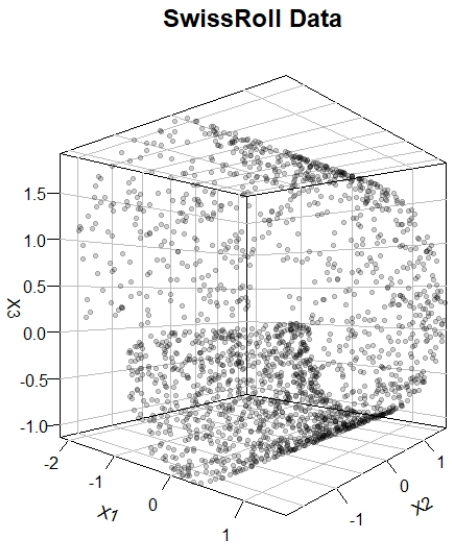
Skeleton Regression: A Graph-Based Approach to Estimation on Manifold

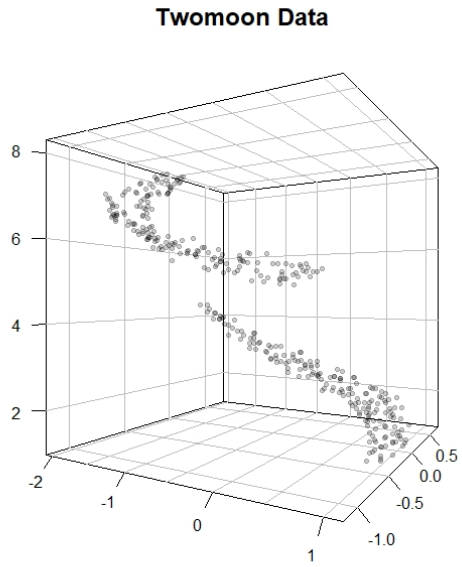
Jerry Wei
Department of Statistics, University of Washington
and
Yen-Chi Chen
Department of Statistics, University of Washington

Outline

Background

Many data nowadays have a geometric structure that the input data lies on a low dimensional manifold embedded inside the large-dimensional vector space.

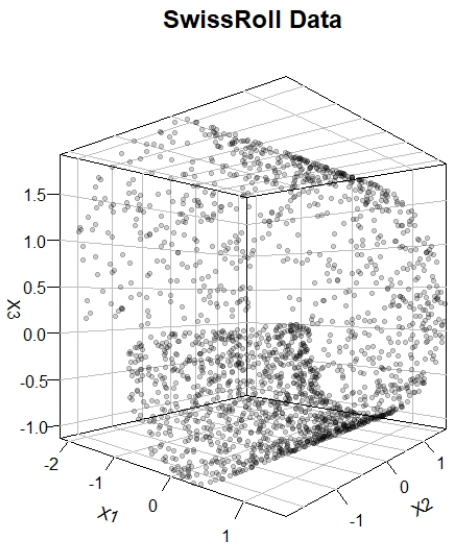


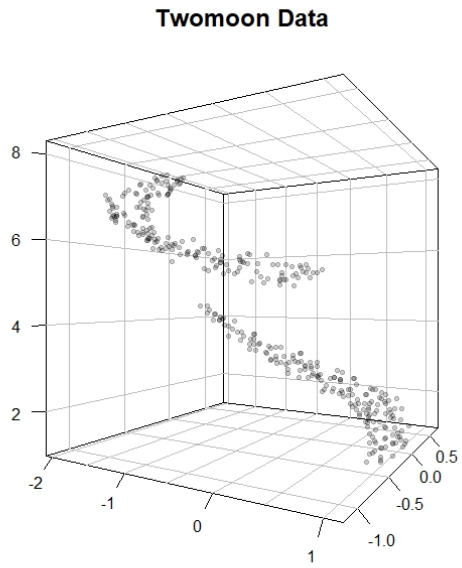


For various data analysis tasks to perform well, we need to understand such manifold structures of the data.

Background

Many data nowadays have a geometric structure that the input data lies on a low dimensional manifold embedded inside the large-dimensional vector space.

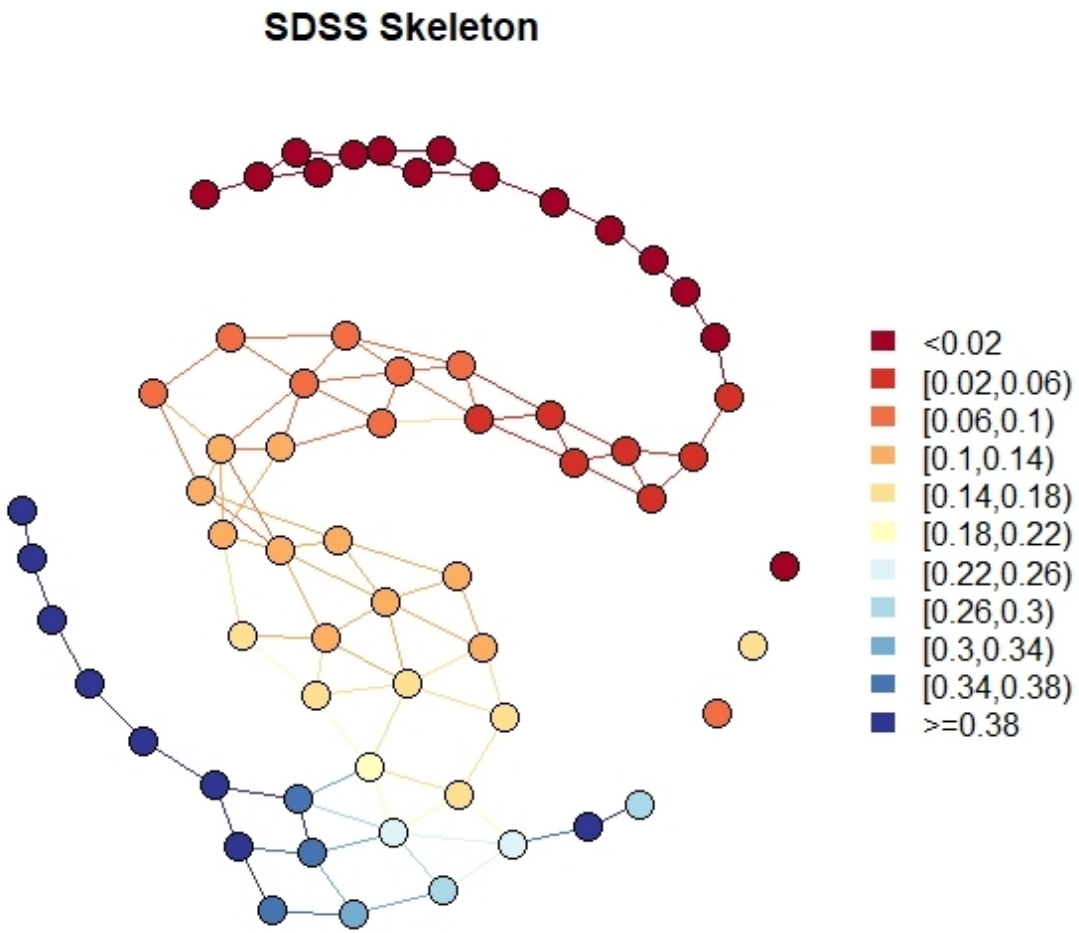




For various data analysis tasks to perform well, we need to understand such manifold structures of the data.

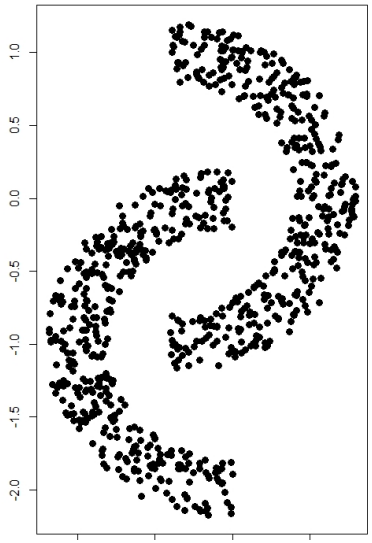
Our line of work propose to use a graph, called *Skeleton*, to summarize the manifold structure and assist various manifold learning tasks.

Example of Skeleton Representation

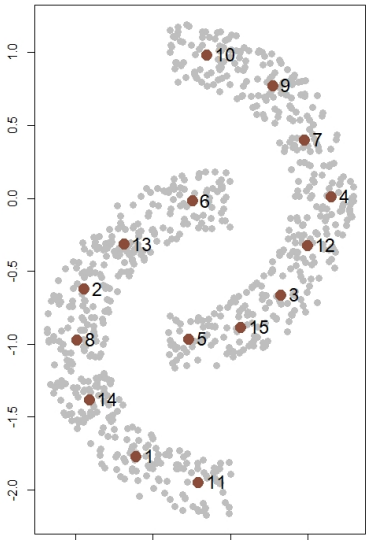


Sloan Digital Sky Survey (SDSS) data with 5 covariates measuring apparent magnitude of stars from images taken using 5 photometric filters. Response is the true redshift.

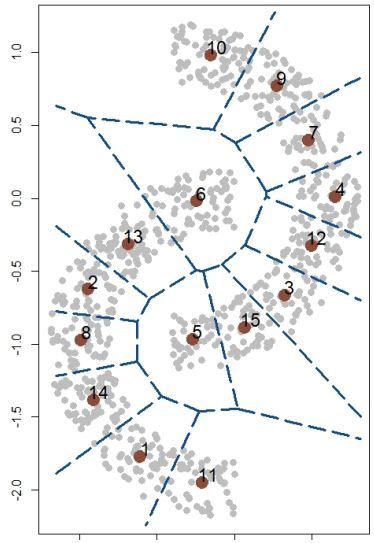
Background: Skeleton Clustering



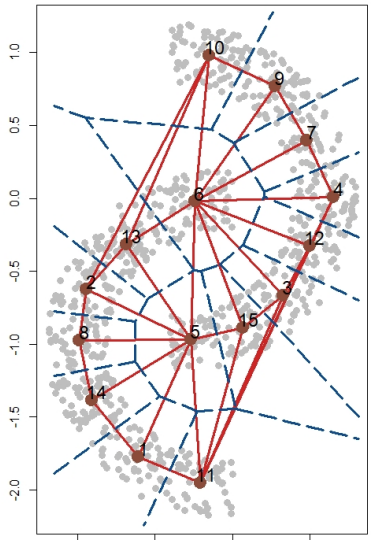
(a) Data



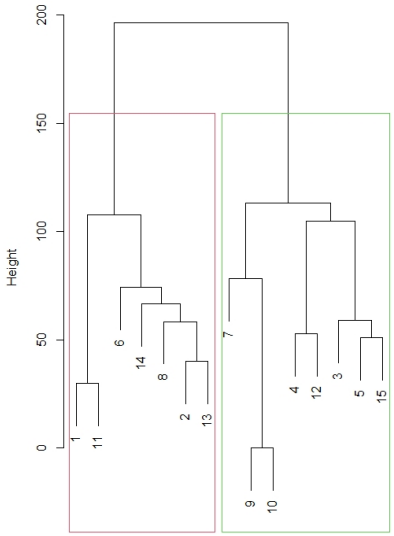
(b) Knots



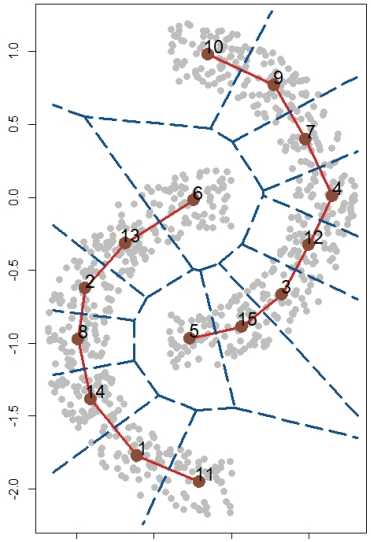
(c) Voronoi Cells



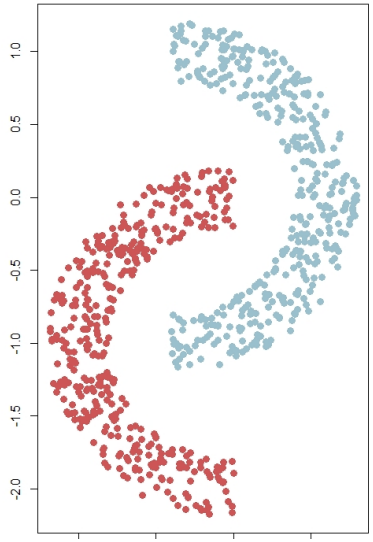
(d) Skeleton



(e) Dendrogram



(f) Segmentation



(g) Clustering

Background: Skeleton Clustering

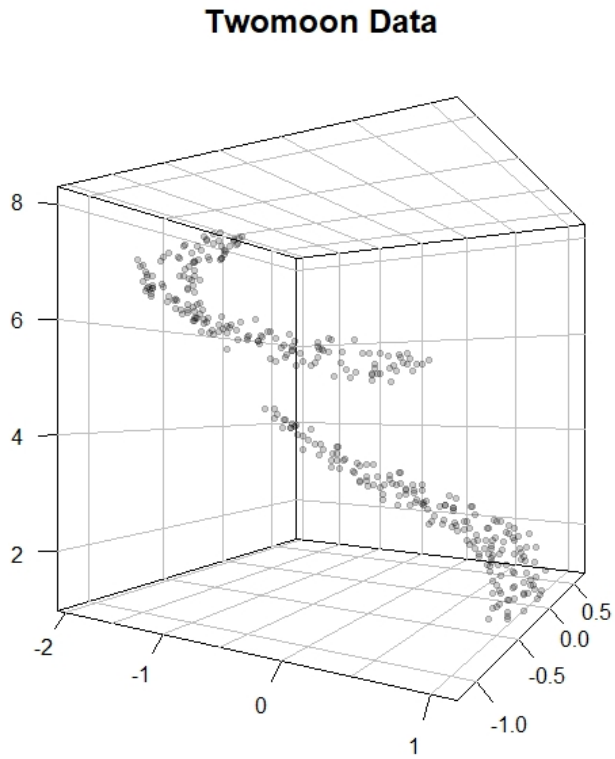
Algorithm Skeleton Clustering

Input: Observations $X_1, \cdots, X_n$, number of knots k

1. **Knot construction.** Perform k-means clustering with a large number of k; the centers are the knots. Generally, we choose $k = [\sqrt{n}]$.
2. **Edge construction.** Apply the Delaunay triangulation to the knots.
3. **Edge weights construction.** Add density-based similarity weights to each edge using Voronoi density (also Face density, Tube density) approach.
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.

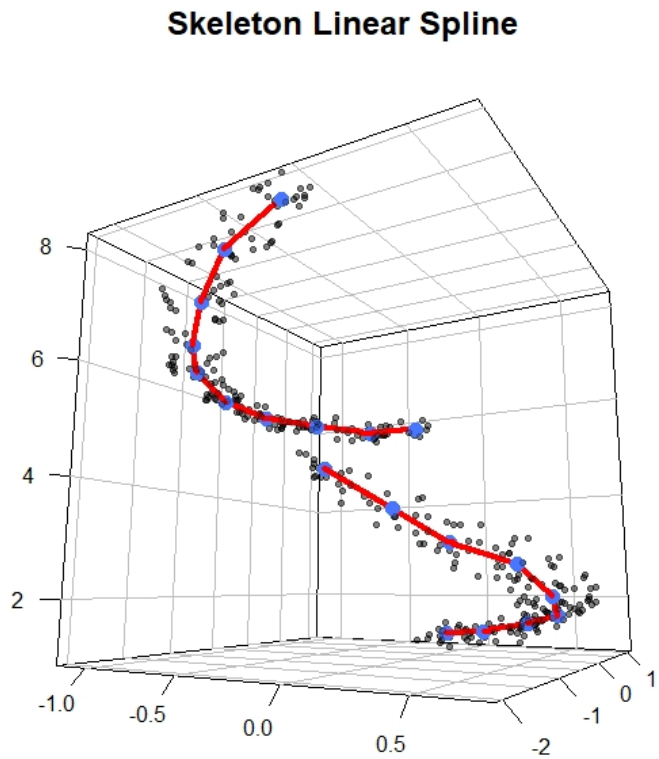
New Task: Regression on Manifold-Valued Input

Instead of clustering the data points, we have scalar response on manifold-valued input space, and we want to do regression.

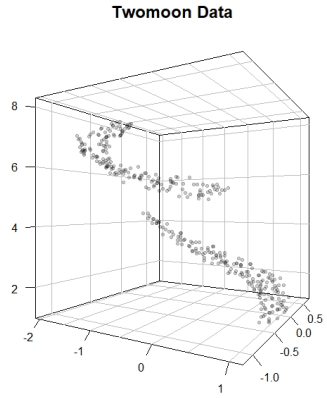


New Task: Regression on Manifold-Valued Input

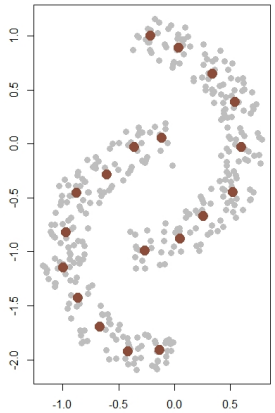
Instead of clustering the data points, we have scalar response on manifold-valued input space, and we want to do regression.



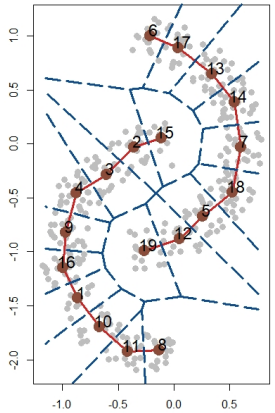
Our Approach: Skeleton Regression Framework



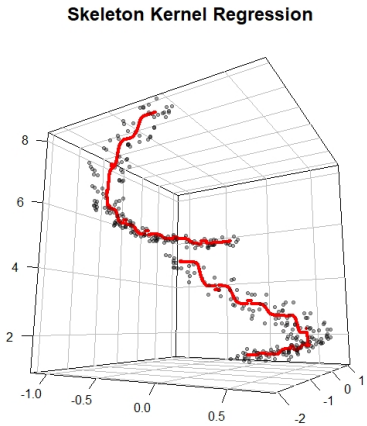
(a) Data



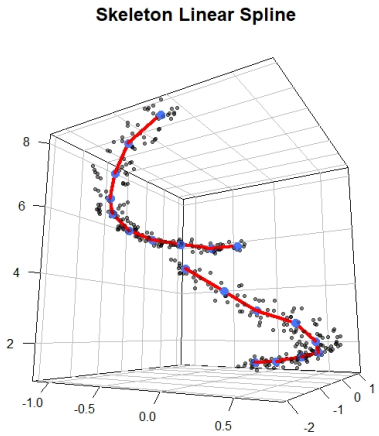
(b) Knots



(c) Skeleton



(d) S-Kernel Regression



(e) Linear Interpolation

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

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 using kernel regression, linear interpolation, or additional methods
4. **Prediction.** Project the feature vectors of new data onto the learnt skeleton structure and use the estimated regression function for prediction.

Brief Literature Review

Classical approach to explicitly account for geometric structure takes two steps:
(1) map the data to the tangent space or some embedding space and then
(2) run usual regression methods with transformed data

- Pioneered by the Principle Component Regression (PCR) (Massy, 1965) and the Partial Least Squares (PLS) (Wold, 1975)
- Aswani et al. (2011) relate the regression coefficients to exterior derivatives
- Cheng and Wu (2013) propose the Manifold Adaptive Local Linear Estimator for the Regression (MALLER)

Some **nonparametric regression approaches** can also deal with the manifold structure of the data

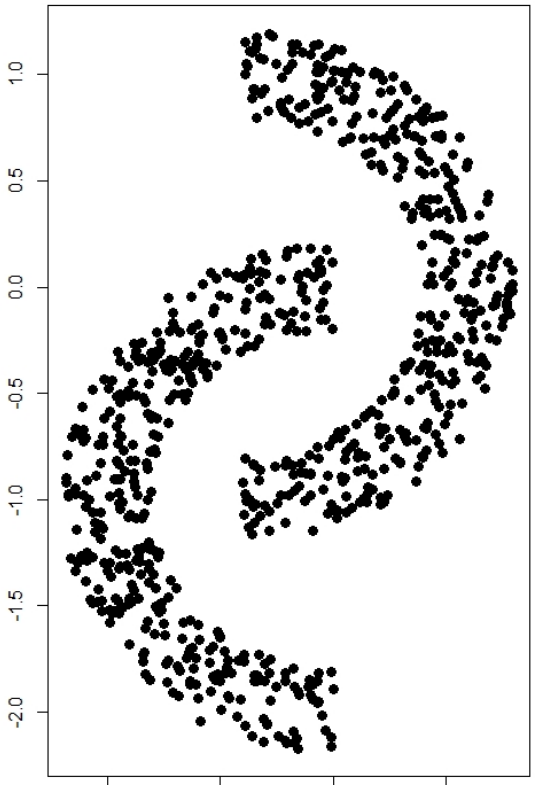
- kernel machine learning (Schölkopf and Smola, 2002)
- manifold regularization (Belkin et al., 2006)
- spectral series approach (Lee and Izbicki, 2016)

In recent years many nonparametric regressors, particularly kNN regression and kernel Regression were shown to be adaptive to the manifold structure that they converge at **rates that depend only on the intrinsic dimensions** of data (Kpotufe, 2009a,b, 2011; Kpotufe and Garg, 2013; Kpotufe and Verma, 2017).

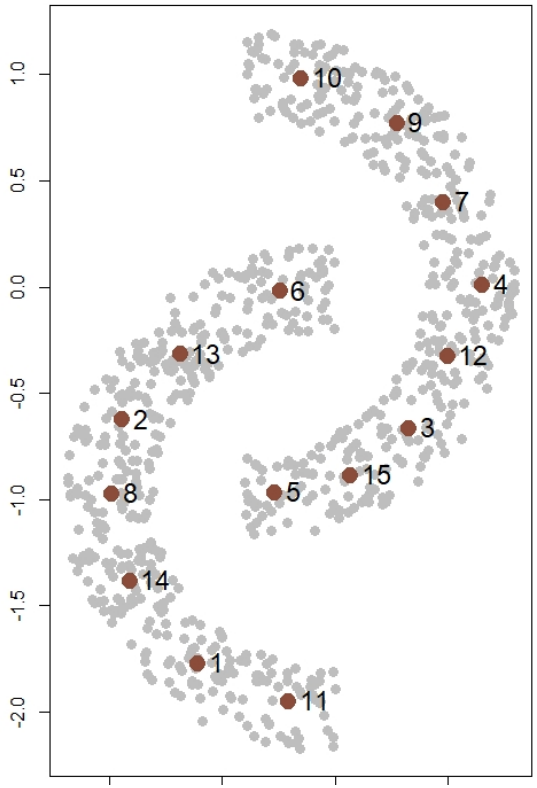
Skeleton Construction

Knots Construction

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



(a) Data



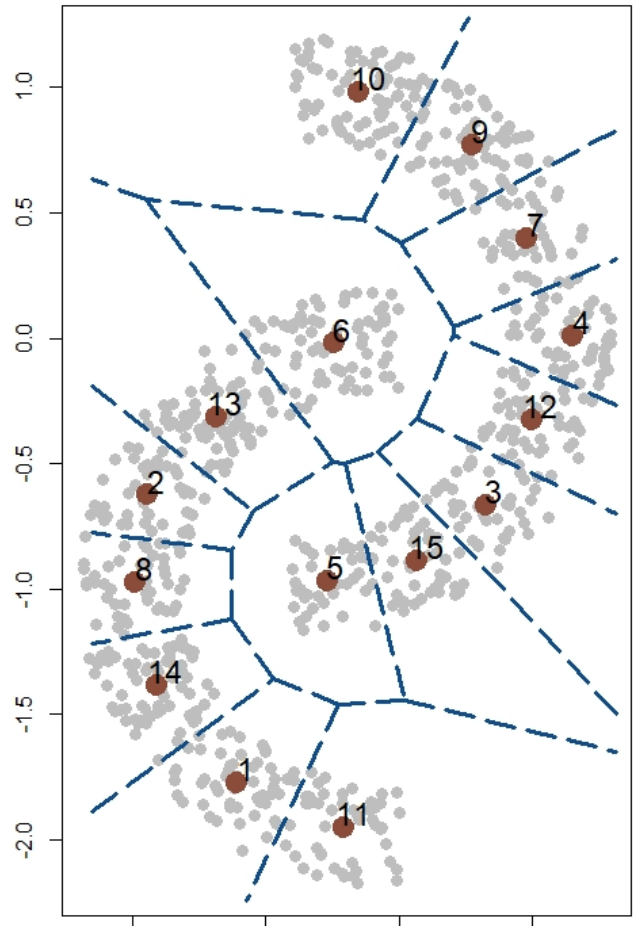
(b) Knots

Edge Construction, Voronoi Cells

The Voronoi cell (Voronoi, 1908), $\mathbb{C}_j$, 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 is,

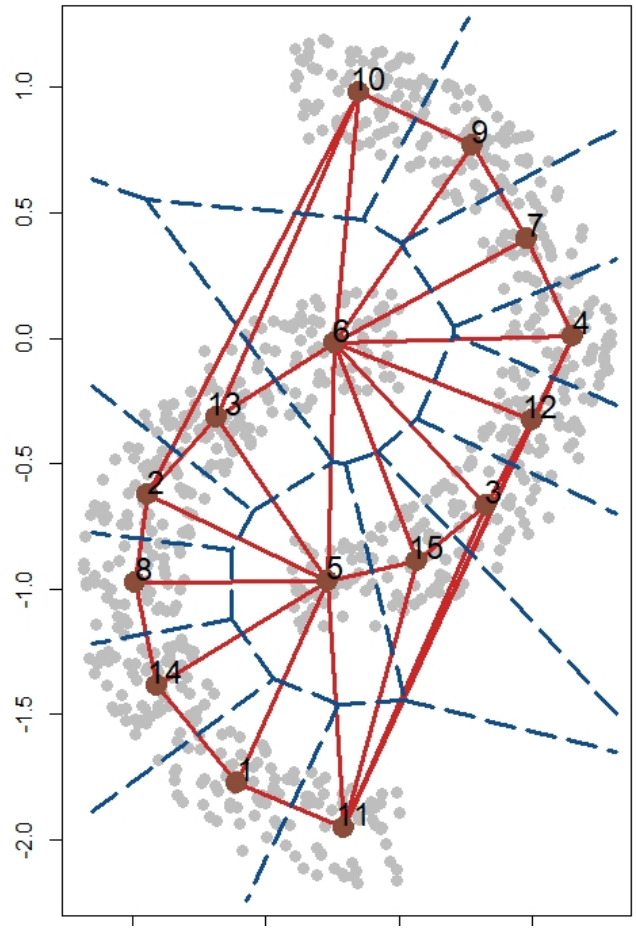
$$\mathbb{C}_j = \{x \in \mathcal{X} : d(x, c_j) \leq d(x, c_\ell) \ \ \forall l \neq j\},$$

where $d(x, y)$ is the usual Euclidean distance.



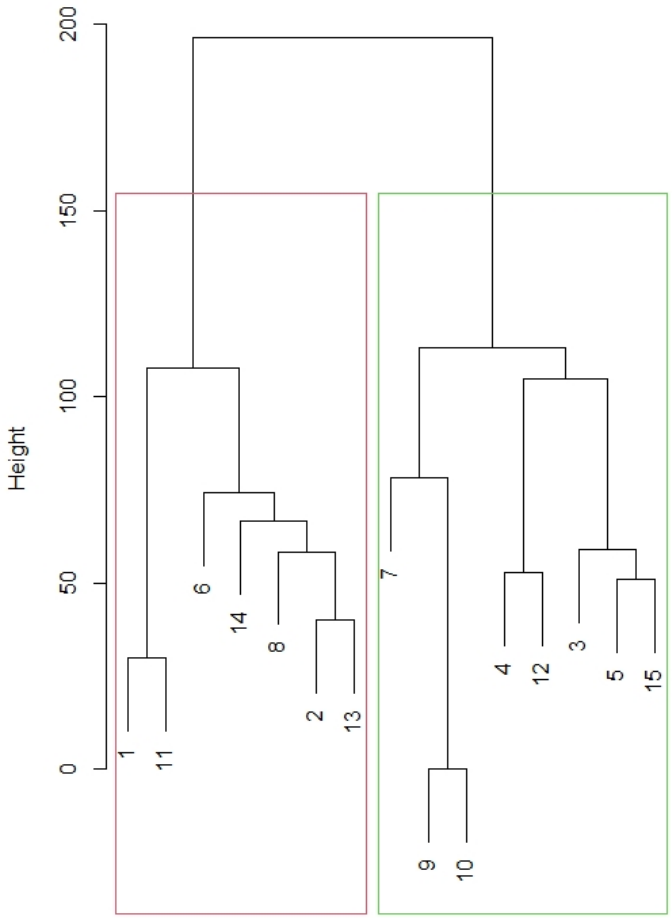
Edge Construction, Delaunay Triangulation

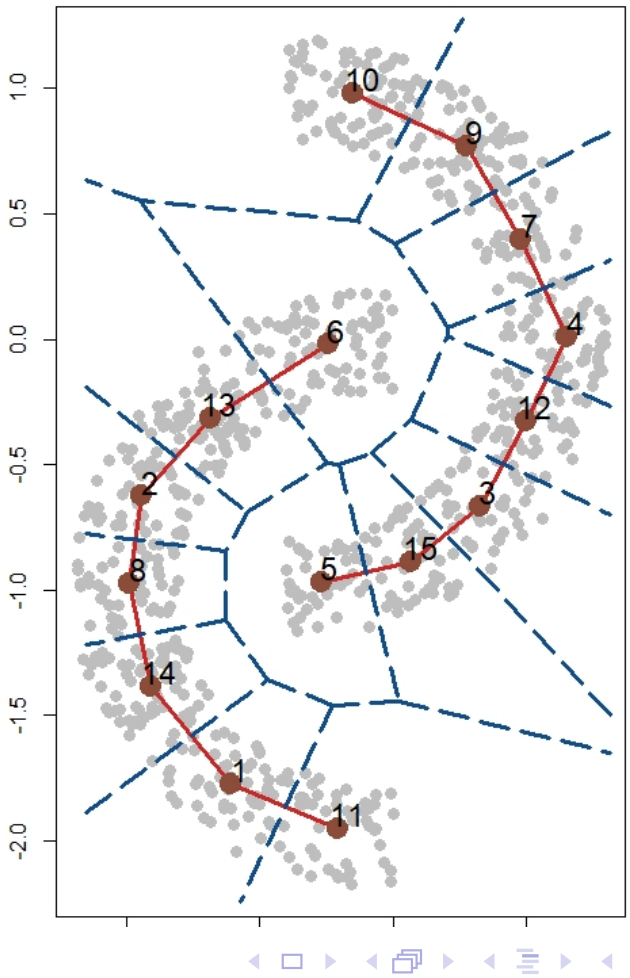
- Add an edge to a pair of knots if they are neighboring with each other. In other words, an edge between (c_i, c_j) is added if $\bar{\mathbb{C}}_i \cap \bar{\mathbb{C}}_j \neq \emptyset$.
- Resulting graph is the Delaunay triangulation $DT(\mathcal{C})$ (Delaunay, 1934) of knots $c_1, \cdots, c_k$



Skeleton Segmentation

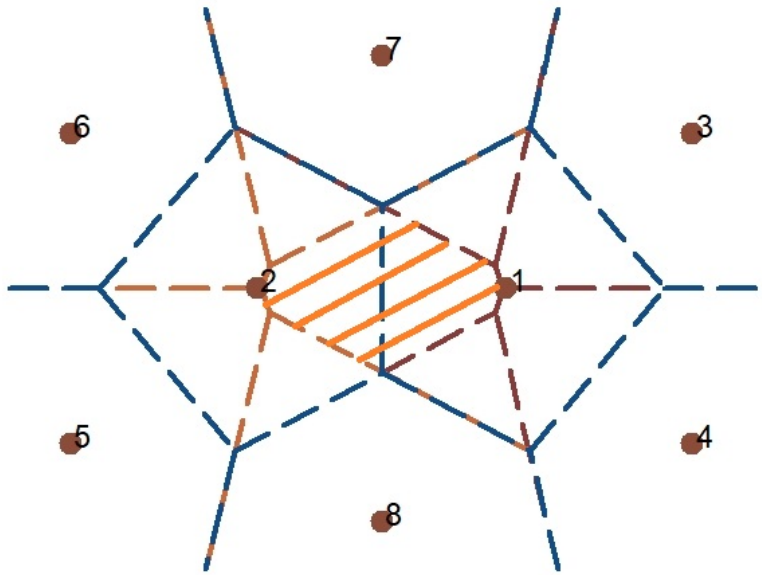
- Density-based weights are assigned to the edges.
- Use traditional clustering/segmentation methods such as the hierarchical clustering to segment the learnt skeleton structure.





Edge Weight: Voronoi Density

- Measures the similarity between knots (c_j, c_ℓ) based on the number of observations whose 2-nearest knots are c_j and c_ℓ.
- 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 $S_{j\ell}^{VD} = \frac{\mathbb{P}(A_{j\ell})}{\|c_j - c_\ell\|}$.

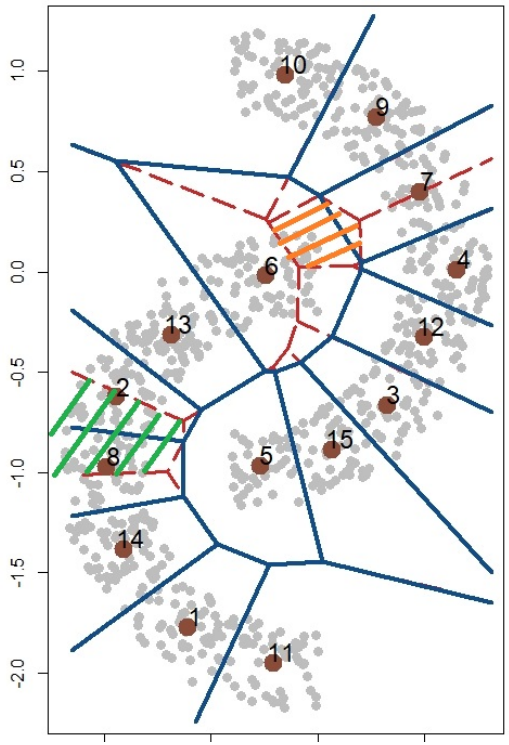


Edge Weight: Voronoi Density Estimation

- 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

$$\hat{S}_{j\ell}^{VD} = \frac{\hat{P}_n(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{1}$$

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



Data Projection

Data Projection

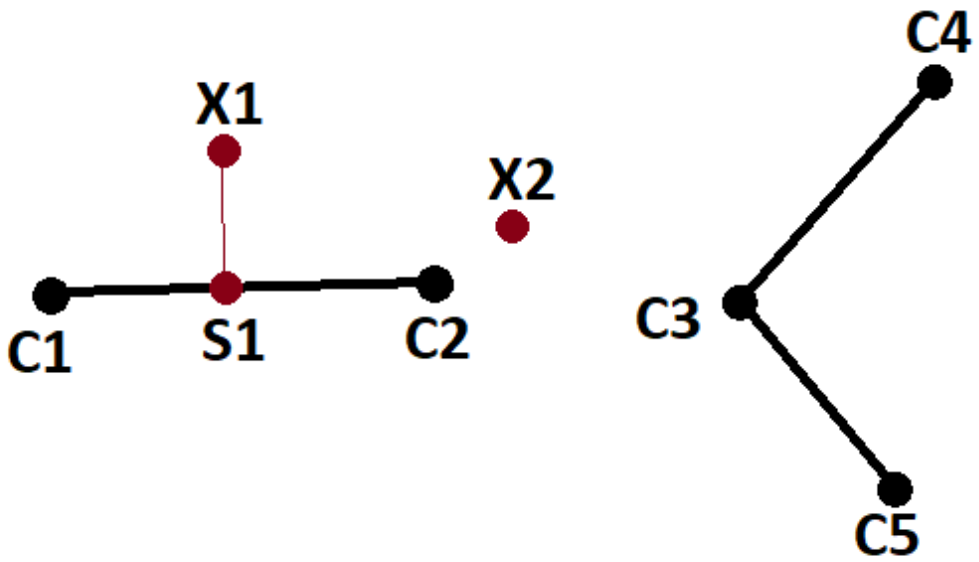


Figure: Illustration of projection to the skeleton. The skeleton structure is given by the black dots and black lines. Data point X_1 is projected to S_1 on the edge between C_1 and C_2. Data point X_2 is projected to knot C_2 as it's two closest neighbors C_2 and C_3 are not connected by an edge in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

Skeleton-Based Distance

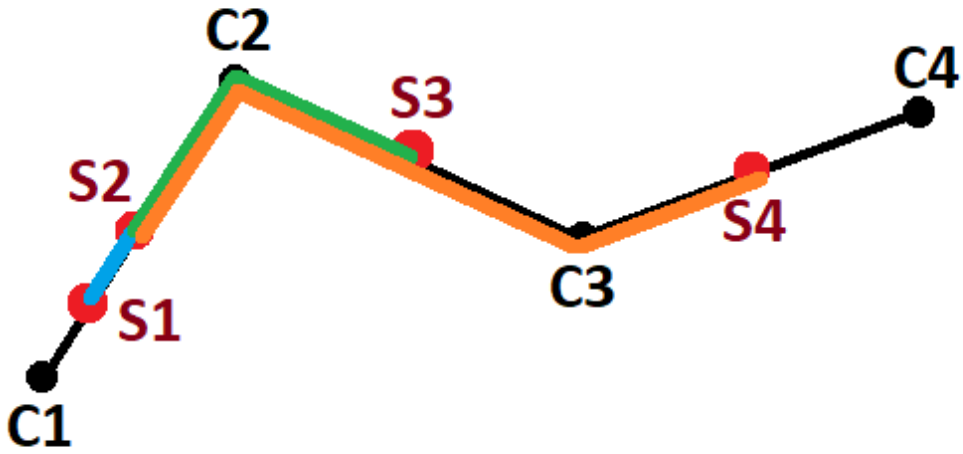


Figure: Illustration of skeleton-based distance. Let C_1, C_2, C_3, C_4 be the knots, and let S_2, S_3, S_4 be the mid-point on the edges E_{12}, E_{23}, E_{34} respectively. Let S_1 bet the midpoint between C_1 and S_2 on the edge. Let $d_{ij} = \|C_i - C_j\|$ denotes the length of the edge E_{ij}.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

Regression Model Fitting

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

$$\hat{m}(\boldsymbol{s}) = \frac{\sum_{j=1}^{N} K_h(d_{\mathcal{S}}(\Pi(\boldsymbol{x}_j), \boldsymbol{s}))\, Y_j}{\sum_{j=1}^{N} K_h(d_{\mathcal{S}}(\Pi(\boldsymbol{x}_j), \boldsymbol{s}))} \tag{2}$$

A concrete kernel function example is the popular Gaussian kernel that

$$K_h(d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell)) = \exp\left(- \frac{d_{\mathcal{S}}(\boldsymbol{s}_j, \boldsymbol{s}_\ell)^2}{h^2} \right) \tag{3}$$

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.

Linear Spline Regression on Graph

Construct a linear regression model on each edge of the graph 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 $\boldsymbol{Z}$, where $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T$ is a $n \times v$ matrix and $\boldsymbol{z}_j$ is the length v transformed data vector for $\boldsymbol{x}_j$ encoding proportional weights on the corresponding vertices.

$$\hat{y}_j + p_j^i(\hat{y}_\ell - \hat{y}_j) = (1 - p_j^i)\hat{y}_j + p_j^i \hat{y}_\ell = \boldsymbol{z}_i^T \hat{\boldsymbol{y}} \tag{4}$$

The S-Lspline model in matrix form can be written as

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{Z}) = \boldsymbol{\beta}^T \boldsymbol{Z} \tag{5}$$

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

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T \boldsymbol{Z})^{-1} \boldsymbol{Z} \boldsymbol{y} \tag{6}$$

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

Empirical Results

Yinyang Data

The input space 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, and two small circles each with 200 points
fit trigonometric function on the ring and constant function on the other structures with random Gaussian error
add iid random $N(0, 0.1)$ variables to the input space to increase the dimension of the input space to a total of 100 dimensions.

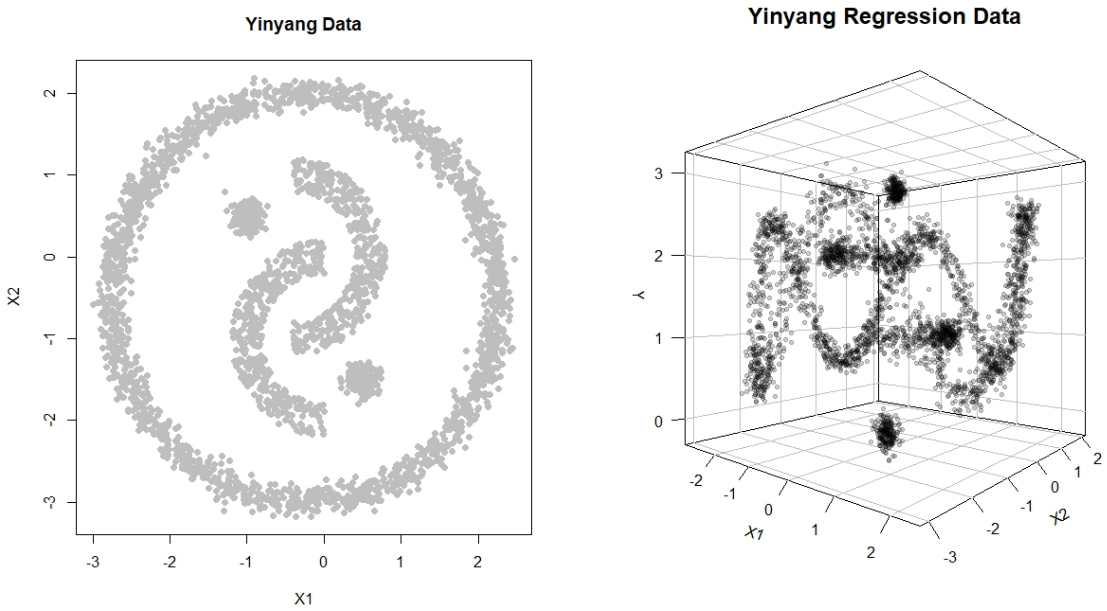


Figure: Yinyang Regression Data

Yinyang Data

We use 5-fold cross-validation to calculate the sum of squared errors (SSE)

- skeleton-based kernel regression (S-kernel) with varying bandwidths
- Skeleton linear model(S-Lspline) and higher-order splines (S-Qspline, and S-Cspline) fitted according to the proposed parametrization
- Euclidean-distance based k-Nearest-Neighbors (kNN) regressor and skeleton-distance based kNN (S-kNN) with varying numbers of neighbors
- Lasso and Ridge regressions are also fitted with hyper-parameter tuning.

Method	SSE	Number of knots	Parameter
kNN	80.58	-	neighbor=21
Ridge	1359.62	-	$\lambda = 0.001$
Lasso	1351.70	-	$\lambda = 0.0025$
S-Kernel	77.77	63	bandwidth $= 8\ h_{hns}$
S-kNN	85.48	76	neighbor $= 36$
S-Lspline	67.98	51	-
S-Qspline	70.57	51	-
S-Cspline	73.18	38	-

Table: Regression results on Yinyang $d = 100$ data. The best SSE from each method is listed with the corresponding parameters.

Lucky Cat Data

Set of 72 gray-scale images of size 128×128 pixels, each to be 2D projections of a 3D lucky cat obtained through rotating the object by 72 equispaced angles on a single axis.

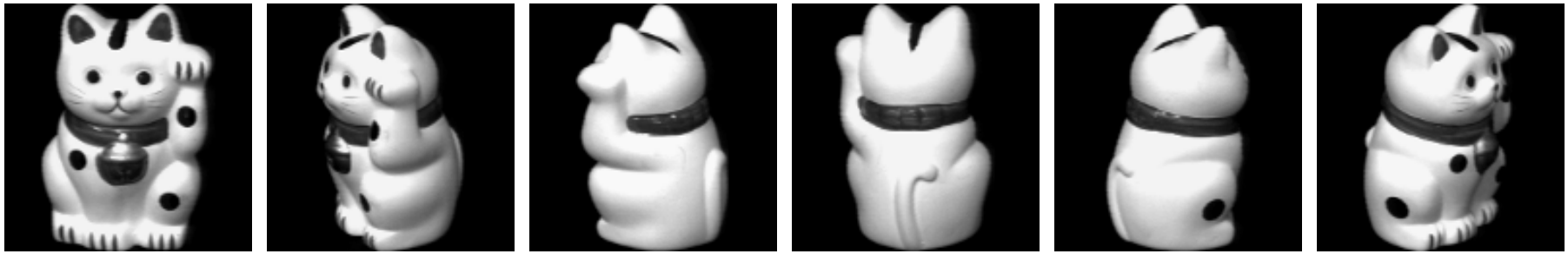


Figure: A part of the lucky cat images from the COIL-20 processed dataset. Each image is of size 128 pixels.

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.

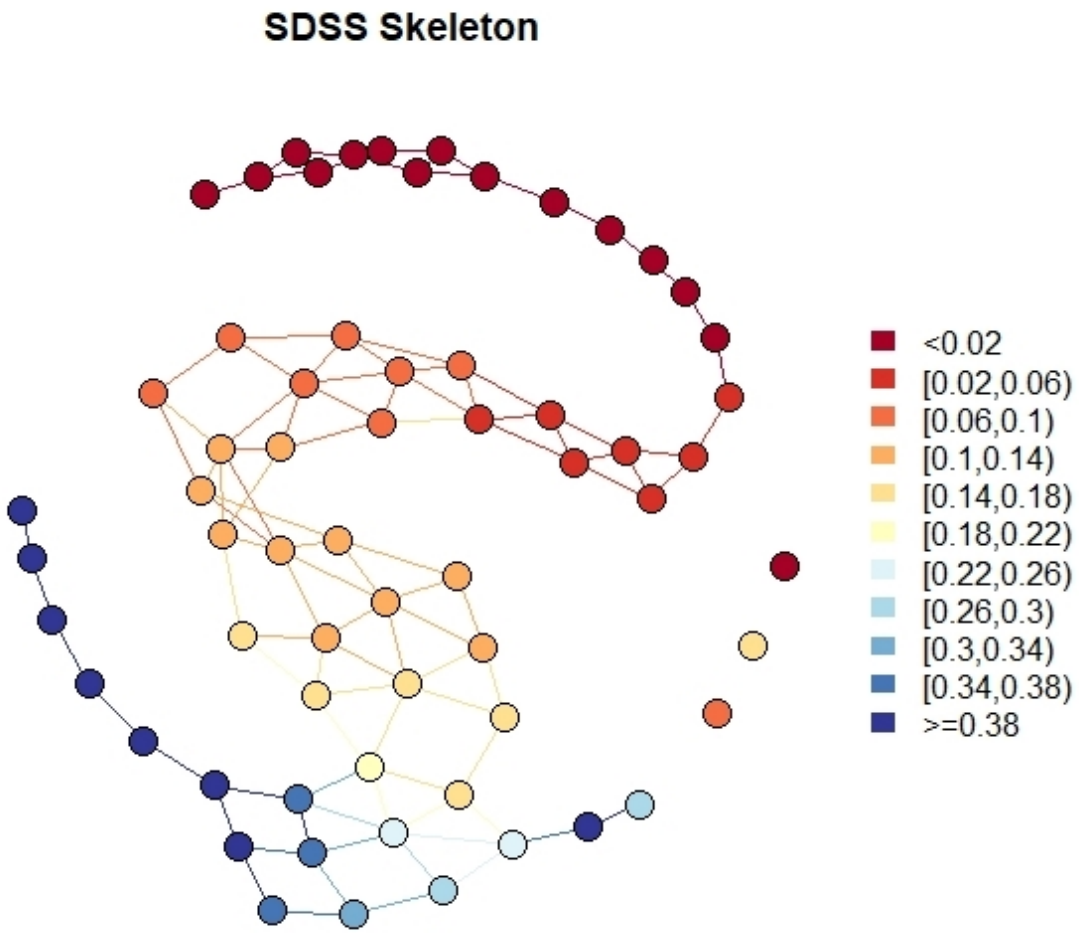
Lucky Cat Data Performance

Used the leave-one-out cross-validation scheme and same procedure as simulated dataset

Method	SSE	Parameter
Knn	888.89	neighbor=9
S-Kernel	1753.43	bandwidth $= 4r_{hns}$
S-kNN	2604.17	enighbor $= 6$
S-Lspline	338.12	-
S-Qspline	2143.47	-
S-Cspline	9449425341	-

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

SDSS Data



Sloan Digital Sky Survey (SDSS) data with 5 covariates measuring apparent magnitude of stars from images taken using 5 photometric filters. Response is the true redshift.

Conclusion

Contribution:

- Skeleton to represent the geometry of data and assist in various data analysis tasks
- Apply nonparametric regression techniques on graphs (kernel regression, splines, kNN)
- Discuss the feasibility of higher-order splines on graphs and the issue of edge directions
- Empirical results demonstrating the usefulness of our framework

Some possible future directions:

- Relate skeleton construction to Persistent Homology
- Other regression approaches applied on graph representations
- Theoretical analysis of the constructed skeleton
- Longitudinal data/ online updating of the skeleton representation

Thanks for listening!

Reference I

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.

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.

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.

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. 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.

Reference II

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`.

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`.

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 III

B. Schölkopf and A. J. Smola. Learning with kernels: Support vector machines, regularization, optimization, and beyond adaptive computation and machine learning. page 626, 2002.

G. Voronoi. Recherches sur les paralléloèdres primitives. *J. reine angew. Math*, 134: 198–287, 1908.

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.

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.