Applications of Diffusion Maps

Aneesh Malhotra, Orton Babb

Introduction

Let $A = \{x_i\}_{i=1}^N$ be a set of data in $\mathbb{R}^n$. We can define a *kernel* function by $k : A \times A \to \mathbb{R}$ by

$$k(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|}{\epsilon}\right).$$

If we define the matrix $K_{ij} = k(x_i, x_j)$, and the diagonal matrix $D_{i,i} = \sum_{j=i}^{N} K(x_i, x_j)$, a result by Coifman and Lafon [1] says that the Laplacian of the manifold approximated by the data is given by

$$L = \frac{D - K}{\epsilon}.$$

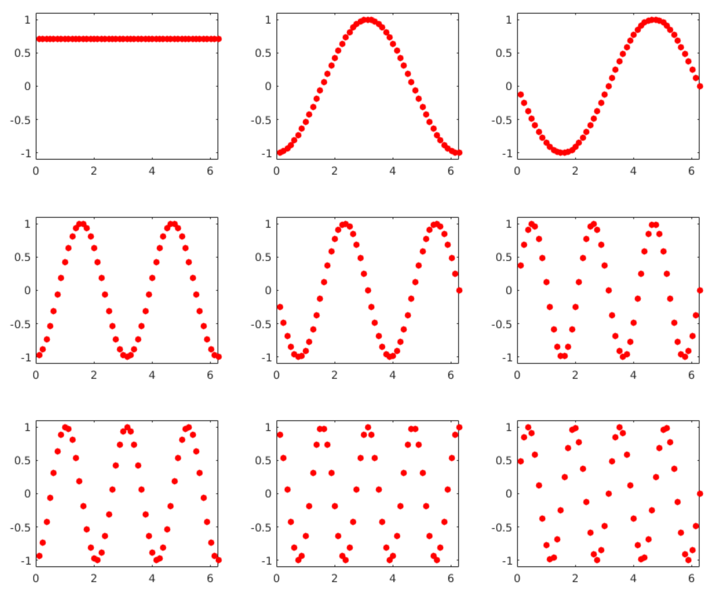
The Laplacian describes the geometry of the data.

Examples

Lets consider a set of data on a circle. We know that the Laplace operator is

$$\Delta = \frac{d^2}{d\theta^2} \qquad \phi_k(\theta) = c_1 \cos(k\theta + c_2)$$

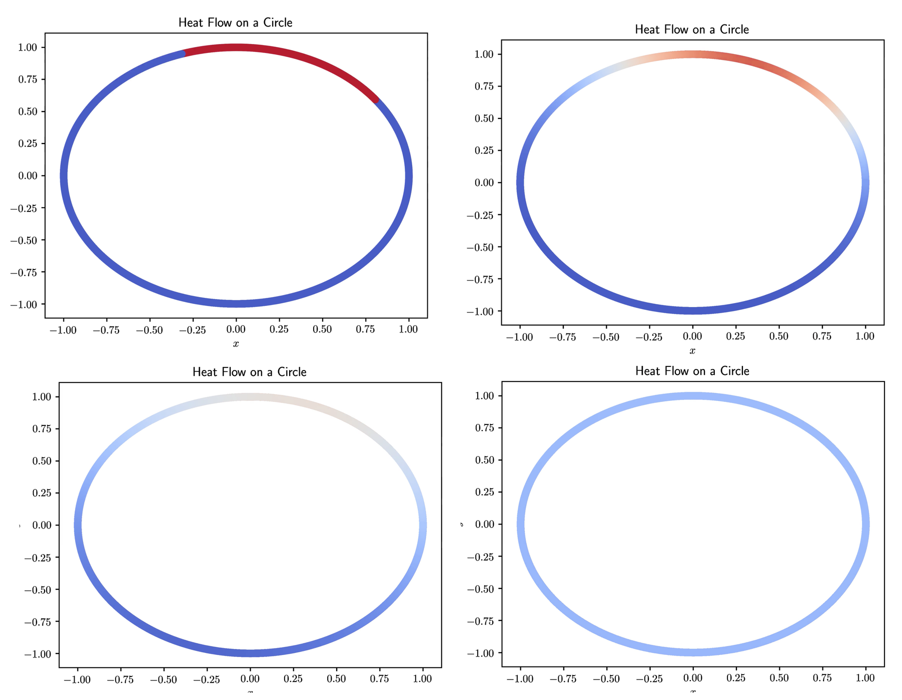
Where $\Delta\phi_k = -k^2\phi_k$ are the eigenfunctions.



The eigenfunctions of the Laplacian are approximated well using Diffusion Maps

Heat Equation

The heat equation in $\mathbb{R}^n$ is defined as the PDE $u_t = \Delta u$, where Δ is the Laplace operator. We solve this PDE numerically using a finite difference method $u(t + \tau) = u(t) + \tau\Delta u(t)$. On a manifold we solve by using the Laplacian approximation from Diffusion Maps. Below is a heat flow on a circle.

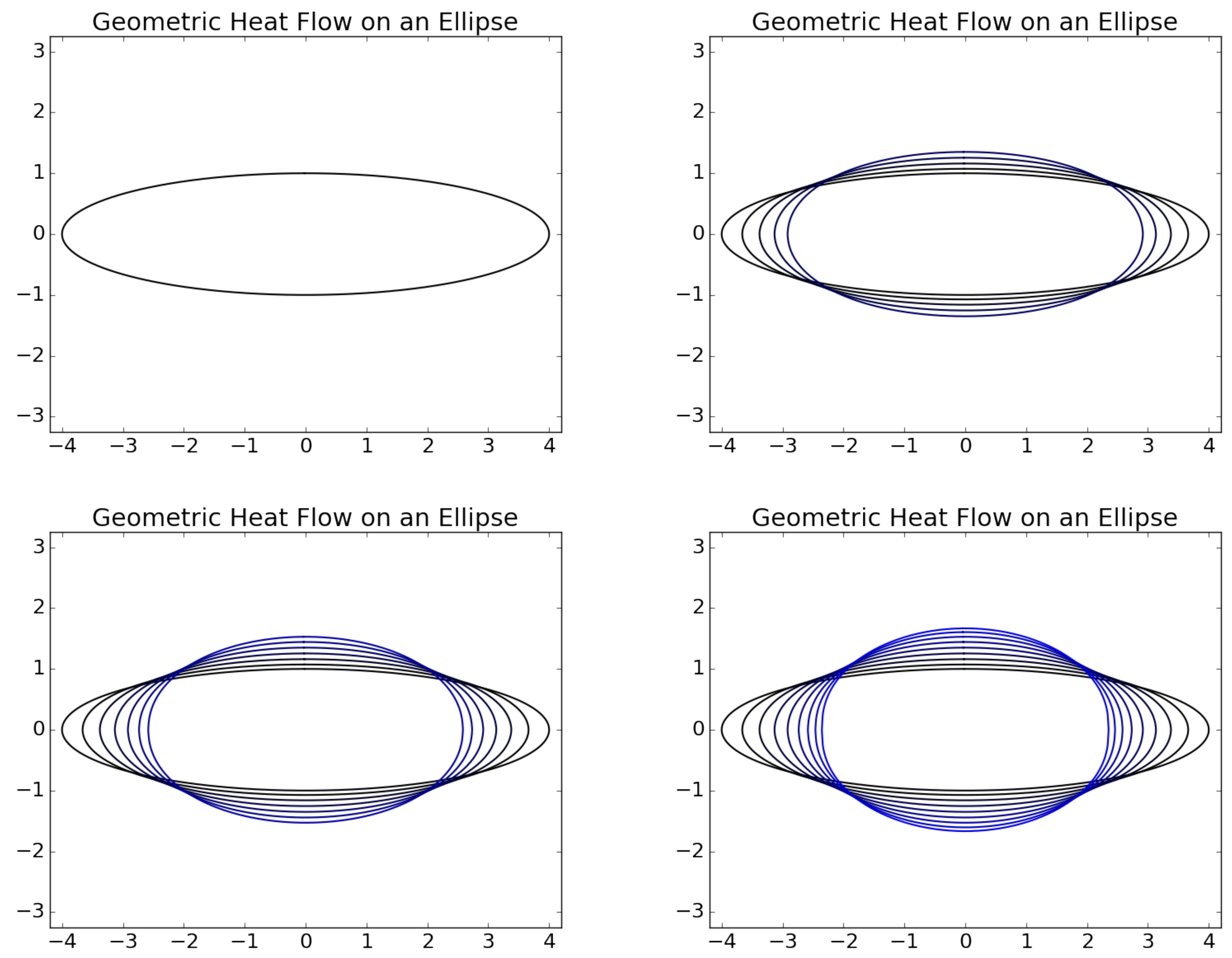


Geometric Heat Flow

Instead of solving the heat equation on functions of the dataset, we can solve the heat equation on the data itself by using the embedding functions of the data. This is summarized in the following steps:

1. Data is embedded into $\mathbb{R}^n$ using embedding functions $(f_1 \ldots f_n)$.
2. We compute Δ on the data and run a heat flow on the embedding functions for a time step dt.
3. Now we have a new dataset. We recompute Δ and repeat the heat flow.

We expect that this kind of flow will give us a dataset with more regularity. Moreover, the circle should be a fixed point of this flow in $\mathbb{R}^2$



Singularities

When running this flow for a long time, we see that perturbations due to numerical error result in clustering of the points, and therefore a singular Kernel matrix:

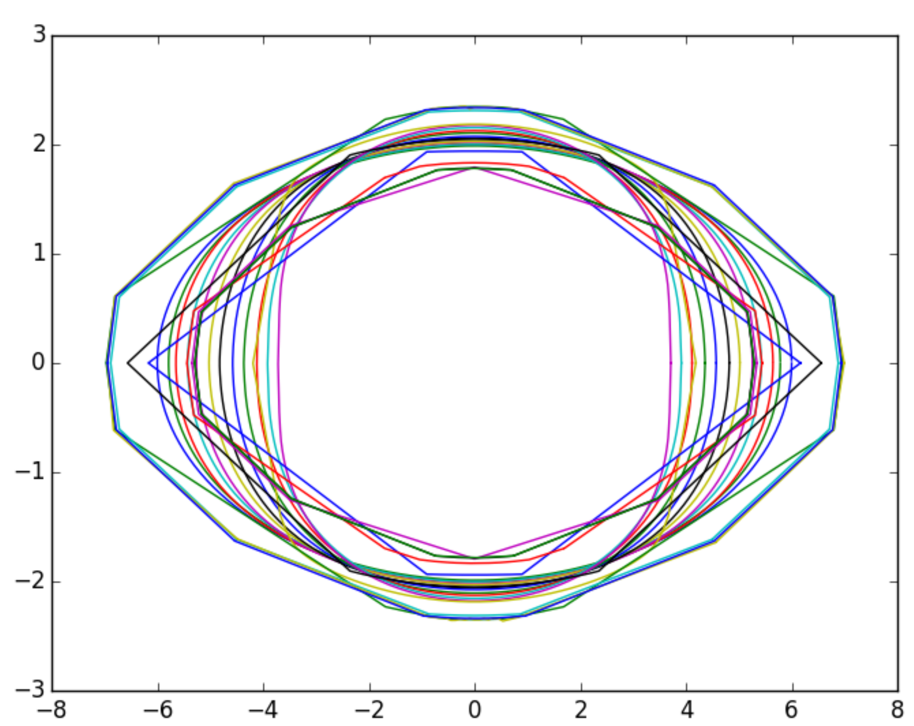
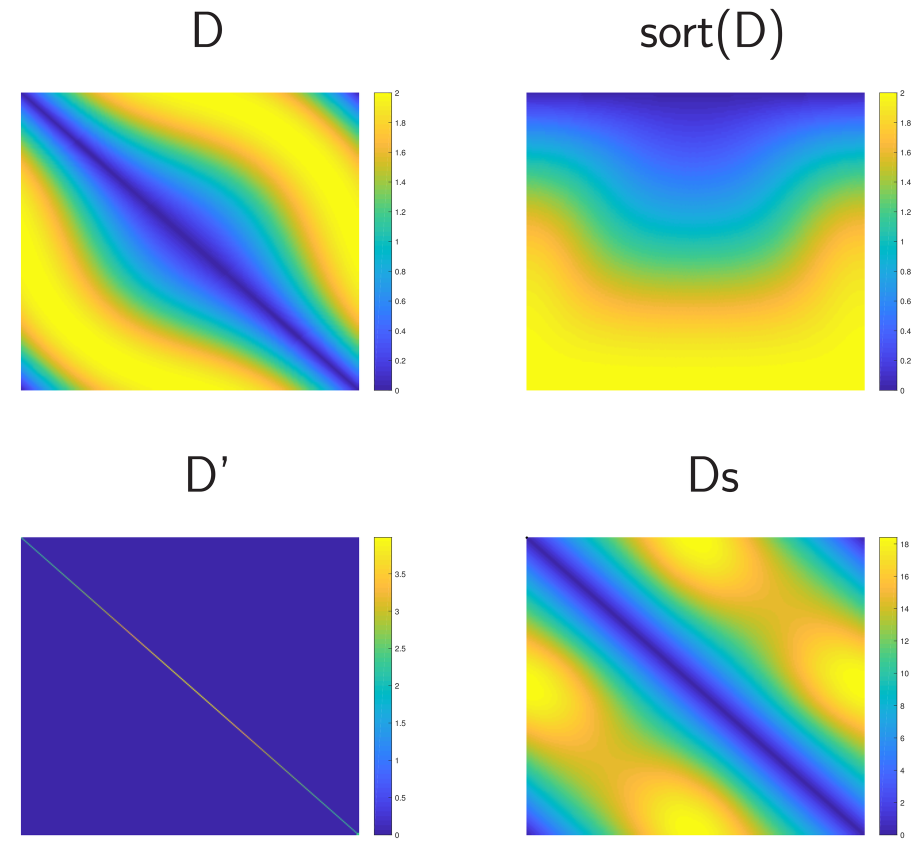


Figure: The geometric flow crashes due to instability

Resampling

In order to address this the singularity, we attempted to resample points uniformly across the manifold with each step of the geometric flow. We thought we could a normalization of the data to its k nearest neighbors.

1. Build the distance/kernel matrix
2. Perform normaliziation
3. Apply inverse to retrieve distance matrix from K_X
4. Perform MDS to recover center version of the points.

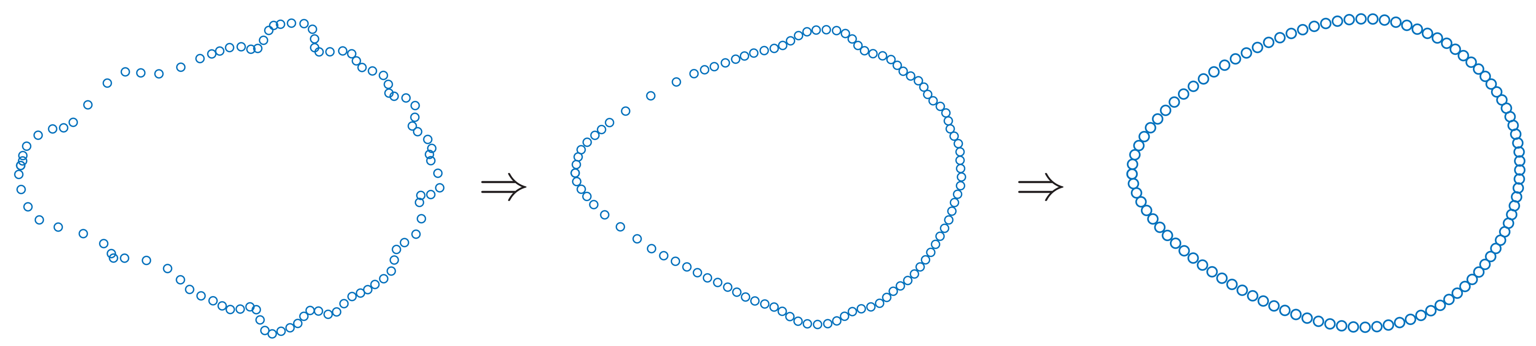


This process, however, distorts the geometry of the data.

Dimensionality Reduction

We can also use diffusion maps to do a nonlinear dimensionality reduction. The benefit of this as opposed to linear techniques is that the reduced data is isometric to the original data. In order to do this we did a gradient descent, where we minimized the objective function:

$$\tilde{C} = \text{argmin}_C \|K_X - K_{\tilde{X}}(C)\|_{fro}$$



Progressive improvements in reconstruction for $m = 2$

Acknowledgements

We would like to thank the Mason Experimental Geometry Lab, Tyrus Berry, and the math department.

References

[1] Coifman, R.R., S. Lafon. (2006). "Diffusion maps". Applied and Computational Harmonic Analysis. 21: 530. doi:10.1016/j.acha.2006.04.006

Department of Mathematics, George Mason University, Fairfax, VA 22030
amalhot4@gmu.edu, obabb@gmu.edu