**Diffusion Maps**

Patrick H. Rupp

E-mail: prupp1@jhu.edu

Abstract

Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd …

Code: <https://github.com/PHRupp/sde>

1. Background

Data is growing at an alarming rate as well as becoming more complex in nature. It is becoming common to have datasets with high number of dimensions. Analyzing this type of data is difficult and cumbersome often making it impossible to find the underlying correlations in the data.

Dimensionality reduction is a technique that attempts to identify dimensions which are correlated and reduce them to a smaller number of dimensions that can still represent the original shape of the data. There exists many different techniques each with their own advantages and limitations including both linear and non-linear approaches.

1.1 PCA

One of the most common dimensionality reduction technique is Principle Componenet Analysis (PCA). The PCA technique uses an approach that identifies features that are linearly correlated and attempts to reduce those dimensions by projecting the data in lower dimenions over new features axes. PCA is widely used within exploratory data anlaysis because visualizing correlations in high dimensional data is a difficult problem.

1.2 Diffusion Maps

Another technique that is becoming popular is called Diffusion Maps. This technique attempts to perform dimensionality reduction by attempting to find an underlying non-linear relationship in the data and projects the data to the lower dimension.

The special highlight with the diffusion map technique is that it is a non-linear approach unlike the PCA which is a linear approach. In practice, it is hard to find linearly correlated data which makes the diffusion maps a good option. It is able to perform the non-linear approach by assuming there is some type of lower dimension manifold tying the data together.

2. Dimension Reduction Experiment

The first experiement will be to test multiple differnet data sets to see how well the different dimensionality reduction techniques perform. The two main experiements will consist of using PCA as well as diffusion maps. In addition, the diffusion maps have three different implementations in this study to see how well they perform.

The first experiment consists of reducing four differnet data sets. The data sets will be repsented in differnet mathematical shapes such as the helix, sine wave, torus, and spiral.

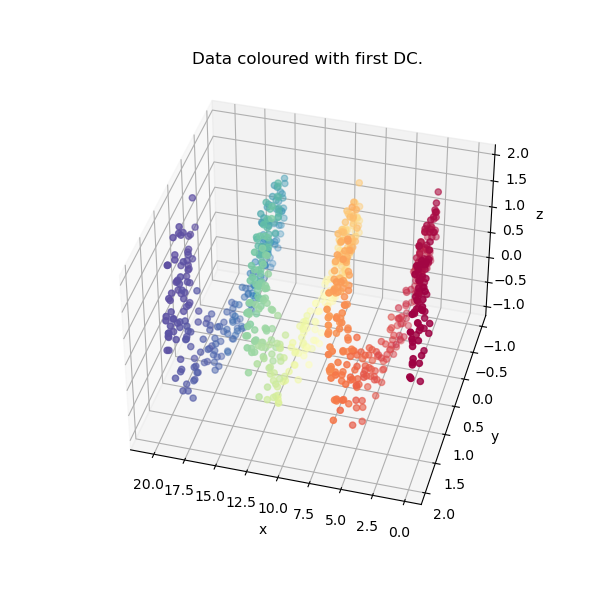


Figure I – Displays the original 4D data set in the form of a helix with noise. The helix is visible visually in 3D and the 4th dimenion is linear with the ‘z’ axis. The underlying manifold in this case would be the helix shape.

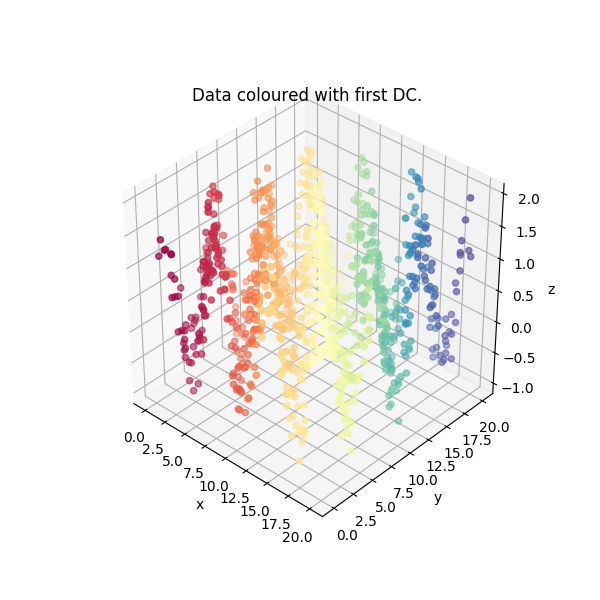


Figure II – Displays the original 4D data set in the form of a sine wave with noise. The sinewave is visible visually in 3D and the 4th dimenion is a “termperature” value corresponding to cos(x). The underlying manifold in this case would be the sine wave.

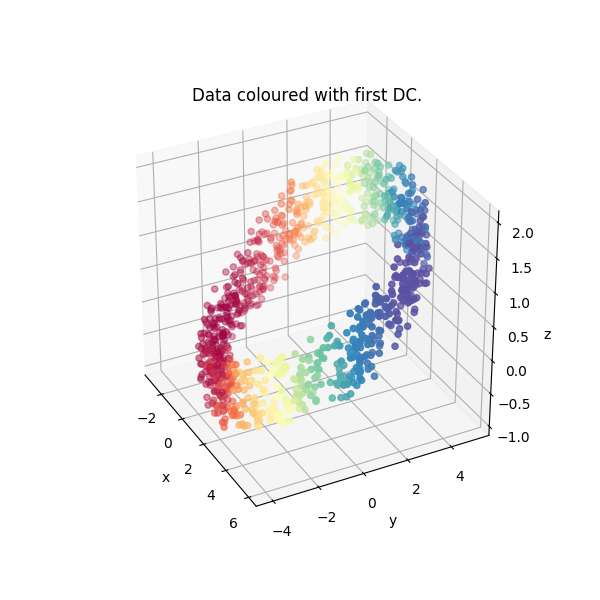


Figure III – Displays the original 4D data set in the form of a tilted torus with noise. The torus is visible visually in 3D and the 4th dimenion is a “termperature” value corresponding to relationship with y. The underlying manifold in this case would be the torus.

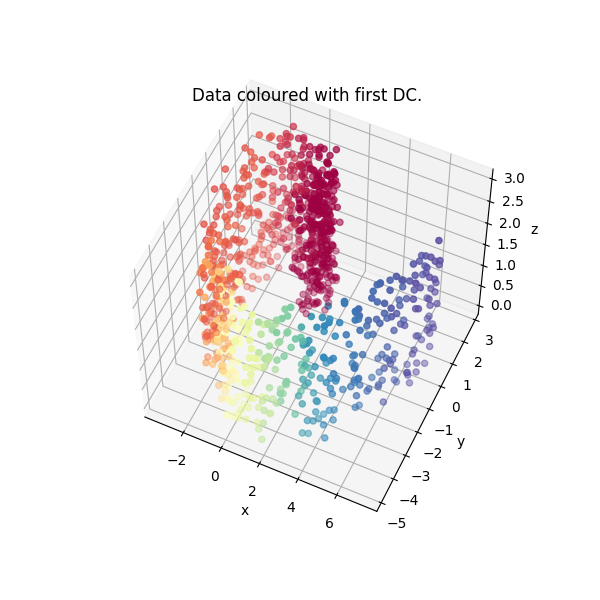


Figure IV – Displays the original 4D data set in the form of a spiral with noise. The spiral is visible visually in 3D and the 4th dimenion is a “termperature” value corresponding to cos(x). The underlying manifold in this case would be the spiral. The spiral is also commonly known as the “swiss role”.

Write stuff

2.1 Diffusion Maps

For our diffusion maps experiment, two different implementation are used. One implementation used can be found within the pydiffmap python library whereas the other was developed on KDNuggets website for an article detailing diffusion maps [5].

*2.1.1 pydiffmap*. Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd.

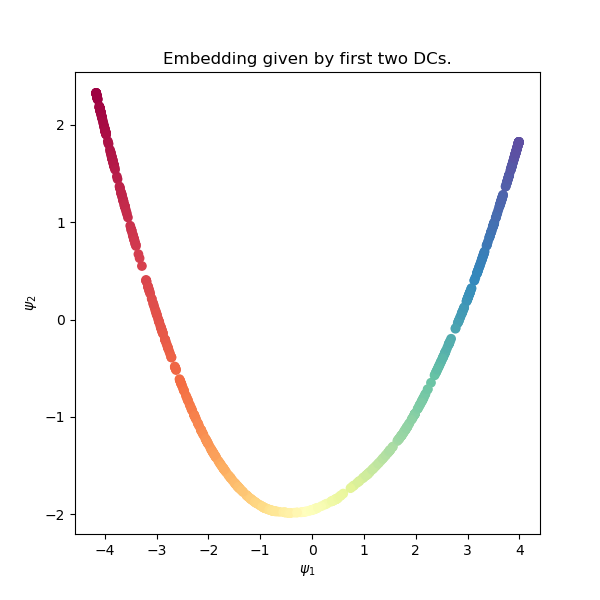


Figure II – Displays the reduced data from the pydiffmap implementation of the diffusion maps of the **helix**. This technique found a lower dimenion manifold reminescent of a quadratic function. The change of color can be correlated with the increase in value of the x-axis verifying that this data is indeed compresed to 2 dimensions from 4.

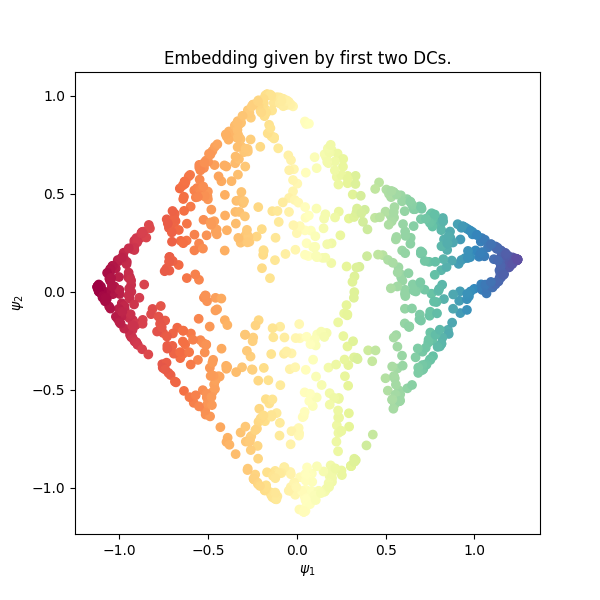


Figure II – Displays the reduced data from the pydiffmap implementation of the diffusion maps of the **sine wave**. This technique was able to flatten the sine wave from the 3D ripples to the 2D sheet that the data represented.

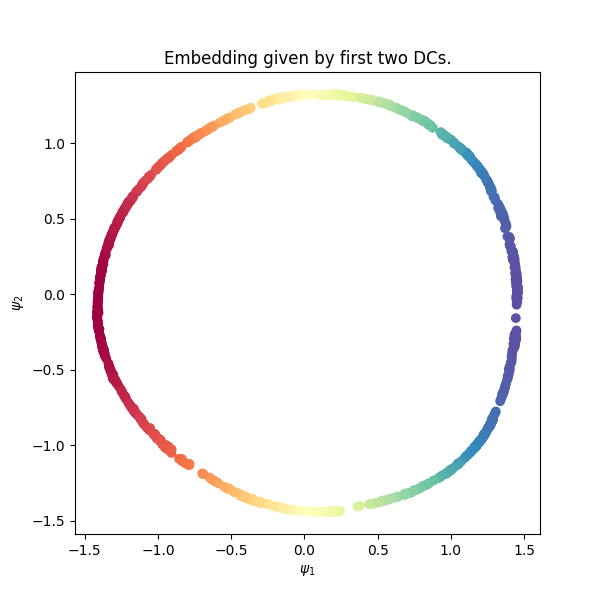


Figure II – Displays the reduced data from the pydiffmap implementation of the diffusion maps of the **torus**. This technique found a lower dimenion manifold reminescent of a quadratic function.

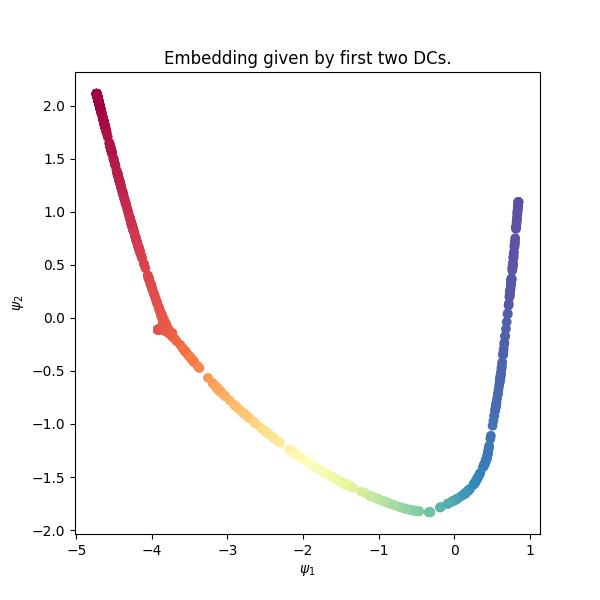


Figure II – Displays the reduced data from the pydiffmap implementation of the diffusion maps of the **spiral**. This technique found a lower dimenion manifold loosely reminescent of a quadratic function.

*2.1.2 KD Nuggets*. Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd.

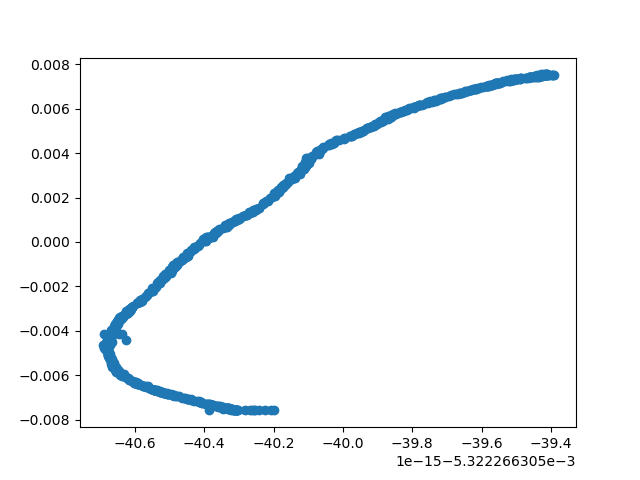


Figure III – Displays the reduced data from the kdnuggets implementation of the diffusion maps for the **helix** data. This technique found a lower dimenion manifold reminescent of a quadratic function also, but it is less clean.

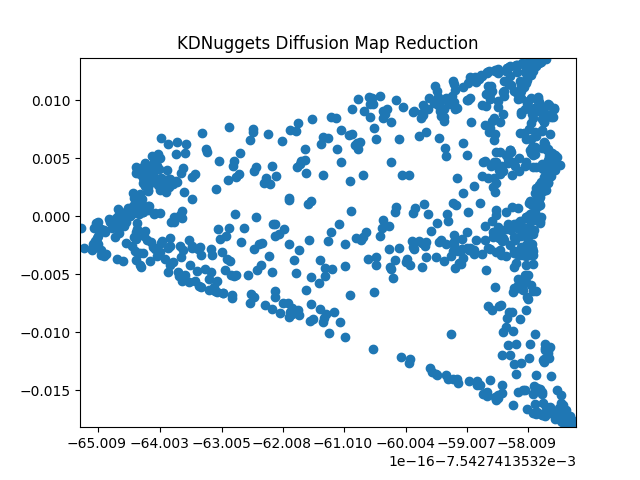


Figure III – Displays the reduced data from the kdnuggets implementation of the diffusion maps for the **sine wave** data. This technique was able to flatten the sine wave from the 3D ripples to the 2D sheet that the data represented, but the shape is not as clean compared to the pydiffmap implementation or the original data set.

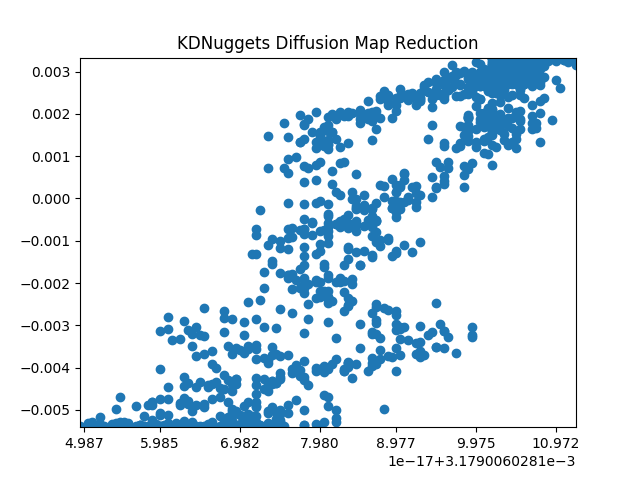


Figure III – Displays the reduced data from the kdnuggets implementation of the diffusion maps for the **torus** data. This technique was able to flatten the the torus but the output does not resemble the higher dimensional data at all. This might be due to the complex shape influenced by all the points used.

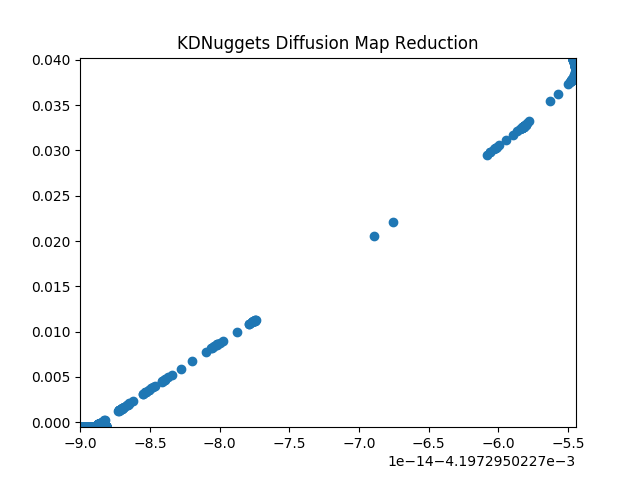


Figure III – Displays the reduced data from the kdnuggets implementation of the diffusion maps for the **spiral** data. This technique was able to flatten the spiral into a linear plot excluding the ends where there is a high concentration of points. We expect the high concentration one end but not both.

3. Resource Usage Experiment

While accuracy for a numerical model is of the upmost importance, it is not the only thing that must be considered when evaluating potential solutions. Two main factors that are considered when employing data solutions are Computational comlexity and Space complexity.

Computational complexity deals with the number of computations that the algorithm needs to perform for a given size of data. There exists a relationship between the number of data points given to the model and the time it takes to complete the training of the model.

Space complexity deals with the physical memory space needed by the algorithm to perform the computation. If an algorithm requires more space than the hardware can provide, the algorithm cannot perform its operations on the entire set of data. Hence reviewing both of these points are crucial for identifying the capabilities of various algorithms and methods.

Tbd. Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd.

3.1 Computational Complexity

Tbd. Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd.

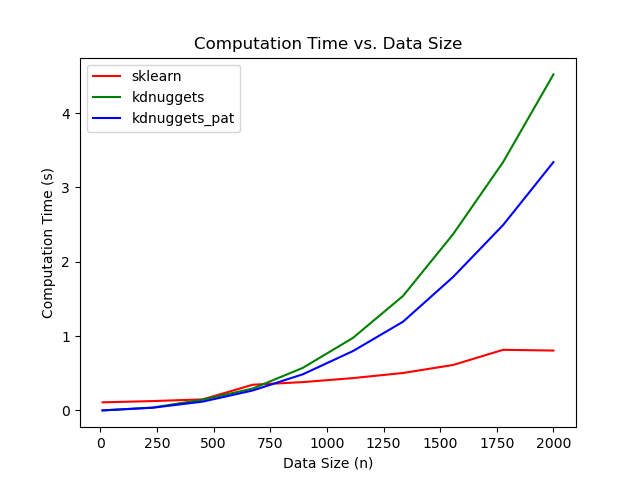


Figure IV – Displays the computation time as a function of the size of the data for training the diffusion map. The knuggets has the fastest increase over time whereas the sklearn avoids the exponential growth due to not building the distance map between all points.

3.2 Space Complexity

Tbd. Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd.

Figure IV – Displays the memory usage for the differnet algorithms as the number of records within the data increases. This gives us the insight into what the space complexity of the algorithms require. The KD Nuggets is the worst because their algorithm requires storing six N by N matrices whereas the Pydiffmap stores a nearly constant amount due to using a limited set of neighbors in an efficient sparse matrix.

4. Conclusion

Tbd. Tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd tbd.

References

[1] Banisch, Ralf, Zofia Trstanova, Andreas Bittracher, Stefan Klus, and Peter Koltai. "Diffusion maps tailored to arbitrary non-degenerate Ito processes." arXiv preprint arXiv:1710.03484 (2017).

[2] Porte, Herbst, Hereman, Walt, An introduction to Diffusion Maps <https://inside.mines.edu/~whereman/talks/delaPorte-Herbst-Hereman-vanderWalt-DiffusionMaps-PRASA2008.pdf>

[3] Coifman, Lafon, Lee, Maggionni, Nadler, Warner, Zucker. “*Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*” <https://www.pnas.org/content/pnas/102/21/7426.full.pdf>

[4] Wolf, Guy. “Introduction to Data Mining: Diffusion Maps”. (2016). <http://cpsc445.guywolf.org/Slides/CPSC445%20-%20Topic%2010%20-%20Diffusion%20Maps.pdf>

[5] Raj, R., 2020. *Diffusion Map For Manifold Learning, Theory And Implementation - Kdnuggets*. [online] KDnuggets. Available at: <https://www.kdnuggets.com/2020/03/diffusion-map-manifold-learning-theory-implementation.html> [Accessed 7 July 2020].

[6] Raj, R., 2020. *Diffusion Map For Manifold Learning, Theory And Implementation - Kdnuggets*. [online] KDnuggets. Available at: <https://readthedocs.org/projects/pydiffmap/downloads/pdf/master/>

[Accessed 11 Aug 2020].