Segmenting Leaf Images using Spectral Clustering

Selina Wang

February 28, 2021

Abstract

I perform image segmentation using spectral clustering to segment leaf images and identify tar spots on leaves. I explore the theoretical aspect of constructing a normalized Laplacian, as well as the impacts of different parameter choices on the segmentation. I find that spectral clustering can indeed successfully segment leaf images from the background, and segment tar spots form the leaf. The success of the segmentation depends heavily on parameter choices as well as the level of contrast between the leaf and background.

1 Introduction

The concept of spectral clustering is widely applicable to many problems, with image segmentation being one of them. The goal of image segmentation is to cluster an image into different components, like segmenting the subject of image from the background. One of the most common applications of image segmentation is in video processing, where we can use image segmentation to separate the background from the subject and then replace it with a virtual background. Image segmentation can also be applied to analyzing medical images, and this application inspired me to explore the problem proposed in this paper.

For this project, I focus on using spectral clustering to perform image segmentation on pictures of maple leaves. I begin by introducing the images collected as sample data: normal maple leaves, and maples leaves with tar spot disease. I then explain the theoretical framework of my spectral clustering algorithm, and explore the process of finding optimal parameters. The next section presents the segmentation results of both a simple leaf as well as leaves with tar spot disease. Finally, I discuss the implications of the results and suggestions for improving the segmentation.

2 Problem

This project aims to use spectral clustering to segment images of leaves. In particular, I want to analyze leaves that show signs of tar spot disease, a fungal infection that causes leaves to display dark spots. Although this disease is completely benign, it is quite commonly observed, and these disease spots serve as a good case study to explore the use of image segmentation to analyze plant health. All of the data used in this project are images of leaves I collected.

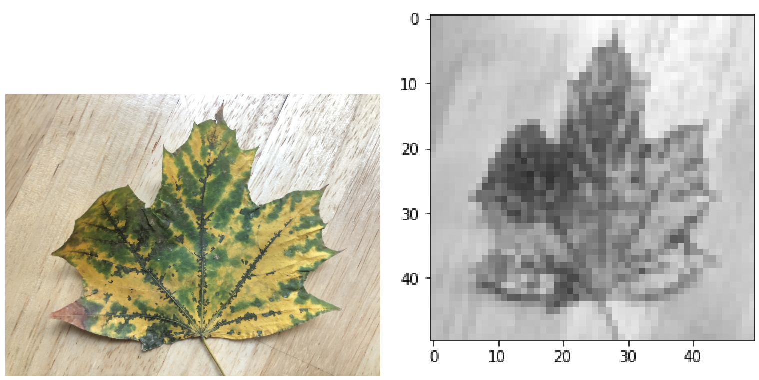
I first explore the theoretical aspect of spectral clustering and image segmentation using an image with a simple leaf and a background. My goal is to separate the image into two clusters: the leaf and the background. I then apply the same algorithm to segmenting leaves with tar spots, with the goal of of separating the images into three clusters: the leaf, the spot and the background.

2.1 Methodology

Data Preparation:
The first step to solving the problem involves preparing the data for spectral clustering. To transform these images into matrices, I take each data point has a pixel on the image, and its value is the grey scale value of that pixel on a scale of 0 to 255. Spectral clustering requires information on the location of each data point in the image, as well as the feature value of each data point. The feature can include values like brightness or color, but for the sake of simplicity I choose to only use the grey scale value of each pixel as the feature value.

I downsize the image to $50 * 50$ pixels to reduce computational complexity, and also convert it to a grey scale image. I then reshape the matrix to a $2500 * 3$ matrix, which gives 2500 data points in $\mathbb{R}^3$, call this. The first two columns contain the the coordinates of each pixel in $\mathbb{R}^2$, and the third column contains the grey scale value of each pixel.

Figure 1: Simple Leaf Image



Construct Weight Matrix:
I then construct the weight matrix for this image to form a graph with the data points as nodes. The weight of the edge between the ith and jth data point is defined as follows[1]:

$$w_{ij} = e^{\frac{-\|F_i - F_j\|^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|X_i - X_j\|^2}{\sigma_X^2}} & \text{if } \|X_i - X_j\| < r \\ 0 & \text{otherwise.} \end{cases}$$

The first component of the weight matrix forms a Gaussian Kernel using the difference in grey scale value between data points, $\|F_i - F_j\|$, while the second component considers the L_2 distance between data points on the image, $\|X_i - X_j\|$. Combining these two factors gives a weight matrix that takes into account both the "color" difference and the distance.

r is a distance threshold, where points on the image that are greater than r units apart automatically have an affinity of zero. Adding this distance threshold makes the weight matrix quite sparse and significantly reduces the computational cost. σ_X and σ_I are parameters for the Gaussian Kernel.

Form Laplacian:
For this problem, I consider a graph partition using a normalized cut[1]. Let $cut(A, B)$ be the weight of the edges connecting group A and group B, and let $assoc(A, V)$ be the total connection from nodes in A to all the nodes in the graph, then the normalized cut is defined as:

$$Ncut = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B).}{assoc(B, V)}$$

. Unlike the minimal cut, the normalized cut considers the cut value between different groups, relative to the association between nodes within one group to the rest of the group. We consider the optimal partition to be the one that not only minimizes the normalized cut, but also maximizes the association between points within the same group. The advantage of using this method is that the spectral clustering will do a better job at capturing the big structure of the data, as opposed to clustering out small groups of outliers.

The optimal partition is the one that minimizes the $Ncut$ values. In fact, minimizing this value simultaneously maximizes the association of points within A and B[1].

The above optimization problem can be relaxed to a real value problem, and translated into the following Laplacian:

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

where W is the weight matrix, and D is the degree matrix. We can then perform spectral clustering using this normalized Laplacian.

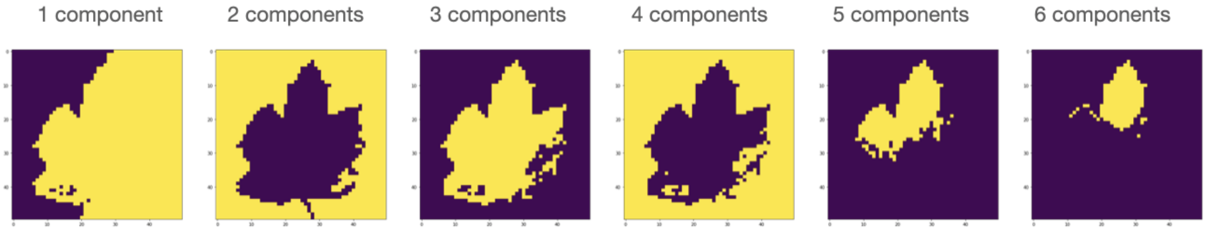
Choices for Spectral Clustering:
The spectral clustering step involves forming a spectral embedding using the eigenvectors of L and then performing K-means clustering on the embedding. Data points labeled as the same cluster belong to the same partition. We can then visualize this partition by linking each data point back to its associated pixel on the image, and see which parts of the image are segmented into the same group.

There are many choices that need to be made when designing the spectral clustering model.
The first is the number of clusters we want for the K-means clustering. In the case of the simple leaf, since the goal is to partition the image into a leaf and a background, the natural choice for the number of clusters is 2.

Choosing the number of embedded components to use for the clustering is less intuitive. I performed the segmentation using different numbers of components, experimenting with different parameter combinations, and using 2 components consistently gave the best results. The figure below compares the outcomes.
The choice of the number of components is quite important, as using only the Fiedler vector does not give desirable results, but using more than 2 components also give worse segmentation results. This is likely due to the fact that more components add extraneous information that mislead the clustering algorithm.

The parameters σ_I and σ_X also have a significant impact on the segmentation outcome. A systematic way of choosing the parameter values would be to compare the quality of segmentation for different pairs of parameter choices. I attempt to find a way to quantify how "good" a segmentation

Figure 2: Segmentation for Different Number of Components ($\sigma_I = 10, \sigma_X = 2$)



is, and to decide on an optimal parameter choice by comparing these values. I choose 5 different values for σ_I and σ_X and perform the spectral clustering (with two components and two clusters) using each combination of parameter values. I then calculate the cut value for each graph partition. This value can be calculated using the following:

$$N(cut) = \frac{y^T(D-W)y}{y^T D y}$$

where $y = (1+X) - b(1-X)$, $b = \frac{\frac{\sum_{x_i>0} d_i}{\sum_i d_i}}{1-\frac{\sum_{x_i>0} d_i}{\sum_i d_i}}$ and X is a vector containing the label for each data point.

However, since the parameters are part of the original weight matrix formula, the sizes of the parameters directly influence the size of the final cut value. To normalize this, I form a new matrix that excludes these parameters when calculating the cut value. It is defined as:

$$w_{ij} = e^{-\|F_i - F_j\|^2} * e^{-\|X_i - X_j\|^2}$$

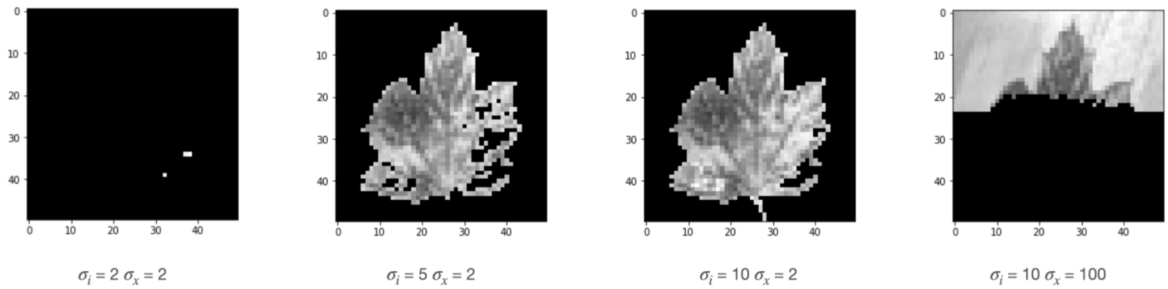
.

Figure 3: Cut Values for Different Parameter Choices

sigma_x	**2.0**	**5.0**	**10.0**	**50.0**	**100.0**
sigma_i					
2.0	0.0000	0.0001	0.0000	0.0001	0.0001
5.0	0.0001	0.0000	0.0002	0.0001	0.0002
10.0	0.0000	0.0001	0.0000	0.0001	0.0001
50.0	0.0020	0.0021	0.0018	0.0012	0.0020
100.0	0.0023	0.0024	0.0026	0.0018	0.0019

The figure above shows the cut values for each pair of parameter choice. Intuitively, a smaller cut value would indicate a better segmentation as this means that the connectivity between nodes in different groups is smaller. To visualize this, I plot the segmented images for a few parameter choices below.

Figure 4: Segmentation for Different Parameter Choices

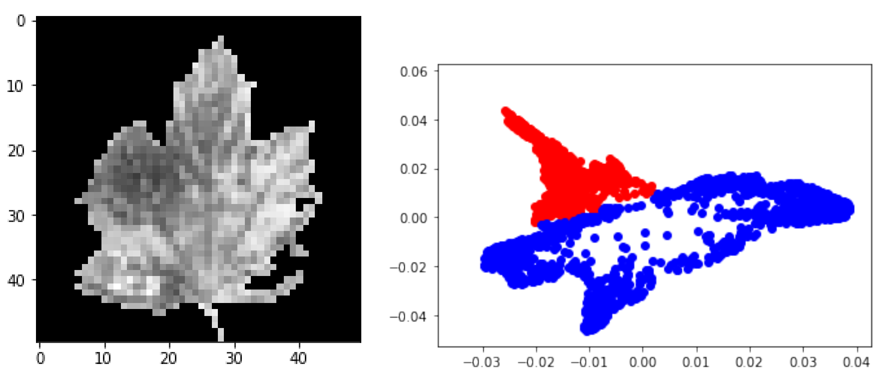


Surprisingly, a lower cut value does not guarantee a better segmentation. For example, both $\sigma_I = 10$ $\sigma_X = 2$ and $\sigma_I = 2$ $\sigma_X = 2$ give the lowest cut values, but the outcomes are completely different. However, parameter choices that give higher cut values generally are associated with worse a segmentation. This suggests that although the cut value does provide some information about the quality of segmentation, it cannot be used as the sole indicator. So, a good approach could be to pick combinations with lower cut values as potential candidates, and then use a different method to identify which gives the best result.

3 Results

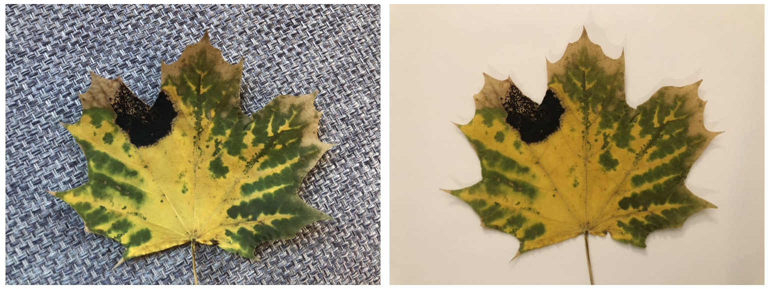
The figures below show the final segmentation result for an image with a simple leaf and the associated clusters in the embedded space. The parameter choices are: $\sigma_I = 10$, $\sigma_X = 2$ and $r = 20$, using two embedded components and two clusters.

Figure 5: Segmentation of Simple Leaf Image



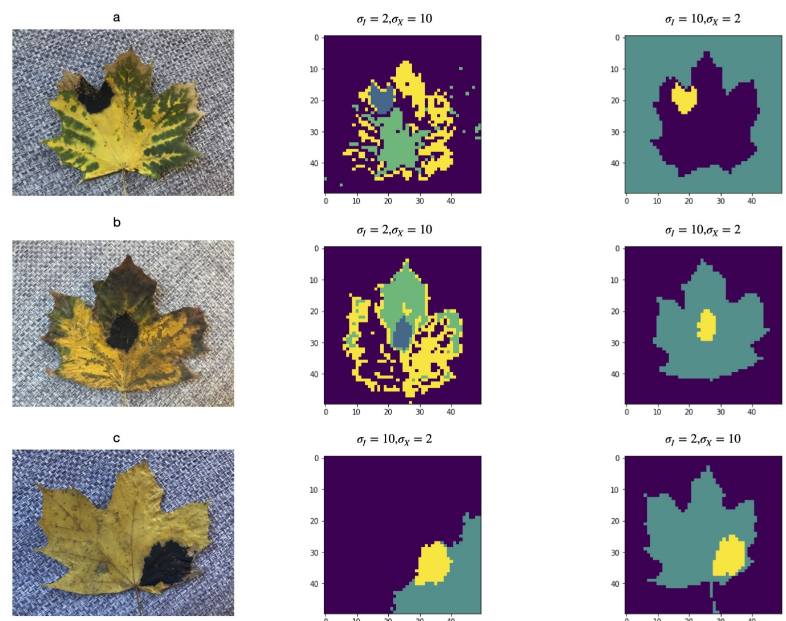
Applying the same algorithm to leaves with tar spots generates interesting but unexpected results. I chose three leaves that vary in shape, color and positioning of the tar spot to test the success of spectral clustering in segmenting the tar spot. I experimented with two different backgrounds: a grey textured background, and a plain white background.

Figure 6: Leaf Against Different Backgrounds



The original leaf image along with the segmentation result for different backgrounds are shown in the figure below. From left to right, the first segmentation image shows the result using a grey background, and the second shows the result using a plain white background. The plain white background consistently gives very good segmentation results: the images are segmented in to three clean groups: the leaf, the spot and the background. However, this algorithm performs less well with pictures with a textured background. Regardless of the choice of background, the dark spot was always successfully segmented.

Figure 7: Final Leaf Segmentation Results



4 Discussion

Choice of parameters:
As demonstrated in the methods, it is important to carefully choose the parameters. However, the

algorithm is not overly sensitive to parameter choices, so a coarse grid like the one I constructed earlier should be sufficient for finding a pair of reasonably good parameters.

The optimal parameter choice does vary from case to case. The choice of $\sigma_I = 10$, $\sigma_X = 2$ worked well for the simple leaf, and for some of the segmentations above, but not all. This might make it difficult to process large quantities of images that are not very similar, since the parameters need to be adjusted for each one.

Choice of background:
The segmentation result for the simple leaf is very good. However, it becomes difficult to segment images into three groups when trying to identify the spot, especially when the background is noisy and the color contrast between the leaf and background is weak. However, a plain white background resulted in surprisingly clean segmentations.

More specifically, using a grey background caused the clustering to become more sensitive to color differences within the leaf, and tend to view the leaf as multiple groups of colors. However, on a white background, the clustering algorithm was able to successfully identify the leaf as one whole component, despite the stark color differences within the leaf.

One explanation for this could be that, with the grey background, the difference in grey scale value between the background and the leaf is quite small, this makes the color difference within the leaf relatively larger. In the example of image a in Figure 7, this means that the connections between data points in the green section and yellow section of the leaf are relatively weak, compared to the connection between these points and the background. The algorithm is thus going to favor partitioning these sections of the leaf into two groups since the cut value is relatively low. With a plain background, however, the connections between points in the leaf and points in the background are significantly smaller, making the connection of points within the leaf relatively stronger. So the algorithm would prefer to keep these points in the same group, and partition the leaf and background as two separate groups.

Choice of cluster number:
The natural choice of clusters in leaves with tar spots is three clusters. However, for cases with a grey background, it seems that using four clusters gave a much better clustering result. This is most apparent in leaves a and b in Figure 7. Since the algorithm favors separating the leaf into multiple groups, adding another cluster actually prevents partitioning parts of the leaf into the same group as the background, and allows the final result to capture most of the structure of the leaf.

These results reveal the difference between an intuitive human approach to segmenting the leaf and a spectral clustering approach. Even though we naturally see the image as having three clusters based on the structure of the leaf, the algorithm only considers the location and color of points, and thus finds it more natural to segment it into four clusters.

The results also suggest potentials to improve the segmentation method. One natural step to take is to construct a weight matrix that takes into consideration the colors of each pixel on an RGB scale, which can bring out the color contrast between the leaf and the background. It is also worth exploring other methods of image segmentation, such as edge detection, which might work better in the case of the grey background.

5 Conclusion

In this project, I use a spectral clustering algorithm to perform image segmentation on leaf images. Factors like parameter choices, the number of clusters and components, and choice of background all have significant impacts on the segmentation outcome.The final results are very encouraging. Both the leaf and the spot are accurately segmented under a plain white background. A combination of spectral clustering with other methods may allow for successful image segmentation that require less strict backgrounds. Methods like these can be further applied to analyze colors and structures of plants.

References

[1] J. M. Jianbo Shi, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.