Laboratory exercise 7: Segmentation benchmarking

Santiago Enrique Cortés
Bárbara & Frick
Av. Calle 82 # 10 18A - 12, Bogotá, Colombia
santiago.cortes@barbara.net.co

Juan Felipe Cerón Uribe
Universidad de los Andes
Cra 1 # 18A - 12, Bogotá, Colombia
jf.ceron10@uniandes.edu.co

1 Introduction

In this exercise, we compared the quality of the segmentations obtained from the UCM algorithm to those obtained from two older algorithms. We used our own implementation of the last two. The algorithms were compared on the basis of the contours which can be inferred from the segmentations resulting from each. These contours were compared to human-drawn contours by treating them as a classification of each pixel (as contour or non-contour).

2 Dataset

For the experiments shown in this article the BSDS-500 (The Berkeley segmentation and benchmark) data set was selected. The former consists in 500 images all with segmentations done by 4 different subjects. The two implemented methods are unsupervised classifiers, hence there was no need to use the train samples. Following this train of tough, for the purpose of the exercises expose here, only the test set was used, it consists of 200 images.

3 Algorithms

3.1 k-means

The standard heuristic algorithm to the NP-complex problem of k-means consists of an initialization, followed by the iterated repetition of an assignment and an update step until convergence is reached:

- **Initialization:** k different points in $\mathbb{R}^d$ (if the data is d-dimensional) are chosen at random. They are called centroids.
- **Assignment:** Each point in the data is assigned to the *cluster* of its nearest centroid. Ties should be rare.
- **Update:** The centroid of each cluster is recalculated from its points.

3.2 Watershed clustering

This algorithm is better explained by a flooding analogy. For this purpose, we conceive a gray-scale image as the topological surface where each point's height represents that pixel's intensity. The algorithm proceeds by:

1. Make a hole in several regional minima.
2. Gradually submerge the surface in water.
3. Whenever two lakes meet, construct a damn to prevent them from mixing. The final set of lakes are considered the segments of the image.

Following a recommendation from a scikit-image example [1], we carried this process out on a gradient representation of each image. We obtained this representation by averaging the gradient representation of the images in each of the channels. Markers (where we poke holes in the surface) were chosen by taking connected sets of low gradient pixels. We did not take into account spatial dimensions for this algorithm, as they (and their gradients) draw trivial topological surfaces.

3.3 UCM

4 Parameter tuning

One tuning parameter, which is ubiquitous across almost all clustering algorithms, is the number k of clusters to be consolidated. We now present a few ways in which we might choose it.

1. We might use domain knowledge of how many objects are present in the dataset's images to fix this parameter.
2. If we have an annotated training set, we could obtain a list of the number of segments in each of the images. If the numbers have a low standard deviation, we could fix k as the integer closest to their mean. We could also cluster the images in search for a clustering in which images in one cluster have a similar number of segments. Then, for a test image, we could associate it with one of those clusters, and use the cluster's mean number of segments as parameter k for the segmentation of the image.

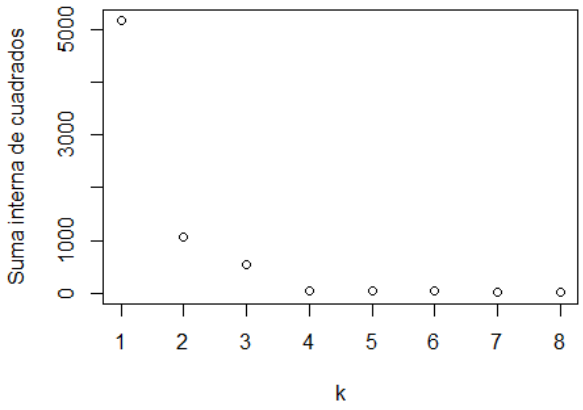

Figure 1. Graph-based approach to selecting k in the k-means algorithm.

3. There is a famous way to pick k in the k-means algorithm. It consists of graphing out the internal sum of square errors between points and their clusters' centroids (after performing the algorithm) against the parameter k, as shown in figure 3. When the sum of square errors ceases to decrease fast, this is a sign that we are near the appropriate k. We can use this method for any clustering algorithm, since we can calculate centroids regardless of how we grouped the points.

4. Drawing inspiration from the previous method, we could perform the clustering algorithm for a long sequence of consecutive k, and measure (and graph out) the mutual information between consecutive clusters. When this distance metric begins to become very small between consecutive k, we can take this as a sign that a clustering with more clusters is simply catching a new, small, irrelevant cluster, and thus we are near a good k.

5 Evaluation methodology

For each image in the dataset, we also have human-determined contours of the objects in the image. We took these as the true contours and compared them to the ones we could infer from each of the segmentation algorithms. This inference was made by declaring that pixels that are close to the border between different segments are contour pixels, and the rest aren't. The comparison between the ground truth and our algorithms' results could then be made by treating the problem as a classification problem, in which we try to classify each pixel as contour or non-contour.

	K-means	Watershed	UCM
ODS	F(0.86,037)=0.52	F(0.18,0.59)=0.27	F(0.73,0.73)=0.73
OIS	F(0.86,0.40)=0.55	F(0.18,0.63)=0.28	F(0.75,0.77)=0.76
AREA_PR	0.10	0.05	0.73
Threshold	2	5	0.13

5.1 Metrics

Transporting the segmentation problem to the classification setting means we can use the typical evaluation metrics for this kind of problem. We thus measure precision (ratio of true positives to true contour pixels) and recall (fraction of detected contour pixels) for each of the algorithms at different parameter choices. Specifically, we let each of our algorithms find 2, 3, 4, 5 and 6 segments in each image. We then graphed the precision and recall of each run as an ordered pair, the result is presented in figure 6.

Additionally, we report the ODS, OIS and AP measures, which we now define:

- The **ODS** (optimal dataset scale) is the is the maximum F-measure obtained by the algorithm at some parametrization when trying to classify each pixel as contour or non-contour. Notice how this measure ignores the fact that pixels belong to different images.
- The **OIS** (optimal image scale) is the aggregation of the best F-measure attained by the algorithm when classifying the pixels of each image separately.
- The **average precision** is exactly that, taken over the recall range (we did not have enough time to produce the full recall range for all algorithms, as evidenced by the short colored lines in figure 6).

5.2 BSDS benchmark

6 Results

Given the set-up of the experiment with a few clusters for every segmentation it can be observed that watershed tends to output fewer borders than k-means. Further more watershed is biased towards segmentation with fewer borders and conversely, k-means is biased towards segmentation with numerous regions. The former can be reflected in the ROC curve where it can be appreciated that the watershed algorithm have a low recall in contrast to the k-means method that has a low precision.

References

[1] Markers for watershed transform. http://scikit-image.org/docs/dev/auto_examples/segmentation/plot_marked_watershed.html. Accessed: 2010-09-30. 1

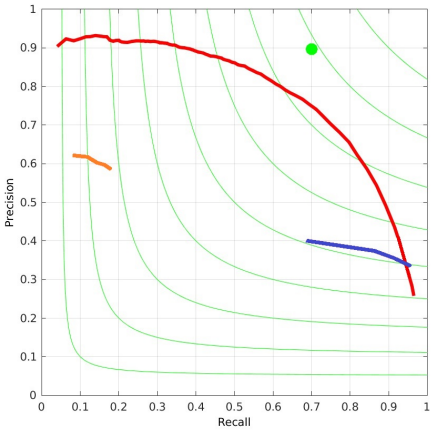

Figure 2. The image of the 3 ROC curves obtained by the three methods. The red on is the ucm method, the orange one is the watershed and finally the blue one corresponds to k-means.

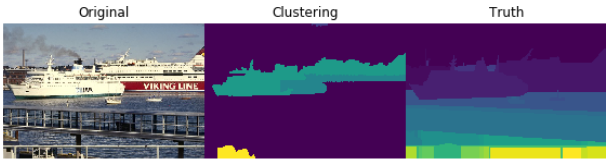

Figure 3. An example of a segmentation with watersheds. Notice how there is little contour.

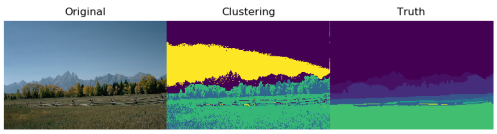

Figure 4. An example of a segmentation with k-means clustering. Notice how there are many contours.