SPECTRAL CLUSTERING AND VISUALIZATION: A NOVEL CLUSTERING OF FISHER'S IRIS DATA SET*

DAVID BENSON-PUTNINS†, MARGARET BONFARDIN‡, MEAGAN E. MAGNONI§, AND DANIEL MARTIN¶

Advisors: Carl D. Meyer[1] and Charles D. Wessell[2]

Abstract. Clustering is the act of partitioning a set of elements into subsets, or clusters, so that elements in the same cluster are, in some sense, similar. Determining an appropriate number of clusters in a particular data set is an important issue in data mining and cluster analysis. Another important issue is visualizing the strength, or connectivity, of clusters.

We begin by creating a consensus matrix using multiple runs of the clustering algorithm k-means. This consensus matrix can be interpreted as a graph, which we cluster using two spectral clustering methods: the Fiedler Method and the MinMaxCut Method. To determine if increasing the number of clusters from k to $k+1$ is appropriate, we check whether an existing cluster can be split. Finally, we visualize the strength of clusters by using the consensus matrix and the clustering obtained through one of the aforementioned spectral clustering techniques.

Using these methods, we then investigate Fisher's Iris data set. Our methods support the existence of four clusters, instead of the generally accepted three clusters in this data.

Key words. cluster analysis, k-means, eigen decomposition, Laplacian matrix, data visualization, Fisher's Iris data set

AMS subject classifications. 91C20, 15A18

1. Introduction. Clustering is the act of assigning a set of elements into subsets, or clusters, so that elements in the same cluster are, in some sense, similar. For many, the internet is a tool used to do everything from shopping to paying bills. One can shop for clothes, groceries, movies, and more. A common theme throughout these websites is the product suggestions that appear when you buy or view an item. These product suggestions form one of the many applications of data mining and cluster analysis. Companies such as Netflix use the concept of cluster analysis to create product suggestions for their customers. The better the suggestions, the more likely the customer is to buy products.

Cluster analysis can be applied to many areas including biology, medicine, and market research. Each of these areas has the potential to amass large amounts of data. There are dozens of different methods used to cluster data, each with its own shortcomings and limitations. One significant problem is determining the appropriate

*This research was supported in part by NSF Grant DMS 0552571 and NSA Grants H98230-08-1-0094 and H9823-10-1-0252

†Department of Mathematics, University of Michigan, Ann Arbor, MI 48104, USA (dputnins@umich.edu)

‡Department of Mathematics, Washington University, St. Louis, MO 63105, USA (mlbonfar@wustl.edu)

§Department of Mathematics, Rensselaer Polytechnic Institute, Troy, NY 12180, USA (magnom@rpi.edu)

¶Department of Mathematics, Davidson College, Davidson, NC 28035, USA (dnmartin@davidson.edu)

[1]Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA (meyer@ncsu.edu,http://meyer.math.ncsu.edu/)

[2]Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA (cdwessel@ncsu.edu)

number of clusters, which will be denoted throughout this document as k. The correct choice of k is often ambiguous. If an appropriate value of k is not apparent from prior knowledge of the properties of the data set, it must somehow be chosen. Another important issue is visualizing the strength, or connectivity, of clusters. It is often impossible to visualize the original data if it is high-dimensional, so the ability to visualize the strength of a clustering would benefit the applications of data mining and cluster analysis.

1.1. Contributions. Our paper addresses the issues of determining an appropriate number of clusters and of visualizing the strength of these clusters. We begin by creating a consensus matrix (see Section 2.2) using multiple runs of the clustering algorithm k-means (see Section 2.1). This consensus matrix can be interpreted as a graph, which we cluster using two spectral clustering methods: the Fiedler Method and the MinMaxCut Method. To determine if increasing the number of clusters from k to $k+1$ is appropriate, we check whether an existing cluster can be split. Finally, we visualize the strength of clusters by using the consensus matrix and the clustering obtained through one of the aforementioned spectral clustering techniques.

We then use our methods to investigate the Iris flower data set. Our results are surprising; the Iris flower data set is generally accepted to have three clusters, but our methods support the existence of four clusters. We determine that the cluster corresponding the *Iris setosa* flowers can be split into two clusters. This clustering is shown to be strong through the use of the visualization tool.

The remainder of the paper is organized as follows. Section 2 introduces the concept of consensus clustering, Sections 3 and 4 discuss the Fiedler Method and the MinMaxCut Method, respectively. We describe our method to determine k in Section 5 and how we visualize the strength of clusters in Section 6. In Section 7, we describe our experimental results and in Section 8, our conclusions.

2. Consensus Clustering. Over time, dozens of different clustering methods have been developed for data analysis. It is frequently unclear which clustering method to use on a particular data set, since each method has shortcomings and limitations [11]. Furthermore, it can be difficult to be confident in the accuracy of the clusters provided by the clustering method since many clustering algorithms, such as k-means, give non-unique answers. Consensus clustering has emerged as a potential solution to these problems [14]. The goal of consensus clustering is to find a single (consensus) clustering that is stronger than the existing clusterings. Specifically, the goal is to find a clustering such that, for elements in a particular cluster, the frequency with which these elements were placed together in previous clusterings is maximized. In addition to representing the consensus over multiple runs of a clustering algorithm such as k-means, this method changes the nature of the clustering problem from points in Euclidean space to a graph (see Section 2.3).

2.1. The *k*-means Clustering Method. The k-means clustering method requires the use of a distance metric. In our investigation of the Iris flower data set, we use k-means clustering with the cosine metric. The cosine metric is defined as

$$\cos\,\theta = \frac{\langle \mathbf{x}|\mathbf{y}\rangle}{||\mathbf{x}||\;||\mathbf{y}||} \tag{2.1}$$

Equation 2.1 measures the angle between nonzero vectors $\mathbf{x}$, $\mathbf{y}$ in a real inner-product space V. For more information about angles, inner products, and vector norms, see [13].

The k-means clustering algorithm partitions n elements into k clusters. The first step in the k-means clustering algorithm is to initialize k centroids. Centroids are n-dimensional points in Euclidean space. The initial centroid locations are chosen randomly among the coordinates of the data points themselves. The second step is to measure the distance from every data point to each centroid. Each data point is then clustered with the nearest centroid and the mean of the values of the elements in each cluster becomes the new centroid. The previous two steps are repeated until the clustering converges (i.e. the centroid locations do not change) [11]. Since the final clustering depends on the choice of initial centroid locations, the clustering algorithm does not give a unique answer.

2.2. Building the consensus matrix. In order to do consensus clustering, we must first build a consensus matrix. In the consensus matrix, the (i, j) entry reflects the proportion of clusterings in which element i was clustered with element j. For example, consider two different clusterings of elements A, B, C, and D. In the first clustering, elements A, B, and C cluster together and element D is a singleton cluster. In the second clustering, elements B, C, and D cluster together and element A is a singleton cluster. The square, symmetric matrix below is the consensus matrix for these clusterings:

$$\begin{array}{c} \\ \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \end{array} \begin{array}{c} \begin{array}{cccc} \text{A} & \text{B} & \text{C} & \text{D} \end{array} \\ \begin{pmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 1 & 0.5 \\ 0.5 & 1 & 0 & 0.5 \\ 0 & 0.5 & 0.5 & 0 \end{pmatrix} \end{array}$$

Elements A and B cluster together in the first, but not the second, clustering. Thus, there is a 0.5 in the (A,B) and (B,A) entries of the consensus matrix. A similar process is used to obtain the remaining entries in the consensus matrix. There are zeroes on the diagonal by convention.

2.3. Visualizing the consensus matrix. We can interpret the consensus matrix as an undirected, weighted graph. For more information on graphs, see [3]. If we consider A, B, C, and D to be vertices, we can interpret the consensus matrix from Section 2.2 to be the weighted graph in Figure 2.1. The (i, j) entry of the consensus matrix represents the weight of the edge connecting vertices i and j. The Fiedler Method and the MinMaxCut Method are clustering algorithms that partition this weighted graph to create clusters (see Sections 3 and 4).

3. Fiedler Method. The Fiedler Method partitions graphs to form clusters. This method and the ideas behind it were developed over several decades, beginning in 1968 with Anderson and Morely's paper on the eigenvalues of the Laplacian matrix, a special matrix in graph theory that will be defined in equation (3.1) [2]. In 1973 and 1975, M. Fiedler published papers on the properties of the eigensystems of the Laplacian matrix [7, 8]. More recently, in 1990, Pothen, Simon, and Liou published a paper applying Fiedler's ideas to the field of clustering [15]. These papers are the

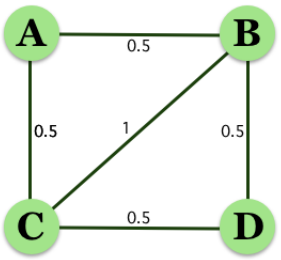
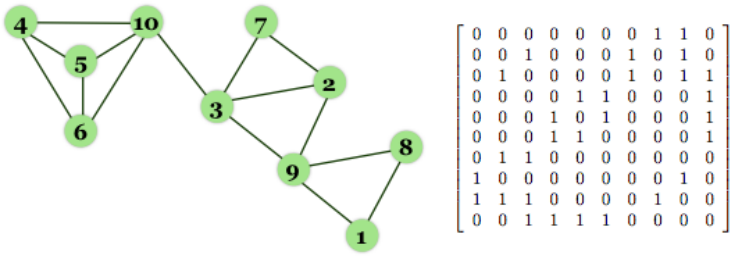


FIG. 2.1. Graph corresponding to consensus matrix from Section 2.2

origins of spectral graph partitioning methods; methods that use the spectral, or eigen, properties of a matrix to identify clusters.

3.1. Example of Fiedler Clustering. Consider the small graph in Figure 3.1 with 10 vertices along with its associated adjacency matrix. Note: each edge has weight 1.



$$\begin{bmatrix} 0&0&0&0&0&0&0&1&1&0\\ 0&0&1&0&0&0&1&0&1&0\\ 0&1&0&0&0&0&1&0&1&1\\ 0&0&0&0&1&1&0&0&0&1\\ 0&0&0&1&0&1&0&0&0&1\\ 0&0&0&1&1&0&0&0&0&1\\ 0&1&1&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&0\\ 1&1&1&0&0&0&0&1&0&0\\ 0&0&1&1&1&1&0&0&0&0 \end{bmatrix}$$

FIG. 3.1. Graph with 10 vertices and its adjacency matrix

The corresponding Laplacian matrix $\mathbf{L}$ is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{3.1}$$

where $\mathbf{A}$ is the adjacency matrix, or matrix of weights, and $\mathbf{D}$ is a diagonal matrix containing the row sums of $\mathbf{A}$. Figure 3.2 shows the Laplacian matrix for the graph in Figure 3.1.

$$\underbrace{\begin{bmatrix} 2&0&0&0&0&0&0&-1&-1&0\\ 0&3&-1&0&0&0&-1&0&-1&0\\ 0&-1&4&0&0&0&-1&0&-1&-1\\ 0&0&0&3&-1&-1&0&0&0&-1\\ 0&0&0&-1&3&-1&0&0&0&-1\\ 0&0&0&-1&-1&3&0&0&0&-1\\ 0&-1&-1&0&0&0&2&0&0&0\\ -1&0&0&0&0&0&0&2&-1&0\\ -1&-1&-1&0&0&0&0&-1&4&0\\ 0&0&-1&-1&-1&-1&0&0&0&4 \end{bmatrix}}_{\mathbf{L}} = \underbrace{\begin{bmatrix} 2&0&0&0&0&0&0&0&0&0\\ 0&3&0&0&0&0&0&0&0&0\\ 0&0&4&0&0&0&0&0&0&0\\ 0&0&0&3&0&0&0&0&0&0\\ 0&0&0&0&3&0&0&0&0&0\\ 0&0&0&0&0&3&0&0&0&0\\ 0&0&0&0&0&0&2&0&0&0\\ 0&0&0&0&0&0&0&2&0&0\\ 0&0&0&0&0&0&0&0&4&0\\ 0&0&0&0&0&0&0&0&0&4 \end{bmatrix}}_{\mathbf{D}} - \underbrace{\begin{bmatrix} 0&0&0&0&0&0&0&1&1&0\\ 0&0&1&0&0&0&1&0&1&0\\ 0&1&0&0&0&0&1&0&1&1\\ 0&0&0&0&1&1&0&0&0&1\\ 0&0&0&1&0&1&0&0&0&1\\ 0&0&0&1&1&0&0&0&0&1\\ 0&1&1&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&0\\ 1&1&1&0&0&0&0&1&0&0\\ 0&0&1&1&1&1&0&0&0&0 \end{bmatrix}}_{\mathbf{A}}$$

FIG. 3.2. Finding Laplacian matrix for adjacency matrix in Figure 3.1

It is generally known that the Laplacian $\mathbf{L}_{n\times n}$ is a symmetric and positive semidefinite matrix whose rank is $n-1$ if and only if the associated graph is connected. Furthermore, $\mathbf{Le} = \mathbf{0}$ when $\mathbf{e}$ is a column of ones. A complete discussion on the Laplacian of a graph is contained in the text by Chung [5]. M. Fiedler proved that the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix can be used to partition a graph into maximally intraconnected components and minimally interconnected components. This eigenvector is referred to as the Fiedler vector.

For example, the Fiedler vector associated with the graph in Figure 3.1 is:

$$v_2 = \begin{bmatrix} 0.38 \\ 0.19 \\ 0.09 \\ -0.40 \\ -0.40 \\ -0.40 \\ 0.16 \\ 0.38 \\ 0.28 \\ -0.29 \end{bmatrix}$$

Fiedler's theory says to cluster the graph using the signs of this eigenvector. The rows with the same sign are placed in the same cluster, i.e. rows with a positive sign are placed in one cluster while rows with a negative sign are placed in another. Thus, for the 10 vertex graph, vertices 4, 5, 6 and 10 are placed in one cluster while vertices 1, 2, 3, 7, 8, and 9 are placed in another cluster. The results can be seen in Figure 3.3.

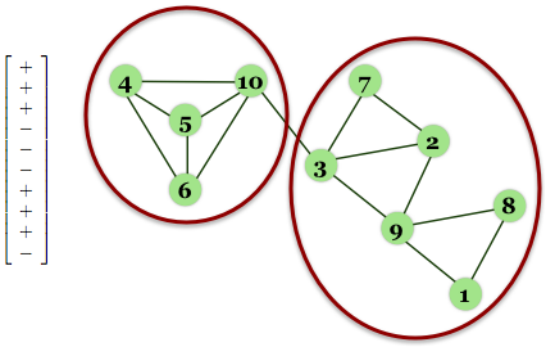


FIG. 3.3. Signs of the Fiedler vector and the partition made by the first iteration of the Fiedler Method

It is, however, possible to have zeros in the Fiedler vector. One method for handling these zeros is to arbitrarily assign the corresponding vertices to the cluster with either positive or negative entries. This has the drawback of turning the Fiedler Method into a non-unique clustering technique. Unfortunately, there is no uniform agreement on how to classify vertices corresponding to zeros in the Fiedler vector.

The next step in the Fiedler Method is to take each subgraph and partition it using its own Fiedler vector. For our example, this second iteration works well on the right hand part of the graph (see Figure 3.4). For this small graph, two iterations are sufficient to provide well connected clusters, but as graphs become larger, more iterations become necessary. There are different ideas for when one should stop partitioning the subgraphs. We created an algorithm that creates a specified number of clusters, k.

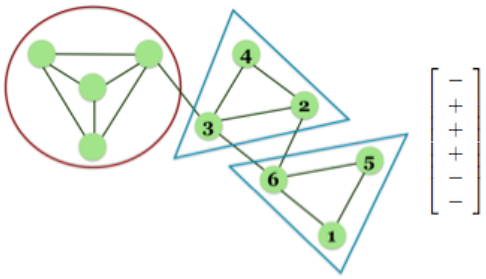


FIG. 3.4. Partition made by the second iteration of the Fiedler Method

3.2. Limitations. The Fiedler Method has gained acceptance as a viable clustering technique for simple graphs. There are, however, some disadvantages to this method. First, the Fiedler Method is iterative, so if any questionable partitions are made, the mistake could be magnified through further iterations. Second, new eigendecompositions must be found at every iteration; this can be expensive for large data sets. Finally, this method was designed for undirected, weighted graphs. Unweighted graphs can be considered by assigning each edge to have weight 1. Directed graphs can be considered as well through utilization of additional eigenvectors of $\mathbf{L}$, but we will not make use of these techniques.

4. MinMaxCut Method. Like the Fiedler Method, the MinMaxCut Method is a spectral clustering method that partitions a graph. Spectral clustering methods use the spectral, or eigen, properties of a matrix to identify clusters. There are a number of spectral clustering methods, several of which are given in detail in [12]. Although the Fiedler Method and the MinMaxCut Method are both spectral clustering methods, the MinMaxCut Method can create more than two clusters simultaneously while the Fiedler Method creates two clusters with each iteration.

4.1. Background. Before explaining the MinMaxCut Method, we describe a similar, more intuitive, algorithm: the Ratio Cut Method. The goal of this method is to partition an undirected, weighted graph into k clusters through the minimization of what is known as the ratio cut. Given a graph, such as a consensus matrix, broken into k clusters $X_1, X_2, \ldots, X_k$, the ratio cut is defined to be

$$\sum_{i=1}^{k} \frac{w(X_i, \overline{X_i})}{|X_i|} \tag{4.1}$$

where $|X|$ is the number of vertices in X, $\overline{X}$ is the complement of X and, given two subgraphs X and Y, $w(X, Y)$ is the sum of the weights of edges between X and Y.

Finding the minimum ratio cut of a graph by checking every possible collection of clusters is computationally prohibitive, but an approximation of the minimum can be found using linear algebra. Given a choice of k clusters in a graph with n vertices, let $\mathbf{H}$ be the $n \times k$ matrix with entries $h_{ij} = \frac{1}{\sqrt{|X_j|}}$ if the ith vertex is in the jth cluster, and 0 otherwise. Minimizing the ratio cut is then equivalent to minimizing $Tr(\mathbf{H}^T\mathbf{L}\mathbf{H})$ over the set of matrices $\mathbf{H}$ whose columns form an orthonormal basis [12], where $\mathbf{L}$ is the Laplacian matrix defined in equation (3.1) and Tr indicates the trace of a matrix.

The conditions on $\mathbf{H}$ being constructed with entries based on the clusters can be relaxed to be $\mathbf{H}^T\mathbf{H} = \mathbf{I}$, i.e. the columns of $\mathbf{H}$ form an orthonormal set of vectors (see [13]). With this new condition, it is known that the minimum of $Tr(\mathbf{H}^T\mathbf{L}\mathbf{H})$ over the set of matrices $\mathbf{H}$ whose columns form an orthonormal basis occurs when the columns of $\mathbf{H}$ are the eigenvectors of $\mathbf{L}$ corresponding to the k smallest eigenvalues [12].

With the original conditions for $\mathbf{H}$, it is easy to determine which vertices are in the same cluster: the ith and jth vertices are clustered together if and only if the ith and jth rows of $\mathbf{H}$ are the same. The solution for $\mathbf{H}$ under the relaxed conditions does not have equal rows, but one can use a clustering technique to group the rows into k clusters. These clusters of rows correspond to clusters in the graph. Figure 4.1 shows the matrix $\mathbf{H}$ with both the original and relaxed conditions for the 10 vertex graph in Figure 3.1 with $k = 3$. With the original conditions, we can see that vertices 1, 8, and 9 cluster together, vertices 2, 3, and 7 cluster together, and vertices 4, 5, 6, and 10 cluster together. With the relaxed conditions, we cluster the rows of $\mathbf{H}$ using k-means with $k = 3$ and find the same clusters as before. This is consistent with the clustering found in Section 3.1, where the Fiedler Method was used to determine the clusters.

H with original conditions:

$$\begin{bmatrix} 0.58 & 0.00 & 0.00 \\ 0.00 & 0.58 & 0.00 \\ 0.00 & 0.58 & 0.00 \\ 0.00 & 0.00 & 0.50 \\ 0.00 & 0.00 & 0.50 \\ 0.00 & 0.00 & 0.50 \\ 0.00 & 0.58 & 0.00 \\ 0.58 & 0.00 & 0.00 \\ 0.58 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 \end{bmatrix}$$

H with relaxed conditions:

$$\begin{bmatrix} 0.31 & -0.38 & -0.43 \\ 0.31 & -0.19 & 0.38 \\ 0.31 & -0.09 & 0.29 \\ 0.31 & 0.40 & -0.11 \\ 0.31 & 0.40 & -0.11 \\ 0.31 & 0.40 & -0.11 \\ 0.31 & -0.16 & 0.59 \\ 0.31 & -0.38 & -0.43 \\ 0.31 & -0.28 & -0.06 \\ 0.31 & 0.29 & -0.01 \end{bmatrix}$$

FIG. 4.1. $\mathbf{H}$ with original conditions (left) and relaxed conditions (right)

4.2. Clustering with MinMaxCut Method. The algorithm we will use to cluster data, the MinMaxCut Method, is a variation of the Ratio Cut Method. We use the MinMaxCut Method because it creates clusters such that elements in the

same cluster are similar and elements in different clusters are dissimilar, whereas the Ratio Cut Method only creates clusters so that elements in the same cluster are similar. The MinMaxCut Method was developed in 2003 by Ding, He, Zha, Gu, and Simon [6]. Instead of minimizing the ratio cut, we minimize what is referred to as the MinMaxCut, defined as:

$$\sum_{i=1}^{k} \frac{w(X_i, \overline{X_i})}{w(X_i, X_i)} \tag{4.2}$$

where $w(X, X)$ is the sum of the weights of edges within the cluster X.

Similar to the minimization problem described in Section 4.1, an approximate solution can be found by creating a matrix $\mathbf{H}$ whose columns are the k eigenvectors corresponding to the k smallest eigenvalues of $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal matrix used to create the Laplacian matrix. Consider the graph in Figure 3.1. In Section 3, we clustered this graph using the Fiedler Method. To create clusters using the MinMaxCut Method, the rows of $\mathbf{H}$, and therefore the corresponding vertices on the graph, are clustered using the k-means algorithm. When we cluster the rows of $\mathbf{H}$ using $k = 2$ and $k = 3$, we end up with the same clustering found by the Fiedler Method (see Figures 3.3 and 3.4).

4.3. Limitations. As with every clustering method, the MinMaxCut Method has some disadvantages. First, the solution depends on the eigenvectors of $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Some of the eigenvectors of $\mathbf{L}$ may not be continuous with respect to small changes in its entries [13], so small changes in weights of edges may lead to significantly different clusterings. Second, in the final step of this method, the rows of $\mathbf{H}$ must be clustered. This means that any weaknesses in the clustering method used on $\mathbf{H}$ will become a weakness of the MinMaxCut clustering. Finally, the MinMaxCut Method, like the Fiedler Method, only works for square, symmetric matrices.

5. Determining *k*. For the clustering techniques that we have discussed it is necessary to decide the value of k before running the algorithm. How do we know what value of k should be chosen? This is an important question in cluster analysis, one that is separate from the problem of actually clustering the data. The correct choice of k is often ambiguous. If an appropriate value of k is not apparent from prior knowledge of the properties of the data set, it must somehow be chosen.

5.1. The Algorithm. When applying spectral clustering methods to graphs, one technique to determine the value of k is to use the eigengap, that is, to look for where the jump from the kth to the $(k+1)$th eigenvalue is relatively large. The method we use to determine k is based on the MinMaxCut minimization method. The MinMaxCut is generally monotonic in the number of clusters; that is, if you make more clusters, the MinMaxCut will be larger. We compare the increase in the MinMaxCut from k to $(k+1)$ clusters to what one would expect the increase to be if one split a perfectly clustered graph to achieve the $(k+1)$th cluster.

For example, suppose the perfectly clustered graph G has been clustered into k clusters $X_1, X_2, \ldots, X_k$. Further, assume X_1 is split into clusters B and C. The MinMaxCut changes from
$\sum_{i=1}^{k} \frac{w(X_i, \overline{X_i})}{w(X_i, X_i)}$ to

$$\frac{w(B,\overline{B})}{w(B,B)} + \frac{w(C,\overline{C})}{w(C,C)} + \sum_{i=2}^{k} \frac{w(X_i,\overline{X_i})}{w(X_i,X_i)} \tag{5.1}$$

where $\overline{B}$ and $\overline{C}$ are complements with respect to all vertices. In the case of an even number of vertices in X_1, half of the vertices are in B and half are in C. In the case of an odd number of vertices, nearly half of the vertices are in both B and C. Since B and C each constitute approximately half the vertices of X_1, half of the edges between X_1 and $\overline{X}_1$ will be edges between B and $\overline{B}$. In addition, half of the edges contained entirely in X_1 will be edges between B and C, so $w(B,\overline{B}) = \frac{1}{2}(w(X_1,\overline{X_1}) + w(X_1,X_1))$. One quarter of the edges contained in X_1 will be entirely contained in B, so we conclude that

$$\frac{w(B,\overline{B})}{w(B,B)} = \frac{\frac{1}{2}(w(X_1,\overline{X_1}) + w(X_1,X_1))}{\frac{1}{4}w(X_1,X_1)} = 2\frac{w(X_1,\overline{X_1})}{w(X_1,X_1)} + 2 \tag{5.2}$$

The same analysis can be applied to C. Combining equations 5.1 and 5.2, we see that MinMaxCut changes from $\sum_{i=1}^{k} \frac{w(X_i,\overline{X_i})}{w(X_i,X_i)}$ to

$$3\left(\frac{w(X_1,\overline{X_1})}{w(X_1,X_1)}\right) + 4 + \sum_{i=1}^{k}\left(\frac{w(X_i,\overline{X_i})}{w(X_i,X_i)}\right). \tag{5.3}$$

Therefore, equation 5.3 gives the expected change in the MinMaxCut by assuming we split a perfectly clustered graph to achieve the $(k+1)$th cluster. We define the weight ratio to be the ratio of the actual change in the MinMaxCut to the expected change in the MinMaxCut. If the weight ratio is high, then the cut made to create the $(k+1)$th cluster is not ideal. If the weight ratio is low, then the cluster may be split further. While we do not have a specific cutoff value to determine whether or not a cluster should be split, high values are numbers greater than 0.5 and low values are numbers between 0 and 0.5.

To determine if increasing the number of clusters from k to $k+1$ is appropriate, we check whether an existing cluster can be split by looking at the weight ratio value for each cluster. For example, if there are two clusters, we examine the weight ratio value for splitting the first cluster and the value for splitting the second cluster. We illustrate this method on the well known Leukemia data set.

5.2. Leukemia Data Set. The Leukemia data set first appeared in a 1999 article in *Science* [10]; it is well known in the DNA microarray cluster analysis literature. The data set contains bone marrow samples of 38 cancer patients. For each sample, the gene expression levels for 5000 genes are given [4]. The samples in the Leukemia data set can be broken into three groups, corresponding to three different types of leukemia. Patients 1-19 were diagnosed with acute lymphoblastic leukemia, B-cell subtype (ALL-B), patients 20-27 were diagnosed with acute lymphoblastic leukemia, T-cell subtype (ALL-T), and patients 28-38 were diagnosed with acute myelogenous leukemia (AML). Since we know how each patient was diagnosed, this data set is

frequently used to evaluate the accuracy of a clustering method using either $k = 2$ (ALL/AML) or $k = 3$ (ALL-B/ALL-T/AML).

Since there are 38 patients and gene expression levels for 5000 genes, the Leukemia data set can be expressed as a 5000×38 matrix. We created a 38×38 consensus matrix from 1000 runs of k-means with $k = 2$ using the cosine metric. We first applied the MinMaxCut Method to the consensus matrix with $k = 2$. One cluster contained the AML patients, with the exception of patient 29, and the other cluster contained the ALL patients with the exception of patients 6 and 17. Next, we found the weight ratio values of the two clusters. The ALL cluster had a weight ratio value of 0.2892 and the AML cluster had a weight ratio value of 1.4872. This indicates that we could split the ALL cluster but we should not split the AML cluster.

Next, we applied the MinMaxCut Method to the consensus matrix with $k = 3$. The first cluster contained the ALL-B patients, with the exception of patients 6 and 17. These patients clustered with the AML patients. The second cluster contained the ALL-T patients and the third cluster contained the AML patients, with the exception of patient 29. This patient clustered with the ALL-B patients. We then found the weight ratio values of each cluster. The ALL-B cluster had a weight ratio value of 4.5638, the ALL-T cluster had a weight ratio value of 0.7270, and the AML cluster had a weight ratio value of 1.4872. These values indicate that none of the clusters should be split.

These results agree with the experimental data. It is known that there are three types of leukemia: ALL-B, ALL-T, and AML, and it is known how the 38 patients should cluster. Our algorithm indicated that there should be three clusters and our clustering is consistent with the diagnoses of the patients with only three exceptions [10].

6. Visualizing the Clusters. Data visualization is an important area because visuals frequently help us see patterns or trends that we might have missed otherwise. The goal of the tool we developed is to help us visualize the strength of the clusters found through the consensus clustering.

6.1. The Algorithm. The algorithm to visualize the consensus matrix is simple. It looks at every nonzero entry in the consensus matrix $\mathbf{C}$ and creates a colored point on what we refer to as the heat map. Recall that the (i, j) entry of the consensus matrix reflects the proportion of clusterings in which element i was clustered with element j. The color of a particular point depends on the value of the corresponding entry in the consensus matrix. The colors are chosen as follows:

- Blue if $0.05 < \ \mathbf{C}(i,j) \ < 0.1$
- Cyan if $0.1 \leq \ \mathbf{C}(i,j) \ < 0.2$
- Green if $0.2 \leq \ \mathbf{C}(i,j) \ < 0.3$
- Yellow if $0.3 \leq \ \mathbf{C}(i,j) \ < 0.7$
- Magenta if $0.7 \leq \ \mathbf{C}(i,j) \ < 0.8$
- Red if $0.8 \leq \ \mathbf{C}(i,j) \ < 0.9$
- Black if $0.9 \leq \ \mathbf{C}(i,j) \ \leq 1$

We obtain a clustering of the consensus matrix either through the use of the Fiedler Method or the MinMaxCut Method. The heat map is then arranged so that

elements belonging to the same cluster are adjacent to each other; the same element order is used to index both the x and y axes of the heat map. The heat map is characterized by colored blocks on the diagonal that correspond to clusters. This idea was also developed in Monti, Tamayo, Mesirov, and Golub's paper on consensus clustering [14].

6.2. Heat Map of Leukemia Data. In Section 5.2 we clustered the Leukemia data set into three clusters. Figure 6.1 shows the heat map for this clustering.

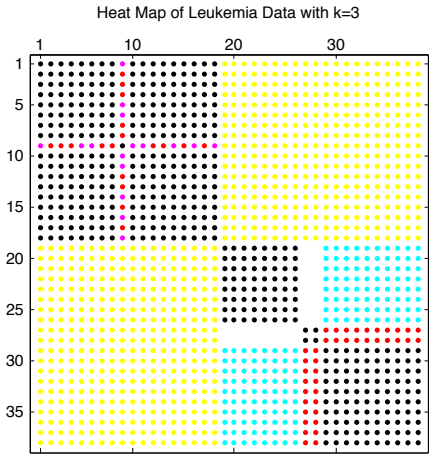


FIG. 6.1. Heat Map of Leukemia Data Set

On the x and y axes are the patient numbers (1-38). Each point in the heat map shows the proportion of clusterings in which the corresponding patients on the x and y axes were clustered together. Observe the first block on the diagonal. Most of the points in this block are black, indicating that these points clustered together many times. The second block on the diagonal is composed of all black points and the third block on the diagonal contains some red points. These three blocks correspond to the three clusters ALL-B, ALL-T, and AML. Because the points within these clusters are mostly black, we can see that these clusters are strong. The yellow and blue points outside of the blocks indicate that, occasionally, some unusual clusterings took place. One such point in the top right corner of the heat map is yellow, indicating that patients 1 and 38 were occasionally clustered together.

7. Experimental Results with Fisher's Iris Data. Fisher's Iris data set is a multivariate data set introduced by R. Fisher in his 1936 paper about discriminant analysis [9]. It is sometimes called Anderson's Iris data set because E. Anderson collected the data [1]. The data consists of 150 samples from three different types of iris flower. Samples 1-50 are *Iris setosa*, samples 51-100 are *Iris versicolor*, and samples 101-150 are *Iris virginica*. In each sample, four features were measured: the length and width of the sepal and petal, in centimeters. Because this data clusters fairly well into three clusters, it has become a standard measure of the strength of a clustering algorithm.

7.1. Iris Data with k=3. We first constructed a consensus matrix using 1000 runs of k-means with $k = 3$. After creating the consensus matrix, we applied the

Fiedler Method and the MinMaxCut Method to the consensus matrix using $k = 3$. The resulting clusters correspond, for the most part, to the known species of each flower. The exceptions were *Iris versicolor* flowers 67, 71, 73, 84, and 85, which clustered with the *Iris virginica* flowers. These five misclassifications occurred with both the Fielder clustering and the MinMaxCut clustering. Despite these misclassifications, the resulting clusters were fairly strong, as indicated by the heat map of the consensus clustering (see Figure 7.1).

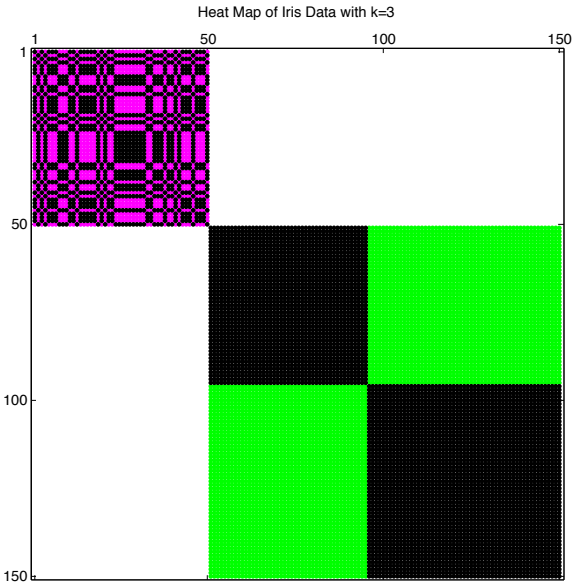


FIG. 7.1. Heat Map of Iris Data Set with $k = 3$

In Figure 7.1, the x and y axes are the flower numbers (1-150). The first block on the diagonal corresponds to the *setosa* cluster, the second to the *versicolor* cluster, and the third to the *virginica* cluster. The *versicolor* and *virginica* blocks are entirely black, indicating that these clusters are very strong. The green points between these two blocks indicate that the flowers in these clusters were occasionally clustered together. The *setosa* cluster, however, has many magenta points. This indicates that this cluster is not as strong as the *versicolor* and *virginica* clusters (see Section 6.1).

To investigate this further, we found the weight ratio value of each cluster. The *setosa* cluster had a weight ratio value of 0.3755 while the *versicolor* and *virginica* clusters had weight ratio values of 0.6037 and 0.5881, respectively. These values indicate that the *setosa* cluster may be split further while the *versicolor* and *virginica* clusters should not be split further.

7.2. Iris Data with *k*=4. When we found the weight ratio values of the three Iris clusters, there was an indication that there could be a fourth cluster by splitting the *setosa* cluster. We then applied the Fiedler Method and MinMaxCut Method to the consensus matrix using $k = 4$. Both clustering methods returned the same result: the *versicolor* and *virginica* clusters remained as they were when $k = 3$, while the

setosa cluster was split in two. These two clusters, referred to as *setosa*1 and *setosa*2, are as follows:

- *setosa*1: 2, 4, 8, 9, 10, 13, 19, 21, 24, 25, 26, 27, 28, 29, 30, 31, 32, 35, 36, 37, 39, 40, 42, 46, 50
- *setosa*2: 1, 3, 5, 6, 7, 11, 12, 14, 15, 16, 17, 18, 20, 22, 23, 33, 34, 38, 41, 43, 44, 45, 47, 48, 49

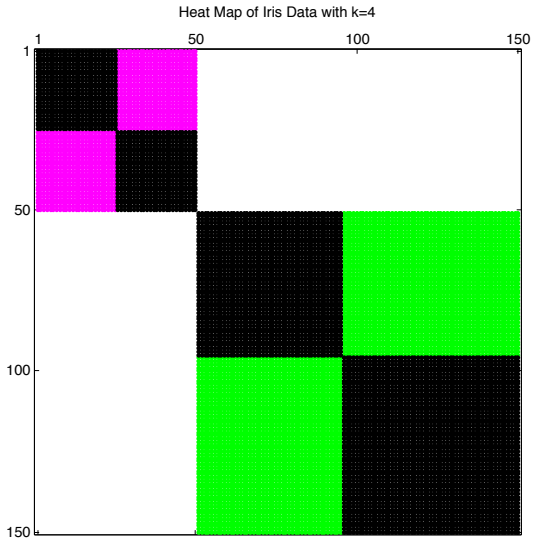


FIG. 7.2. Heat Map of Iris Data Set with $k = 4$

Figure 7.2 shows the heat map of the consensus clustering with $k = 4$. The first block corresponds to the *setosa*1 cluster, the second to the *setosa*2 cluster, the third to the *versicolor* cluster, and the fourth to the *virginica* cluster. Consider the first two blocks in Figure 7.2. This portion of the heat map is distinctly different from the first block in Figure 7.1. While there were magenta and black points interspersed within the block in Figure 7.1, the blocks in Figure 7.2 are solid black, with magenta points in the surrounding area. These clusters appear to be quite strong. We confirm this by finding the weight ratio values of these clusters. The *setosa*1 cluster had a weight ratio value of 3.3177, the *setosa*2 cluster had a weight ratio value of 1.6209, and the weight ratio values for the *versicolor* and *virginica* clusters were 0.6037 and 0.5881, respectively. These values indicate that none of the clusters should be split further.

7.3. Splitting the *setosa* flowers. In the previous section, we found evidence that the cluster containing the *setosa* flowers can be split further. To verify these results, we clustered the *setosa* flowers themselves, without the *versicolor* and *virginica* flowers. We first constructed a consensus matrix for the 50 *Iris setosa* flowers by running k-means 1000 times using $k = 2$. We then applied the Fiedler Method and the MinMaxCut Method to the consensus matrix. We found the same clusters, *setosa*1 and *setosa*2, that we found in Section 7.2. Figure 7.3 shows the heat map for this consensus clustering. This figure shows two strong clusters, the first of which corresponds to the *setosa*1 cluster, the second of which corresponds to the *setosa*2

cluster.

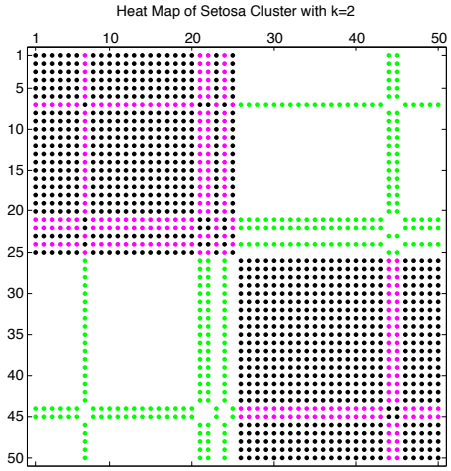


FIG. 7.3. Heat Map of *Iris setosa* Data with $k = 2$

According to the USDA's PLANTS database, there is only one subspecies of the *Iris setosa* flower growing in the area in which the samples were collected [16]. Why, then, do the *Iris setosa* flowers split into two clusters so well? Consider the comparison of the sepal length and sepal width of the 50 *Iris setosa* flowers given as a scatter plot in Figure 7.4. The red points (+) correspond to flowers in the *setosa*1 cluster while the blue points (·) correspond to flowers in the *setosa*2 cluster.

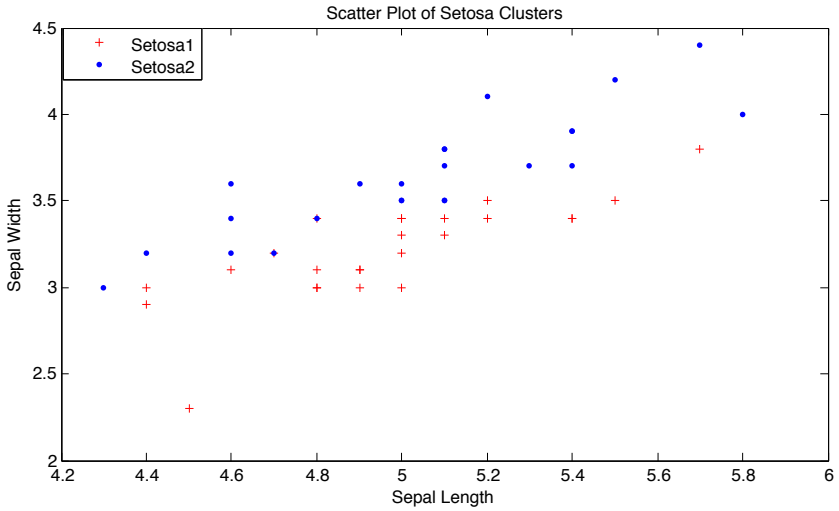


FIG. 7.4. Scatter Plot of *Iris setosa* clusters

Figure 7.4 shows that, despite the slight overlap between the red and blue points, there is a noticeable separation between the *setosa*1 cluster and the *setosa*2 cluster.

Flowers in the *setosa*2 cluster have a greater sepal width to sepal length ratio than flowers in the *setosa*1 cluster. Although we will not make any definitive conclusions on this matter, one possible explanation for this data is the existence of a different subspecies of iris flower.

8. Conclusion. In clustering, there are several issues, including determining the appropriate number of clusters for a particular dataset and visualizing the strength of these clusters. We have described methods to deal with these problems, and we have made a practical application of the proposed method studying what is known as Fisher's Iris Data Set. Through the use of our methods for determining the appropriate number of clusters and for visualizing the strength of these clusters, we concluded that there could be four clusters, instead of the generally accepted three clusters, by splitting the *Iris setosa* cluster in two. These two sub-clusters differ significantly in their sepal width to sepal length ratio, perhaps indicating the existence of another subspecies of iris flower.

Acknowledgments. We would like to thank our advisors C. Meyer and C. Wessell for guiding us through the research process and the process of writing this paper. They gave many hours of their time, reading multiple drafts and listening to much conjecture. We would also like to thank North Carolina State University for providing this research experience and the National Science Foundation and National Security Agency for providing funding to the REU program that made this experience possible.

REFERENCES

[1] E. Anderson, *The irises of the Gaspé Peninsula*, Bulletin of the American Iris Society, 59 (1935), pp. 2–5.
[2] W. N. Anderson Jr. and T. D. Morley, *Eigenvalues of the laplacian of a graph*, 1968.
[3] J. A. Bondy and U. S. R. Murty, *Graph Theory*, Springer, 2008.
[4] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, *Metagenes and molecular pattern discovery using matrix factorization*, PNAS, 101(12) (2004), pp. 4164–4169.
[5] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society,1992.
[6] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, *A MinMaxCut Spectral Method for Data Clustering and Graph Partitioning*, Lawrence Berkeley National Laboratory, Tech. Rep. 54111 (2003).
[7] M. Fiedler, *Algebraic connectivity of graphs*, Czechoslovak Mathematical Journal, 23 (1973), pp. 298–305.
[8] M. Fiedler, *A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*, Czechoslovak Mathematical Journal, 25 (1975), pp. 619–633.
[9] R. A. Fisher, *The use of multiple measurements in taxonomical problems*, Ann. Eugen., 7 (1936), pp. 179–188.
[10] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring*, Science, 286(5439) (1999), pp. 531–537.
[11] J. Kogan, *Introduction to Clustering Large and High-Dimensional Data*, Cambridge University Press, 2007.
[12] U. Luxburg, *A tutorial on spectral clustering*, Statistics and Computing, 17(4) (2007), pp. 395–416.
[13] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.
[14] S. Monti, P. Tamayo, J. Mesirov, and T. Golub , *Consensus Clustering: A resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data*, Machine Learning, 52 (2003), pp. 91–118.
[15] A. Pothen, H. D. Simon, and K. P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM Journal on Matrix Analysis and Applications, 11(3) (1990), pp. 430–452.
[16] USDA, NRCS. 2010. The PLANTS Database (http://plants.usda.gov, 15 July 2010). National Plant Data Center, Baton Rouge, LA 70874-4490 USA.