Laboratory 7: Berkeley Segmentation Dataset and Benchmark (BSDS500)

Ramón Emiliani
Universidad de los Andes
201125694
rd.emiliani689@uniandes.edu.co

Alejandro Posada
Universidad de los Andes
201227104
a.posada10@uniandes.edu.co

Abstract

Image segmentation is one of the principal tasks of computer vision. In this laboratory, we used the function we implemented in the previous laboratory to segment images from BSDS500. We used the methods that showed best performance —k-means and GMM— with lab+xy as the representation space. GMM outperformed k-means but UCM performed better that both of our methods.

1. Introduction

Image segmentation is a fundamental but challenging computer vision task. The goal of image segmentation is to cluster pixels into regions that correspond to individual objects, surfaces or natural parts of objects. Segmenting an image has two objectives. The first one is to decompose the image into parts that can be analyzed later. The second one is to perform a change of representation, *i.e.* to organize the pixels of the image into higher-level units that are more meaningful or can be analyzed more efficiently [4]. In the past decades, a wide variety of segmentation methods have been proposed. Roughly, these can be grouped in four categories [3]:

- Threshold based segmentation: histogram thresholding and slicing techniques are used to segment the image.
- Edge based segmentation: detected edges are assumed to be object boundaries in order to identify these objects.
- Region based segmentation: it takes an opposite approach to edge based segmentation. This technique "starts" in the middle of objects and "grows" outwards until it meets the objects' boundaries.
- Matching: this technique is based on the *a priori* knowledge of the shape of the object.
- Clustering methods: these techniques group together high-dimensional data points according to a criteria of similarity such as distance.

In this laboratory, we focus on the latter category to perform image segmentation. In order to compare the performance of different segmentation methods, in this laboratory we tested the function we implemented in the previous laboratory. We chose the two best methods implemented in the function and evaluated them in the Berkeley Segmentation Data Set and Benchmark 500 (BSDS500).

2. Materials and methods

We used the segmentation function implemented in the previous laboratory, whose inputs are an RGB image, a feature space (RGB, LAB, HSV, RGB+XY, LAB+XY or HSV+XY), the segmentation method (k-means, gaussian mixture model, hierarchical or watershed) and a number of clusters.

```
segmentByClustering(rgbImage, featureSpace,
    clusteringMethod, numberOfClusters)
```

In order to select the two best segmentation methods, in the previous lab we tested our function on a subset of the BSDS500. The two methods that showed best performances are k-means and gaussian mixture model (GMM). K-means is a very intuitive and well-known method to group a training set $x_1, ..., x_m$ into k clusters, where each data point $x_i \in \mathbb{R}^n$. The algorithm works by initializing the cluster centroids $\mu_1, \mu_2, ..., \mu_k \in \mathbb{R}^n$ randomly. Then, every x_i is assigned the label of the closest μ_j according to a similarity measure such as euclidean distance. Finally, each centroid μ_j is recalculated by averaging all the x_i that belong to the cluster. These steps are repeated until convergence.

GMM assumes that the data points come from a gaussian mixture distribution which consists of k gaussians with their own means and variances. Thus, GMM can be seen as a generalization of k-means that incorporates information about the covariance structure of the data and the centers

of the latent gaussians. GMM is based on the Expectation-Maximization (EM) algorithm and is resumed in the following steps:

- Initialize the centers $\mu_1, \mu_2, ..., \mu_k \in \mathbb{R}^n$, the covariance matrices $\Sigma_1, \Sigma_2, ..., \Sigma_k$ and the mixing coefficients $\pi_1, \pi_2, ..., \pi_k \in \mathbb{R}$, which are estimations of the fraction of points in each gaussian.

- E step: for each point x_i and gaussian compute the responsabilities $z_{ik} = \frac{1}{z_i}\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)$

- M step: compute the parameters $\mu_k = \frac{1}{N_k}\sum_i z_{ik}x_i$, $\Sigma_k = \frac{1}{N_k}\sum_i z_{ik}(x_i - \mu_k)(x_i - \mu_k)^T$ and $\pi_k = \frac{N_k}{N}$, where $N_k = \sum_i z_{ik}$.

  The E and M steps are repeated until convergence.

Both methods have the same hyperparameter, the number of clusters k. There is no "right" way to choose k, only heuristics and rules of thumb. For example, the number of clusters can be chosen by visually inspecting the data, by computing the sum of squared error (SSE) for some values of k and choosing a value of k such that the SSE drops abruptly (elbow method), calculating the silhouette to determine how well each object lies within its cluster, etc. In this laboratory, we inspected the images and determined that most of the images had between 3 and 9 discernible objects/regions. Thus, the set of k we used in our methods is $3, 5, 7, 9$.

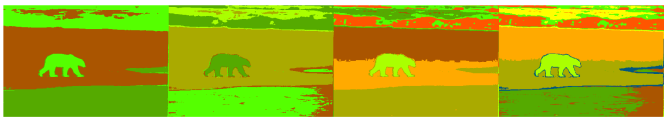
For both segmentation methods the same representation space, lab+xy, was chosen. Once again, this decision was taken based on the results observed in the previous laboratory, *i.e.* performing segmentations in a subset of BSDS500. In most cases, the best segmentations were obtained when the representation space was lab+xy. This can be explained because lab is a perceptually uniform color space and, additionally, the x and y components allow the spatial information to be taken into account when segmenting the image.

We tested both methods in the test subset of the BSDS500, which is composed by 200 images. We also ran the benchmark on the provided Ultrametric Contour Map (UCM) segmentations. The results were evaluated using a precision-recall curve. Precision is a measure of result relevancy and is defined as $P = \frac{TP}{TP+FP}$, while recall indicates how many truly relevant results are identified and is defined as $R = \frac{TP}{TP+FN}$ (TP = true positives, FP = false positives and FN = false negatives). Thus, precision-recall curves illustrate the tradeoff between precision and recall. The ideal result would be a precision of 100% for all recall values. This would imply a F score, which is defined as $F = 2 \times \frac{P \times R}{P+R}$, of 1.

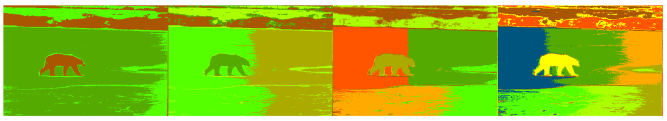
3. Results

Figures 1, 2 and 3 show examples of segmentations obtained with the k-means and GMM methods with $k = 3, 5, 7$ and 9.

(a) Original image.



(b) GMM segmentation.



(c) K-means segmentation.

Figure 1: Segmentations with both methods and $k = 3, 5, 7$ and 9.

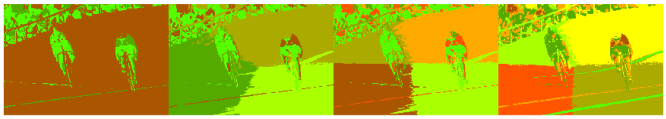
(a) Original image.

(b) GMM segmentation.

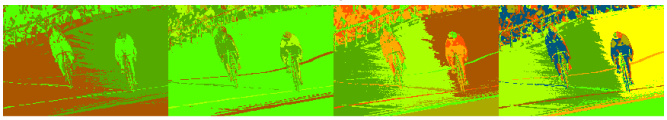
(c) K-means segmentation.

Figure 2: Segmentations with both methods and $k = 3, 5, 7$ and 9.

(a) Original image.



(b) GMM segmentation.



(c) K-means segmentation.

Figure 3: Segmentations with both methods and $k = 3, 5, 7$ and 9.

It is clear that each method produces different segmentations. However, both k-means and GMM tend to segment uniform regions into different clusters. Note, for example, the ground in figure 1, the children's clothes in figure 2 and the race track in figure 3. The elements/surfaces that were not correctly segmented share a common trait: texture. Thus, a possible improvement to the segmentation methods could be introducing texture in the space representation. A representation that took into account color, position and textons would probably avoid segmenting uniform regions into different clusters.

Figure 4 shows the precision-recall curve of segmentations obtained with the k-means, GMM and UCM methods.

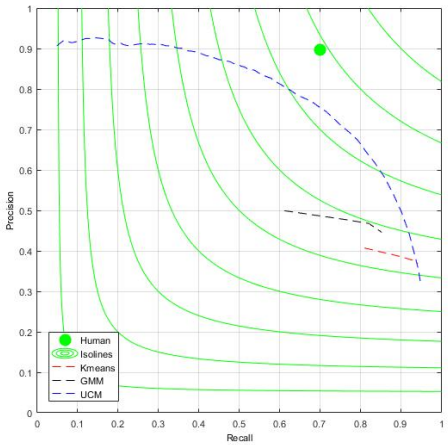


Figure 4: Precision-Recall curve for K-means, GMM and UCM.

Our methods curves' domains are limited (the curves obtained with our methods are *shorter* than the UCM curve) because the number of clusters k was limited and small (4), so possible recalls (*i.e.* the curves' domains) are also limited.

In all cases, the results from UCM beat k-means and GMM. GMM outperformed k-means (however, note that the results are not always comparable since the PR curves are not defines for the same values of recalls). This can be explained since GMM doesn't depend on the L2 norm because it is based on expectation, *i.e.* the probability of a point belonging to a cluster. Thus, unlike k-means, it is not biased towards spherical clusters. UCM outperformed our methods because it approaches segmentation in a more hollistic way: it uses a higher-level representation space because it assumes the image is a graph $G(P_o, K_o, W(K_o)$, where the nodes are the regions P_o, the links are the arcs K_o and the weights $W(K_o)$ are a measure of dissimilarity between regions [1]. Thus, UCM incorporates information that our methods do not use in order to segment the image. Also, both k-means and GMM highly depent on the choice of k. Given that the initialization of the cluster centroids is random, repeteability in our results is very unlikely. Our results could be improved by using a better initialization technique. For example, k-means++ finds optimal initial cluster centers [2], making k-means a more robust technique.

4. Conclusions

Even though our methods were outperformed by UCM, we saw that simple clustering methods such as k-means and GMM can achieve decent segmentation results. Several improvements could be made to our methods: modifying the representation space, using edge detection methods to determine candidate regions, improving the initialization of the centroids, selecting k in a more rigurous way by using specific initialization algorithms like k-means++, etc. Also, pre-processing the images could help: an average filter would eliminate noise and would smooth textures, so objects and surfaces would become more uniform and therefore could be better segmented. These possible improvements point in the same direction: clustering raw images is not enough to obtain state of the art segmentation results.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[4] G. Stockman and L. G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.