Geometric Learning - Final Project
The Dynamic Laplacian

Adi Pahima – 201346954
(parental relief)

March 10, 2021

Abstract

The following work is focusing on understanding the geometry of transport by using Diffusion maps for Lagrangian trajectory unravel coherent sets. The work will try to reconstruct the results from [1] with the two mention datasets. The first one is the Double Gyre dynamic system and the second one is the Bickley Jet flow regime. The innovative of this approach is to determine a coherent sets from possibly sparse Lagrangian trajectory data. In addition, the method expand the idea of diffusion map not only for the space domain but also for the time domain and create connectivity between space and time, thus creating "dynamic coordinates", which reveal the intrinsic low-dimensional organization of the data with respect to transport.

1 Introduction

The increasing in computational and storage capabilities, just as improving measurement techniques, supply us with large amount of data. Analysis of the data can provide a lot of information for the system we would like to investigate. Recently, different approaches emerge that compute coherent sets and coherent structure based on Lagrangian trajectory data. The mentioned paper [1] introduce a method based on Lagrangian trajectory data, which uses only local distances between data points by using affinity matrix with nearest-neighbour algorithm. The more interesting things is that they develop algorithm which takes the time domain into consideration and create dynamic coordinates that help to understand coherent sets in dynamic systems and how they transport over time. They called the base matrix as Space-time diffusion matrix.

Below are the governing equations using to construct ,the paper results:

The kernel which used to construct the affinity matrix with nearest-neighbour and controled with ε parameter is:

$$k_\varepsilon(x^i, x^j) = h(\frac{\|x^i, x^j\|^2}{\varepsilon}) \quad [1.1]$$

Where $h(x) = Cexp(-x)$

The Normalized affinity matrix is:

$$P_\varepsilon = \frac{k_\varepsilon(x^i, x^j)}{d_\varepsilon(x^i)} \quad [1.2]$$

Where:

$$d_\varepsilon = \sum_{j=1}^{m} k_\varepsilon(x^i, x^j) \quad [1.3]$$

The Forward-backward diffusion maps define as:

$$B_\varepsilon = (diag(P_\varepsilon^T 1))^{-1} P_\varepsilon^T P_\varepsilon \quad [1.4]$$

And finally the Spacetime Diffusion Map transition matrix Q_ε define as:

$$Q_\varepsilon(i, j) = \frac{1}{T} \sum_{t \in I_t}^{m} B_{\varepsilon,t}(i, j) \quad [1.5]$$

Equation 1.5 is the key matrix for all the analysis done for constructing the paper results

We asked to reconstruct results of two datasets that was present by the paper authors [3] [2]. The first dataset is Double Gyre flow and the second one called Bickley Jet

2 Double Gyre Flow

2.1 Normalized Laplacian eigenvalues

The first step was to investigate the Affinity matrix eigenvalues and learn from that how the data looks like. The Normalized Laplacian matrix is sparse affinity matrix which is ε depended i.e. the neighbors which takes into consideration depend on the ε radius. Figure 1 represent L_ε eigenvalues [1] for different ε values. By investigating the eigenvalue trends, we can identify a gap after three eigenvalues it is indicate that we can clustering the data to three different groups by using the first three eigenvectors (We will see later this assumption become true). The results are slightly differ from the original paper but the trend is similar, the reason for that could be numerical errors which depend on the algorithm structure or scaling factors which not describe by the authors.

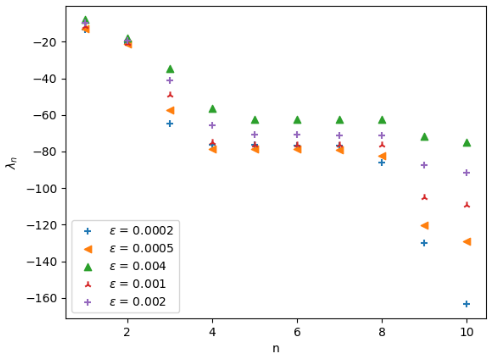

Figure 1: Double Gyre Flow Normalized Laplacian matrix eigenvalues

2.2 Double Gyre Clustering and Eigenfunctions

To proof the assumption of ability to cluster the data for three clusters just by the first three eigenvalues, Clustering algorithm (K-means) apply on the data for the initial and the final time (t=0 & t=19.5) as can bee seen in Figure 2. Even though the eigenvalues where slightly different from the authors values, the trend was similar and also the results are similar to what they presented.

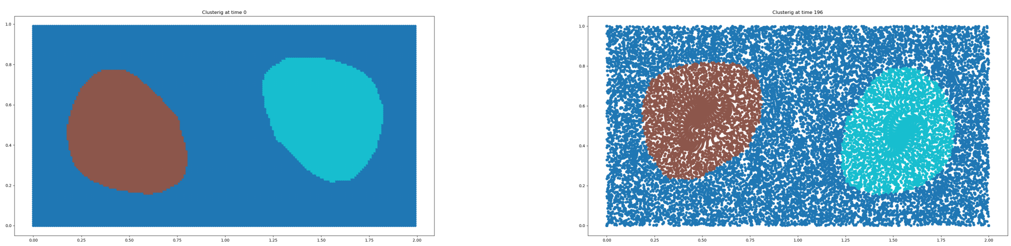

Figure 2: Left: Results of 3-clusters shown at initial time. Right: The same but with the final time - 19.5 time units

The next step was to understand the sensitivity of choosing an ε value. As mention before the ε control the number of neighbour which takes into consideration. As can be seen from

Figure 3 while choosing relatively small ε we get the middle Gyre clustering. But for larger ε the clustering map is totally different, it could be understand from eigenfunctions map which are correlate to the specific ε value. For small ε values the Gyre are concentrate at the middle ("*gyre core*") and therefore I success to cluster the middle Gyre but for the large value more neighbour takes into consideration and the diffusion map is pretty diffuse so the clustering in this case is in-between the two Gyres. The parameter ε is playing a physics role in this analysis. We can think of the ε as diffusion rate and simulate different behaviours just by changing the ε values, it can save computational time and simulate phenomenon that cannot been simulated by tests or even measured.

From numeric calculation point of view, since the Gyre frame is symmetric, the clustering for $\varepsilon = 0.0002$ could be opposite i.e clustering left Gyre and instead of the right one as presented. It's totally depend on the numeric calculation accuracy.

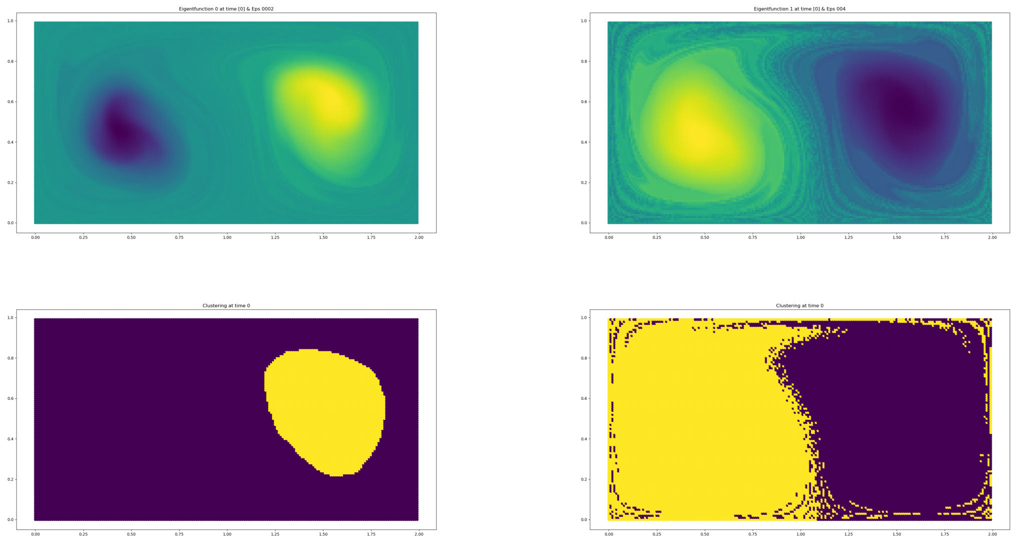

Figure 3: Left-Up:2nd Eigenfunction at initial time with $\varepsilon = 0.0002$. Right-Up: 2nd Eigenfunction at initial time with $\varepsilon = 0.004$. Left-Down: 2 Dim. clustering with the same time stamp using $\varepsilon = 0.0002$. Right-Down: 2 Dim. clustering with the same time stamp using $\varepsilon = 0.004$

The next step was to try cluster the Gyre map into more than two or three clusters. Figure 4 represent the results. The interesting in this section was to construct the same results as the authors presented. By using the clustering convection i.e using n-eigenvalues to cluster n groups, the results for $\varepsilon = 0.0005$ is different than what the authors report. But, by using more eigenvectors (more than n) to cluster n-clusters (in this case n=4) we get the same results as can be shown in Figure 5 (Using 5 eigenfunction to cluster 4 groups).

2.3 Double Gyre Clustering with missing data

The last and the most interesting case the paper present is success to cluster coherent sets even though the provided raw data is uncompleted. The algorithm overcome the missing data by assign the distance between two nodes that has no data - to infinity and than it's assign zero value 1.1 to affinity matrix, i.e. no connections for missing data points. That is allowed the algorithm to continue running. The small changes that needs to be done while using missing data approach is increasing the ε since the missing data is sparse, and to be able to construct

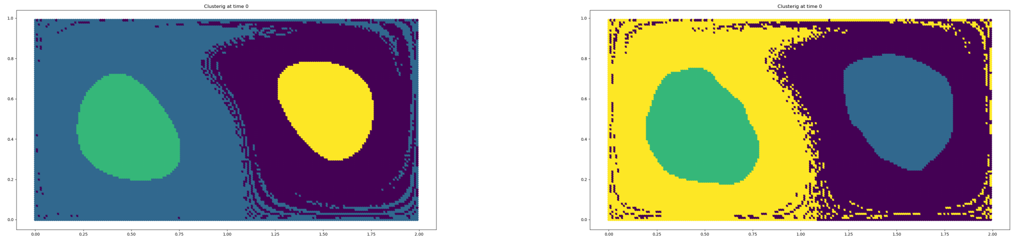

Figure 4: Left: Results of 4-clusters shown at initial time with ε = 0.0005. Right: The same but with ε = 0.0004

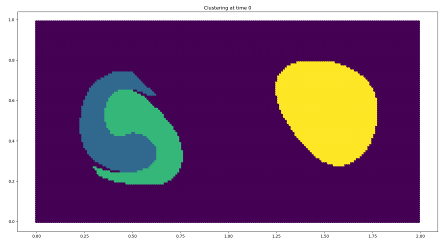

Figure 5: 4-clusters a t=0 and ε = 0.0005 with 5 eigenvectors decomposition

an efficient affinity matrix, needs to maintain sufficient amount of neighbours that connected to each node.

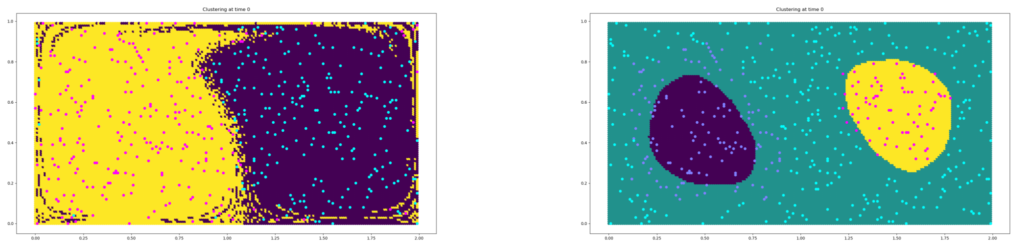

Figure 6: Missing data clustering with on top of the original clustering with fully raw data

3 Bickley Jet

3.1 Normalized Laplacian eigenvalues

Same calculation as 2.1 was done. Figure 7 represent L_ε matrix eigenvalues, similar to previous the eigenvalues results are slightly different from the author results but the eigenvalues trends is similar. It can be explained by numerical calculations accuracy or maybe multiplication factor that not mentioned by the author. The important is that the coherent sets results are similar to what they presented.

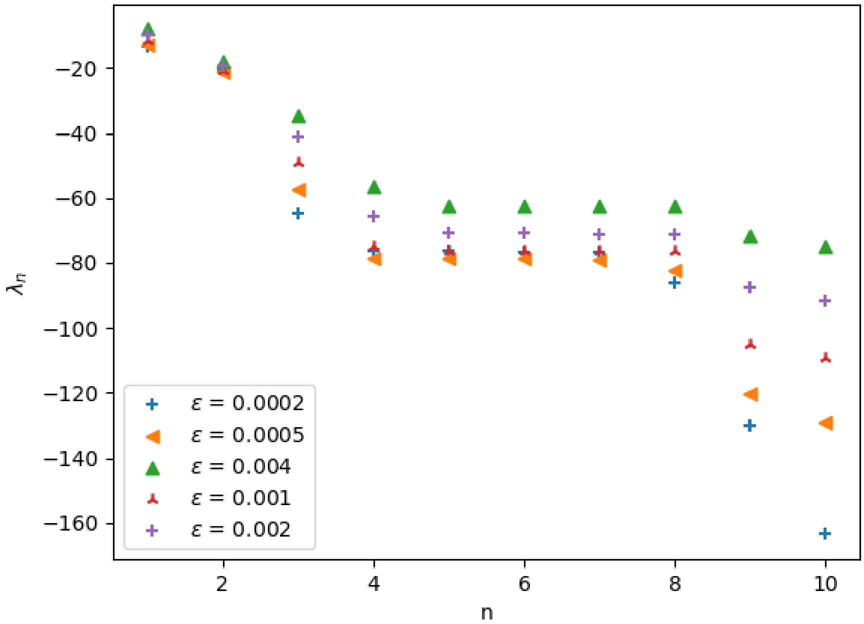

Figure 7: Bickley Jet Normalized Laplacian matrix eigenvalues

3.2 Embedding using first eigenfunctions

Figure 8 represent the embedding of the 2,4,5 eigenfunction which colored by the classification color code for 9 coherent sets. The embedding is highlight the connectivity structure of the cluster. We can infer what are the closest coherent sets and if there is a physics reason why they are close or far from each other.

3.3 Bickley Jet clustering over time

Figure 9 represent clustering which done for different time units (from t=5 to t=35). The vortexes flow along X axis can be distinguish and that is indeed reflect the reality. This is very interesting results since flow regime simulations are very heavy and sometimes cannot provide sufficient results. In this case we have light model with Space-Time connectivity that can provide good results in very short time. It could save simulation time and converge to the right solution quite quick.

3.4 Bickley Jet Eigenfunctions over time

Figure 10 represent the Eigenfunctions. In this case the first and the third eigenfunction is similar to author eigenfunction. But surprisingly the second eigenfunction is totally different, instead of indicating the main stream line, in my implementation case the second eigenfunction is indicating the top vortexes and not the main stream line. For my opinion it is more realistic since the first eigenfunction describe the streamline in general, the third eigenfunction describe the bottom vortexes, so the second eigenfunction for my opinion needs to describe the top vortexes as I got and not like what they get. I tried many different approaches to construct there results without success. For my understanding general map, top and bottom vortexes are the first three governing mode in this problem.

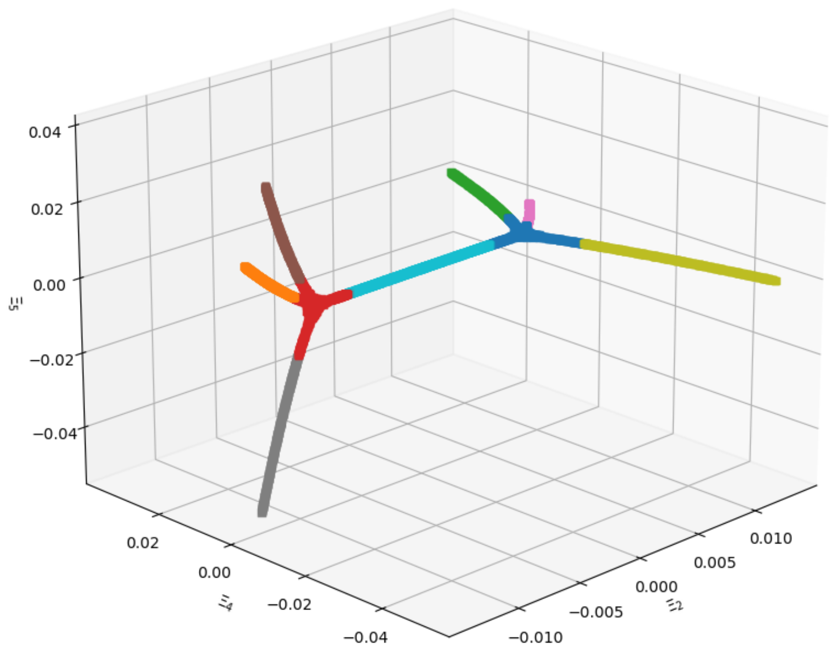

Figure 8: 3D embedding for Space time diffusion matrix using the first 3rd eigenfunctions

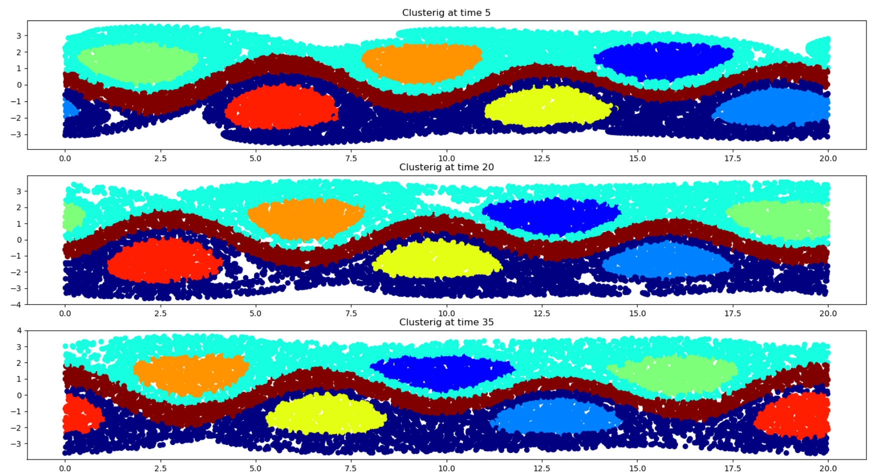

Figure 9: Bickley Jet 9-cluster at different time slots. From up to down t=[5,20,35]

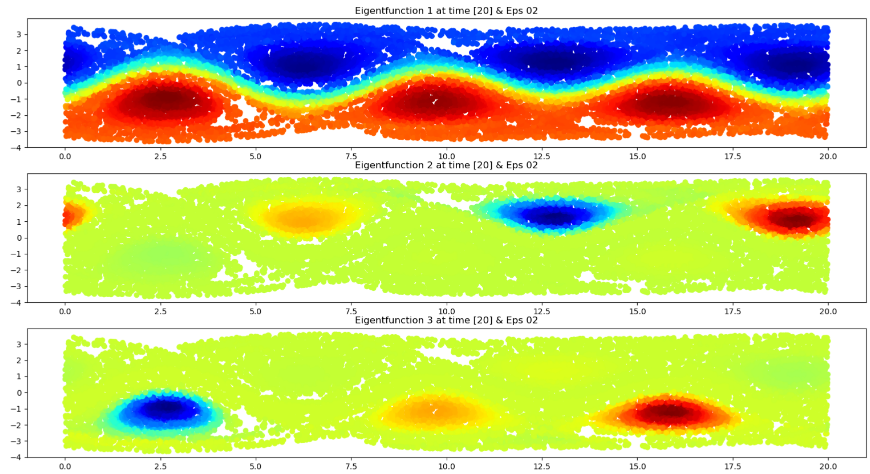

Figure 10: Bickley Jet Eigenfunctions at t=20. From up to down Ξ_2,Ξ_3,Ξ_4

4 Conclusions

The paper presented a method which based on provided data to find a coherent sets. This is something new for me in particular which I not recognize before. It can helps to solve a lot of complex problems that consume a lot of computational resources especially for dynamic problem and diffusion cases, even though the solution is not 100% correct, it could helps to converge to the right set point and then simulate with more complex tools. The most interesting is by using missing data to extrapolate and find a coherent sets although there are some missing trajectories. Moreover, the ε parameter uses as diffusion rate for the dynamic system and can simulate different cases and find the best coherent sets. The algorithm does not have any physical background of the problem, but it is success to solve the problem in a good manner, and this aspect make this algorithm very unique and robust for many science applications

References

[1] Ralf Banisch and Péter Koltai. "Understanding the geometry of transport: Diffusion maps for Lagrangian trajectory data unravel coherent sets." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.3 (Mar. 2017), p. 035804. ISSN: 1089-7682. DOI: 10.1063/1.4971788. URL: http://dx.doi.org/10.1063/1.4971788.

[2] Clementi Group. *Space and State Space Decomposition*. https://github.com/ClementiGroup/S3D. 2017.

[3] Adi Pahima. *Diffusion Map*. https://github.com/Apahima/GeoLearn. 2021.