Assignment 4: Image Segmentation

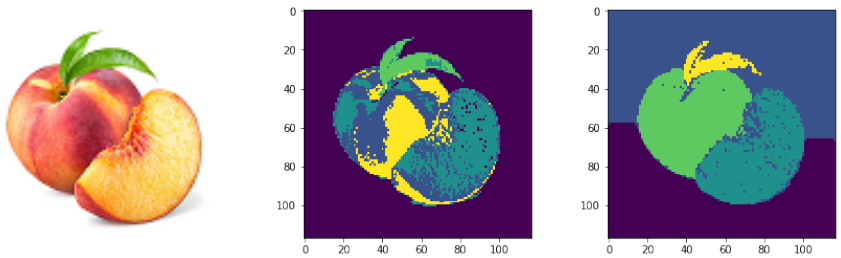
2019 Fall EECS205002 Linear Algebra

Due: 2020/1/13

Image segmentation is an important process to partition an image into multiple segments. One technique for image segmentation is spectral clustering, which works as follows

1. Create a similarity graph between pixels. Assume there are N pixels.
2. Let A be the adjacency matrix of the similarity graph, and D be a diagonal matrix, whose element $D[i,i] = \sum_{j=1,..,N} A[i,j]$. The size of matrix A and D are $N \times N$.
3. Compute the eigenvectors corresponding to the m smallest eigenvalues of the matrix $L = I - D^{-1/2}AD^{-1/2}$, which is called normalized Laplacian matrix, and store them into an $N \times m$ matrix V.
4. For the ith pixel, let $V[i, 2 : m]$ be its feature. Run k-means on these features to separate objects into k classes.

Figure 1 shows an example of image segmentation using RGB KMeans and spectral clustering. As can be seen, if you only use RGB as features, the segmentation ignores the geometry relation among pixels. But spectral clustering can take care about the geometry and color information.



(a) Original image (b) RGB Kmeans (c) Spectral Clustering

Figure 1: An example of image segmentation by Kmeans and spectral clustering.

The idea of spectral graph theory and spectral clustering can be found on Dr. Dan Spielman's webpage (`http://www.cs.yale.edu/homes/spielman/561/`).

The `A4_test.py` file has implemented RGB Keamns and the partial code of spectral clustering. You only need to finish the function `SpectralClustering`. In the code, we use a sparse matrix format, called coordinate format (COO), to store the Laplacian matrix L, because it is too large to store and to compute efficiently. For example, if the size of an image is 100×100, the matrix size will be $10^4 \times 10^4$. Besides storing the matrix size, the COO format only uses three lists, RowIndex, ColumnIndex, and Value, to record the nonzero elements of a matrix. For example, the matrix A can be stored by

$$A = \begin{bmatrix} 3 & 0 & 0 & 7 \\ 0 & 6 & 2 & 0 \\ 4 & 0 & 5 & 1 \\ 0 & 8 & 0 & 9 \end{bmatrix}$$

$$\text{RowIndex} = [1, 1, 2, 2, 3, 3, 3, 4, 4]$$

$$\text{ColumnIndex} = [1, 4, 2, 3, 1, 3, 4, 2, 4]$$

$$\text{Value} = [3, 7, 6, 2, 4, 5, 1, 8, 9]$$

1 Assignment in Python

1. Understand and finish the partial code `A4_test.py`. Look up how to use the function `scipy.sparse.linalg.eigs`. Explain how K-means algorithm works.

2. Find two images (best in .png), whose sizes are larger than 32×32, but smaller than 100×100 to evaluate two different methods. The first image must have better segmentation result using RGB KMeans than using Spectral Clustering; the second image must have better segmentation result using Spectral Clustering than using RGB KMeans. Discuss what kind of images are suitable for which segmentation method.

3. For the image with better result of using Spectral Clustering, try different (a) number of eigenvectors and (2) number of clusters to see the differences. Can you conclude their influence?

4. Compare the storage and computation cost of using COO format and traditional 2D arrays to store a matrix A and to compute $A\vec{x}$.

5. (Bonus) In the code, we use the function $f(i, j) = \exp(\frac{\|I_i - I_j\|}{\sigma})$ as the weight for the edge of pixel I_i and I_j if they are neighbors on image. Can you think a better way to assign the weight or to construct the graph. Use an example to show your idea.

2 Submission

1. Write a report in PDF file that includes
   (a) Explanation of how to use the function `scipy.sparse.linalg.eigs`.
   (b) Explanation of how K-means algorithm works.
   (c) Your original images and the figures after running two segmentation methods.
   (d) Discussion about what kinds of images prefer which segmentation methods.
   (e) Figures of segmentation results for different number of eigenvectors and clusters.
   (f) Discussion about the influence of different number of eigenvectors and clusters.
   (g) Discussion about the COO format.
   (h) (option) Result and discussion on the bonus question.
2. Python codes of all results.
3. Zip them and submit to iLMS system