Point Cloud Data Recovery with Diffusion Map

Chuxiangbo Wang

December 10, 2023

Abstract

This project presents a study on the application of diffusion maps in the recovery of point cloud data, especially focusing on datasets that after non-linear transformations. The problem addressed is the challenge in recovering the original structure of data that has been transformed. The technique being used is the diffusion map algorithm, which involves constructing a weight matrix and graph Laplacian to reveal the intrinsic geometry of the data. The project conducts a detailed analysis using datasets with distinct clustering, particularly a half-moon dataset, to examine the behavior of eigenvalues in the diffusion map and their influence on data recovery. A key part of the research involves applying a non-linear transformation to a square-shaped pointcloud dataset, transforming it into a mushroom shape, and then employing diffusion maps to recover the original structure. The results demonstrate the effectiveness of diffusion maps in accurately recovering the original data structure, highlighting the importance of parameter selection in the process. The Project providing insights into the use of diffusion maps for complex data structure analysis and recovery in point cloud datasets.

All the code utilized in this project are accessible at this link.

1 Diffusion Map

Diffusion maps represent an advanced technique in the field of data analysis, particularly effective in uncovering the intrinsic geometry of high-dimensional data. This method belongs to the family of dimensionality reduction techniques, aiming to simplify the complexity of large datasets while preserving their essential features.

1.1 Definition of Diffusion Map

The diffusion map algorithm begins with a set of data points $\{x_i\}_{i=1}^N$. It involves constructing a weight matrix W, where each element W_{ij} encodes the similarity between data points x_i and x_j. This similarity is typically defined by a Gaussian kernel:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\epsilon}\right)$$

where ϵ is a scaling parameter controlling the influence of neighboring points.

A degree matrix D is then formed, which is a diagonal matrix defined as follow:

$$D_{ii} = \sum_j W_{ij}$$

The graph Laplacian L, defined as $L = D^{-1}W$, is crucial in this framework. The eigenvectors and eigenvalues of L are computed to form the diffusion map:

$$\vec{x}_i \to \begin{pmatrix} \lambda_1^t \psi_1(i) \\ \lambda_2^t \psi_2(i) \\ \vdots \\ \lambda_n^t \psi_n(i) \end{pmatrix}$$

Here, λ_i are the eigenvalues, $\psi_i(i)$ are the eigenvectors, and t represents the diffusion time, capturing the multiscale geometry of the data.

1.2 Motivation

A critical aspect of the diffusion map method revolves around the eigenvalues and eigenvectors derived from the graph Laplacian. Typically, the first eigenvalue λ_1 in a diffusion map is always 1, representing the steady-state distribution of the system. In practice, this first eigenvalue is often omitted in the analysis, as it generally does not provide meaningful insights into the underlying data structure. Consequently, the diffusion map is usually expressed as:

$$\vec{x}_i \to \begin{pmatrix} \lambda_2^t \psi_2(i) \\ \lambda_3^t \psi_3(i) \\ \vdots \\ \lambda_{n+1}^t \psi_{n+1}(i) \end{pmatrix}$$

However, intriguing questions arise when considering eigenvalues other than λ_1 that are also close to 1, such as λ_2, λ_3, or even λ_4. What implications do these eigenvalues have on the data representation and the insights that can be derived from it? Moreover, what specific characteristics of a dataset and its diffusion map setup might lead to such a scenario where subsequent eigenvalues are very close to 1?

These question is interesting because eigenvalues near 1 can significantly influence the diffusion map's ability to model and reveal the data's intrinsic geometry. They may suggest nearly disconnected components within the dataset or indicate possibly a separation in the manifold's structure. Understanding the circumstances that give rise to these scenarios is essential for interpreting the diffusion map's results accurately and for exploiting this technique's full potential in data analysis.

2 Analysis of Diffusion Maps with Clustered Data

The phenomena of having eigenvalues close to 1, other than the first eigenvalue, we assume it will occur in datasets with distinct clustering. In such scenarios, each cluster can be viewed as a separate entity, where transitioning from one cluster to another is relatively rare or difficult. This clustering effect in the data leads to eigenvalues that are close to 1, reflecting the isolated nature of these clusters within the overall data structure.

To exam this assumption, we constructed a dataset that visually represents this scenario. The dataset is designed with separate clustering, where each cluster exhibits a round Gaussian distribution with the same covariance matrix. This setup mimics the situation where within each cluster, the data points are tightly grouped, and there is a clear separation between different clusters. The following image illustrates the half-moon dataset with distinct clustering:

This visual representation serves as a basis for our subsequent analysis, where we apply the diffusion map technique to understand how it captures and represents these distinct clusters and investigates the behavior of eigenvalues in such a context.

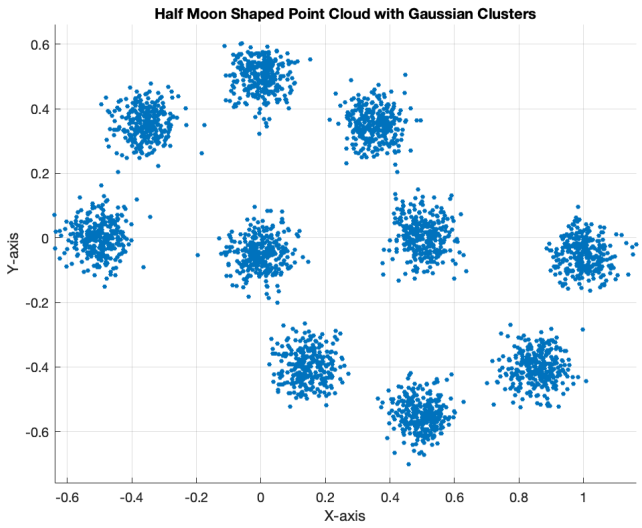


Figure 1: Half Moon Pointcloud Dataset

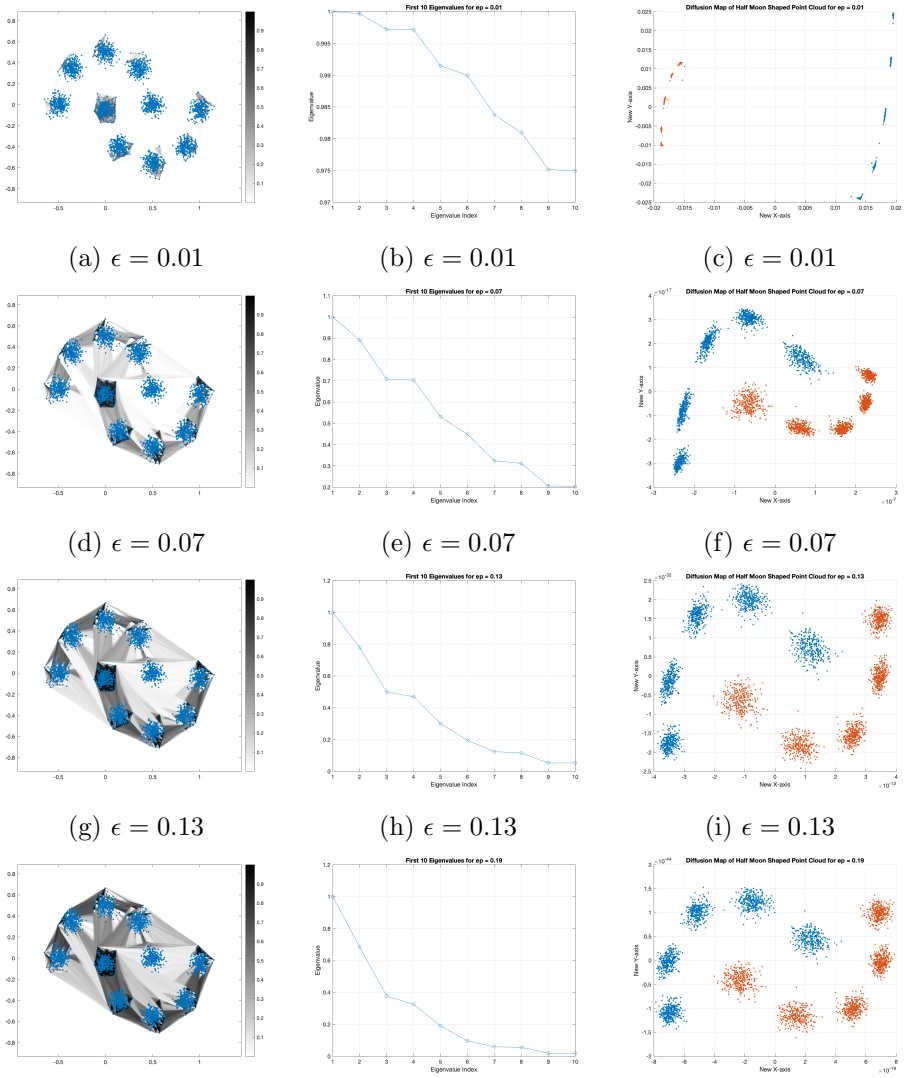
Upon applying the diffusion map algorithm to the half-moon dataset, we explored the impact of varying the epsilon value. The epsilon parameter in the Gaussian kernel significantly influences the diffusion map's ability to capture the data's geometry. Therefore, analyzing the eigenvalues and the first two principal components of the diffusion map for different epsilon values provides insights into how this parameter affects the data representation.

The figures below display the results of this analysis. Each row corresponds to a different epsilon value used in the diffusion map. The first image in each row shows a plot of the distance graph, while the subsequent two images illustrate the eigenvalues and the first two principal components derived from the diffusion map. These visualizations offer a comprehensive view of how changing epsilon values alter the eigenvalues and principal components, thereby affecting the overall structure captured by the diffusion map.

With a very small epsilon, the eigenvalues tend to be very close to 1(Figure 2(b)), reflecting a scenario where the diffusion map captures predominantly local transitions within one cluster. These clusters behave almost like isolated systems, leading to several 'near-steady-state' conditions within the local neighborhoods. Furthermore,with small epsilon, the two principal component plots do not accurately represent the global structure of the data (figure 2(c)).

In contrast, increasing the epsilon value allows for a more interconnected graph structure. which gives a more holistic view of the original data (Figure 2(l)) , leading to a diffusion map that better captures the inherent geometry of the data.

In summary, the epsilon parameter in diffusion maps plays a pivotal role in determining the graph's connectivity and, by extension, the diffusion map's ability to accurately represent the data's underlying structure. The choice of epsilon thus becomes a balancing act between capturing local versus global structures within the data.



(a) $\epsilon = 0.01$ (b) $\epsilon = 0.01$ (c) $\epsilon = 0.01$

(d) $\epsilon = 0.07$ (e) $\epsilon = 0.07$ (f) $\epsilon = 0.07$

(g) $\epsilon = 0.13$ (h) $\epsilon = 0.13$ (i) $\epsilon = 0.13$

(j) $\epsilon = 0.19$ (k) $\epsilon = 0.19$ (l) $\epsilon = 0.19$

Figure 2: Distance Graph(Left), Eigenvalues(Middle), First Two Principal component(Right)

3 Data Recovery with Diffusion Map

3.1 Introduction

The ability of diffusion maps extends beyond capturing the intrinsic geometry of datasets; they are also potent in recovering data structures after transformations. To illustrate this ability, consider a dataset that, after undergoing a nonlinear transformation, adopts a distinct mushroom shape. The objective here is to demonstrate how diffusion maps can effectively reverse this transformation, guided by the principles outlined in Amit Singer's work on "Non-linear ICA with Diffusion Maps."

3.2 Nonlinear Transformation and Recovery

Assuming our original dataset forms a square-shape, we apply a nonlinear transformation

$$y_1 = x_1 + x_2^3$$

$$y_2 = x_2 - x_1^3$$

that transform the squared data into a mushroom-shaped data (Figure 3). Compare with regular linear trasnformation (shifting, rotating), this transformation serves as a more complicated transform of the data, posing a challenge for recovery using traditional linear methods.

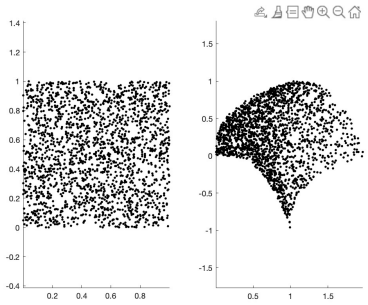
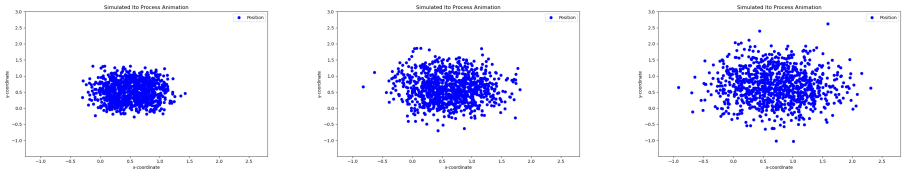


Figure 3: Original Data(Left) and Transformed Data(Right)

In the scenario presented, our primary dataset (the square-shaped data) is unknown, and the observable dataset is the mushroom-shaped transformed data. The goal is to utilize diffusion maps to recover the square-shaped structure from the mushroom-shaped data.

To perform the recovry, we first simulate an ito process our unknown dataset to create a reasonable number of bursts in a small time interval (Figure 4). while simulate the ito process, we can observe the change in the observable data set. With this process, we can construct a covariance matrix for each point in the observable manifold with the transformed ito burst points that corresponding to it.



(a) (b) (c)

Figure 4: Ito Process Simulation

Following the method proposed by Amit Singer, we employ a formula

$$\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|^2 = \frac{\delta^2}{2d+2}(\mathbf{y}^{(j)} - \mathbf{y}^{(i)})^T \left[C_i^\dagger + C_j^\dagger\right](\mathbf{y}^{(j)} - \mathbf{y}^{(i)}) + O(\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|^4).$$

as the foundational concept for this recovery process.

The main idea of this equation is that the euclidean distance between points in the unknown data set $\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|^2$ can be approximated by $\frac{\delta^2}{2d+2}(\mathbf{y}^{(j)} - \mathbf{y}^{(i)})^T \left[C_i^\dagger + C_j^\dagger\right](\mathbf{y}^{(j)} - \mathbf{y}^{(i)}) + O(\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|^4)$. Where d is the dimension of the data and δ is the parameter that $\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\| \leq \delta$

This formula encapsulates the essence of the diffusion map's ability to reverse nonlinear transformations by identifying and leveraging the intrinsic geometric properties of the data.

3.3 Methodology and Results

Once we have successfully approximated the Euclidean distances between points in our origional dataset, we can use this approximate distance to construct the weight matrix:

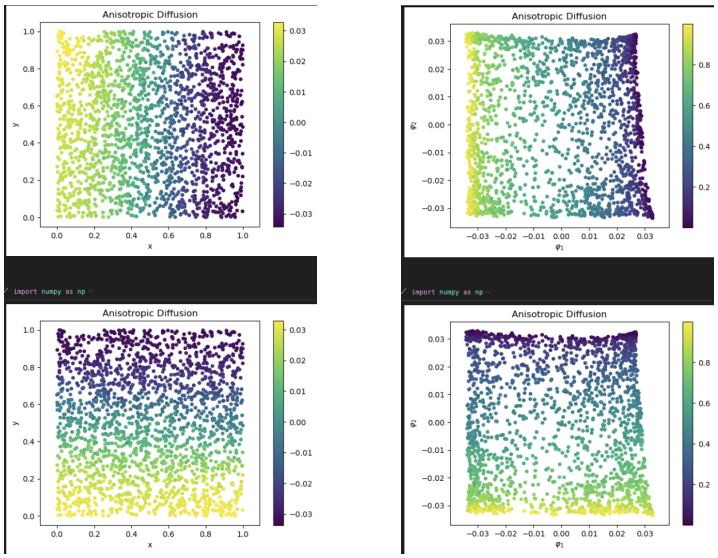
$$W_{ij} = \exp\left\{-\frac{\delta^2}{d+2}(\mathbf{y}^{(j)} - \mathbf{y}^{(i)})^T \left[\mathbf{C}^{\dagger}_{i,\delta} + \mathbf{C}^{\dagger}_{j,\delta}\right] (\mathbf{y}^{(j)} - \mathbf{y}^{(i)}) \frac{1}{4\epsilon}\right\}.$$

and then compute our graph Laplacian:

$$L = D_{\text{inv}} \cdot W$$

where $D = \text{diag}\left(\sum_{j=1}^{P} W_{i,j}\right)$.
With the graph Laplacian in place, we then computing its eigenvalues and eigenvectors to form the diffusion map.

In Figure 5, we illustrate this process. We plot the second and third eigenvectors and color-code them according to the original data. It reveals that the data recovered from these eigenvectors not only replicates the original dataset's squared shape but also maintains a one-to-one correspondence with it. This outcome highlights the effectiveness of our method in capturing the essential features of the dataset using the diffusion map.



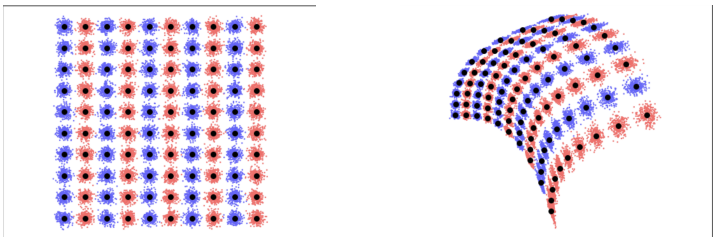
(a) Origional Data (b) Recovered Data

Figure 5: Data recovered with Dissusion Map

4 Point Cloud Data Recovery

Now, let's delve into our analysis with a focus on point cloud data. We've had a technique that successfully recovered data that is uniformly distributed using Diffusion Map. Our next goal is to merge these two aspects of our analysis by exploring whether the diffusion map can be used to effectively reconstruct point cloud data that has undergone a non-linear transformation.

In Figure 6, we present an illustrative example. Here, we start with point cloud data that is uniformly distributed. We then apply a transformation to this data, resulting in it taking on a distinct, mushroom-like shape.



(a) Original Poincloud Data

(b) Transformed Poincloud Data

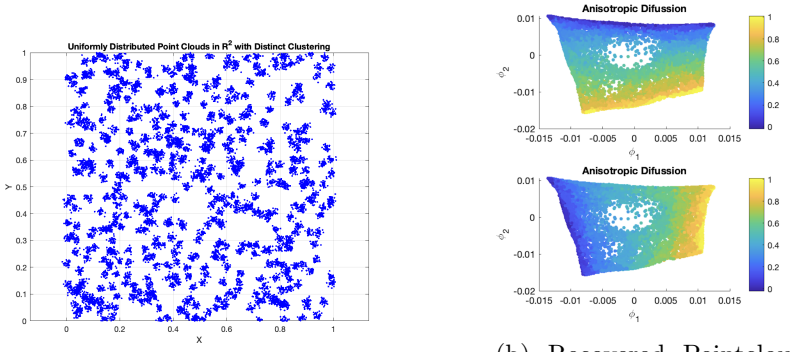
Figure 6: Original And Transformed Pointcloud Data

4.1 Non-linear Point Cloud Data Recovery

In Figure 5, the point cloud we see represents a uniform 10x10 grid. However, in real-world scenarios, generating data uniformly doesn't always result in such a precise grid structure.

In our next numerical example, we create a point cloud dataset by establishing 500 different centers, or 'means'. Around each of these means, we generate a set of points that follow a Gaussian distribution, with each set containing 20 points (Figure 7(a)).

Once we've created this dataset, we apply the Ito process and compute the graph Laplacian. Then, we plot the first two principal components to analyze the results. Interestingly, the data we recover retains the squared shape and has a one-to-one correspondence with the original dataset. However, it does not distinctly shows the Gaussian clusters we initially created (Figure 7(b)). I believe this happens because the Gaussian clusters in our dataset are quite densely packed. This density likely makes it challenging for the diffusion map to accurately capture and differentiate each cluster.



(a) Original Pointcloud Data

(b) Recovered Pointcloud Data

Figure 7: Numerical Example 1.

n our second point cloud dataset, we aim to modify our previous approach to enhance the diffusion map's ability to capture distinct clusters during data recovery. To achieve this, we adjust the dataset's structure.

This time, we reduce the number of means to 50. However, around each mean, we significantly increase the number of points, generating a Gaussian distribution with 300 points for each. This adjustment in the data generation process is designed to create more space between the clusters.From the visualization in the figure 8, we can observe a noticeable difference in the dataset's structure compared to our previous attempt. The clusters in this new dataset are more spread out and

separated from each other

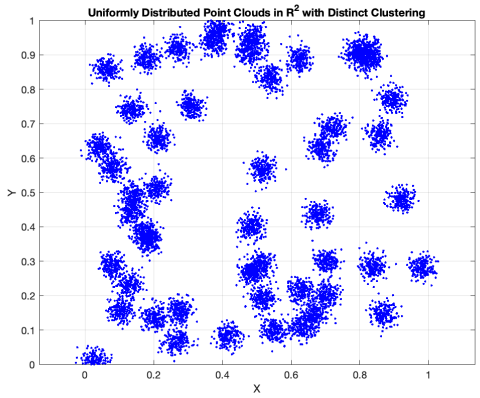
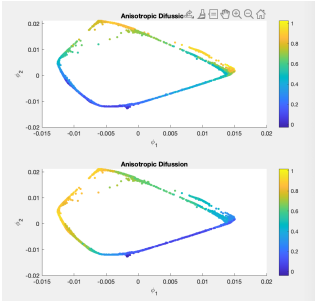


Figure 8: Test dataset 2

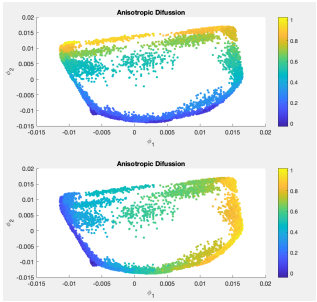
To begin the recovery process, we start with a very small value for epsilon, the parameter that influences the diffusion map's sensitivity. Gradually, we increase this value (as shown in the figure 9).

From these experiments, we notice that the recovered data primarily highlights the boundary of the original dataset. The second epsilon value (Figure 9(b)), in particular, seems to outline a more accurate boundary. For the first epsilon value (Figure 9(a)), we believe it captures the boundary shape while excluding some isolated Gaussian clusters that are further apart.

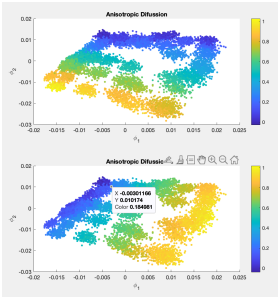
As we increase epsilon to a larger value (Figure 9(c)), the recovery results improve significantly. The data now clearly shows the separate clustering structure (as seen in another figure). By color-coding this recovered data to match the original dataset, we observe that each cluster corresponds one-to-one with the original, despite a slight apparent rotation. This one-to-one correspondence, even with the visual rotation, indicates that our recovery process is effective in reconstructing the distinct cluster formations from the original data.



(a) Very Small ϵ



(b) Small ϵ



(c) Larger ϵ

Figure 9: Numrical Experiment 2.

5 Conclusion

In conclusion, our exploration of the Diffusion Map technique has highlighted its capability in handling point cloud datasets, particularly in recovering those with well-defined clustering. This method stands out for its effectiveness in a variety of scenarios, showcasing its ability to capture the overall shape and structure of datasets, even when conditions are less than ideal. A critical factor in leveraging the full potential of Diffusion Maps is the careful selection of parameters. When these parameters are optimally chosen and certain conditions are met, the method can go beyond general approximations to achieve highly precise reconstructions of the point cloud structure. This versatility and accuracy make Diffusion Maps a valuable tool in the realms of data science and analytics, especially for tasks that involve complex data structures and require detailed cluster analysis.

References

[1] Amit Singer, Ronald R. Coifman, "Non-linear independent component analysis with diffusion maps," *Applied and Computational Harmonic Analysis*, Volume 25, Issue 2, 2008, Pages 226-239, ISSN 1063-5203, DOI: 10.1016/j.acha.2007.11.001. Available: ScienceDirect.