

# BSDS by clustering

Carlos Andres Reyes Rivera  
Universidad de los Andes

ca.reyes1787@uniandes.edu.co

Lina Maria Fierro Zambrano  
Universidad de los Andes

lm.fierro1340@uniandes.edu.co

## Abstract

*The present lab is about clustering methods and its used Berkeley Segmentation Data and Benchmarks 500 (BSDS500). This data set contain 500 natural images. The principal objective of this lab is to select model with the best parameters to cluster images and finally evaluate results and compare models. The best models are K-means and GMM in space Labxy, so with these methods was made the test stage.*

## 1. Clustering Methods

### 1.1. K-means

k-means is one of the simplest unsupervised learning algorithms in order to cluster a whole of pixels. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The method starts with the assign of each data to a centroid and these are replaced until it finds the centroids and clusters to minimize squared distances between elements and centroids.[1]

$$\operatorname{argmin} \sum_{i=1}^k \|x_j - \mu_i\|^2 \quad (1)$$

In general, K-means use the Euclidean distance to find how far are the data pixels from the centroids. The number of dimension of these distances depends on the number of features per pixel[4]. In this way, the ruggedness of the classification depends on the number of feature that it has each pixel.

For this lab we use 6 cases, 3 of them with 3 features and the others with 5 features. In the 3 feature cases, we use as feature the value of each pixel in each channel of RGB, LAB and HSV respectively. In the 5 feature cases, we use the same channel values but this time we add the position X and Y of each pixel.

Initially, when we were in the training state, we realized that the results were biased towards position. In other words, the results only depends of the position, so it doesn't take into account the value of the channels. In order to solve this situation, we decide to normalize the value and the range (0 to 1) of the feature. For reach this goal, we divide the process in two sub-process. The first part was make the minimum value of the channels equal to 0. Which meaning subtract the minimum pixel value to the others and itself. Once we made that we could normalize the value of the channels. For this process we find the maximum value of each feature and divide all the value for this. In this way all the features were in the range of 0 - 1 (all have the same weight) and conserved the differences between them.

### 1.2. Gaussian Mixture Model

GMM is one of the simplest unsupervised learning algorithms in order to cluster a whole of pixels. This is the generalization of k-means but the idea of this algorithm is to represent each group with a Gaussian distribution or a parametric distribution [1]. As k-means the procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The method starts with the formation of cluster by representing the probability density function of observed variables as a mixture of multivariate normal densities. The goal of this method is to estimate the optimal parameters of a mixture of Gaussians in order to explain the observed data. For reach this goal, the algorithm estimate the responsibilities:

$$Z_{ik} = \frac{1}{Z_i} \pi_k N \left( x | \mu_k, \sum_k \right) \quad (2)$$

and the parameters:

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i \quad (3)$$

$$\sum_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k) (x_i - \mu_k)^T \quad (4)$$

$$\sum_k = \frac{N_k}{N} \quad (5)$$

Where

$$N_k = \sum_i Z_{ik} \quad (6)$$

Finally the algorithm iterate this responsibilities until reach the optimal.

GMM use the Mahalanobis metric to find how far are the data pixels from the centroid of the fixed distribution.

$$d(x_i, \mu_k, \epsilon_k) = \|x_i - \mu_k\|_{\sum_k^{-1}} = (x_i - \mu_k)^T \sum_k^{-1} (x_i - \mu_k) \quad (7)$$

As k-means the number of dimension of these distances depends on the number of features per pixel. In this way, the ruggedness of the classification depends on the number of feature that it has each pixel.

For this lab (as in the case of k-means) we use 6 cases, 3 of them with 3 features and the others 3 with 5 features. In the 3 feature cases, we use as feature the value of each pixel in each channel of RGB, LAB and HSV respectively. In the 5 feature cases, we use the same channel values but this time we add the position X and Y of each pixel.

### 1.3. Watershed

Watershed is an unsupervised classification method, this method also called watershed transform. The idea is to consider image in gray level as a topographic relief and it is put through to immersion process. In other words, each pixel of an image represents the elevation at this point [3]. The idea is imagine that the image is immersed in a lake with holes that are local minima. So, gradually submerge the surface into water and water coming from the holes form lakes; when two lakes meet, a dam is constructed. Finally when de surface is submerged all dams are watersheds lines [2].

This method is different to K-means and GMM because the parameter k do not exist, the parameter is h-minima transform, is like a threshold used to prevent the problem of over-fitting, this variant called watersheds markers-based where impose extended Regional Minima as markers. In this lab was used this variant but in this case was considered only RGB space because As mentioned already watershed transform use the image in gray scale.

## 2. Test Methodology

The data set contain 500 natural images. There are different size of images, some are 321x481 and others are 481x321. The data set are divided in three parts: 200 train images, 100 validation images and 200 test images. Furthermore, there are human annotations that are used like ground-truth. In other hand, benchmark evaluated segmentation, this by measuring Precision/Recall curve.

In the present lab was used first the train images to select the best two clustering models with the corresponding feature space. This selection was based the larger area under curve. After that, validation images was used to confirm the results of training subset phase Finally with the test subset of data and evaluated performance with Precision/Recall curve.

## 3. Results

Initially we implemented the three methods mentioned ( K-means, GMM and Watersheds) to cluster the train images. Some examples are presented below:



Figure 1: Train Image

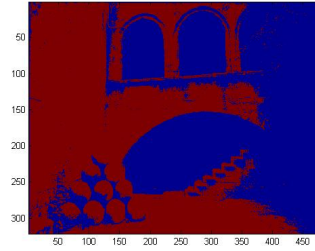


Figure 2: Train Image. K-means, k=2

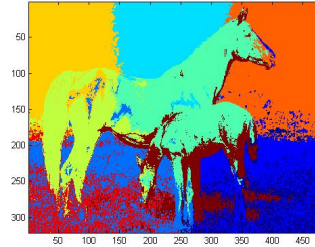


Figure 3: Train Image. K-means, k=10

The figure 1 show an example of train image and the figure 2-3 are clustering image with K-means k=2 and k=10 respectively. This clustering was made in feature space RGB-XY.

The next example show the difference in clustering between K-means (Figure 5) and GMM (Figure 6) with  $K=3$ .



Figure 4: Train Image

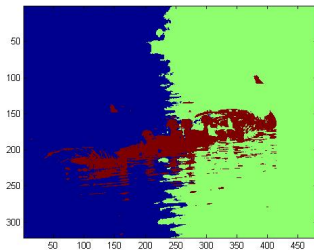


Figure 5: Train Image. K-means,  $k=3$

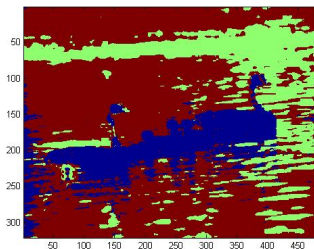


Figure 6: Train Image. GMM,  $k=3$

Finally, the figure 7-9 show a watersheds example with different  $h$ .



Figure 7: Train Image

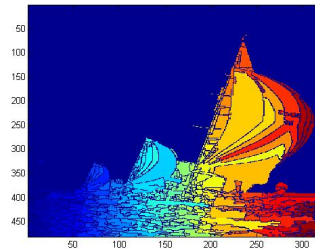


Figure 8: Train Image. Watersheds  $h=30$

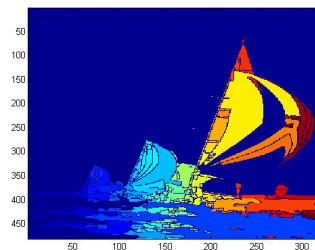
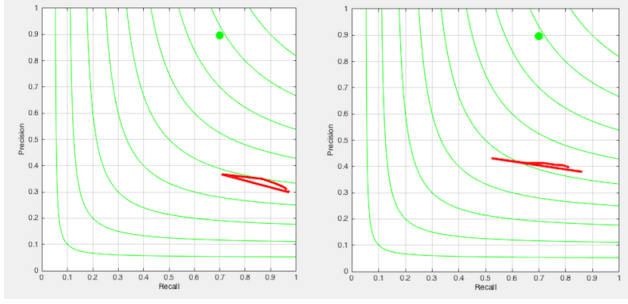
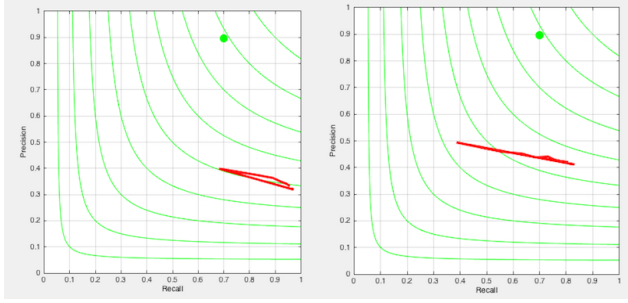


Figure 9: Train Image. Watersheds  $h=50$

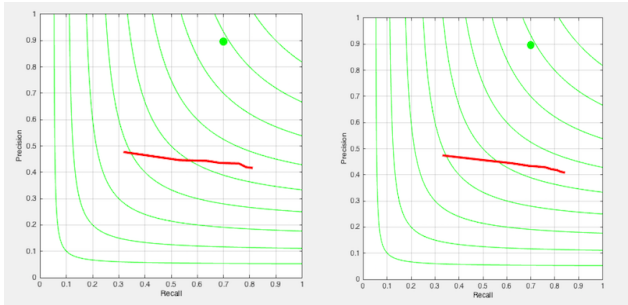
In the other hand, the selection of the best two clusters, as we explained above, was made based the larger area under curve. This area was found with the help of code provided by the professor Pablo Arbelaez. Some results of this area are presented then:



(a) RGB



(b) HSV



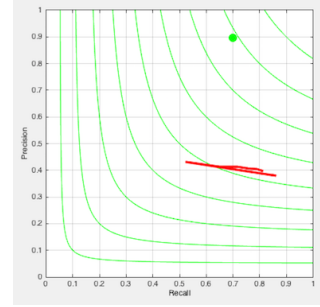
(c) LAB

Figure 10: Precision-Recall curves

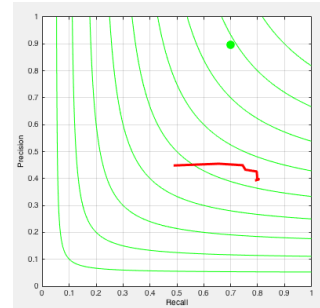
The figure 10 represents the results obtained for Kmeans in each one of the spaces. The first row correspond to the RGB space. The first Image, with a cover area of 0.11, correspond to the case with 3 feature. On the other hand, the second image, with a cover area of 0.14, correspond to a case with 5 features. The second row correspond to the lab space. The first Image, with a cover area of 0.20, correspond to the case with 3 feature. As the RGB space, the second image, with a cover area of 0.22, correspond to a case with 5. Finally the last row correspond to the HSV space. As the others spaces the first image correspond to the case with 3 feature and the second to the case with 5 features.

Once we obtained these image, we realized that the case with 5 features, as expected, is better than the case with 3 features. For this reasoned we decided to evaluate this

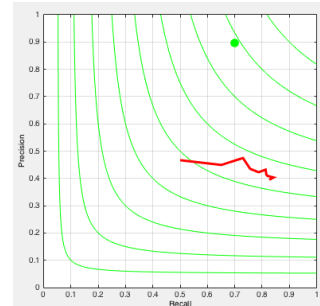
case only in GMM. The results of that are presented bellow (figure 11):



(a) RGB



(b) HSV



(c) LAB

Figure 11: Precision-Recall curves

As we said the figure 11 represents the results obtained for GMM in each one of the spaces with the position xy. The first row correspond to the RGB space, with a cover area of 0.11, The second image correspond to HSV space, with a cover area of 0.14. Finally the last row correspond to the Lab space , with a cover area of 0.15.

Following these results, we chosen Kmeans in labxy and GMM in labxy methods. The results obtained in the test files are presented bellow:

Figure 12 show the performance of Gaussian Mixture Model in feature space Labxy. In this case the cover area

0.14.

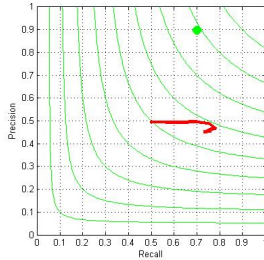


Figure 12: Precision/Recall Curve. Method: GMM in feature space Labxy

## 4. Discussion

These algorithms, especially k-means and GMM, have a strong dependence of the number of dimension for each pixel. In this way, the three channels of spaces or the position of the pixel on the image are not enough to achieve a good segmentation. As we can see, the segmentation is only good to images with a uniform background and an object that is clearly separated from the rest of the image. In other case, some patterns like texture or shape can influence the result and reduce the effectively of our method. A space like lab or hsv which take into account other features different to the color like luminosity are more effective and allow obtain a nearer distance to the human segmentation. Other important aspect is that watersheds is not a good method to cluster because much depends on  $h$ , so the results vary greatly between images. Is evident in train results that this methods do not have the best performance compared with humans, but visually you would think that are good methods.

## 5. Improvements

The influence of amount of dimension in this method is evident. For this reason we considered that the best way to improve our algorithm is evaluate more feature like texture. Similarly we should consider the problem of segmentation as a problem of contour. If we find the contour and use the gradient to find and eliminated the weak contour we could find a good approximation of the segmentation. Now well, if we use this contour like a feature of k-means is possible to improve the method with a simple process like apply canny and take the maximum of the gradient.

## References

- [1] P. Arbelaez. Lecture 5: Clustering. *Computer Vision*, page 27, 2016.
- [2] P. Arbelaez. Lecture 7: Perceptual Grouping 01. *Computer Vision*, page 105, 2016.

- [3] M. Bicego, M. Cristani, A. Fusiello, and V. Murino. Watershed-based unsupervised clustering. Technical Report Mm.
- [4] C. Guestrin. Unsupervised learning or Clustering Gaussian mixture models. *Machine Learning*, 2007.