Berkeley Segmentation Dataset and Benchmark

Maria Camila Escobar
Universidad de los Andes
Bogotá D.C, Colombia
mc.escobar11@uniandes.edu.co

Laura Gongas
Universidad de los Andes
Bogotá D.C, Colombia
l.gongas10@uniandes.edu.co

1. Introduction

The Berkeley Segmentation Dataset and Benchmarks 500 (BSDS500) is a dataset created by the Computer Vision group at Berkeley university. The goal of the BSDS500 is to provide a basis for research on image segmentation and boundary detection [2]. Before the BSDS was introduced, segmentation and boundary detection were thought to be ill-posed problems, since every human can create a different segmentation of the same image. However, the data in the BSDS includes ground-truth segmentations from various humans and it was found that, even though the ground-truth segmentations were not exactly the same for each case, there were regions that all ground-truths had in common.

Another contribution to research made by the BSDS was the creation of a standardized metric for judging the quality of a segmentation method. Martin et. al formulated boundary detection as a classification problem where the goal is to tell apart boundary pixels from non boundary pixels. A precision-recall curve was then made for varying hyperparameters of the algorithm and the maximum F-measure ($\frac{2\times Precision\times Recall}{Precision\times Recall}$) was calculated with respect to human ground-truth boundaries [3]. The ideal result of a precision-recall curve would be of a 1 in precision and 1 in recall, this would mean that the segmentation predicted every border in the ground-truth. However, given the nature of the segmentation problem, the consistency of the humans segmentation was calculated as 0.8 so this would be the best result that a segmentation algorithm could aim for.

K means is a partitional algorithm that assumes that the number of clusters is known. The main idea of the algorithm is to represent clusters with their centroids and find the centroids and clusters that minimize squared distances between elements and centroids. Lloyd's algorithm is an approximation that consists of: choosing k random centroids, assigning elements to clusters using a Voronoi diagram, computing new centroids and iterating until convergence is reached. The convergence is defined when there's no reassignments of elements to other clusters, no variance of centroids or when there's a minimum decrease in the sum of squared error [1].

Other algorithms of partitional clustering are model based like mixture of gaussians (gmm). The main idea of this method is to represent groups with gaussian distributions. In other words, clusters are formed by describing the probability density function of variables by a mixture of multivariate normal densities. Therefore, the objective is to find the parameters of a mixture of Gaussians that explain best the data [1].
The objective of this paper is to use the method proposed by [3] to evaluate the results of the segmentation algorithm that was made. The training, validation and evaluation phases will be made in the BSDS500 dataset.
The BSDS500 dataset consists of 300 training and validation images and 200 test images. The images are in jpg format and there are some in horizontal orientation (481 x 321) and others in vertical orientation (321x481). The dataset comes with a ground-truth for each image in .mat format, inside the ground-truth there are segmentations performed by five different subjects. Performance is evaluated by measuring Precision-Recall on detected boundaries, it also uses covering of ground-truth segmentations, Probabilistic Rand Index and Variation of Information benchmarks.

2. Materials and methods

The development of this method is divided in three parts. In the first part, the algorithm made in the previous lab was tested for the train dataset. This was done in order to select the clustering methods and color spaces that gave the best results. The criteria to determine the best configuration were the Best covering criteria given by the evaluation method and the time that the algorithm took running. Once the selection of the best configuration was made, we ran the algorithm through the validation dataset to find the optimal

range of clusters to obtain a full Precision-Recall curve. Lastly, we ran the test dataset with our two best configuration. For each of these we obtained the metrics of the evaluation method and compared them with each other and with the Ultrametric Contour Map segmentation algorithm.

3. Results

The algorithm was first tested for each of the clustering methods. Here we found that Hierarchical and Watersheds gave a better result than kmeans and gmm. However, the difference in results were of only 0.03 and the first two methods took three times more than kmeans and gmm. Since the algorithm was going to be tested in 200 images we considered the time to be a defining criteria so we evaluated our method with kmeans and gmm clustering.

The results of Optimal data scale (ODS), Optimal image scale (OIS) and Best coverage for the region benchmarks using the rgb, lab and hsv color spaces with gmm are shown in table 1.In figure 1 is shown the precision-recall curve for the best gmm configuration (rgb).

Table 1. Comparison of color spaces using gmm

	BDSD500						
	Covering			PRI		VI	
	ODS	OIS	Best	ODS	OIS	ODS	OIS
hsv	0.43	0.47	0.49	0.70	0.79	2.40	2.40
rgb	0.49	0.52	0.55	0.72	0.84	2.01	2.01
lab	0.42	0.44	0.47	0.76	0.78	2.35	2.32

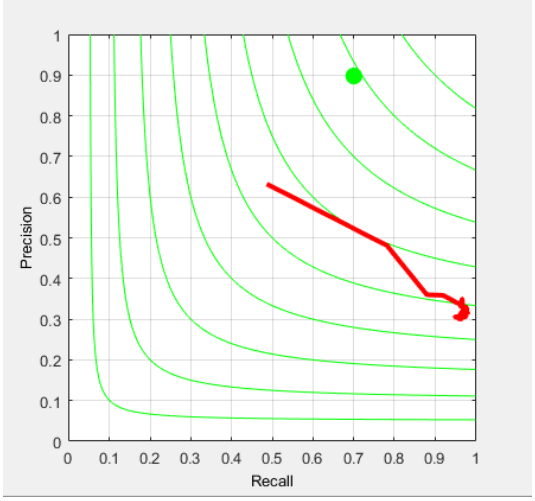

Figure 1. Precision-recall curve for gmm with rgb color space

The results of Optimal data scale (ODS), Optimal image scale (OIS) and Best coverage for the region benchmarks using the rgb, lab and hsv color spaces with k-means are shown in table 2.In figure 2 is shown the precision-recall curve for the best k-means configuration (rgb).

Table 2. Comparison of color spaces using kmeans

	BDSD500						
	Covering			PRI		VI	
	ODS	OIS	Best	ODS	OIS	ODS	OIS
hsv	0.43	0.46	0.48	0.69	0.78	2.49	2.49
rgb	0.46	0.48	0.52	0.73	0.79	2.27	2.22
lab	0.41	0.44	0.47	0.71	0.77	2.43	2.41

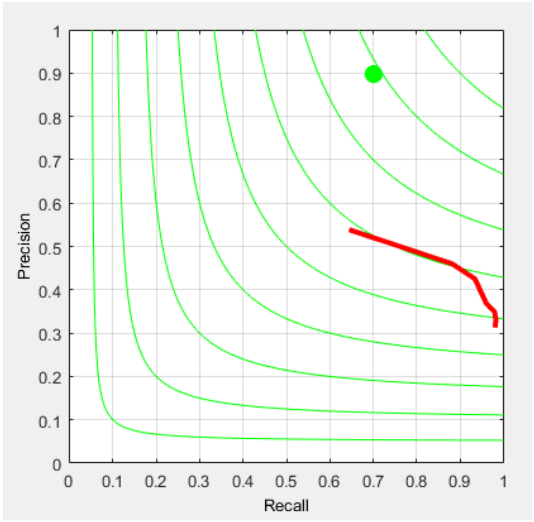

Figure 2. Precision-recall curve for kmeans with rgb color space

Finally, table 3 shows the results for the best configuration of gmm, k-means and the ucm algorithm. Additionally, in figure 3 it can be seen the precision-recall curves for all three algorithms.

Table 3. Comparison of color spaces using kmeans

	BDSD500						
	Covering			PRI		VI	
	ODS	OIS	Best	ODS	OIS	ODS	OIS
gmm	0.36	0.39	0.41	0.68	0.70	2.57	2.56
kmeans	0.35	0.38	0.40	0.70	0.72	2.67	2.66
ucm2	0.59	0.65	0.74	0.83	0.86	1.69	1.48

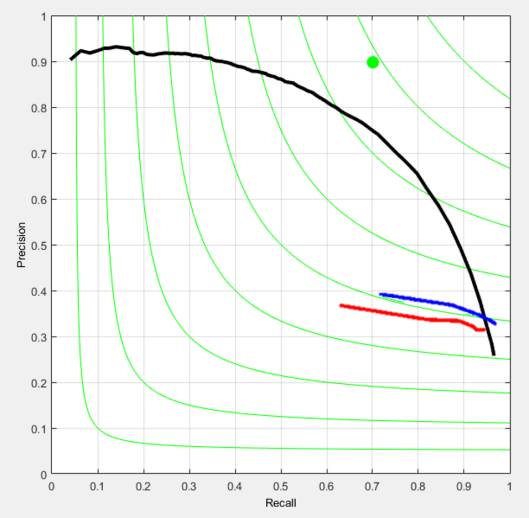

Figure 3. Precision-recall curve for the three algorithms

4. Discussion and Conclusions

According to table 1 the best configuration for gmm was found with the rgb colorspace. This result was consistent with the initial evaluations we made on our past laboratory. Additionally, in the figure 1 it can be seen that the recall goes in a range from 0.5 to 1. Additionally, in table 2 we can see that the best color space for k-means was rgb.

In table 3 it is possible to observe the results for all three algorithms. The best method among the ones we made was gmm. However, the methods we made do not come close to the result given by ucm. This can be attributed to different factors, for example the fact that ucm uses texton maps to discriminate between classes while our methods use only color representations. Additionally, our methods do not have as much recall as the ucm algorithm, we changed the range of clusters used trying to fix this but it did not make any difference. A recommendation to widen the Recall range would be to use hierarchical and watersheds clustering methods even though these take more time running.

Finally, something that is relevant to analyze is the fact that our methods gave a better result for the training dataset than for test. Since our training results are not optimal either, we can conclude that a limitation of our methods are that they are underfitting the data. We could improve this pattern by incorporating in our algorithm a more complex model, including textures and various features and a classifier such as Random Forest.

References

[1] P. Arbelaez. Lecture 5: Clustering. *Universidad de los Andes*, 2018.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[3] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004.