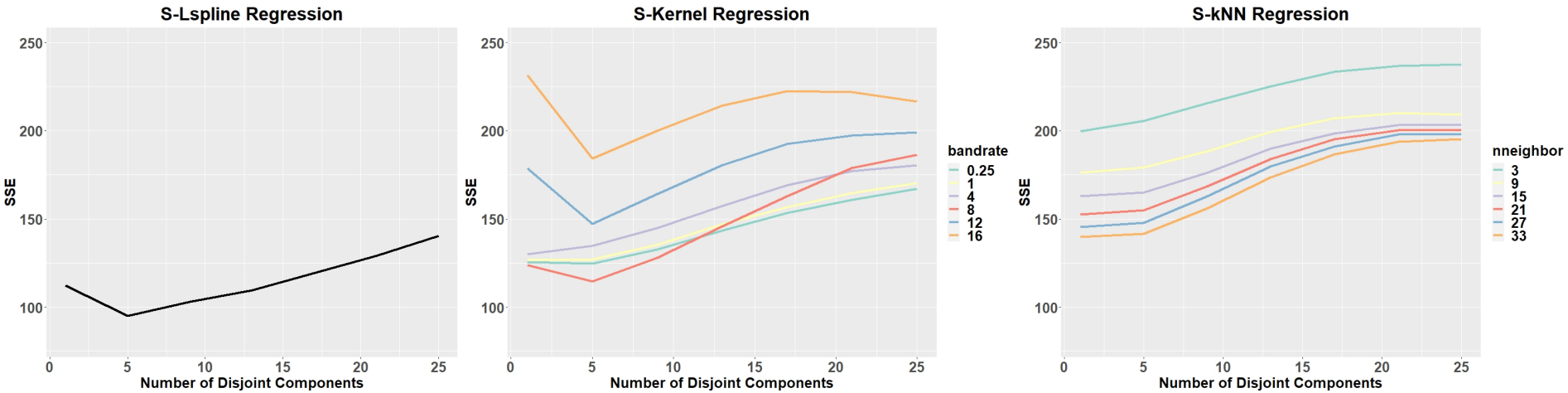


Figure: Yinyang $d = 1000$ data skeleton regression results with the number of knots fixed as 38 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

SwissRoll Vary Knots

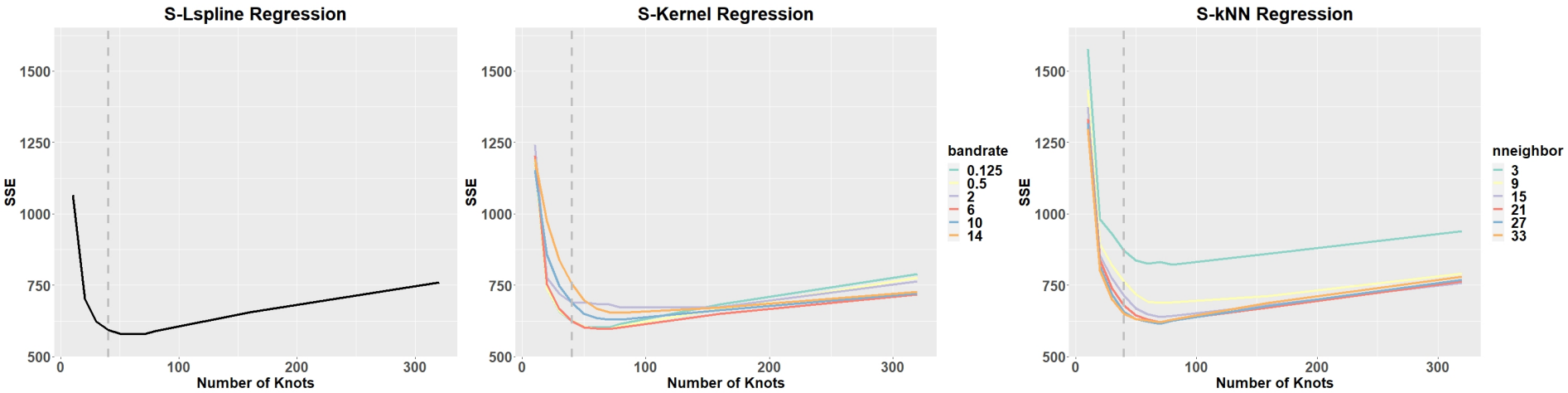


Figure: SwissRoll $d = 1000$ data regression results with varying number of knots. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

SwissRoll Vary Cuts

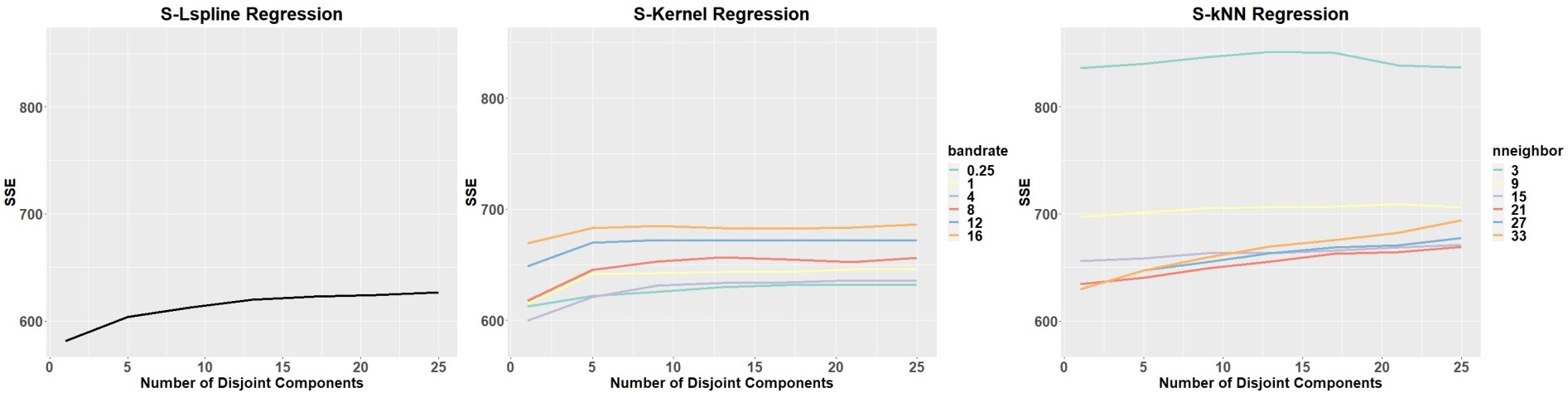


Figure: SwissRoll $d = 1000$ data skeleton regression results with the number of knots fixed as 70 but segmented into varying numbers of disjoint components. The medium SSE across the 100 simulated datasets with each given parameter setting is plotted.

S-Kernel Consistency

We estimate the *projected* regression function

$$m_{\mathcal{S}}(\boldsymbol{s}) = \mathbb{E}(y|\pi(\boldsymbol{x}) = \boldsymbol{s}), \boldsymbol{s} \in \mathcal{S} \tag{10}$$

on the skeleton domain $\mathcal{S}$.
We treat the edges and knots as separate domains.
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 $\int x^2K(x)dx < \infty$

S-Kernel Consistency: Edge

Theorem (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 $n \to \infty$, $h_n \to 0$, $nh_n \to \infty$,

$$|\hat{m}_n(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})| = O(h_n) + O_p\left(\sqrt{\frac{1}{nh_n}}\right) \tag{11}$$

S-Kernel Consistency: Knots with Nonzero Mass

Theorem (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 $nh_n \to \infty$,

$$|\hat{m}(\boldsymbol{s}) - m_{\mathcal{S}}(\boldsymbol{s})| = O(h_n) + O_p\left(\sqrt{\frac{1}{n}}\right) \tag{12}$$

S-Kernel Consistency: Knots with Zero Mass

Proposition

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 $nh_n \to \infty$,

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

Higher-Order Spline on Graph: Edge Directions

- Odd-degree derivatives are directional and are dependent on the directions of the edges on the graph.
- However, many graphs, including the skeleton built in our framework, do not have built-in directions.
- Different edge directions do lead to different spline models on the graph and do give different empirical performances
- Many works on graph-based estimations that use derivatives implicitly assumed the directions as given a prior Wang et al. (2016).
- Further study on how the change of edge directions can affect such derivative-related models on graphs can be interesting and can help address this concern.

Higher-Order Spline on Graph: Feasibility

Classical spline methods use degree $p+1$ polynomial functions to achieve continuity at p-th order derivative.
Example: univariate cubic splines use polynomial function up to degree 3 to ensure that up to the second derivatives of the regression function agree at each knot.

However, on a graph, degree $p+1$ polynomial functions may fail to achieve continuity at p-th order derivative.
Example: For $p=1$, we fit a quadratic polynomial function on each edge, and we want the 1st-order derivatives of the models to agree on shared knots. Assume we have a complete graph with 6 knots and $\binom{6}{2}=15$ edges. For each quadratic function, we have 3 degrees of freedom and hence there are a total of $3\times 15=45$ degrees of freedom to spare. Then for the constraints, for each vertex there are $(r_j-1)\times 2=(5-1)\times 2=8$ conditions to satisfy, where $r_j=5$ is the degree of the vertices on the complete 6-knots graph. Consequently, we have $8\times 6=48$ conditions in total, larger than the degrees of freedom, and hence the specified quadratic spline model is infeasible.

Higher-Order Spline on Graph: Feasibility

- For the p-th order smoothness spline model to be feasible on general graphs (including complete graphs), we need $2p + 1$ degree polynomials.
- For any polynomial with degree less than $2p + 1$ the degrees of freedom can be negative on some graphs.
- Requiring degree $2p + 1$ polynomials may be too high a demand and can lead to regression functions that are more flexible than desired. For known sparse graphs that have only a few edges and loops smaller degree polynomials can be employed.
- Can also be parametrized by fitted values and derivative values on the knots and be estimated by ordinary least squares regression