

Image Segmentation and Clustering techniques

Ding-Chih Lin

May 7, 2021

Abstract

This is the midterm report for the course of statistical computing. The topics presented here is **image segmentation** and its related clustering techniques. In the first section, I introduce the idea of image segmentation via some practical applications. In the second section, I formally elaborate the formulations of these related techniques: K-means Clustering, Spectral Clustering and Hidden Markov Model. In the next section, I design a simulation study to acquire knowledge of the performance of different clustering methods under different circumstances. Advantages and disadvantages of the different clustering algorithms are discussed here. In the last section, I utilize these methods to real world image data, including color quantization and brain tumor segmentation. This project is implemented by using Python, and the code is presented on <https://github.com/davidlinn89222/image-segmentation>.

1 Introduction

In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple segments. The main goal of image segmentation is to simplify and alter the representation of an specific image into something which is meaningful or easier to analyze with. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label shared certain characteristics.

Here I present some practical applications of image segmentation

- Tumor segmentation from medical images such as X-ray, MR image and CT image.
- Object detection for autonomous vehicles or satellite.
- Color quantization, the process of reducing the number of colors in an image without loss of quality, in the research of data compression.
- Optical character recognition (OCR) for text-based image.

Several algorithms and techniques for image segmentation have been developed using domain-specific knowledge to solve segmentation problems in that specific application area. In this midterm project, I will utilize K-means clustering algorithm, spectral clustering algorithm and Hidden Markov Model to solve segmentation tasks.

2 Clustering Techniques

Clustering is an important branch in the field of unsupervised learning. The goal is to cluster the objects into the same subset utilizing some algorithms according to the context of applications. For example, there are hundreds of type of music in Spotify, and we could cluster the similar songs together based on the information of the song such as language, the year of publication, signal data obtained from digitizing the song, etc. Based on the content in section 1, the goal of image segmentation can be completed using clustering techniques. In designing the algorithm for solving clustering problem, the question need to be answered is what kind of objects should be cluster in a same group. Intuitively, we can think this question via two aspects - compactness and connectedness. With the aspect of compactness, we tend to cluster the objects which share lower distances. While with the aspect of connectedness, we tend to cluster the objects which are connected to each other. K-means algorithm is a clustering technique considering the first aspect

while spectral clustering is a technique consider the second aspect. HMM is quite special, it is not designed only for solving clustering problem but a parametric model can accomplish the objectives of both clustering and classification.

2.1 K-means Clustering

K-means clustering aims to partition n observations into k clusters, where the observations in each cluster share the similar characteristics. The original k-means problem is NP-hard computationally. The most common algorithm to solve k-means problem is described as follows, which is so-called naive k-means clustering algorithm, as a faster alternative but with the drawback of local optimal.

Given the data $x = \{x_1, \dots, x_n\}$ and an initial set of k means $m_1^{(1)}, \dots, m_k^{(1)}$, the algorithm proceeds by alternating between the following two steps

1. Assign each observation to the cluster with the nearest mean in t step

$$S_t^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2, \forall 1 \leq j \leq k\}, \quad p = 1, \dots, n \quad (1)$$

2. Recalculate means for observation assigned to the new cluster in t step

$$m_i^{(t+1)} = \frac{1}{|S_t^{(t)}|} \sum (x_j) \quad (2)$$

The above algorithm proceeds until the assignments no longer change.

The most commonly used methods to decide an initial set of k means are *Random Partition* and *Forgy method*. The random partition method randomly assigned a cluster to each observation and calculate the initial set of k means, then proceeds the algorithm. The Forgy method randomly choose k observations from the data x and use their value as the initial set of k means, then proceeds the algorithm.

2.2 Spectral Clustering

Spectral clustering is a technique with roots in graph theory. This approach is used to identify the clusters of nodes in a graph based on the edges connecting them. It utilizes information from the eigenvalues, which is so-called spectrum, of the similarity matrix built from the graph. Different from K-means using the absolute locations, it utilize affinity to determine what points fall under which cluster. Ultimately, the problem of clustering can now be reformulated using the graph and similarity matrix. The algorithm wants to find a partition of the graph such that the edges between clusters have low weights (similarity) and the edges within a cluster have high weights.

Given the data $x = \{x_1, \dots, x_n\}$, we can build a complete undirected graph $G(V, E)$ with corresponding similarity (weights) w_{ij} , $i, j = 1, \dots, n$. The similarity matrix can be defined as a symmetric matrix $W = [w_{ij}]_{i,j=1,\dots,n}$. There are different ways to construct w_{ij} : *the ϵ -neighborhood graph*, *the k -nearest neighborhood graph* and *the fully connected graph with Gaussian kernel*. I only elaborate the formulation for fully connected graph with Gaussian kernel here, define weights as

$$w_{ij} = e^{\frac{-\|x_i - x_j\|_2^2}{\sigma^2}} \quad (3)$$

These weights are greater than and equal to zero, and represents a measure of the similarity between data point x_i and x_j .

The math tools for spectral clustering are the Laplacian matrices, which is defined as

$$L = D - W \quad (4)$$

where D is the degree matrix with diagonal elements $d_{ii} = \sum_{j=1}^n w_{ij}$ represents the degree of a vertex x_i , and W is the weight matrix of the graph.

The general approach to spectral clustering is to use a standard clustering method (for example, K-means) on relevant eigenvectors of a Laplacian matrix of G . There are several ways to define Laplacian matrix with different interpretations, and then these clustering results are applicable in different scenarios. A popular spectral clustering technique is **normalized cuts algorithm**, proposed in *normalized cuts and image segmentation*, Jianbo Shi and Jitendra Malik. It partition data points into two sets based on the eigenvector corresponding to the second-smallest eigenvalue of the L^{norm} , where the symmetric normalized Laplacian matrix is defined as

$$L^{\text{norm}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (5)$$

The algorithm of spectral clustering is described as follow

1. Calculate the Laplacian matrix L
2. Calculate the first k eigenvectors
3. Consider the matrix formed by the first k eigenvectors; the l -th row defines the features of graph node l
4. Using K-means technique to cluster the graph nodes based on these feature defined in (3)

Basically, spectral clustering techniques utilize eigenvalue of the weight matrix to perform dimension reduction before clustering in lower dimensions. It is known as segmentation-based object categorization.

2.3 Hidden Markov Model

Hidden Markov Model is a discrete-time doubly embedded stochastic process, where the process X being modeled is assumed to be a Markov process with unobserved (hidden) states Y that depends on X . The goal is to learn about X in order to observe or obtain Y .

Assume we observe sequential data $x = \{x_1, \dots, x_t\}$ which is assumed to be a Markov process. u is generated by a chain of hidden states $s = \{s_1, \dots, s_T\}$, and each s_t can take M states with initial probability π_k , $k = 1, \dots, M$. The distribution of x conditional on s is represented as $b_k(x)$, which is called emission probability

$$x_t | s_t = k \sim b_k(x_t) \quad (6)$$

The changes of states between consecutive hidden state is specified by transition probability

$$a_{k,l} = P(s_{t+1} = l | s_t = k) \quad (7)$$

and denote the transition probability matrix by A . We can further assume time-homogeneous transition probabilities property to make the structure of A simpler.

The goals now can be summarized into two parts based on HMM

- Parameter estimation: $\hat{\lambda} = \{\hat{\pi}_k, \hat{b}_k(x), \hat{a}_{k,l}\} = \arg \max_{\lambda} P(x|\lambda)$ via EM algorithm
- Cluster problem: probabilities of the underlying states, given the observations, $P(s|x)$.

$$\hat{s} = \arg \max_s P(s|\lambda, x) \quad (8)$$

In order to solve the first problem, we can make parametric assumptions of the emission probabilities $b_k(x_t)$, which normally distributed with parameters μ_k and σ_k^2 . Therefore, the parameters in HMM are

$$\lambda = \{\pi_k, \mu_k, \sigma_k^2, a_{k,l} : k = 1, \dots, M\} \quad (9)$$

λ can be estimated from the observed likelihood function, but it is unreasonable since it involves M^T possible paths. However, we can utilize backward and forward algorithms to simplify the calculation of observed likelihood given λ .

Ignoring some derivation of forward and backward recursive calculation, the observed data likelihood can be represented as forward probability $\alpha_k(t)$ and backward probability $\beta_k(l)$

$$\begin{aligned} P(x_1, \dots, x_T) &= \sum_{k=1}^M \alpha_k(T) = \sum_{k=1}^M \beta_k(1) P(x_1|s_1=k) P(s_1=k) \\ &= \sum_{k=1}^M \alpha_k(t) \beta_k(t) \end{aligned} \quad (10)$$

We have prepared everything we need in the estimation of λ through EM algorithm. The EM algorithm for HMM with Gaussian assumption is described as follows

- **E-step:** evaluate $L_k(t)$ and $H_{k,l}(t)$ given λ_{t-1} , where

$$L_k(t) = P(s_t = k|x) = \frac{P(x, s_t = k)}{P(x)} = \frac{\alpha_k(t) \beta_k(t)}{\sum_{k=1}^M \alpha_k(t) \beta_k(t)} \quad (11)$$

$$H_{k,l}(t) = P(s_t = k, s_{t+1} = l|x) = \frac{\alpha_k(t) a_{k,l} b_l(y_{t+1}) \beta_l(t+1)}{\sum_{k=1}^M \alpha_k(t) \beta_k(t)} \quad (12)$$

- **M-step:** update parameters

1. For initial probability π_k with constraint $\sum_{k=1}^M \pi_k = 1$

$$\hat{\pi}_k = \frac{L_k(1)}{\sum_{k=1}^M L_k(1)} \quad (13)$$

2. For transition probability $A = [a_{kl}]$ with constraint $\sum_{l=1}^M a_{kl} = 1$ for $k = 1, \dots, M$

$$\hat{a}_{k,l} = \frac{\sum_{t=2}^T P(s_{t-1} = k, s_t = l|x, \lambda^{t-1})}{\sum_{l=1}^M \sum_{t=2}^T P(s_{t-1} = k, s_t = l|x, \lambda^{t-1})} = \frac{\sum_{t=2}^T H_{k,l}(t)}{\sum_{l=1}^M \sum_{t=2}^T H_{k,l}(t)} \quad (14)$$

3. For emission probability $b_k(x) = N(x_t|\mu_k, \sigma_k^2)$

$$\hat{\mu}_k = \frac{\sum_{t=1}^T L_k(t) x_t}{\sum_{t=1}^T L_k(t)}, \quad \hat{\sigma}^2 = \frac{\sum_{t=1}^T L_k(t) (x_t - \hat{\mu}_k)^2}{\sum_{t=1}^T L_k(t)} \quad (15)$$

Once the parameter estimated from EM algorithm reach the convergence condition, we can utilize HMM to cluster the dataset into M groups. The detail for clustering with HMM and its variations is presented in *Clustering Hidden Markov Models with Variational HEM, Coviello et al., Journal of Machine Learning Research, 2014*.

3 Experimental Design

In this section, I artificially design three data scenarios, and make a simulation study to visualize the clustering results obtained from the methods introduced in section 2. Beside the visualizations, I also provide the Rand index (accuracy rate) among repeated random samples. The functions and scripts required to reproduce the same results in this section are presented in `Simulation.py` and `tools.py` file on Github. Define number of observations $n = 1000$ and number of repeated sample $M = 100$.

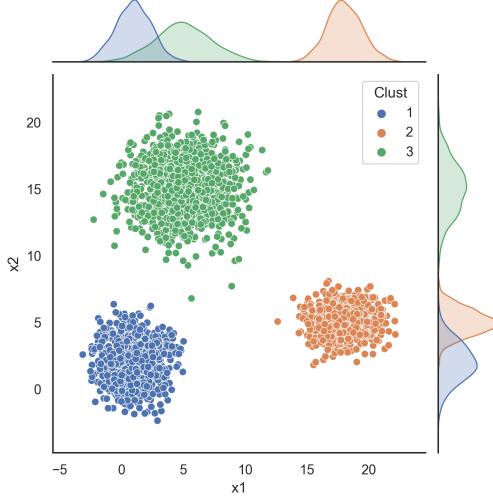


Figure 1: Simulation result from equation (16) and its true labels

3.1 Scenario one: two-dimensional independent normal distribution

Consider the following data generating process with three clusters in the first scenario:

$$x_{1k}, \dots, x_{nk} \sim_{iid} N_2(\mu_k, \Sigma_k), \quad k = 1, 2, 3 \quad (16)$$

where $\mu_1 = (1, 2)', \mu_2 = (18, 5)', \mu_3 = (5, 15)', \Sigma_1 = (\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix})$, $\Sigma_2 = (\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix})$ and $\Sigma_3 = (\begin{smallmatrix} 5 & 0 \\ 0 & 4 \end{smallmatrix})$

The simulations from the DGP in equation (16) is presented in Figure 1.

Next apply the K-means clustering, spectral clustering and HMM clustering method to the data we just generated, respectively. The clustering results are presented in Figure 2. By the fact that the three clusters are space-separated enough, and the data in each cluster are independent distributed. The performances of clustering are equally good as in these methods..

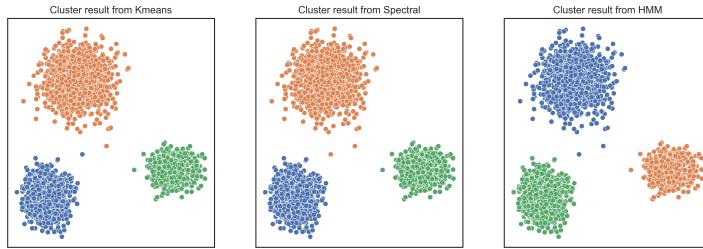


Figure 2: Cluster results from K-means Clustering, Spectral Clustering and HMM in scenario 1

3.2 Scenario two: two-dimensional correlated normal distribution

Consider the following data generating process with three clusters in the second scenario:

$$x_{1k}, \dots, x_{nk} \sim_{iid} N_2(\mu_k, \Sigma_k), \quad k = 1, 2, 3 \quad (17)$$

where $\mu_1 = (1, 2)', \mu_2 = (20, 20)', \mu_3 = (10, 20)', \Sigma_1 = (\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix})$, $\Sigma_2 = (\begin{smallmatrix} 6 & 6 \\ 6 & 8 \end{smallmatrix})$ and $\Sigma_3 = (\begin{smallmatrix} 6 & 6 \\ 6 & 8 \end{smallmatrix})$

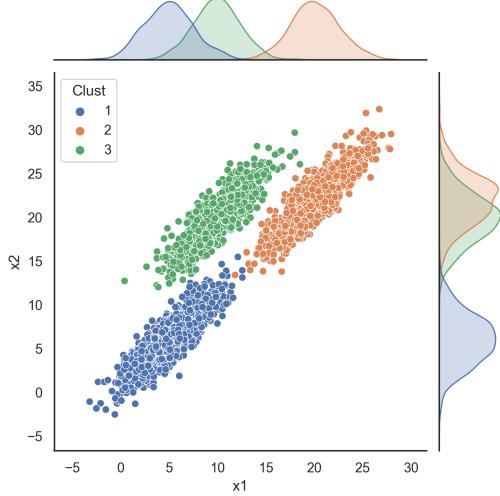


Figure 3: Simulation result from equation (17) and its true labels

The simulation from the DGP in equation (17) is presented in Figure 3.

Next apply the K-means clustering, spectral clustering and HMM clustering method to the data we just generated, respectively. The clustering results are presented in Figure 4. We can see that the K-means performs the worst because it considers only the Euclidean distance without any parametric assumption, and cannot deal with non-linear clustering problem. Spectral clustering performs better than K-means clustering but still get some miss-classifications. HMM outperforms other two algorithm since it assume the emission probability is normally distributed. By doing this, the algorithm will take statistical distance into considerations rather than Euclidean distance.

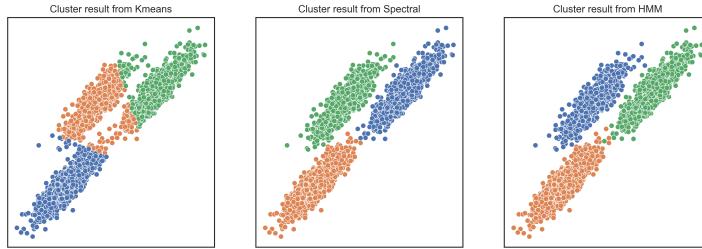


Figure 4: Cluster results from K-means Clustering, Spectral Clustering and HMM in scenario 2

3.3 Scenario three: large circle containing a smaller circle

Consider the following data generating process with two clusters in third scenario:

$$\mathbf{x}_{1i} = \begin{pmatrix} x_{11} \\ x_{21} \end{pmatrix} = \begin{pmatrix} z_{11} \\ z_{21} \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{21} \end{pmatrix} \quad s.t. \quad z_{11}^2 + z_{12}^2 = 1 \text{ and } \epsilon_{11}, \epsilon_{21} \sim_{iid} N(0, \sigma^2) \quad (18)$$

$$\mathbf{x}_{2i} = \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} = \begin{pmatrix} z_{12} \\ z_{22} \end{pmatrix} + \begin{pmatrix} \epsilon_{12} \\ \epsilon_{22} \end{pmatrix} \quad s.t. \quad z_{21}^2 + z_{22}^2 = \frac{1}{2} \text{ and } \epsilon_{12}, \epsilon_{22} \sim_{iid} N(0, \sigma^2) \quad (19)$$

I make an advantage of the `make_circles()` function with $\sigma^2 = 0.08$ in `sklearn` module to generate the data from equation (18) and (19). The simulations from the DGP in equation (18) and (19) is presented in Figure 5.

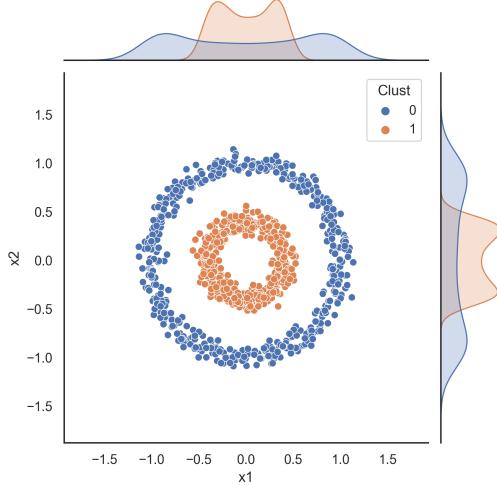


Figure 5: Simulation result from equation (18) and (19), and its true labels

Next apply the K-means clustering, spectral clustering and HMM clustering method to the data we just generated, respectively. The clustering results are presented in Figure 6. It suggests that K-means clustering and HMM clustering algorithm performs poorly at both around 50% accuracy rate. The reason is that these two algorithms essentially designed for the aspect of compactness. K-means defines closeness as Euclidean distance in most situations, and HMM defines closeness as statistical distance via probability assumption. On the other hand, spectral clustering is another broad approach called connectivity. Even if the distance between each point is very small, if they are not connected in the graph build from the data, they are not clustered together. Therefore, spectral clustering outperforms others because the data generating process clusters the data based on connectivity rather than compactness.

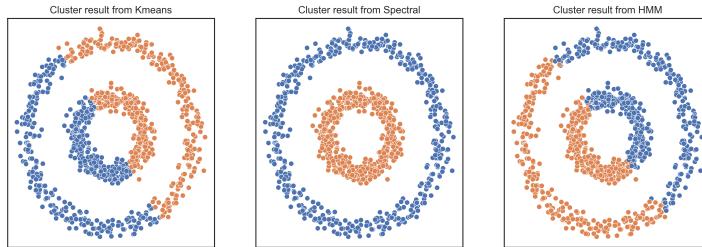


Figure 6: Cluster results from K-means Clustering, Spectral Clustering and HMM in scenario 3

3.4 Rand index

Table 1 presents the Rand index of three clustering algorithms over $M = 100$ repeated random sample. Rand index is a measure of the similarity between two data clusters. I use Rand index rather than accuracy rate because the order of labels from the results of algorithm is not guaranteed to be the same as true order of labels. I am not going to elaborate the formulation of Rand index here.

	K-means clustering	Spectral clustering	Hidden Markov Model
Scenario 1	1	0.99	1
Scenario 2	0.95	0.99	1
Scenario 3	0.50	1	0.50

Table 1: Rand index of K-means clustering, spectral clustering and HMM

4 Practical Example

I've divided this section into two parts. I am going to start with the applications in area of image compression, particularly in *vector quantization* or *color quantization*, and using K-means clustering algorithm to achieve the task. Next, I utilize the image segmentation techniques to segment a brain tumor based on magnetic resonance imaging (MRI).

4.1 Color Quantization

The K-means clustering algorithm is a commonly used technique to achieve color quantization in digital image processing. It is a process that reduces the number of distinct colors used in an image, i.e. the number of combination of values in each pixel with RGB-color space. The left image in Figure 7 is a digitized photograph of *London Bridge*. It consists 1200 x 1200 pixels, where each pixel is RGB-scale value ranging from 0 to 255 in 3-dimensional space, and hence requires 32 bits of storage per pixel. Therefore, this image needs 4 Megabyte for the storage. The remaining images in Figure 7 are VQ-compressed version of original image obtain from K-means, which K equals to 64 and 10, respectively. These images require much less storage space but at some loss in quality, and the loss depends on the option of K . The loss is severer as the K parameter decreasing.



Figure 7: *London Bridge in London, UK*. The left panel is original image. The middle panel is the compressed image using 64 code vectors (64 clusters in K-means). The right panel is the compressed image using 5 code vectors (5 clusters in K-means)

I demonstrate another example of color quantization in Figure 8. The left image in Figure 8 is photograph of three *cute Shiba*. It consists 533 x 800 pixels in RGB-color space. The remaining images in Figure 8 are VQ-compressed version of original image obtained from K-means either, which K equals to 64 and 10, respectively. By the same token, we can use the compressed image with 64 code vectors to represent the original image without losing its quality too much.



Figure 8: *three cute Shiba being happy on the grass*. The left panel is original image. The middle panel is the compressed image using 64 code vectors (64 clusters in K-means). The right panel is the compressed image using 5 code vectors (5 clusters in K-means).

4.2 Medical Image Segmentation

Medical image segmentation is a task of automatically segmenting the targets of interest in a medical image, such as X-ray or magnetic resonance images. It has an essential role in computer-aided diagnosis system. Specifically, I will segment brain tumor from MRI in this section. A brain tumor is a mass of abnormally growing cells in the brain or skull. MRI provide a detained image of the brain, and it is one of the most common tests used to diagnose brain tumors. Brain tumor segmentation from MR image can have great impact for the improvement of diagnostics, growth rate prediction and treatment planning. U-Net is a NN-based method to perform image segmentation, and it outperforms the other techniques in most situation, especially in medical imaging problem. Due to the constraint of topics and time allowed for this midterm project, I only use techniques introduced in section 2 to perform the segmentation task: K-means clustering, spectral clustering and HMM.

The brain tumor image is presented in Figure 9, which is obtained from Stanford brain tumor center. There is a obvious white part representing the tumor in this MRI. The goal is trying to automatically segment the pixels containing the brain tumor.

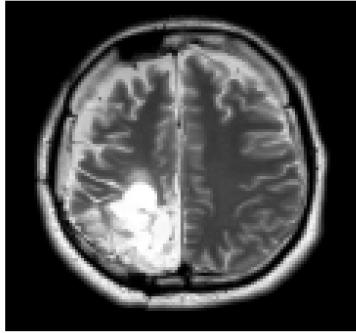


Figure 9: *tumor brain image* obtained from Stanford brain tumor center

Next, I perform $K = 5$ clusters image segmentation using K-means clustering, spectral clustering and HMM with the following settings, respectively, as a demonstration to show the clustering results from these methods.

- K-means clustering algorithm with Forgy method to decide initial set of k means
- Spectral clustering algorithm with radial basis weights and k-means to cluster
- Hidden Markov Model with normally-distributed emission probabilities

The segmentation results obtained from these methods are shown in Figure 10. It suggests that the spectral clustering algorithm successfully segment the pixels of brain tumor in the MRI, while k-means clustering and HMM performs poorly at segmenting the tumor.

After demonstrating the power of spectral clustering, and limitations of k-means clustering and HMM in medical image segmentation. I decided to utilize only spectral clustering algorithm to segment a set of brain tumor images, which is obtained in *Brain MRI Images for Brain Tumor Detection, Kaggle*. The results are shown in Figure 11. Left panel is original brain MRI and right panel is segmentation results obtained from spectral clustering algorithm with $K = 3$. It suggests that the spectral clustering algorithm successfully segment tumor from original brain MRI. There are still some work to do, such as filtering, eliminating background or detecting brain border. Besides, I lower the dimensions of images before segmenting them due to the constraint of memory space and storage.

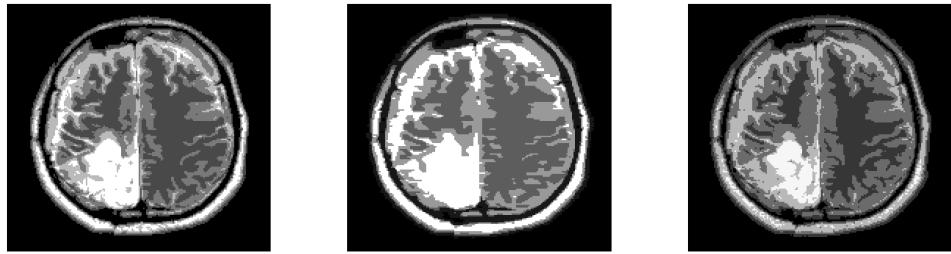


Figure 10: The results obtained from K-means, HMM and spectral clustering from left to right.

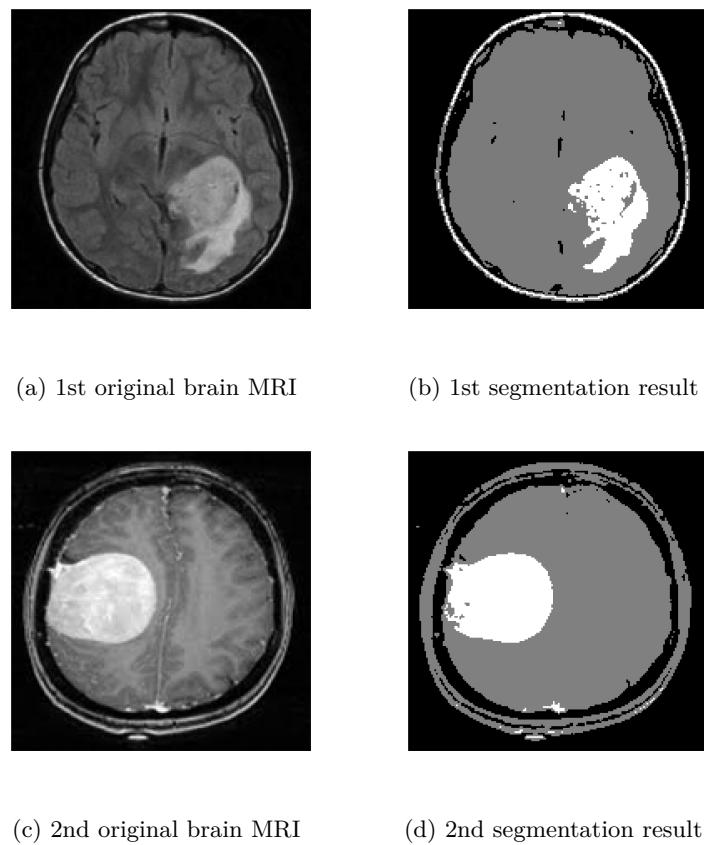


Figure 11: Original image and segmentation results from spectral clustering algorithm