Revisiting Spectral Clustering Techniques on Images: 2-k-way vs. k-way Partitioning

1st Laura E. Kaplan
Computer Science Department
College of Stanten Island (CUNY)
Staten Island, USA
Laura.Kaplan@cix.csi.cuny.edu

2nd Louis Petingi
Computer Science Department
College of Staten Island (CUNY)
Staten Island, USA
Louis.Petingi@csi.cuny.edu

***Abstract*—In this paper we compare two Spectral Clustering techniques, 2-k-way, and k-way partitioning, obtained by the application of k-means on the eigenvectors of the unnormalized Laplacian of images. Classical implementations of Spectral Clustering rely on algorithms that recommend the use of k eigenvectors as the training set for the multi-partitioning of data into k-clusters. Our empirical testing of cluster and vector combinations reveals that choosing k vectors for k clusters does not necessarily yield best-performance clustering. Rather, it appears that limiting the eigenvector matrix to just two eigenvectors (specifically including the Fiedler eigenvector), while partitioning the data into k-clusters, yields more balanced clustering results.**

***Index Terms*—Fiedler eigenvalue, Fiedler eigenvector, k-means, Laplacian, Spectral Clustering.**

I. Introduction

In this paper, we revisit Spectral Clustering techniques and we present computational results to illustrate how the inclusion of more eigenvectors of the unnormalized Laplacian of images, as the training set for the multi-partitioning of data into k-clusters, does not predict better clusters. We follow the algorithms described in [2]. Clustering is an important research field in data-mining and graph theory. The purpose of clustering is to divide a dataset into classes so that data points in the same class are similar to each other. Spectral Clustering [2]–[8] treats the data clustering problem as a graph partitioning problem and relies on the construction of an undirected weighted graph where each point in the dataset represents a vertex, and the similarity value between any two points represents the weight of the edge connecting the two vertices. Here we will apply the k-means clustering method to find k-clusters given a training set of j, $2 \leq j \leq k$ eigenvectors; each j-dimensional point will be mapped to the spectral space to construct the clusters. Using our implementation of Spectral Clustering on the unnormalized Laplacian matrix, we generate both visual displays and quantitative measures of clustering performance. Our results suggest that when k-means is trained on $j = 2$ (i.e., just two eigenvectors), better quality clusters are constructed, while increasing the number of eigenvectors and generating k clusters produces the opposite effect.

The paper is structured as follows. In Section II we present a background concerning Spectral Clustering methodology and introduce the 2-k-way clustering algorithm, a modification of the classical k-way clustering algorithm. In Section III, empirical results are shown to illustrate the quality of clustering performed on several images by our algorithmic techniques. In addition to the visual clustering results, we provide quantitative measures by which clustering performance can be compared.

This research was partially supported by PSC-CUNY grant award 66165-00 54 from the City University of New York Research Foundation.

II. Background

A graph $G = (V, E)$, is composed of finite sets V and E, of vertices and edges, respectively. An image can be described as a graph $G = (V, E)$, in which the pixels correspond to the vertices of its underlying graph; the weight of the edge between vertices (i, j), $w_{i,j}$, is determined by the euclidean distance between pixels i and j, and the similarity of pixels' RGB color values. A cut is a set of edges that, if deleted from the graph, disconnect a graph into connected components. There are a variety of traditional graph cut methods such as minimum cut, ratio-cut, normalized cut, and min/max cut. Regardless of the type of cut applied, the objective is to minimize the weights of the edges of the cut C so when deleted, the remaining components correspond to clusters of highly related data points (vertices). However, for general graphs, the determination of a min-cut is often NP-hard. With the help of Spectral Clustering, the original problem can be solved in polynomial time by relaxing the original discrete optimization problem to the real domain.

The Laplacian of an image (graph) of order n, $L = [m_{i,j}]_{n \times n}$, is defined as

$$m_{i,j} = \begin{cases} d(i) & : \quad i = j \\ -w_{i,j} & : \quad (i,j) \in E \\ 0 & : \quad (i,j) \notin E \end{cases}$$

where $d(i)$, the degree of the vertex i, is

$$\sum_{j \neq i} w_{i,j}.$$

The earliest research on algebraic graph theory is made by *Fiedler* [1] who derived the algebraic criterion about the connectivity of graphs. The eigenvalues of the Laplacian, $\lambda_1, \lambda_2, \ldots, \lambda_n$, are real with $\lambda_1 = 0 \leq \lambda_2, \ldots, \lambda_n$. The connectedness of a graph can be judged by the second smallest eigenvalue of the Laplacian, also referred to as the *algebraic*

979-8-3503-2759-5/23/$31.00 ©2023 IEEE
DOI 10.1109/CSCE60160.2023.00313

connectivity, whose eigenvector is called the *Fiedler eigenvector*.

A cut $C(A, \bar{A})$ of a graph $G = (V, E)$ is a sum of the weights of the edge set that when removed from G, partitions the set of vertices V into two sets, A and $\bar{A}$. The *ratio-cut* $\Omega(A, \bar{A})$ of the cut $C(A, \bar{A})$ is defined as $C(A, \bar{A})/min(|A|, |\bar{A}|)$. The ratio-cut can be derived by the Fiedler eigenvector which divides a graph into two parts. In fact, the set of vertices corresponding to the negative values of the Fiedler eigenvector correspond to one of the clusters (set A), while the remaining set of vertices ($\bar{A}$) comprise the second cluster (i.e., bi-partition), thus minimizing the ratio-cut [6]. The relevance of this bi-partition approach is that the ratio-cut is minimized whenever the cardinality of A and the cardinality $\bar{A}$ do not differ significantly.

The following k-way partitioning clustering algorithm was described in [2]. Here k-partitioning is performed on k-eigenvectors corresponding to the first k-eigenvalues $\lambda_1 = 0 \leq \lambda_2, \dots, \lambda_k$.

k-way partitioning algorithm

- Input: Laplacian matrix L and the k of clusters to be constructed.
- Compute the first k eigenvectors $u_1, \dots, u_k$ of L.
- Let $U \in R$ be the $n \times k$ matrix containing the eigenvectors $u_1, \dots, u_k$ as columns.
- For $i = 1, \dots, n$ let $y_i \in R$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i), i = 1, \dots, n \in R$ with the k-means algorithm into clusters $C_1, \dots, C_k$.
- Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j | y_j \in C_i\}$

Also in [2], similar k-way clustering algorithms to the one aforementioned are described, by replacing the unnormalized Laplacian by a normalized version.

In the following algorithm, the eigenvector matrix is limited to just two eigenvectors (specifically including the Fiedler eigenvector) for k-means training.

2-k-way partitioning algorithm

- Input: Laplacian matrix L and the k of clusters to be constructed.
- Compute the first 2 eigenvectors u_1, u_2 of L.
- Let $U \in R$ be the $n \times 2$ matrix containing the eigenvectors u_1, u_2 as columns.
- For $i = 1, \dots, n$ let $y_i \in R$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i), i = 1, \dots, n \in R$ with the k-means algorithm into clusters $C_1, \dots, C_k$.
- Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j | y_j \in C_i\}$

In the next section, we visualize the segmentation of images yielded by these algorithms and measure the clustering balance generated by the multi-partitioning approach.

III. Empirical Results

The main purpose of Spectral Clustering based on the unnormalized Laplacian is to minimize the k-ratio-cut (a generalization of the bi-partition ratio-cut mentioned in Section II) so that the number of data points distributed over the k clusters are balanced. According to Malinen and Franti [10], there are two main approaches for achieving balance in clustering.

- Minimizing Squared Mean Error (SME).
- Balancing cluster sizes.

For example, minimizing SME is the objective of the traditional k-means clustering. On similarity data alone, k-means often fails to reliably balance cluster sizes, but combined with Spectral Clustering, yields improved performance.

Following this optimization objective, we apply a metric to measure the overall performance for clustering balance. Let N_i denote the number of data points in cluster C_i, and n be the total number of data points to be distributed over $k > 0$ clusters. We then define the clustering balance of image G, B_G, as

$$B_G = (\prod_{j=1}^{k} (N_j/n))(k)^k, \tag{1}$$

with the rationale that a perfect balance will be achieved when all the clusters contain similar numbers of data points. The following inequality explains why when the number of data points (integers) are equally distributed, Equation 1 is then maximized.

$$\prod_{j=1}^{k} a_j < \prod_{j=2}^{k-1} a_j \times (a_1 + 1) \times (a_k - 1),$$
$$a_1 \leq \ldots \leq a_{k-1} \leq a_k, a_1 < a_k - 1. \tag{2}$$

Thus, increasing the smallest integer by one and decreasing the largest integer of the sequence by one will increase the product $\prod_{j=1}^{k} N_j$. That is, balancing the integers (number of data points belonging to clusters) will increase the value of B_G.

To assess the efficacy of clustering under varying conditions, we implement the classical k-way algorithm for Spectral Clustering as described in Section II, but parameterize the number of eigenvectors, v, used for training, and the number of k clusters generated. This establishes a testing environment encompassing all combinations of the number of eigenvectors and number of clusters. Within this environment, when $v = k$ and $v = 2$, we capture results corresponding to the k-way and the 2-k-way algorithms, respectively.

For visual inspection, Fig. 1 and Fig. 2 display clustering results for two of the randomly selected test images. Fig. 1 highlights clustering results for generated shapes on a black background at 150x150 pixel resolution, while Fig. 2 highlights clustering results for a photographed real object at 50x50 pixel resolution. Green-bordered results on the first column correspond to the 2-k-way algorithm, while red-bordered results along the diagonal correspond to the k-way algorithm.

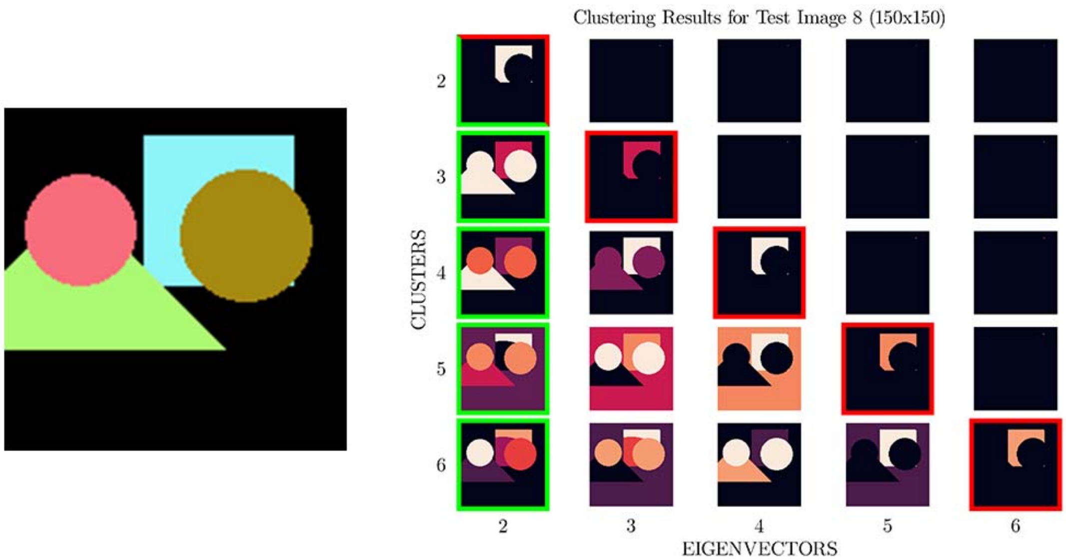

Fig. 1. Multi-partitioning of image data with varying numbers of clusters and eigenvectors on test image 8.

Fig. 2. Multi-partitioning of image data with varying numbers of clusters and eigenvectors on test image 3.

From these visualizations, strong patterns can be clearly observed. Of particular relevance to our study is the comparison between results along the $v = 2$ column and $v = k$ diagonal. Here, we can see that for any specific number of clusters k, the column $v = 2$ contains k easily distinguishable clusters. Conversely, the $v = k$ diagonal tends to have only two easily observable clusters.

As an example, consider Fig. 1 where $k = 3$. On the $v = 2$ column, three colors distinguish between the image's black background, light blue square, and remaining shapes. However, the $v = 3$ result lacks a clear third cluster, combining those remaining shapes with the background. While it may appear a cluster is missing, in actuality they have been assigned an extremely small number of pixels (often just one or two) preventing them from being easily distinguished in the visualization.

Since visual distinction becomes difficult when clustering group sizes are so small, we further explore these trends with

TABLE I
CLUSTERING BALANCE VALUES FOR TEST IMAGE 8 (150×150)

Clustering Balance Values for Test Image 8 (150×150)						
	v = 2	**v = 3**	**v = 4**	**v = 5**	**v = 6**	**v = k**
k = 2	3.365E-01	1.778E-04	1.778E-04	1.778E-04	1.778E-04	-
k = 3	4.722E-01	1.009E-04	5.333E-08	5.333E-08	5.333E-08	-
k = 4	3.399E-01	1.989E-04	4.249E-08	2.247E-11	2.247E-11	-
k = 5	9.607E-02	1.843E-04	1.078E-07	2.304E-11	1.220E-14	-
k = 6	6.984E-02	6.368E-05	1.222E-07	7.153E-11	1.530E-14	-
μ	2.629E-01	1.451E-04	3.562E-05	3.556E-05	3.556E-05	6.731E-02
$*\mu$	2.629E-01	1.369E-04	9.086E-08	4.729E-11	1.530E-14	6.731E-02

$*\mu$ is the result of the lower-triangular average (where $v \leq k$)

Fig. 3. Set of 8 randomly selected test images corresponding to summary Table II

TABLE II
AVERAGE BALANCE VALUES FOR 8 TEST IMAGES

Average Balance Values for 8 Test Images						
	v = 2	**v = 3**	**v = 4**	**v = 5**	**v = 6**	**v = k**
Image 1	3.231E-04	3.233E-04	3.207E-04	3.207E-04	3.207E-04	3.207E-04
Image 2	3.546E-01	1.549E-03	3.255E-04	3.224E-04	3.207E-04	1.621E-01
Image 3	5.456E-01	1.421E-03	3.285E-04	3.208E-04	3.207E-04	1.823E-01
Image 4	1.715E-01	1.481E-03	3.258E-04	3.208E-04	3.207E-04	1.707E-01
Image 5	1.122E-02	3.372E-04	3.209E-04	3.207E-04	3.207E-04	6.681E-03
Image 6	3.220E-04	3.220E-04	3.207E-04	3.207E-04	3.207E-04	3.207E-04
Image 7	5.711E-02	4.746E-04	3.213E-04	3.207E-04	3.207E-04	5.497E-02
Image 8	2.629E-01	1.451E-04	3.562E-05	3.556E-05	3.556E-05	6.731E-02

the quantitative approach described above by Equation 1.

Table I displays the balance values obtained for each eigenvector-cluster combination corresponding to Fig. 1. As an example, consider the value 4.722E-01 at $k = 3$ and $v = 2$. When v increases by one such that $k = 3$ and $v = 3$, the balance drops by -4.721E-01 to a value of 1.009E-04. We observe similar diminution table-wide as v increases for any fixed value of k.

To summarize this behavior, the last two rows of Table I display the average of eigenvector selection across all clusters. On these averages, an additional column is provided which describes the diagonal $v = k$ performance. The first row of averages includes all values from the table and the second row of averages is computed by the lower-triangular portion.

The lower triangular average (seen in the last row of Fig. 1) represents the average with omission of values where $v > k$. To justify the relevance of this omission, we note that when $v > k$, clustering balance tends towards failure (where clusters contain only one pixel each). Without these cases, the average more strongly captures the trending decrease in quality as v increases relative to k. Therefore, we include these lower triangular averages to call attention to the clustering results most strongly pertaining to the comparison of k-way and 2-k-way algorithms. We leave any further exploration of the distinction between lower-triangular and upper-triangular clustering behavior for future research.

In Table II, we display a summary of the average balance values corresponding to 8 randomly chosen test images (including the images shown in Fig. 1 and Fig. 2). These averages demonstrate that results are consistent across multiple test cases with varying resolution (between 50x50 and 150x150). For cross-reference, thumbnail versions of these images are available in Fig. 3.

IV. CONCLUSIONS AND ONGOING RESEARCH

In this paper, we have parameterized the classical k-way clustering algorithm to run all possible combinations on the number of eigenvectors and the number of clusters. It is particularly notable that across all images, for a fixed number of clusters k, best clustering balance is achieved by choosing $v = 2$ (i.e., 2-k-way clustering). Additionally, for any fixed k, increasing the number of vectors in the training matrix $v = 3, 4, \ldots, k$, leads to a decrease in clustering balance. Conventional application of k-way Spectral Clustering recommends choosing $v = k$. However, when considering any $v \leq k$, our empirical results suggest that this may be the worst choice for v. Therefore, contrary to conventional recommendations, we suggest that ideal clustering is best achieved when the training set is composed of only the first two eigenvectors (i,e., $v = 2$).

These significant results warrant further exploration; particularly in regard to the relationship between the lower-triangular ($v \leq k$) and upper-triangular ($v > k$) eigenvector-cluster results, which appear to behave differently. Additionally, for k where clustering is successful (i.e., roughly balanced) on $v = 2$, cluster performance decreases predictably as v increases along $v \leq k$, but fails suddenly in other conditions.

Within the domain of Image Processing, Spectral Clustering has classically been used as a standalone tool for segmentation and clustering tasks. Recent research efforts often implement Spectral Clustering as a feature extraction technique preceding deep learning tasks [11]–[13]. Irrespective of its application, reexamining the foundational algorithm appears to hold substantial potential for improved clustering performance on a comprehensive scale.

REFERENCES

[1] M. Fiedler, ”Algebraic connectivity of graphs.” Czechoslovak Math. J., 23,1973, pp. 298 – 305.

[2] U. von Luxburg,” A Tutorial on Spectral Clustering.” Stat Comput, vol. 17, 2007, pp. 395–416.

[3] J. Shi, Jianbo, and M. Jitendra, “Normalized Cuts and Image Segmentation.” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22 (8), 2000, pp. 888–905.

[4] F. Chung, “Four proofs for the Cheeger inequality and graph partition algorithms.” In Proceedings of ICCM, vol. 2, 2007, pp. 378.

[5] Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. “On Spectral Clustering: Analysis and an algorithm.” Advances in Neural Information Processing Systems, vol. 2, 2002, pp. 849–856.

[6] Spielman DA, Teng SH. Spectral partitioning works: Planar graphs and finite element meshes. Linear Algebra and Its Applications, vol. 421, 2007, pp. 284–305.

[7] J. R. Lee, S. O. Gharan, and L. Trevisan. “Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities.” Journal of the ACM (JACM), vol. 61 (6), 2014, pp. 1–30.

[8] R. Peng, H. Sun, L. Zanetti. “Partitioning well-clustered graphs: Spectral clustering works!”. Conference on Learning Theory, 2015, pp. 1423–1455. PMLR.

[9] V. Estivill-Castro, Vladimir. ”Why so many clustering algorithms – A Position Paper.” ACM SIGKDD Explorations Newsletter, vol. 4 (1), 2002, pp. 65–75.

[10] M. I. Malinen, P. Franti. ”Balance k-Means for Clustering.” Lecture Notes in Computer Science, 8621, 2014, pp. 32–41.

[11] L. Melas-Kyriazi, C. Rupprecht, I. Laina and A. Vedaldi, ”Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization,” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 8354-8365.

[12] Y. Zhao and X. Li, ”Spectral Clustering With Adaptive Neighbors for Deep Learning,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 34 (4), pp. 2068-2078, April 2023.

[13] Y. Zhao and X. Li, ”Spectral Clustering With Adaptive Neighbors for Deep Learning,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 34 (4), pp. 2068-2078, April 2023.