K-means and GMM in the Berkeley Segmentation Dataset and Benchmark

Catalina Gómez
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
c.gomez10@uniandes.edu.co

Diana Herrera
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
ds.herrera10@uniandes.edu.co

Abstract

Segmentation is one of the main problems studied within the field of Computer Vision. This problem aims to group pixels corresponding to the same object on an image, depending on its representation within a high-dimensional space, obtaining regions that are delimited by contours. Using the BSDS500, performance between k-means and GMM was compared based on Lab feature space representation. Evaluation was made based on the benchmark provided with the dataset and the best results were obtained when choosing equally spaced number of clusters from 3 to 23 with k-means methods and an average precision of 0.09. Based on the segmentation methodology proposed by [1], it was analyzed why our two models did not perform well when evaluating segmented regions and contours.

1. Introduction

Segmentation is one of the main problems studied within the field of Computer Vision. This problem aims to group pixels corresponding to the same object on an image, that are delimited by contours. In order to address this problem, each pixel is a candidate for assigning it a semantic label, which depends on its representation within a high-dimensional space.

Clustering algorithms have been used for addressing segmentation problems, since these algorithms require data, but not labels to group pixels together into clusters, or regions. The goal is to assign pixels to each cluster based on a measure of similarity, that must be greater within pixels of the same cluster, and lesser between pixels from different clusters. The input for these algorithms requires a representation of the pixels with some features that allow for an accurate discrimination. Once these features are computed, the grouping result depends on the method employed by the algorithm to cluster.

Perhaps the best known segmentation method is K-means algorithm, which belongs to a family of unsupervised learning techniques. In addition, it falls into the group of hard clustering, where each data point belongs to only one cluster. It works partitioning data into k clusters, parameter that must be known a *priori*. Every point is assigned into a cluster based on its distance to the center of the cluster, called centroid, in such a way that centroids and clusters are chosen to minimize the squared distances between points and centroids [2].

A similar unsupervised segmentation method is Gaussian Mixture Model (GMM), but unlike k-means this one performs a soft assignment, giving each point the probability to belong to one or more clusters, each one represented by a Gaussian distribution and without a unique stated limit. Hence, groups in this algorithm are assumed to be normally distributed and are represented by multivariate Gaussian distributions, configured to optimize the model through the Expectation Maximization algorithm (EM). To do so, the first step (E step) is to use random parameters for the Gaussian distributions and estimate the responsibilities of each point, called soft assignments. Afterwards, M step is performed in which parameters of the Gaussian distributions are estimated again until a minimum is found. An important limitation is that the minimum found can be local and hence the result obtained would not be the optimal result [3].

Although these methods are computationally expensive compared with other clustering methods developed, they showed the best performance on a trial made with 25 images from the dataset used here when images were represented in a Lab feature space. Both are simple but yet powerful clustering methods because there is a unique hyperparameter needed to configure them: the number of clusters. Nevertheless, it is hard to choose a unique k to run the algorithms because annotations have different amount of clusters depending on the subject that made it. Therefore, we repeated the segmentations using values from 3 to 7 number of clusters to have different segmentations and compare performance. In this opportunity, only these two methods will be used, both of them based on a Lab feature space representation.

Additionally, it is imperative to introduce the dataset

used in this work. The Berkeley Segmentation Dataset and Benchmark 500 (BSDS500) is the extension of the previous version (BSDS300), which is the most used dataset for segmentation worldwide. It was made public on 2011 thanks to an effort made at Berkeley to put together 200 images for training, 100 images for validation and other 200 images for testing segmentation algorithms, each one annotated by five different subjects on average, to be evaluated through precision recall on region contours and three additional evaluation metrics based on regions [1].

In fact, although the algorithm is focused on segmentation and precision recall is a method mostly used in detection problems, there is a duality between them. When segmentation is made and objects are recognized, each one of them has a boundary that separates it from the others. The set of these object boundaries form the contours of the image. On these contours, evaluation can be done through precision recall just as it is made on any detection task, where each contour pixel from the ground truth is taken as a point that should be detected or obtain when segmentation is done.

Therefore, the precision-recall curve is an especially useful evaluation method for detection problems because it shows the performance of the algorithms with different thresholds, giving an broad idea of its quality. When varying the threshold, a trade off between precision (how many points identified as positives were actually positives) and recall (how many of the image positives were identified) happens; using a low threshold will increase the number of pixels identified as positives, increasing recall but decreasing precision by rising the amount of false positives. Contrastingly, using a high threshold will decrease the number of pixels identified as positives, increasing precision but decreasing recall by rising the amount of false negatives. For each threshold, precision is calculated as True Positives divided over the sum between True Positives and False Positives, and recall as True Positives divided over the sum between True Positives and False Negatives. Hence, the ideal algorithm would have an equilibrium between precision and recall, reaching 1 in the curve, which would mean no False Negatives or False Positives given by the algorithm.

2. Materials and Methods

K-means and GMM algorithms were applied for train, validation and test images of BSDS500. Evaluation was made through precision recall curve on contours based on the bench fast evaluation method available on the Berkeley Computer Vision web page.

We evaluated five scenarios for each method on the whole test set:

- 5 K's from 3 to 7 with no preprocessing.
- 5 K's from 3 to 7 preprocessing each image with a gaussian filter of size 15 and sigma=10.
- 5 K's from 3 to 23 increasing in 5 units each.
- 12 K's from 3 to 15 preprocessing each image with a gaussian filter of size 7 and sigma=5.
- 5 K's from 3 to 7 preprocessing each image with a median filter on a 5 window size.

The only hyperparameter needed for both methods is *number of clusters*, which was varied to obtain and compare different results and to recreate a larger piece on the precision recall curve. This number must be chosen according to the computational power available, because using a bigger k parameter increases significantly the number of calculations required and hence the computational power. The parameter must also be chosen depending on the detail of the segmentations that wants to be obtained, therefore, if lots of clusters comprising small details want to be obtained, a higher k should be used; but if only big structures and no details want to be segmented, then a small k should be used to avoid over segmentation, as was the purpose of this paper. It was necessary to use a broad range of number of clusters because using only five values similar between them, for example from 3 to 7, only a small part of the precision recall curve can be calculated, whereas using different values separated among them or more values, allowed us to build a larger part of the precision recall curve, although computational power increased massively.

3. Results

The following figures show the precision recall curves for k-means on train, validation and test sets using number of cluster parameter from 3 to 7.

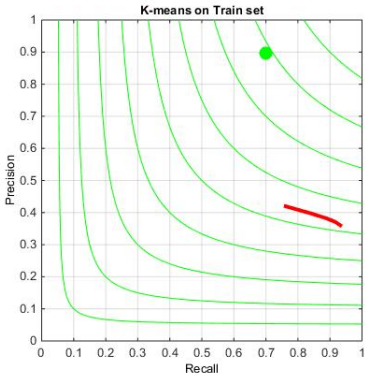

Figure 1. Precision-Recall for k-means method on Training set. k=3:7

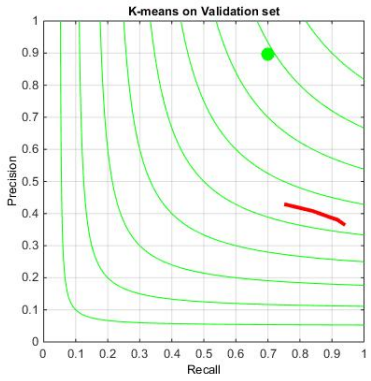

Figure 2. Precision-Recall for k-means method on Validation set. k=3:7

The results on the test set for both methods are presented in Fig. 3, together with the curve from [1] and the human annotations. The blue curve corresponds to k-means, the red one to GMM, the purple one to gPB-owt-ucm method, the green dot to human annotations and the green lines to the different isolines for F-measure. This convention is kept during the article. From the curve, it can be observed that k-means yielded better results when running the segmentation with 5 consecutive k's.

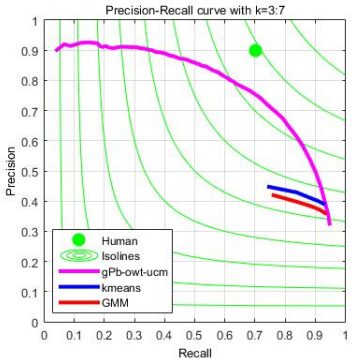

Figure 3. Precision-Recall for k-means and GMM method on Test set. k=3:7

A pre-processing step was applied to the test set, in which we applied a gaussian filter with size 15 and sigma=10 in order to improve the results from the segmentations. The results for both methods are shown in Fig. 4.

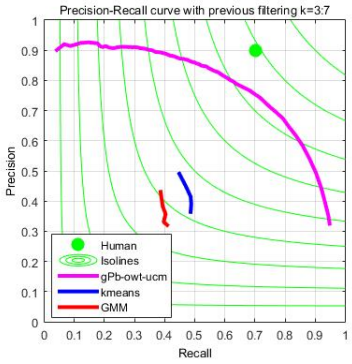

Figure 4. Precision-Recall for k-means and GMM method with previous filtering on test set. k=3:7

In addition, we evaluated the results of the segmentations with and increased number of clusters(k), beginning at 3 clusters and increasing it by 5 until 23. The results are shown below in Fig. 5, where they are compared with the curve of [1] and the human segmentations, that correspond to the green dot. There is little differences between the curves for the k-means and GMM methods.

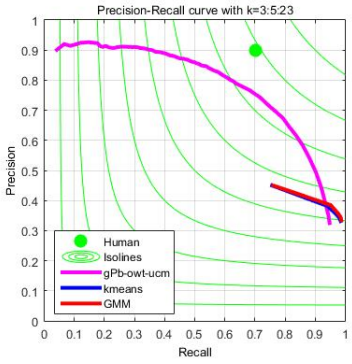

Figure 5. Precision-Recall for k-means and GMM method with spaced k=3:5:23.

Likewise, the methods were evaluated with a large amount of k's (12 per method) from 3 to 15 preprocessed with a gaussian filter of size 7 and $\sigma = 5$. Results are shown in Fig. 7 for both k-means and GMM methods:

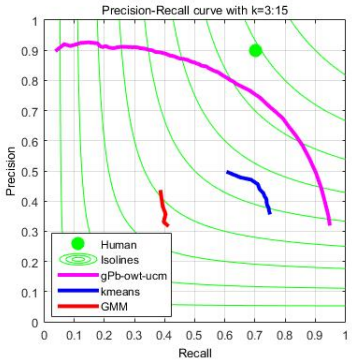

Figure 6. Precision-Recall for k-means method with 12 k's from 3 to 15.

Finally, the methods were evaluated with 5 k's ranging from 3 to 7 but images were previously filtered with a median filter in a 5 window size. Results are shown in the following image.

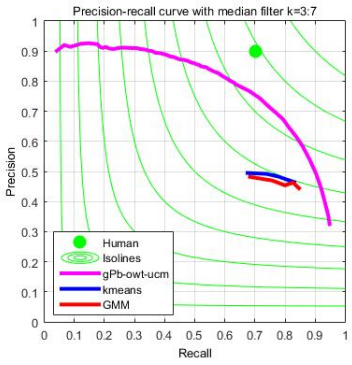

Figure 7. Precision-Recall for k-means method with 5 k's from 3 to 7 with median filtering.

The additional information of the Precision-Recall curves (for all the different experiments) related with boundary and region evaluation results is shown in tables 1 and 2, respectively.

Table 1. Boundary evaluation results.

Case	Boundary ODS	Boundary OIS	Area PR
K-means,Train,k=3:7	F(0.78,0.41)=0.54 th=1.21	F(0.88,0.43)=0.58	0.07
K-means,Validation,k=3:7	F(0.84,0.41)=0.55 th=1.93	F(0.86,0.45)=0.59	0.07
K-means,Test,k=3:7	F(0.84,0.43)=0.57 th=2	F(0.87,0.46)=0.6	0.08
GMM,Test,k=3:7	F(0.78,0.41)=0.54 th=1.21	F(0.88,0.43)=0.58	0.07
K-means,Test Filtered,k=3:7	F(0.45,0.49)=0.47 th=1	F(0.51,0.48)=0.49	0.02
GMM,Test Filtered,k=3:7	F(0.39,0.43)=0.41 th=1	F(0.46,0.43)=0.45	0.01
K-means,test,k=3:5:23	F(0.76,0.45)=0.56 th=1	F(0.86,0.44)=0.58	**0.09**
GMM,test,k=3:5:23	F(0.77,0.45)=0.57 th=1.03	F(0.87,0.44)=0.58	**0.09**
kmeans,test,k=3:15	F(0.69,0.47)=0.56 th=3	F(0.71,0.51)=0.59	0.06
GMM,test,k=3:15	F(0.39,0.43)=0.41 th=1	F(0.46,0.43)=0.45	0.01
K-means,test,k=3:7,median filter	F(0.81,0.47)=0.6 th=3.87	F(0.8,0.52)=0.63	0.08
GMM,test,k=3:7,median filter	F(0.83,0.46)=0.59 th=4	F(0.82,0.51)=0.63	0.07

Table 2. Region evaluation results.

Case	GT covering			Rand Index		Var Info	
Case	ODS	OIS	BEST	ODS	OIS	ODS	OIS
K-means,Train,k=3:7	0.42	0.46	0.49 (TH=1)	0.72	0.75 (TH=4)	2.43	2.38(TH=1)
K-means,Validation,k=3:7	0.41	0.45	0.47 (TH=1)	0.71	0.74 (TH=4)	2.5	2.44(TH=1)
K-means,Test,k=3:7	0.41	0.45	0.48 (TH=2)	0.74	0.76 (TH=5)	2.54	2.47(TH=1)
GMM,Test,k=3:7	0.42	0.46	0.49 (TH=1)	0.72	0.75 (TH=4)	2.43	2.38(TH=1)
K-means,Test Filtered,k=3:7	0.42	0.46	0.49 (TH=2)	0.73	0.76 (TH=5)	2.48	2.44(TH=1)
GMM,Test Filtered,k=3:7	0.42	0.45	0.49 (TH=1)	0.72	0.75 (TH=5)	2.48	2.44(TH=1)
K-means,Test,k=3:5:23	0.41	0.43	0.48 (TH=1)	0.74	0.77 (TH=2)	2.54	2.53(TH=1)
GMM,test,k=3:5:23	0.43	0.45	0.50 (TH=1)	0.74	0.77 (TH=3)	2.48	2.46(TH=1)
K-means,Test,k=3:15	0.42	0.46	0.51 (TH=2)	0.74	0.78 (TH=8)	2.48	2.42(TH=1)
GMM,Test,k=3:15	0.42	0.45	0.49 (TH=1)	0.72	0.75 (TH=5)	2.48	2.44(TH=1)
K-means,test,k=3:7,median filter	0.42	0.46	0.49 (TH=2)	0.74	0.76 (TH=5)	2.49	2.4(TH=1)
GMM,test,k=3:7,median filter	0.45	0.49	0.53 (TH=1)	0.73	0.77 (TH=5)	2.34	2.26(TH=1)

4. Discussion

Segmentation methods chosen were K-means and GMM using a Lab feature space, because the overall results obtained with these methods in a 25 image sample from BSDS500 over performed the ones obtained with methods such as watersheds and hierarchical in feature spaces such as RGB and HSV in addition to xy coordinates information. Although these methods are computationally expensive, the other options were extremely sensible to the k parameter chosen and annotations do not have a unique known number of clusters annotated.

The results obtained when applying the previous filtering were the worst ones, since they had the lowest average precision (AP) values, 0.02 with K-means and 0.01 with GMM. This result is consistent with the fact that the evaluation metric takes into account contours from the segmentations and the ones annotated by humans, and the process of filtering smoothens the edges and contours of the original image. The degree of smoothness depends on the parameters chosen for the kernel of the filter, specially on the standard deviation of the gaussian, where the bigger σ, less details are preserved. This is why it is not a good idea to apply filters that smooth the edges, but instead it would be better to apply filters that enhance contrast.

Furthermore, it is noticeable that increasing k increases recall but precision does not increase. This makes sense because if a larger k is used, then more objects will be segmented and so does the amount of contours detected, making it more probable that the annotated contours are detected increasing recall. Nevertheless, precision is not affected because the number of pixels detected as positives grows but probably many of these are false positives, affecting precision result.

Running the segmentations with more spaced k's provided the best results, in terms of AP (0.09). This could be attributed to the fact that increasing the number of k, while keeping differences between them, yielded better discriminative segmentations at the time of evaluating them with respect to annotations. On the other hand, increasing the number of segmentations did not improve the results as much as expected because as it was mentioned before it increased the recall, but including consecutive k's does not lead to significant differences between segmentations to evaluate. In

addition, it must be taken into account that data of the experiment with more k's was pre-processed, and this filtering step was not a good idea, which is reflected in the decreased AP. In addition, it is pertinent to note that when segmenting with spaced k's, the results of k-means and GMM methods differed slightly, with GMM being a little bit better. These two methods can yield similar results when the gaussian distributions adjusted to the data have a low standard deviation, and then the clusters would be as in k-means method.

When comparing the results shown in Fig. 3 and Fig. 5, it must be noted that the curves are similar because the curve of figure 3 contains points that are a subsample of the points plotted in Fig. 5, such as k=3 and k=8 (very close to k=7). Then, one can infer that the extra segment of the curve is due to the larger k values, that increase the recall as it was explained in the previous paragraph. A similar effect was observed when comparing Fig. 4 and Fig. 7 for k-means methods, where increasing the number of k's from 5 to 12 improved the recall (making the curve longer), but not the maximum precision.

The results obtained are not as good as the ones obtained with the gPb-owt-ucm method because it takes into account a segmentation hierarchy, where there is a set of contours at different levels, which are used as thresholds to compute the precision recall curve. These closed contours delimit the regions or objects within each image. Therefore, different segmentations at each level or scale could be retrieved from the UCM by choosing a threshold. Both the information from boundaries and regions is compared with the corresponding groundtruth annotations. The procedures and different techniques included within gPb-owt-ucm improved the performance of th segmentations since it deals with the problem of oversegmenting uniform regions by merging them within the same object.

It is evident that K-means and GMM performances are not satisfactory in the whole dataset, although in all the cases k-means is slightly better than GMM. In the previous test done with more clustering algorithms and feature representation, these two (k-means and GMM in La*b* space) showed better performance, but the result was the opposite because GMM over performed K-means by a small difference. This might be because that time, the test was done only with 25 images from the dataset, using only k=3 and manually choosing the annotation closer to the one we were looking for; but this time, we used the whole 200 image database with several annotations containing various amounts of k's, which is closer to a general case.

The difference among both algorithms is big when filtering preprocessing was done, in the other two cases their performance was similar. This is due to the fact that filtering affects the contours making them blurry, affecting the soft assignment that GMM does to contour pixels, increasing the probability of mixing labels from contiguous objects in them, leading to a bigger error specially represented in a reduced recall, because contours are no longer detected in the specific place they were before filtering the image. On the other hand, k-means does a hard assignment and this reduces the probability of mixing labels from contiguous objects in blurry contours.

Furthermore, on the last laboratory we evaluated the results of our segmentation methods by comparing them with only one of the annotations (chosen by our preference), and computing Jaccard Index average for each clustering method and representation space. For the Jaccard Index we only took into account the three and two biggest objects that were segmented. In addition, the segmentations were computed with only one number of clusters (k=3). In this laboratory, we run the segmentations with more k's in order to have something similar to a hierarchy of segmentations, where the level corresponds to the number of objects or segmented regions. When evaluating the results of the segmentation, this benchmark takes into account not only the regions of the segmented objects, but also its boundaries, and the match with the annotated segmented objects and boundaries. By including different k's to segment each image, this parameter was the threshold value used to generate the points of the precision-recall curve. The evaluation of the regions also considers different selection of thresholds, such us one fixed for all the data set and a fixed scale for each image [1].

One of the limitations of the algorithms used to segment the images was the feature space used to represent the pixels, since they only took into account color information in one color space, that could benefit or not a single image depending on its color palette or discernible differences between intensities. Color information is not enough to segment images, followed by clustering methods based on similarity measurements (such as Euclidean distance) to group data, since an object can have different intensities inside its boundaries, or there could be repetitive colors within an image, without implying they belong to the same object. Another problem of the two clustering methods used to segment is that they require a prior knowledge of the number of regions or partitions within the image, and the dataset is varied enough to define certain k's that adjust well to all the images. Despite the fact that we evaluated the results of the methods with different k's, they were the same for all the testing set and hence the results of the evaluations were poor. These two models belong to the family of unsupervised learning techniques, in which no labels are needed to train the models, and the assignment of labels depends on the initial conditions of the problem.

All the scenarios tested had a serious limitation in precision, therefore improvements focused on increasing precision should be made. A possible cause might be texture segmentation, which leads to a big amount of false positive

contours detected. This problem might be tackled by taking into account regions with the same texture as a single object, through texton information.

To improve the segmentations, probably the most important aspect would be creating a segmentation hierarchy so as to not depend completely on the k used on the annotations, or even better to use this information to segment with the same hyperparameter. This might lead to an important improvement in results because precision would not be affected for segmentations made on objects of low relevance (false positives).

A filtering pre-processing step could be used to improve the results of the segmentations. However, it must be chosen carefully since depending on the type of the filter the results can be worst or better. For instance, using filters that filter homogeneous regions but preserve edges, such as median filter or bilateral, improved the maximum precision obtained compared with the other methods without filtering.

In addition, to improve the results of the segmentations, it is important to take into account local information by computing features not only for individual pixels, but also for the vicinity they lie into. By comparing adjacent windows with a specific shape and orientation, the Global Probability of Boundary method provides a probability of having a contour at a specific location, which is then used to compute global information of the image. Similarly, using features based on different kinds of information would be extremely useful. A clear example is that using texture information leads to avoiding over segmentation caused by textures, and comprises a useful cue because pixels with similar textures are usually part of the same object. Extracting additional features could complement the representation of individual pixels or local patches; however, including many features could overfit the models to train data and this could lead to errors when evaluating the results on the test set.

5. Conclusions

- Basic clustering algorithms such as k-means and GMM did not perform well on the whole BSDS dataset. This might be due to the fact that a specific k must be determined to use the algorithms and if the chosen hyperparameter is not equal to the ones existing in annotations results can not have a big recall.

- If hyperparameter number of clusters is increased, computational power increases significantly and recall might increase, but precision does not because there are more detected pixel contours but many of them do not correspond to true positives, affecting precision.

- Using gaussian filtering is not a good strategy to improve the results of clustering algorithms because evaluation in BSDS benchmark is done based on contours and filtering makes them blurry, reducing dramatically recall.

- Using filtering strategies that filter homogeneous regions but preserve edges, such as median filter or bilateral can improve results.

- Using features based on a broader amount of information (color, texture, spatial, shape, and others) could possibly improve results obtained.

- Evaluation for segmentation tasks remains one of the biggest problems to address. First of all, each human annotation is different and to evaluate either a unique annotation must be chosen or the overlap of them must be calculated. Second, in many clustering methods the label assigned to each region changes on each iteration, hence automatic procedures must be used to match the regions that are evaluated through Jaccard index, and this provides an important error source if regions compared through Jaccard do not belong to the same object. This could be avoiding through checking manually which label belongs to each object but the task would be extremely time consuming and inefficient.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik. Contour Detection and Hierarchical Image Segmentation. 2011. IEEE TPAMI, Vol. 33, No. 5, pp. 898-916.

[2] MathWorks. Machine Learning Matlab. Applying Unsupervised Learning.

[3] Scherrer, B. Gaussian Mixture Model Classifiers. 2007. [online] Available at: http://www.medialab.bme.hu/medialabAdmin/uploads/VITMM225/GMMScherrer07.pdf

[4] Hierarchical Clustering. s.f. Mathworks [online] Available at: https://www.mathworks.com/help/stats/hierarchical-clustering.html

[5] Beucher, S. The Watershed transformation. 2010. [online] Available at: http://cmm.ensmp.fr/~beucher/wtshed.html