#Project1 – Image Color Segmentation

Dian Basit – 260771254 Shaun Soobagrah – 260919063

Abstract

Using Gaussian mixture models (GMMs), we carried out the Expectation- Maximization (EM) algorithm (Reference?). We used Bayesian Information Criterion (BIC) to find the ideal number of clusters per image. We carried out test on a simple image and a more complex one.

1 Problem/Mathematical Setup

The EM algorithm remains the same where we initialize our free parameters, calculate the E-step, the M-step and if keep on doing the E-M steps till the log-likelihood converges.

The Bayesian Information Criterion is defined as:

$$BIC = -2\log(\hat{U}) + \log(N)\, d$$

N is the number of samples, d is the number of free parameters and $\hat{U}$ is the maximum likelihood of the model. For a Gaussian model, the maximum log-likelihood is defined as:

$$\log\left(\mathcal{N}(x|\mu_k, \Sigma_k)\right) = -\frac{1}{2}Nlog(2\pi) - \frac{1}{2}log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T{\Sigma_k}^{-1}(x-\mu_k)$$

where μ_k and Σ_k are the mean and covariance matrix for each cluster respectively. For faster results, we use the Cholesky decomposition of the inverse of the covariance, L. The equation then becomes [1]:

$$\log\left(\mathcal{N}(x|\mu_k, \Sigma_k)\right) = -\frac{1}{2}\left(Nlog(2\pi) + ||L^T(x-\mu_k)||^2\right) + log|L|$$

2 Numerical Methods and Algorithms

The EM algorithm remains the same as explained in assignment 2 [2]. The steps we use to produces our results start by using images as input data. The images are reshaped into a Nx3 matrix to represent the RGB distribution. The pixels are then normalized by dividing by 255 as all the pixels are reduced to domain of [0,1] and also reduces overflowing in exponential operations. The images are then blurred using a Gaussian blur to reduce the noise and to make smooth color transition from one cluster to another.

We then carry out the EM steps and we plot the log-likelihood graph vs the number of EM iteration to see if the clustering converges. The steadier is the plot, the more the means converges to their clusters.

We use the BIC to get the near-ideal number of clusters per image [3]. The smaller the BIC value, the better is the number of clusters. However, when looking at the BIC graphs obtained when running test 3, we can we that after some cluster number, there is no advantage in increasing the number of clusters as the BIC values do not change much.

3 Experiments and Results

There are four tests and four different images. They are shown when you run the code. K is the number of clusters. Close the plots to get to the next test.

In the first test, GMM is tested on two simple images with 3 different colors each. This is done to get an idea if the clustering is working for a picture where we know the ideal cluster values. The results show that indeed the algorithm is able to distinguish 3 clusters and the log-likelihood graphs show that it converges.

The second test shows the color segmentation of the bear image for different number of clusters (K values). The BIC graph shows that after a certain number of clusters (here 6 or 7), increasing the number of clusters will not make considerable difference.

Test 3 carries out the same procedure as test 2 but for the flower image. In this one, since there are more clusters, the BIC curve shows spikes at the end, but we can see that BIC values reaches steady state. Therefore, we take the number of clusters to be 10 (K = 10).

Test 4 is carried out with the two non-trivial spatial-color distribution and there seemingly ideal value of K based on their BIC graph. Their log-likelihood graphs are also plotted. Below is a result for the flower image.

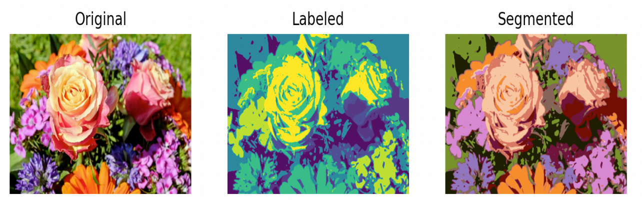

Figure 2: the image color segmentation of the flower image with K = 10.

We can see from the results that the algorithm is able to cluster the similar colored pixel and from the log-likelihood graph, we see that the mean converges to their clusters. The BIC graphs are able to indicate proper K values to choose for clustering.

Bibliography

[1] https://math.stackexchange.com/questions/2661595/explanations-on-scipy-gaussian-mixture-score-computation
[2] https://towardsdatascience.com/how-to-code-gaussian-mixture-models-from-scratch-in-python-9e7975df5252?fbclid=IwAR2tZQwHsVcxkqMrCV5PPvugYRKcbefS3FkRZR890T1gGPuQoKK_tkAoN_8
[3] https://towardsdatascience.com/gaussian-mixture-model-clusterizationhow-to-select-the-number-of-components-clusters-553bef45f6e4