Deriving Political Party Affiliation With Diffusion Maps

Timothy Healy

August 2019

Abstract

Natural language processing has a large computational burden due to the high dimensionality and sparsity of textual data. Diffusion Maps is a mathematical technique that can be used to reduce the dimensionality of data, even if the data has been sampled from a nonliner underlying structure. This paper will employ and evaluate Diffusion maps as a dimensionality reduction technique for textual data, using it to derive the political party for members of the United States Congress based on Twitter activity.

Contents

1 Introduction & Motivation

1.1 Curse of Dimensionality

There are many applications where a "Curse of Dimensionality" can affect one's ability to decipher a dataset. This curse plagues many fields, such as numerical analysis, statistical sampling, combinatorics, machine learning, data mining and databases. Intuitively it makes a lot of sense; as the dimensionality of a dataset increases, so does the complexity of the result space. As more dimensions are added, a larger sample of data is needed to properly represent the training space and get a robust result.

One example of where this issue is relevant is within database systems and data mining. Organizing and searching data often relies finding areas in the space where objects cluster with similar properties. In high dimensional data, objects can appear sparse and thus dissimilar which can make efficient organization of high dimensional data difficult.

In machine learning, models attempt to "learn" about a naturally occurring state or structure from a finite set of data. This data is often derived from a high dimensional feature space, where each value can assume a high range of values. Therefore, machine learning models require an enormous amount of training data in order to see enough combinations to make robust decisions.

Natural Language Processing is a field that is affected by many of aspects as data mining and machine learning face. Textual data is sparse, many documents of text have no terms in common, and are therefore extremely dissimilar. When thinking about the feature space of a collection of documents, we can think of words are being features, and an "importance" score as being it's particular value. We can quickly see how vast such a result space can be, and thus, how vast a textual dataset would need to be in order to adequately train a model.

1.2 Dimensionality Reduction

The goal of an analysis or machine learning model is to better understand some naturally occurring phenomena, which exists in some vast dimensional space. In practice, it is simply not feasible to capture all of the features of this space, so a means of extracting patterns from the feature space is needed, such as Dimensionality Reduction.

In practice there exist many different techniques for reducing dimensionality, and can be divided into two methodologies: feature selection and feature extraction. Feature selection methods attempt to algorithmically find a subset of relevant features, acting under the assumption that the data contains some features that are either redundant or irrelevant. Feature extraction is a process that maps the high dimensional data into a lower dimensional embedding.

1.2.1 Linear vs. Nonlinear Techniques

Feature extraction methodologies can be further divided into techniques accounting for nonlinear and linear underlying data structures. In the real world many phenomena being described by finite data sets are nonlinear in nature.

1.3 A Motivating Example - Swiss Roll Dataset

A good data structure to show how Diffusion Maps can be useful is the "swiss roll". As we can see from Figure 1, 4000 points are sampled along a nonlinear underlying geometry resembling a swiss roll. If we were to "zoom in" on a particular area of this structure, the local space would appear linear; that is, Euclidean distance could be used to accurately measure distance between points.

Globally, Euclidean distance cannot be used since geodesic distance must be taken into account along the swiss roll manifold. Using a visual example, in Figure 1, Euclidean distance will indicate that the distance between a yellow point and a blue point would be shorter than the distance between a yellow point and a green point. We can see from the nonlinear

structure of the data that this is not possible; a distance measure must travel along the swiss roll when measuring distance.

Therefore, a useful reduction in dimensionality would be to "unroll" the swiss roll, since it would preserve the geodesic distances between points along the spiral. Additionally, with an unrolled structure, we are free to use a Euclidean distance measure since the nonlinear geometry of the data is accounted for.

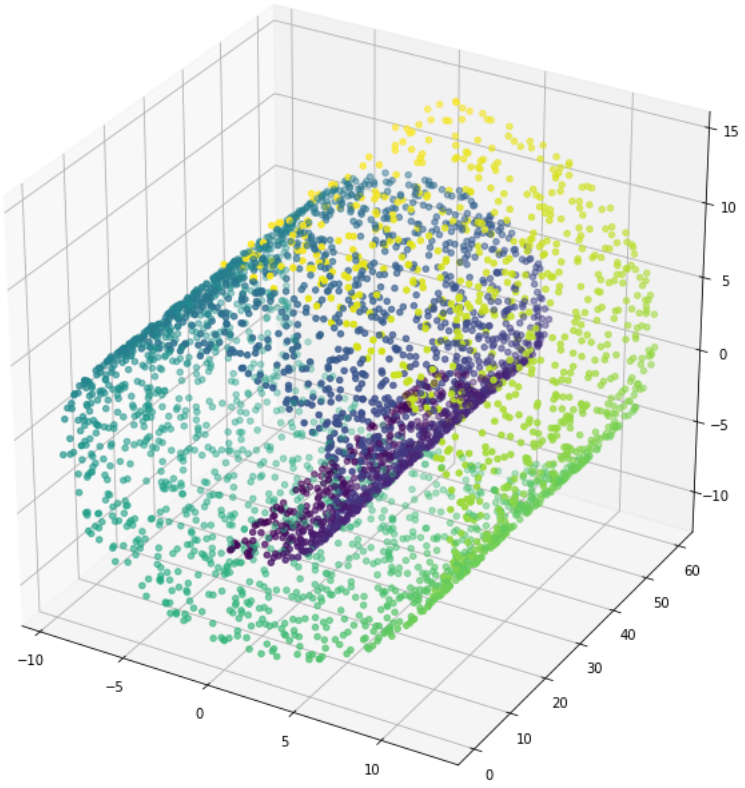

Figure 1: Swiss Roll Dataset

1.3.1 Principal Component Analysis

In Figure 2, Principal Component Analysis (PCA) with $k = 2$ is applied to the swiss roll, and color mappings from Figure 1 are applied to the data. It is clear that geodesic distances are not preserved within PCA's linear subspace. Intuitively, it seems that PCA is simply

"taking a picture" of the data from a linear window along two of the axes. Eucliden distance in this 2-dimensional space would tell us that yellow points are close to blue points, which we know is not the case according to the underlying structure of the data.

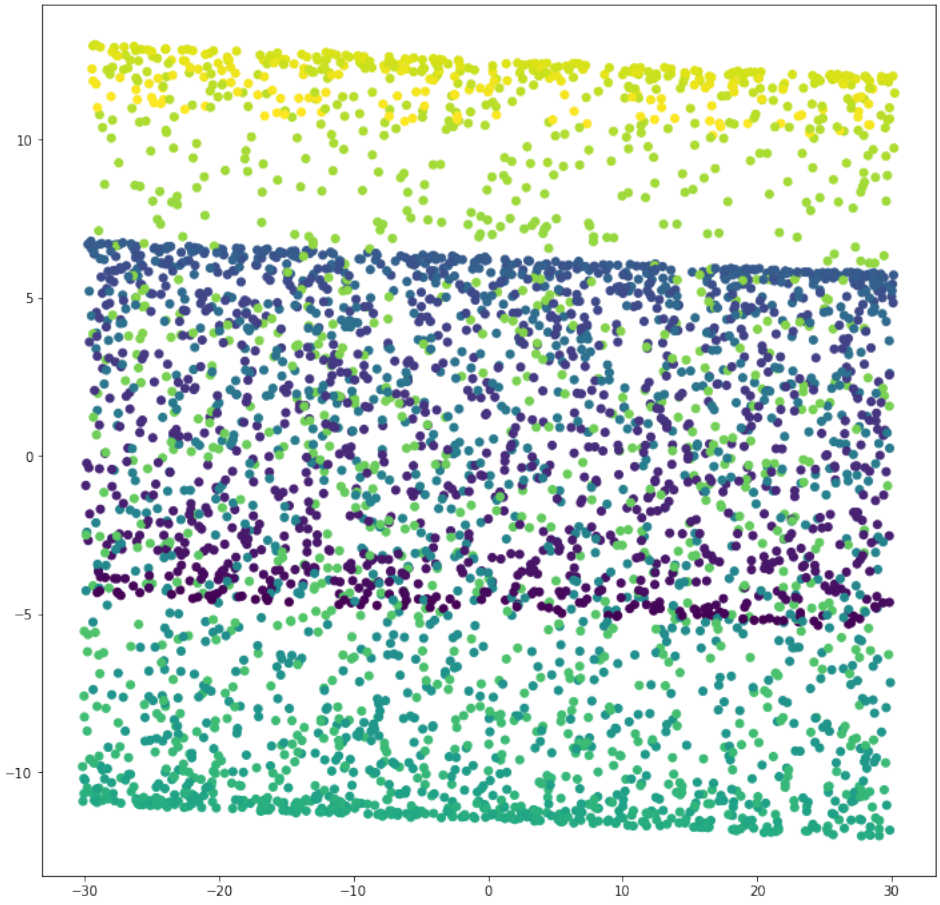

Figure 2: Embedded Swiss Roll - Principal Component Analysis

1.3.2 Diffusion Maps

In Figure 3, we can see that the Diffusion Maps technique successfully "unrolls" the data, preserving the geodesic distances between points. It correctly embeds the data into a lower dimensional subspace along the nonlinear geometry. This reduction of dimensionality is especially useful we are now able to evaluate the data using linear methods, since the subspace is Euclidean.

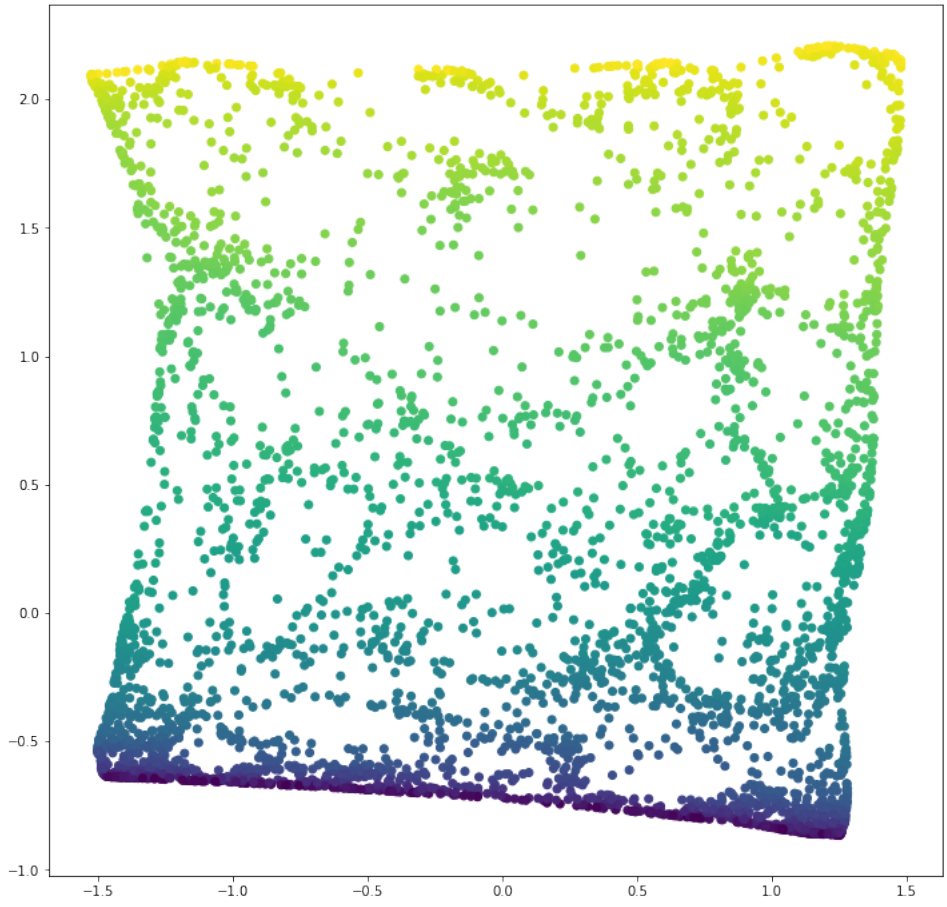

Figure 3: Embedded Swiss Roll - Diffusion Maps

2 Diffusion Maps

2.1 Overview

As seen in the example above, the goal of Diffusion Maps is to reduce dimensionality by re-organizing data according to an underlying geometry. This assumes that this structure can be represented as a manifold, where locally, the neighborhood of each data point resembles a Euclidean space. Within these neighborhoods, a diffusion kernel is used to measure the connectivity of points. At different time-dependent scales, t, the diffusion process integrates local geometry, and globally reveals the nonlinear structure of the manifold.

With this diffusion process a diffusion map is generated and used to embed the data into a lower-dimensional space, such that the Euclidean distance between points approximates the diffusion distance in the nonlinear space. In practice, the diffusion process takes a Random Walk on the data, generating a Markov transition matrix based on the connectivity of points. This Markov transition matrix can then be embedded Euclidean space using linear matrix decomposition.

2.2 Theory

2.2.1 Random Walk

One of the key concepts of Diffusion Maps is taking a random walk on the data. The idea is that the probability of "jumping" between points is dependent on how close the points are. This essentially means that the probability of jumping between points x_i and x_j will be higher if the points are nearby.

Let $(X, \mathcal{A}, \mu)$ be a measure space, where X represents the data set and μ represents the distribution of of the points on X. Based on this measure space, we can define a kernel $k : X \times X \to \mathbb{R}$ that represents the connectivity between data points x and y in one step of the random walk. One example is the following Gaussian kernel, but kernels can be tailored

to the application at hand.

$$k(x,y) = \exp\left(-\frac{||x-y||^2}{\epsilon}\right) \tag{1}$$

Where ϵ is a neighborhood parameter. Such kernels have the following properties.

- k is symmetric: $k(x,y) = k(y,x)$
- k is positivity preserving: $k(x,y) \geq 0$

With (X,k), we can construct a Markov matrix, also known as a graph Laplacian, on X.

$$d(x) = \int_X k(x,y)d\mu(y) \tag{2}$$

Defining,

$$p(x,y) = \frac{k(x,y)}{d(x)} \tag{3}$$

2.2.2 Diffusion Process

In other words, $p(x,y)$ represents the one-step transition probability of the Markov transition matrix at different time steps, t. By taking powers of M, we essentially are increasing the number of steps taken. For example, Let,

$$M = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \tag{4}$$

Each of the elements p_{ij} represents the probability of jumping between data points i and j. When M is squared, it becomes,

$$M = \begin{bmatrix} p_{11}p_{11} + p_{12}p_{21} & p_{12}p_{22} + p_{11}p_{12} \\ p_{21}p_{11} + p_{22}p_{21} & p_{22}p_{22} + p_{21}p_{12} \end{bmatrix} \tag{5}$$

Where (5) represents all of the combinations of two-step "jumps" between points with probability $p(x_i, x_j)$. More generally, M contains all of the possible one step jumps, while M^t

contains all of the possible t step jumps that can be made from x_i to x_j. We can thus define a Diffusion distance is defined as follows.

$$D_t(x_i, x_j)^2 = \sum_{u \in X} |p_t(x_i, u) - p_t(x_j, u)|^2 \tag{6}$$

$$= \sum_{k} |M_{ik}^t - M_{kj}^t|^2 \tag{7}$$

We can see that the diffusion distance is small if there are many high probability paths of length t between points x_i and x_j. Running the diffusion process forward in time, by taking larger and larger powers of M^t will reveal the underlying geometric structure at different scales. The idea of a cluster within the data set is quantified as a region where the probability of escaping is low.

2.2.3 Diffusion Map

M^t provides the re-organization of the data, but no dimensionality reduction. Dimensionality reduction is achieved by decomposing Markov Diffusion matrix, Therefore, the diffusion distances in (7) can be expressed in terms of the eigenvectors and values of M as,

$$Y_i^{'} = \begin{bmatrix} \lambda_1^t \phi_1(i) \\ \lambda_2^t \phi_2(i) \\ \vdots \\ \lambda_n^t \phi_n(i) \end{bmatrix} \tag{8}$$

where $\phi_i(i)$ indicates the i-th element of the first eigenvector of M. Thus, we can use the Euclidean distance between mapped points Y_i and Y_j as a proxy for the diffusion distance. Dimensionality reduction is achieved by retaining the m dimensions associated with the dominant eigenvectors, which ensures that $||Y_i^{'} - Y_j^{'}||$ approximates the diffusion distance $D_t(x_i, x_j)$ best.

3 Application

3.1 Tweet Clustering

This application of Dimensionality Reduction strives to observe a naturally occurring phenomena; political party affiliation, within a high dimensional space; Twitter activity. Since tweets are limited to 280 characters, the corpus of documents to be analyzed is extremely sparse. In this context, we'd like to understand how political discourse from each politician characterizes them as Republican or Democrat.

The central assumption of this application is that the political discourse we view on Twitter is generated from some lower dimensional space. Intuitively, this makes sense because politicians usually have some sort of underlying position or agenda that they are conveying when publishing tweets. In essence, similar politicians will convey a similar message, although they will use different vocabularies. This paper will employ Diffusion Maps to uncover these similarities. For further illustration, let us observe the following fictitious tweets as an example.

- *During my term, unemployment is at an all time low! Elect me again and we will continue to put Americans to work!*
- *Jobs, Jobs, Jobs! The newest statistics from the Bureau of Economic Analysis are looking good!*

We as humans can understand that one topic of these tweets is "Employment". We know this because the terms used in the tweets are frequently used when discussing the subject. To a computer, each of these tweets is just a list of a words. Perhaps Dimensionality Reduction can be used to help the computer better understand the relationships between these sentences. But first, some preprocessing will need to be done.

3.1.1 Data Extraction & Pre-processing

The data source for this application is the Twitter Application Programming Interface (API). While the API does store lists of politicians, it does not directly store political party affiliations. Therefore, a list of politicians and parties from BallotPedia is mapped to the Twitter data. A Venn diagram comparison of each list of politicians follows in Figure 4. Tweets are then pulled from the Twitter API using the final list of 408 politicians and party affiliation labels. In total, 106,921 tweets are extracted from the API.

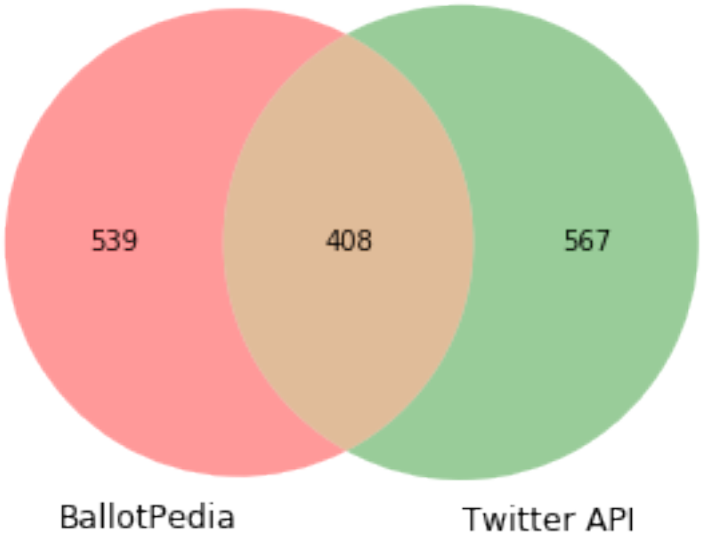

Figure 4: Politician List Comparison

The tweet data contains many "noise" terms that we are not concerned with, and therefore some pre-processing of the text data is needed. This pre-processing will reduce noise within the textual data and help the Diffusion Maps algorithm better understand the become "less confused" with the data. In this analysis, the following steps are completed to clean up the data.

1. Split Tweet strings into lists of words
2. Convert words to lower case
3. Remove punctuation from each word

4. Remove remaining tokens that are not alphabetic
5. Filter out stop words - "the", "to", "and", etc.
6. Remove handles that are at the beginning of retweets
7. Convert back to a string

The end result of this pre-processing converts the following tweet string,

- *Cancelling private and public student debt would boost 45 million Americans and the economy.*

To the following,

- *cancelling private public student debt would boost million americans economy*

3.1.2 Term Frequency - Inverse Document Frequency (TF-IDF)

With a list of pre-processed tweets, we need a way of measuring which terms are important. In information retrieval, TF-IDF, short for term frequency–inverse document frequency, is a metric that indicates how important a word is to a document in a set of text documents. It is often used as a weighting factor in text mining and user modeling. For this analysis, we use the TF-IDF metric to compare key terms between politicians

The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. As an example, a summary of the top 10 terms and TF-IDF scores from US Representative Ilhan Omar is given in Table 1. We can see that the metric represents the Omar's political rhetoric quite well.

Term	TF-IDF
struggle	0.7927974683178989
cosign	0.7833171469235927
suffering	0.7455088452588212
woah	0.7311076349291042
point	0.7311076349291042
coverage	0.7245398195218251
thankful	0.7071067811865475
team	0.7071067811865475
voting	0.6873127582915157
cruelty	0.6822621388794573

Table 1: Top 10 Terms from Representative Ilhan Omar

3.1.3 Term-Document Matrix

As represented by Table 1, we now have vectors of terms for each politician. These vectors are then organized into a Term-Document Matrix, where each "document" is a string of all the terms from the politician's set of tweets. We now have an object that can be used to relate politicians, since each row represents a politician across terms, and each column represents a term across politicians. This matrix is very sparse, since each politician has their own vocabulary. In this analysis we evaluate a 51,253x408 dimensional matrix, with only 525,581 TF-IDF values, a sparsity of 97%!

The TD Matrix is powerful because it will be used to identify clusters of terms by political party. Our central hypothesis is that politicians within each party will be tweeting about similar political positions and topics, which can represented in a lower dimensional space. The hope is that applying Dimensionality Reduction to this matrix will uncover these rhetorical similarities. Diffusion Maps gives us the ability to uncover these embeddings in a nonlinear fashion.

3.1.4 Diffusion Metric - Cosine Similarity

In section 2.2.1, a Gaussian kernel is given as an example of a Diffusion Metric used to measure the connectivity between points in the random walk. This analysis will employ a Cosine Similarity kernel to measure distance, which uses the cosine of the angle between two vectors rather than euclidean distance. More intuitively, Cosine Similarity determines if two vectors are roughly pointing in the same direction, regardless of how many terms the vector includes. It is often used to measure document similarity in text analysis. One example includes Latent Semantic Analysis (LSA), which uses Principal Component Analysis to find a linear embedding for a given Term-Document Matrix.

3.2 Results

The Diffusion Maps algorithm performs quite well at naturally discovering the party affiliations in the list of 408 politicians. Figure 5 shows the results of applying the Diffusion Maps algorithm to our the sparse TD Matrix at different combinatiosn of time scales and ϵ neighborhood values. Each point represents a politician, colored red for Republican and blue for Democrat from the BallotPedia mapping data.

It is really interesting to see how the groups form and the data becomes more linear as t increases. At $t = 3$, it seems that a k-means clustering algorithm with $k = 2$ would create a fairly accurate classifier, as two groups are distinct in the space. It is also interesting to see how spread out the Democratic points are compared to the Republican points. Perhaps the Democrat's points are more spread out because they tend to have different positions on a wider range of subjects. It would be interesting to run clustering algorithms within each of the parties to see if topics emerge, and which politicians are discussing certain topics more.

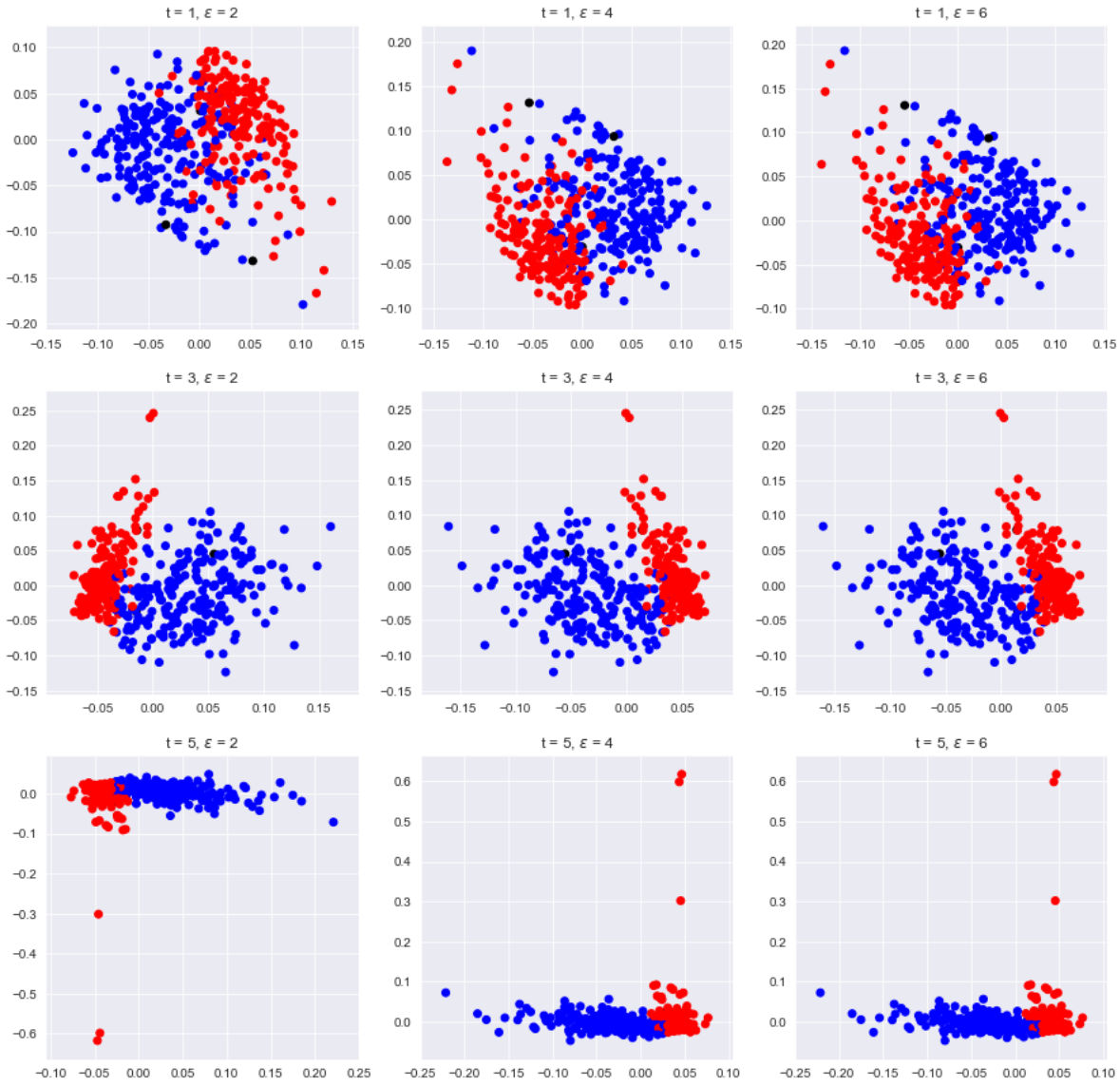

Figure 5: Diffusion Maps Result

4 Conclusion

Diffusion Maps are an effective tool for Dimensionality Reduction. The technique has proven to be quite valuable as a means of preserving complex underlying data structures while reducing dimensionality. The results of this paper's application to Twitter data are extremely interesting and warrant further study. With the curse of dimensionality affecting so many domains leveraging high dimensional data, it is encouraging to see effectiveness of the embeddings that Diffusion Maps can produce.

Bibliography

Banisch, R., Trstanova, Z., Bittracher, A., Klus, S. and Koltai, P. (2018). Diffusion maps tailored to arbitrary non-degenerate Itô processes. Applied and Computational Harmonic Analysis.

J. De la Porte, B. Herbst, W. Hereman, S. van der Walt, An Introduction to Diffusion Maps, Proceedings International, 2008.

Fagette Antoine, Nicolas Courty, Daniel Racoceanu, Jean-Yves Dufour. Unsupervised dense crowd detection by multiscale texture analysis. Pattern Recognition Letters, Elsevier, 2013, pp.1-27.

Lafon, S. and Lee, A. (2006). Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9), pp.1393-1403.

S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multi-cue data matching by diffusion maps. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(11):1784–1797, 2006.

T. Sipola. Knowledge Discovery Using Diffusion Maps. PhD thesis, University of Jyvaskala, Jyvaskala, Finland, 2013.
https://jyx.jyu.fi/dspace/handle/ 123456789/42647?show=full

G. Karypis and E.-H. Han. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. CIKM, 2000.

Shalizi, C. Nonlinear Dimensionality Reduction II: Diffusion Maps. Data Mining, Carnegie Mellon University, United States, 2009.
https://www.stat.cmu.edu/ cshalizi/350/lectures/15/lecture-15.pdf

R. Socher and M. Hein, "Manifold learning and dimensionality reduction with diffusion maps," in Seminar report, Saarland University, 2008.

P. Sharma and A. Raglin. Efficacy of Nonlinear Manifold Learning in Malware Image Pattern Analysis. 2018 17th IEEE International Conference on Machine Learning and Ap-

plications (ICMLA), Orlando, FL, 2018, pp. 1095-1102..

Nadler, B., Lafon, S., Coifman, R., Kevrekidis, I.: Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems 18, pp. 955–962. MIT Press, Cambridge (2006).

G. Mishne and I. Cohen, "Multiscale Anomaly Detection Using Diffusion Maps," in IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 1, pp. 111-123, Feb. 2013.

Rodrigues, P. An introduction to diffusion maps and their use in time series analysis. VIBS team meeting, Mar. 2017. https://plcrodrigues.github.io/riemann-lab/pdfs/presentation_vibs.pdf.

BallotPedia. List of Current Members of the US Congress.
https://ballotpedia.org/List_of_current_members_of_the_U.S._Congress