2017

Daily Traffic Flow Pattern Recognition by Spectral Clustering

Matthew Aven
Claremont McKenna College

Claremont McKenna College

Daily Traffic Flow Pattern Recognition by Spectral Clustering

submitted to
Professor Deanna Needell

by
Matthew Aven for

Senior Thesis
Spring 2017
April 24, 2017

Daily Traffic Flow Pattern Recognition by Spectral Clustering

Matthew Aven

Abstract

This paper explores the potential applications of existing spectral clustering algorithms to real life problems through experiments on existing road traffic data. The analysis begins with an overview of previous unsupervised machine learning techniques and constructs an effective spectral clustering algorithm that demonstrates the analytical power of the method. The paper focuses on the spectral embedding method's ability to project non-linearly separable, high dimensional data into a more manageable space that allows for accurate clustering. The key step in this method involves solving a normalized eigenvector problem in order to construct an optimal representation of the original data.

While this step greatly enhances our ability to analyze the relationships between data points and identify the natural clusters within the original dataset, it is difficult to comprehend the eigenvalue representation of the data in terms of the original input variables. The later sections of this paper will explore how the careful framing of questions with respect to available data can help researchers extract tangible decision driving results from real world data through spectral clustering analysis.

1 Introduction

1.1 Motivation

This paper focuses on the construction of spectral clustering algorithms and their application to road traffic datasets. Clustering algorithms are a class of unsupervised machine learning techniques that that attempt to separate data without known labels into optimal groups based on the relationships between points based on any number of parameters. The end goal is to create groups such that any two points in the same group are as similar as possible while any two points not in the same group are as dissimilar as possible. Where supervised classification techniques offer methods of separating new data based on a training set in which each data point is known to belong to a certain group, unsupervised clustering techniques construct similar methods of separation by assigning more complex data into optimal groups, and determining what characteristics would place new data points into these groups. Specifically, this algorithm will provide an optimal solution to the task of separating high dimensional, non-linearly separable and unlabeled data into optimal natural clusters. Supervised techniques such as linear regressions, linear classifiers, and support vector machines are well understood and provide powerful results, however, they require a large amount of detailed information about our data set to provide such results. Since this information is sparse and difficult to accumulate, there is ample motivation to construct efficient clustering algorithms that can uncover optimal partitions of data without preexisting labels. Further, the tangible implications of unsupervised methods such as the algorithm presented in this paper are encompass a broad scope of important issues that can benefit significantly from these results.

A key breakthrough in the search for such clustering solutions came with the discovery of k-means clustering in 1982 [Llo82]. K-means clustering was developed to solve the problem of choosing k centers and a set of n data points in $\mathbb{R}^d$ in order to minimize the total squared distance of the points in a given cluster from the cluster's center. Due to the complexity of this problem, this challenge is NP-hard to solve precisely. However, Stuart Lloyd provided a breakthrough iterative algorithm in 1982 that has been used in research for decades and inspired numerous innovations towards completely solving the clustering problem [Llo82]. This breakthrough offers a succinct solution that is guaranteed to terminate and often converges fairly efficiently. However, the accuracy of this solution is often left to chance and can be unreliable. David Arthur and Sergei Vassilvitskii provide a key improvement to the algorithm that allows the centers of each cluster to be chosen based on a points squared distance to preexisting centers rather than at random [AV07]. These two breakthroughs provide key context in order to understand the development of more advanced data clustering techniques in addition to providing a powerful tool to separate interesting data sets once they have been projected into a manageable space. This paper will explore the k-means algorithm in more depth in a following section.

While k-means algorithms provide powerful results on many datasets, problems arise when dealing with high dimensional data sets that do not contain trivial clusters, or clusters that can be linearly separated with ease. The algorithm developed in this paper will focus on two key challenges that arise when applying k-means to complex data. First, the run time complexity of the algorithm becomes unmanageable when dealing with high dimensional data sets, so we need a method to efficiently compare the relationships between data points along multiple parameters. Second, k-means clustering relies solely on the euclidean distance between two points in their native space, so the resulting clusters are only accurate when the points in each cluster are in close proximity to each other and distant from any point in another cluster. Figure 1 provides a sample dataset in which k-means fails to accurately partition the data while spectral clustering separates the clusters easily. The key challenge to solving these problems is finding a way to compare individual data points to each other rather than to an arbitrary cluster center. By constructing graphs to represent the relationship between each point in a given dataset, it is possible to compute powerful and accurate clustering results using linear algebra techniques. This paper will focus on a normalized spectral clustering technique developed by Jianbo Shi and Jitendra Malik in 2000 that projects n unlabeled data points in $\mathcal{R}^d$ into an eigenvector space that allows us to effectively separate complex data using more basic techniques such as k-means clustering [SM00]. Later sections of this paper will expound this algorithm and any necessary adjustments before exploring the potential applications of such methods.

To demonstrate the algorithm, I have selected a sample of road traffic data from New York City that captures the traffic flows of intersections during hour intervals throughout the day. This data demonstrates the spectral clustering algorithm's ability to project high dimensional, non linearly separable data into a more manageable dimension that will allow us to easily separate the data by simple clustering methods. This experiment follows previous work applying spectral based clustering methods to similar road traffic issues, network traffic recognition, and even the attrition problem [VKG08, VMP03, CJR$^+$12]. The dataset encapsulates every intersection as a 23 dimensional vector, where each parameter represents the traffic flow during a given time period throughout the day. There is no way to illustrate this type of data, let alone dream up a way to visualize the relationships between data points in this form. Using our eigenvalue interpretation of the data and the subsequent clustering of these points, it is possible to view the relationships between data points and the clear partitions that exists within the data set. This experiment will not incorporate the multi level learning methods used in some previous experiments, this data set offers nearly a million data points to train and test the data using spectral clustering methods in order to uncover a key dimension of short term road traffic patterns.

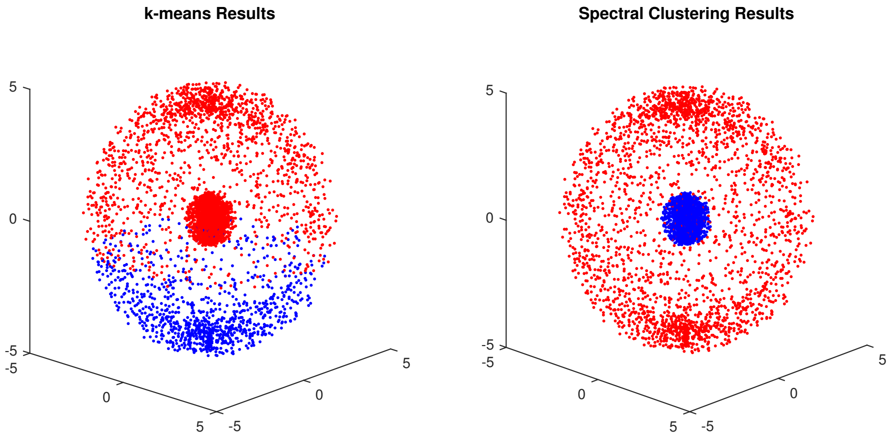

Figure 1: Both figures present attempts to cluster two concentric spheres that clearly form two related groups. The figure on the left used the k-means algorithm which was unable to accurately identify the natural clusters since the points are not linearly separable. The figure on the right demonstrates the ability of the spectral clustering algorithm to accurately identify the related points even though the data is not linearly separable in its original form.

1.2 K-means Clustering

This section will provide a comprehensive overview of the k-means clustering algorithms that were introduced earlier in order to illustrate the broad goal of clustering algorithms and introduce a powerful tool to assist the spectral clustering algorithm that is the focus of this paper. Recall the goal of unsupervised clustering algorithms which is to separate similar clusters of data points and construct an optimal method for separating new data points with the same parameters [Llo82]. k-means clustering attempts to uncover these groups by separating n data points into k clusters such that every point in a cluster is at a distance from the cluster's center similar to the mean distance from the center for the whole cluster. An optimal solution to this problem minimizes the total distance between points and the center of their respective cluster. In other words, this solution finds k centers C that minimizes the potential function

$$\phi = \sum_{x \in X} min_{c \in C} \|x - x\|^2 \tag{1}$$

for any given set of training data X. This is an NP-hard problem to solve given the high number of possible solutions for any data set. However, the first k-means algorithm offers a relatively efficient method for finding these optimal clusters given an initial set of k means and cooperative data [Llo82]. The proposed algorithm iterates between two repeating steps until the data converges to a solution. The algorithm begins with k random cluster centers from our input data. The first step, called the assignment step assigns each data point to the cluster whose mean yields the minimal squared euclidean distance between other points in the cluster and itself. After assigning each point to a cluster, the cluster centers are recalculated to be the means of each point in the clusters generated in the first step. These two steps are repeated until the process converges to a single solution. The algorithm has converged when the clusters stop changing with each iteration. This breakthrough algorithm provides powerful results but still faces some computational challenges.

The k-means algorithm utilized by this paper and many researchers today takes advantage of a simple seeding strategy that vastly improves the performance and accuracy of the k means algorithm [AV07]. Rather than beginning the process by selecting k random cluster centers, this strategy picks the first cluster center c_1 uniformly at random and then precedes to pick the following $k-1$ centers with an adjusted probability density function. The following cluster centers are chosen uniformly with probability $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$ where $D(x)$ is the shortest distance from the point x to the closest center c_i already chosen. Figure 2 demonstrates the power of this k-means algorithm in $\mathbb{R}^2$. This algorithm is helpful for the final classification of our data after performing spectral embedding and provides an idea of the optimizing problems encountered when dealing with unsupervised classification.

2 Spectral Clustering

Spectral clustering builds on the progress of k-means and other clustering algorithms by introducing graphs in order to more accurately characterize the relationships between entries in the data set X. A graph $\Gamma = (V, E)$ is a collection of vertices V and edges E. For the purpose of clustering data points, the data set X will form the vertices of our graph and the set of edges E will represent relationships between each entry in X [VL07,

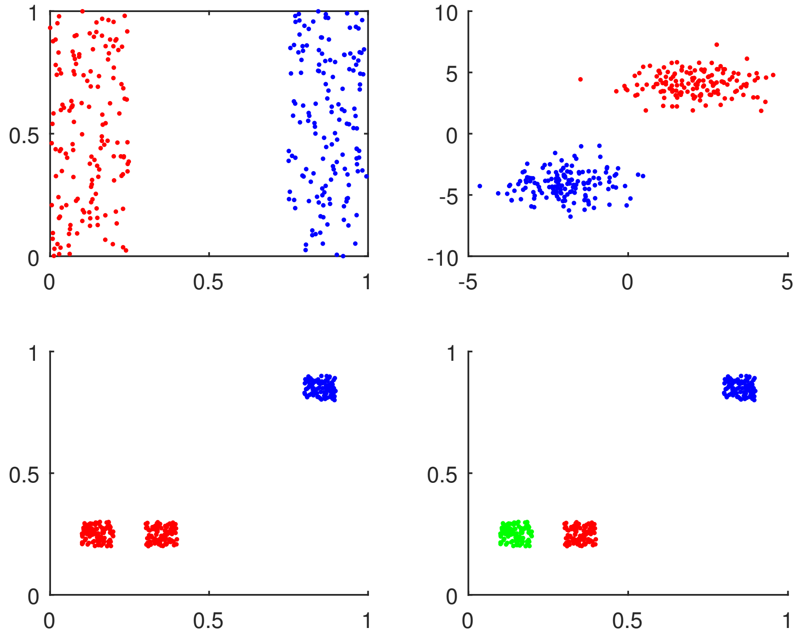

Figure 2: These figures provide examples of globular data clusters that the k-means algorithm can identify with ease. The top figures consist of two obvious point clusters. The bottom two figures contain three clusters of points where two clusters are closely related. The k-means algorithm successfully separates the points into accurate 2 and 3 cluster representations. These are the types of data clusters that we hope to generate with the spectral embedding algorithm in order to classify more complex data using k-means.

SM00]. More specifically, the spectral clustering algorithm presented below utilizes a weighted graph where each entry $w_{i,j}$ represents the degree of similarity between the ith and jth entries in X. Once the data is characterized in this manner, the spectral clustering algorithm can perform simple and effective linear algebra techniques to cluster data with much higher accuracy and flexibility than prior techniques. The technique described in the following section is a normalized method published in 2002 although alternative unnormalized and normalized methods exist. [SM00, NJW$^+$01, VL07]

2.1 Spectral Embedding

While k-means clustering offers an adequate solution for low dimensional data sets with obvious natural partitions, this spectral clustering algorithm attempts to separate high dimensional, non-linearly separable data that would almost always cause k-means methods to fail as illustrated in Figure 1. The key step in this process is called spectral embedding and involves constructing an alternate characterization of our data X that will represent our data in a more tangible and easily separable space. Specifically, the spectral embedding process projects our original data set $X \in \mathbb{R}^d$ into an eigenvector representation of the original data features. Figure 3 depicts an artificial data set in R^3 next to its two-dimensional eigenvector representation. Even though the two groupings in the original data, a k-means algorithm would clearly fail to find this partition given the close proximity of the two clusters. However, after representing the data as a graph and projecting it into a two-dimensional eigenvector space, the resulting characterization of the data reveals two obvious groups of points that could be easily separated by a k-means algorithm.

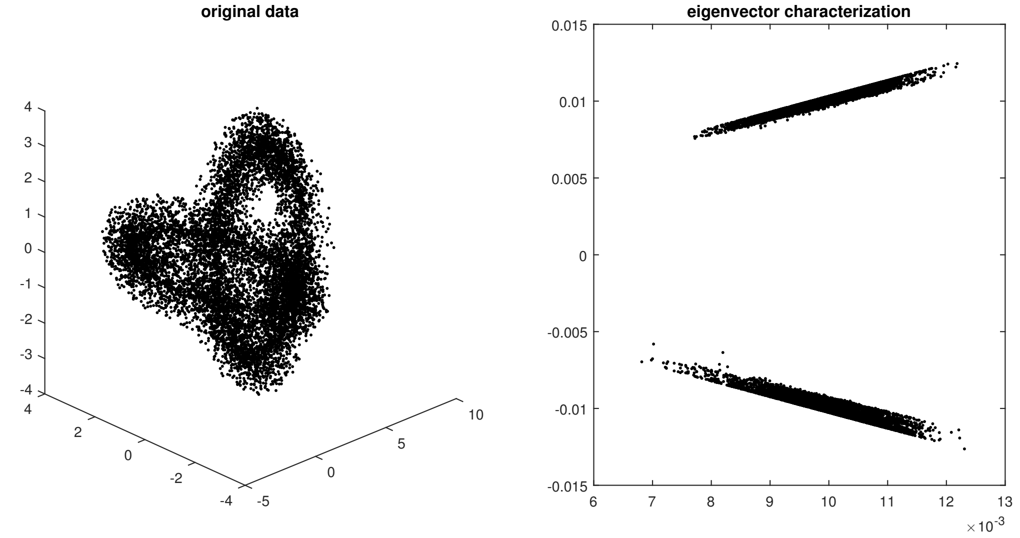

Figure 3: Both figures provide a visual representation of the same data points. the figure on the left portrays the original data in 3 dimensions. The figure on the right illustrates the eigenvector characterization of this data generated by the spectral embedding algorithm. The separate vectors on the right represent the relationships between points in each of the two rings from the original data.

The spectral embedding algorithm that is the focus of this paper is modeled after the algorithm proposed by Shi and Malik [SM00]. This spectral embedding process will take an input data matrix $X \in \mathbb{R}^{n \times d}$ and produce a matrix $U \in \mathbb{R}^{n \times k}$ containing the desired k eigenvector representation of each data point by following four main steps. First, construct a similarity matrix $S \in \mathbb{R}^{n \times n}$. There are numerous methods for constructing such similarity matrices. This discussion of spectral embedding will utilize the Gaussian kernel to construct any necessary adjacency matrices. That is, every entry $s_{i,j} \in S$ is simply the Euclidean distance from the ith point to the jth point or $\|x_i - x_j\|$. Once our data is in this form, we can construct the resulting weighted adjacency matrix

$$W_{i,j} = e\frac{-\|x_i - x_j\|^2}{2\sigma_x^2}. \tag{2}$$

The Gaussian kernel is utilized to project each data point into $\mathbb{R}^n$ space and evaluate any given data points relationship with every other point in X. After constructing the weighted adjacency matrix W, we can compute the the unnormalized Laplacian representation L of our data. The normalized matrix L is computed from the weighted adjacency matrix W so that

$$L = I - (D^{\frac{-1}{2}} W D^{\frac{-1}{2}}), \tag{3}$$

where D is a diagonal matrix that represents the overall level of connectivity of each data point so that

$$D_{i,i} = \sum_{j=1}^{n} W_{i,j}. \tag{4}$$

The Laplacian matrix $L \in \mathbb{R}^{n \times n}$ contains the intricate relationships among entries in the original data set X, however, it is of little help to us in such a dimension. The third step of this spectral embedding algorithm sets up the following generalized eigenvector problem.

$$Lu = \lambda Du \tag{5}$$

Solving for the first k generalized eigenvectors $\{u_1, u_2, ..., u_k\} \in U \in \mathbb{R}^{n\times k}$ projects the matrix L into a more manageable dimension and forms the desired eigenvector characterization of the original data set X. To represent data in two dimensions and make clustering into two groups manageable, construct the matrix A by concatenating the eigenvectors corresponding to the two smallest eigenvalues of W according to the following algorithm.

Algorithm 1 Spectral embedding)

```
procedure ($\boldsymbol{X}$, $\sigma$)        ▷ $n \times d$ training data matrix $\boldsymbol{X}$, tuning parameter $\sigma$
    Initialize Construct the similarity matrix $\boldsymbol{W}$ from (2), degree matrix $D$ from (4), and normalized Laplacian from (3)
    Compute $\boldsymbol{A}$, the two eigenvectors corresponding with the two smallest eigenvalues of $L$
    Output $\boldsymbol{A} \in \mathbb{R}^{n\times 2}$   ▷ $\boldsymbol{A}$ is a two dimensional representation of the original data $\boldsymbol{X}$
end procedure
```

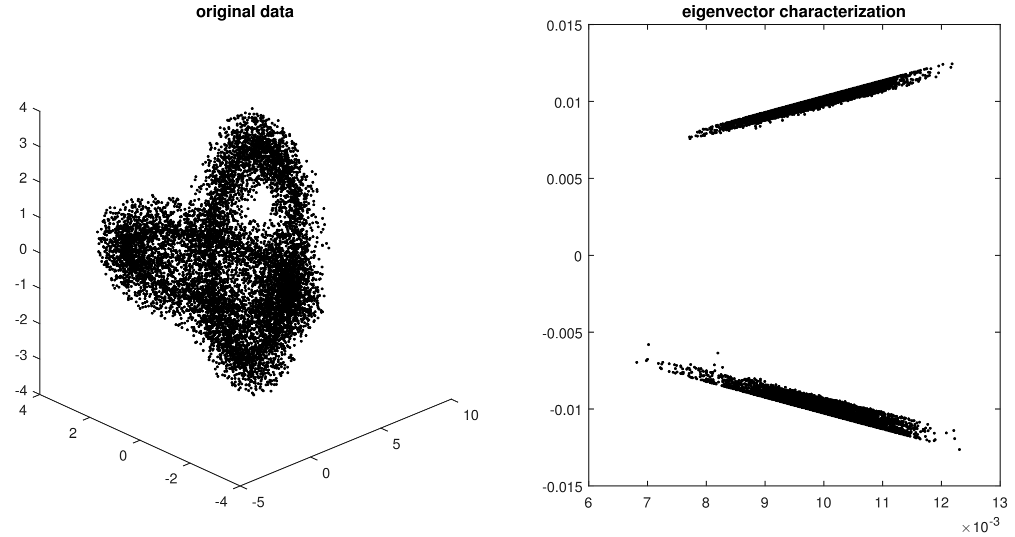

Figure 4: Both figures provide a visual representation of the same data points. the figure on the left portrays the original data in 3 dimensions. The figure on the right illustrates the eigenvector characterization of this data generated by algorithm 1. The separate vectors on the right represent the relationships between points in each of the two rings from the original data.

2.2 Spectral Clustering

After the spectral embedding step is complete, each row vector y_i of $\boldsymbol{A}$ corresponds to a unique entry $x_i \in \boldsymbol{X}$. Now the new point representations y_i can be clustered using k-means in order to form the optimal groups that could not be discerned in the data's original form. The matrix U can also provide a helpful visualization tool for high dimensional data points. Although the eigenvector representation of each data point will not correspond to the input parameters, the 2 and 3 eigenvector representation of a dataset helps visualize the intricate relationships that exist in the original data. At this stage, the spectral embedding algorithm has produced a characterization of the data that allows for convenient visualization and classification by k-means.

Algorithm 2 Normalized spectral clustering according to Shi and Malik(2000))

```
procedure ($\boldsymbol{X}$, $\sigma$)          ▷ $n \times d$ training data matrix $\boldsymbol{X}$, tuning parameter $\sigma$
    Initialize Construct the similarity matrix $\boldsymbol{W}$ from (2), degree matrix $D$ from (4), and normalized Laplacian from (3)
    Compute $\boldsymbol{v}$, the eigenvector corresponding with the second smallest eigenvalue of $L$
    Use k-means to cluster the eigenvector assigning cluster indices to data entries
    Output $\boldsymbol{idx} \in \mathbb{R}^{n\times 1}$          ▷ $\boldsymbol{idx}$ holds the cluster assignment of each data point
end procedure
```

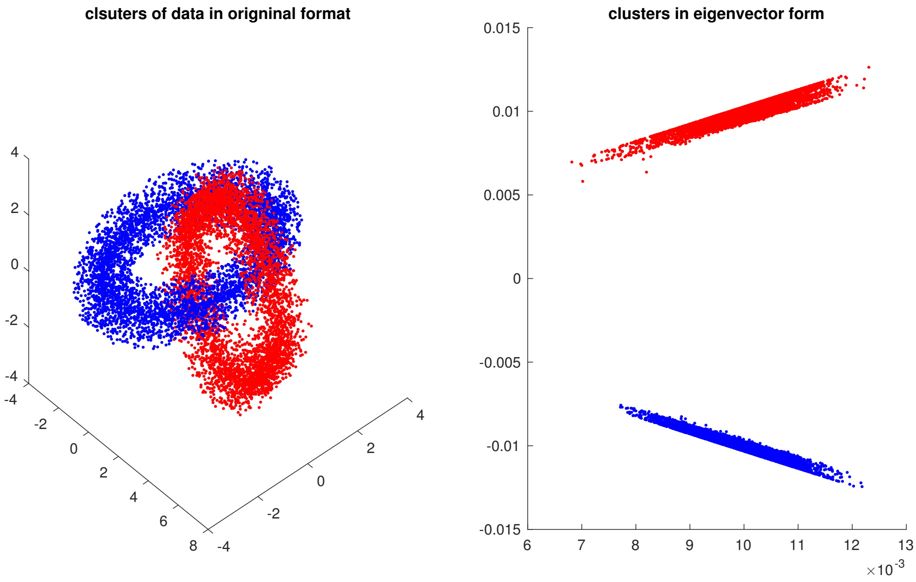

Figure 5: These figures illustrate the output of algorithm 2 in its original form on the right and in eigenvector form on the left. The figure on the right is the data representation that clustering is performed on to generate the corresponding clusters in the original data on the left.

2.3 Linear Separating Hyperplane

After generating the eigenvector representation of the dataset X and classifying each entry into two optimal partitions, the next goal is to define a method for classifying new data points into the groups determined by the training data. In order to accomplish this goal, we will find what is called the best linear separating hyperplane. This hyperplane x takes the form

$$x = A^{-1}b \tag{6}$$

where $A = \begin{bmatrix} A_{C_1} \\ -A_{C_2} \end{bmatrix}$represents the training data set in eigenvector form and $b \in \mathbb{R}^{n\times 1}$ is a column vector that assigns 1 to indices of A that belong to the first cluster C_1 and -1 to elements of the second cluster C_2. Using the clusters generated by our spectral clustering algorithm, simple linear algebra can be used to solve for the separating hyperplane $x \in \mathbb{R}^{1\times d}$ where the dimension d is the number of eigenvectors chosen to represent each data point. The examples in this section will utilize the two smallest eigenvectors so the hyper plane will be represented by a 2-dimensional row vector. Once this vector is calculated, we can pass in new data points a_i in their eigenvector form and classify them into the clusters C_1 and C_2 based on the clustering rule

$$\begin{aligned} a_i \to C_1 \text{ if } \langle a_i x\rangle < 0 \text{ or,} \\ a_i \to C_2 \text{ if } \langle -a_i x\rangle > 0. \end{aligned} \tag{7}$$

Thus we can pass in new data points and classify them into clusters efficiently by projecting them into the eigenvector space and solving a simple inner product with the hyperplane. Figure 6 provides an illustration of how this separating hyperplane can be applied to nonlinearly separable data.

Algorithm 3 Linear separating hyperplane for two clusters)

```
procedure ($\boldsymbol{A}$, $\boldsymbol{idx}$)    ▷ $n \times d$ training data matrix $\boldsymbol{A}$ in eigenvector form, cluster assignments $\boldsymbol{idx}$
    Construct the vector $\boldsymbol{b}$ so that $\boldsymbol{b_i} = 1$ if $X_i \in C_1$ or 0 if $x_i \in C_2$.
    Solve for $\boldsymbol{x}$ by multiplying $\boldsymbol{A^{-1}} * \boldsymbol{b}$ according to equation (6).
    Output $\boldsymbol{x} \in \mathbb{R}^{1\times 2}$▷ $\boldsymbol{x}$ is the hyperplane that allows us to assign new data points to a cluster based on equation (7)
end procedure
```

3 Experimental Data: New York City Intersection Case Study

This section will focus on the analysis of this spectral clustering algorithm when applied to real world data sets. While other statistical models, linear regressions, and econometric methods that utilize linear algebra provide direct inference between their results and the population that they model, the relationships uncovered by this spectral clustering algorithm can be difficult to interpret in terms of tangible results. Unlike an econometric or statistical regression that describes a variable of interest as a result of relationships between parameters within certain degrees of accuracy, the output of this spectral clustering algorithm consists of groups with unknown labels that are determined by intricate relationships that are difficult to conceive. Previous papers have argued that spectral clustering methods are well suited for analyzing traffic networks of different types. A 2003 paper utilizes spectral clustering methods to analyze network traffic data while multiple papers describe spectral clustering based methods for analyzing real time traffic flow [HVA03, AMP06, PRH99]. The following sections will follow the lead of

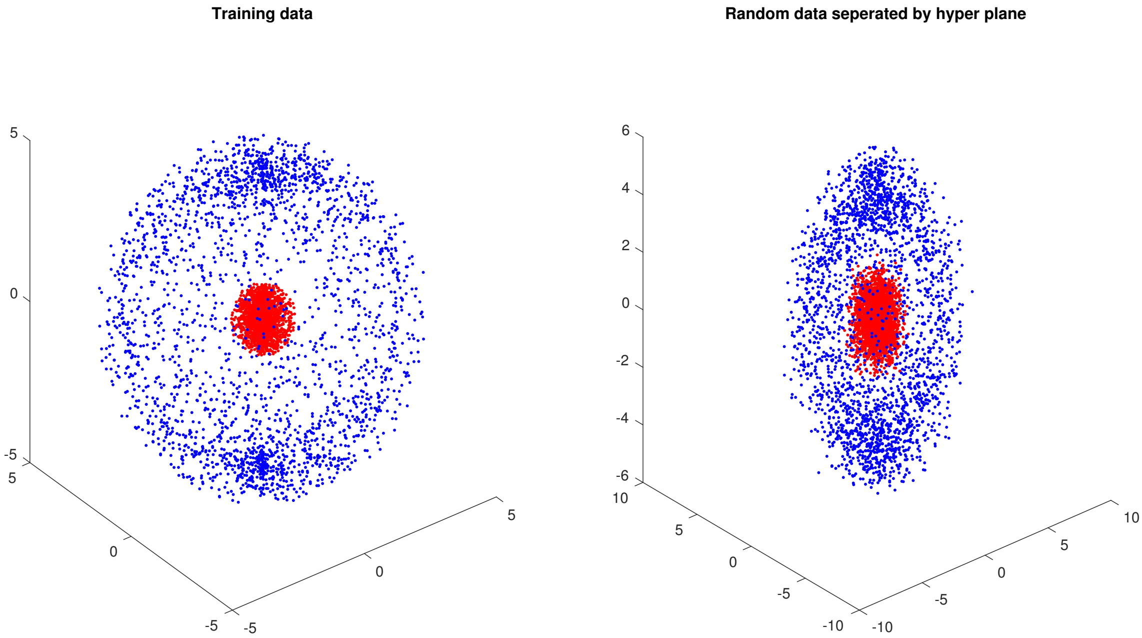

Figure 6: The figure on the left represents the training data used to identify clusters and train a separating hyperplane. The figure on the right presents randomly generated data that has been separated into optimal clusters based on the hyperplane generated by the training data.

these articles by applying the spectral clustering algorithm mentioned above to road traffic data in order to understand the relationships guiding different traffic phenomena.

The dataset of intest consists of simple traffic count data captured by cameras at 1287 intersections throughout New York City. The traffic count is recorded for every hour throughout the day so every entry in our training set A is a 24 dimensional row vector where each parameter represents the average traffic flow for that hour throughout the year 2013. Thus the training set A is a 1287×24 matrix. The high dimension of these data entries make it difficult to visualize and evaluate the relationships between the daily traffic distribution across the intersections in the dataset. Without any prior knowledge about these specific intersections, the spectral clustering algorithm should be able to project the data into its eigenvector representation revealing the relationships in the traffic count data. Figure 7 contains the 2-dimensional eigenvector representation of each entry and the optimal clusters generated by the spectral clustering algorithm. The eigenvector representation of the data is contains three obvious clusters that are easily separated by k-means, however, the illustration provides no insight into what relationships are driving the formation of these clusters.

The methods utilized to project the original data into a manageable format and construct optimal clusters offer almost no assistance in discerning what relationships drive the formation of such clusters. Unless a particular data set and spectral clustering algorithm are designed to analyze a narrowly defined issue, it can be nearly impossible to draw tangible conclusions from the algorithm's results. However, the chronological nature of this particular dataset offers a starting point towards understanding what relationships are driving the formation of the three obvious clusters illustrated above. Figure 8 below helps illustrate what the each cluster represents. Both figures represent

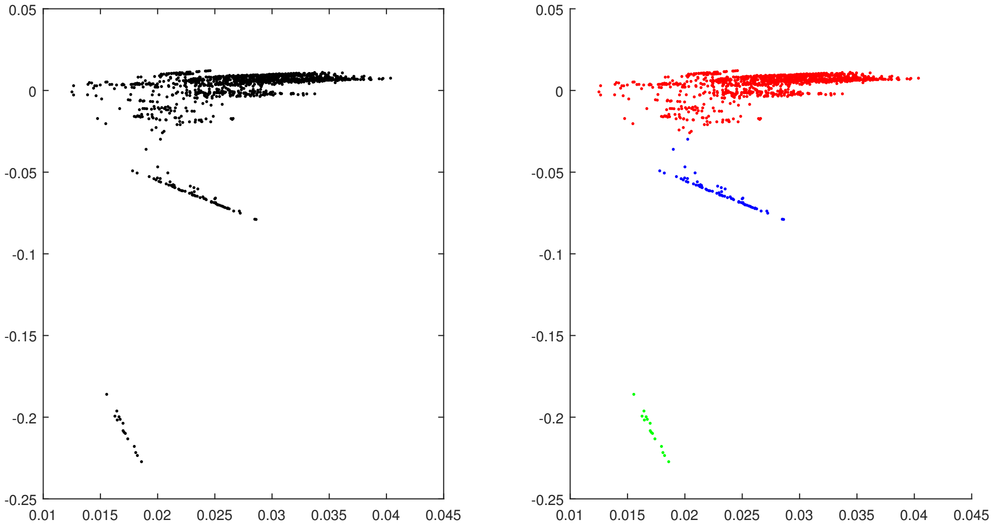

Figure 7: The figure on the left illustrates the un-clustered eigenvector representation of the 24 dimensional traffic count data. The figure on the left presents the same data in the three clustered generated by k-means clustering

the trajectory of traffic flows throughout the day for each intersection in the data set. The trajectory of every intersection is depicted on the left. At first glance, it appears that the algorithm is organizing the intersections into high traffic, medium traffic and light traffic clusters. However, if this was the the core of the cluster generation, then the results cluster results would not be very accurate given the amount of overlapping present in the first figure. However, upon closer inspection one can discern a unique shape to the trajectories in each cluster. The figure on the right plots the average traffic flow of the intersections in each cluster along with the two standard deviation error bound at each time interval. Now the unique shapes of daily traffic fluctuation in each cluster is beginning to take shape with significant statistical evidence. Further, note that 1202 data points were assigned to the red cluster while only 70 and 15 were assigned to the blue and green clusters respectively. Thus, while it is impossible to verify that the clusters are being formed by these fluctuations, this data set demonstrates the spectral clustering algorithm's ability to discern vital relationships in data even when they non obvious and relatively scarce.

4 Conclusion

After exploring the challenges of machine learning and the construction of spectral clustering algorithms, the novel power of the technique is obvious, but the challenge of applying the method to real world problems using real world data is still formidable. While spectral embedding of data and its subsequent clustering can organize complex unlabeled data into groups of similar and dissimilar entries, it is difficult to accurately understand the exact relationships driving such cluster formations without extensive data available. Since the method clusters data based on a representation that does not correspond to the input parameters, it is important to frame questions narrowly based on the data available in order to draw effective conclusions from these results. The

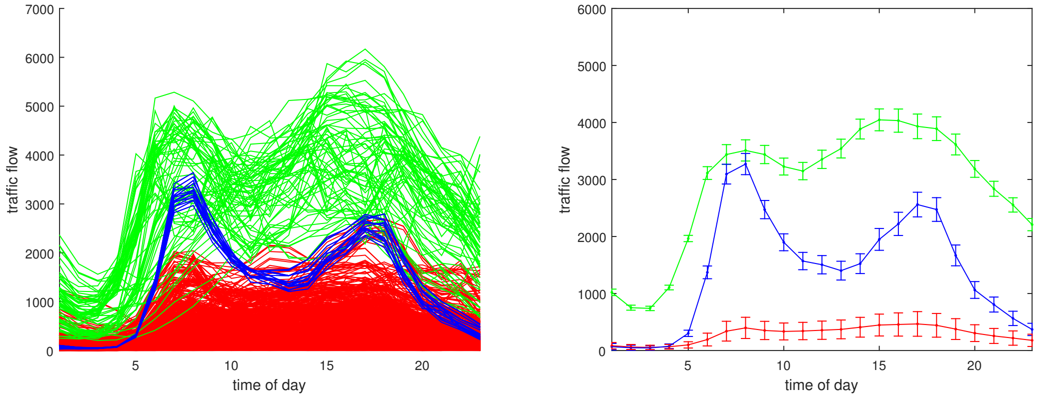

Figure 8: Both figures illustrate traffic density throughout the day. The y-axis in each figure represents the number of vehicles that pass through an intersection during any given hour while the x-axis represents the hours of the day. The figure on the left presents the trajectory of every intersection in the training data set. The figure on the right illustrates the average trajectory of each cluster surrounded by a 1 standard deviation error bound.

traffic applications explored in this paper demonstrate spectral clustering's ability to uncover specific traffic trends based on data that could not be analyzed with traditional statistical supervised learning methods. However, our understanding of the relationships that the method uncovers are cursory at best. The ability to cluster new data entries into the newly constructed clusters is helpful, but it is difficult to turn these results into meaningful knowledge without some guesswork and a little luck. That being said, the future of spectral clustering and other unsupervised machine learning techniques is bright as computing resources and the availability of quality data continues to expand.

Road traffic learning methods continue to face the two-fold challenge of obtaining informative data to analyze and understanding how to frame problems so that the learning methods can generate decision driving information. Fortunately, the availability of both reliable data and effective learning algorithms are increasing steadily. Researchers are using spectral clustering algorithms to analyze diverse traffic datasets, even going as far as to analyze the relationships between intersection trajectories recorded by video cameras [AMP06]. Others are working to incorporate spectral methods in order to capture numerous parameters in a single traffic model through the use of multi-layer neural networks [PRH99, QPL06]. Each of these efforts represent leaps in the ability to identify traffic patterns accurately and efficiently. Moving forward, the increasing access to data and computing power will continue to drive progress in traffic learning, however, the ability to incorporate multiple types of traffic data into single learning methods will vital for realizing the full power of traffic learning algorithms. As these methods advance and resaerchers continue to collaborate and expand their models, traffic learning has the potential to transform the way we travel and help usher in a new age of efficiency on our roads.

References

[AMP06] S. Atev, O. Masoud, and N. Papanikolopoulos. Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *IROS*, pages 4851–4856, 2006.

[AV07] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[CJR+12] B. Cung, T. Jin, J. Ramirez, A. Thompson, C. Boutsidis, and D. Needell. Spectral clustering: An empirical study of approximation algorithms and its application to the attrition problem. *arXiv preprint arXiv:1211.3444*, 2012.

[HVA03] N. Hohn, D. Veitch, and P. Abry. Cluster processes: a natural language for network traffic. *IEEE Transactions on Signal processing*, 51(8):2229–2244, 2003.

[Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[NJW+01] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001.

[PRH99] D. Park, L. R. Rilett, and G. Han. Spectral basis neural networks for real-time travel time forecasting. *Journal of Transportation Engineering*, 125(6):515–523, 1999.

[QPL06] C. Quek, M. Pasquier, and B. B. S. Lim. Pop-traffic: a novel fuzzy neural approach to road traffic analysis and prediction. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):133–146, 2006.

[SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[VKG08] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. Temporal evolution of short-term urban traffic flow: A nonlinear dynamics approach. *Computer-Aided Civil and Infrastructure Engineering*, 23(7):536–548, 2008.

[VL07] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[VMP03] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, 2003.