Comparison of Segmentation Methods in the BSDS500 Benchmark Dataset

Laura Bravo
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
lm.bravo10@uniandes.edu.co

Juan Pérez
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
jc.perez13@uniandes.edu.co

Abstract

Two simple segmentation algorithms based on color and space descriptors, with k-means and a Gaussian Mixture Model, as clustering methods, were evaluated over the test set of the Berkeley Segmentation Dataset (BSDS500). The algorithms were also compared with the gPb-owt-ucm algorithm. The results show that a GMM clustering run over a feature space comprising normalized information of the pixels' color in La^*b^* and spatial information provides better segmentations than k-means run over a normalized space consisting of color information in the RGB space. Nevertheless, these methods are still far behind gPb-owt-ucm, which showed impressive performance in this dataset in a very consistent way. Segmentations produced by these algorithms still present a large number of problems, such as dividing a clearly defined region into various parts, over-segmentation, and generating segmentations with what could be though of as spatial-outliers, which is not consistent with most natural images. Mean-shift with some unique features and a clever selection of its hyperparameter is proposed as a (probably) better performing method for future works.

1. Introduction

Segmentation is one the elementary problems of computer vision and consists of partitioning an image into sections. It can be considered as a problem of classification, since each pixel can be given the 'class' of the object it belongs to. The pixel will be considered to belong to a determined object or region of the image if it satisfies some characteristics of the object, that is, if it is somehow similar. Similarly, it can also be thought of as a sort of *closeness* in some space, therefore, the problem can be broken into two subproblems: find a representation of pixel's characteristics in some space and, define a method for classifying pixels close to each other as belonging to the same class.

The first problem, a good representation space for the object, can be tackled by thinking of attributes of its pixels, such as color and, maybe, position of the pixel. Color is a powerful feature to extract from a pixel for this purpose, since objects depicted in pictures usually have a characteristic range of colors different from other objects. Also, it would be a good assumption to say that colors 'near' to each other are part of the same object. Nevertheless, one can have different objects with the same color, and if color is the only variable being taken into account, misclassification may occur. A way to handle this is to add another variable to the pixel, its spatial position. Just like with colors, it is common for pixels belonging to the same object to be spatially close to each other; however, if spatial distance is the only variable being measured, misclassification will occur, too. Accordingly, it seems a rather good idea to either select color or color *and* position as the variables representing the pixel. The second problem is addressed with clustering algorithms which, in the case of the methods previously developed, are k-means and Gaussian Mixture Model (GMM).

Once a segmentation method is designed, some metric regarding its output must be developed, too. In the present work, the performance of the formerly developed methods [3] was computed according to the BSDS500 standards in the test set of this dataset. Also, the performance of the *gPb-owt-ucm* is computed and, finally, all the methods are compared visually through their *Precision-Recall curve* and numerically through the *maximum F-measure* and *Area below the PR-curve*.

2. Materials and methods

In order to compare two different segmentation methods: K-means and Gaussian Mixture Models (GMM), it was necessary to evaluate them using a common methodology. This includes both the evaluation algorithms, as well as the dataset on which the performance was to be measured. For this reason, the BSDS500 Benchmark and Dataset was used. The dataset is comprised of 500 images, 200 for test and 300 for training and validation. The regions in each of these were independently annotated by 5 of 30 human subjects, providing a variety in the ground truth available for each image.

The segmentation methods selected were both clustering algorithms applied in a specific feature space. For each method, the resulting cluster labels (pixel-wise) determined the grouping of regions. K-means, the first method, was set in the RGB color space and consists of given a number of clusters and a random initialization of the cluster centroids. With this information the algorithm iteratively searches for the best assignment of centroids and clusters, that is, it minimizes the distance amongst elements and between them and their respective centroids [1, 2]. This method requires the fine-tunning of the number of clusters, the parameter *k*. Additionally, the default distance used for the optimization, euclidean distance, can be modified to better fit the data and representation space. Ultimately, both of these parameters must be chosen according to the available data

The second segmentation method subject to the evaluation was GMM. It consists on fitting gaussian distributions to the different clusters of data and the complete model for the data is acquired by mixing those gaussians. Nonetheless, the individual parameters of the gaussians are unknown (mean, and covariance matrix), as well as the mixing coefficients, that is the relevance of each gaussian for attempting to explain the data, expressed as the fraction of the total points in each gaussian. As opposed to k-means, this algorithm allows for the soft assignment of any given point to the available distributions, which in turn provides a model more suited to accepting new data, whose position is different to the training data, by not forcefully fitting it into strict predefined sections of the feature space [2]. The feature space selected for this method was the La^*b^* color space jointly with the *x* and *y* coordinates per pixel. It is noteworthy to mention that the feature space selected for both of the methods was determined previously in [3]. These segmentation methods, and others such as Watershed and Hierarchical clustering, were evaluated in a small subset of images from the BSDS500 dataset. The experiments consisted on varying the feature space from several color spaces (RGB, La^*b^* and HSV), with and without the corresponding spatial coordinates per pixel. The best feature space per method, from this previous work, was taken and fixed in the present one, so as to only vary the parameter *k*.

2.1. Evaluation

The BSDS500 Benchmark introduces a separate evaluation methodology for both contours and regions for a given segmentation. This due to the fact that in a contour it may not be possible to identify errors in a segmentation that are cumbersome when evaluating the corresponding region [6]. In particular, the closed-loop restriction of the segmentations is not applicable when evaluating contours only. The three proposed region evaluations, also used in this work, are *Variation of information*, *Rand index* and *Segmentation Covering*, which will be briefly described below.

Variation of Information This metric compares the distance between the test and ground-truth segmentations by comparing the individual entropies (H) of the aforementioned groups and their similarity (I, know as their joint entropy). See equation (1)

$$VI(S, S') = H(S) + H(S') - 2I(S, S') \qquad (1)$$

By minimizing the individual entropy one can find evermore homogeneous regions [4]. Conversely, because the mutual information between two groups measures the amount of info that one group contains about another, maximizing this is equivalent to finding the best overlapping regions. Nevertheless, due to this being a pixel-wise process, there is still no restriction for producing closed regions [5].

Rand Index It compares the ground-truth and test segmentations by counting the number of pixel pairs in these that have the same label divided by the total number of pixel pairs. For dealing with the multiple ground-truths the final metric is the average of the Rand Index per ground-truth segmentation [6].

Segmentation Covering This last metric is defined as the normalized overlap between two regions, here the segmentation and the annotations. It deals with the multiple human segmentations as by taking the average of these [6].

Ultimately, these metrics are combined to identify the precision and recall of the segmentation method per confidence measure. Particularly, the final curve is the result of varying the number of clusters (*k*) per image and thus having the option of obtaining among a group of segmentations the one that best corresponds to the ground-truth. Said curve is the Precision Recall (PR) curve. It depicts the evaluation of a detection method and its response under the variation of a confidence parameter. In this case the evaluation is done pixel-wise, so the segmentation problem can be thought of as a detection problem and the PR curve can be used. The ideal result of the PR curve depends on the task at hand. In this case, there is a point that represents the outcome of a human group performing this task, to achieve a result as good as a human could be sufficient. The intersection of the curve and the coordinate axes is paramount in understanding the limitations of any given method. For instance, if the recall is significantly lower than the perfect score (1), then, the method could be improved because presently, according to the curve, it would be blatantly disregarding multiple objectives. Additionally, the two most common measures extracted from this curve are the *maximum F-measure*, that graphically is the harmonic average of the coordinates of the highest point in the PR curve (closer to $(1, 1)$). The second measure is the area under the curve or Average Precision (AP).

Method	Max. F-measure	Area
gPb-owt-ucm	0.73	0.72
K-means	0.61	0.04
GMM	0.71	0.13

Table 1. Max. F-measure and area of the curves presented in Fig. 1.

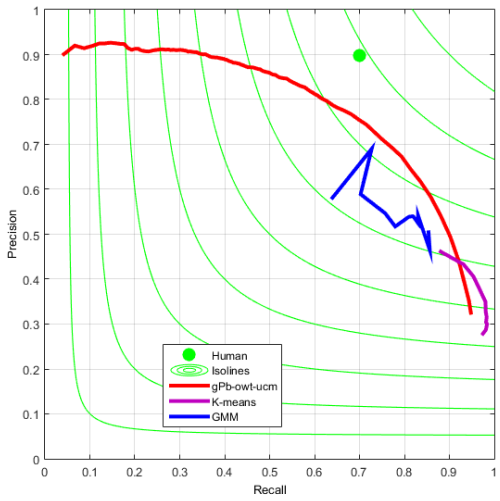

Figure 1. Precision-Recall curve showing the performance of the two methods developed here plus that of (gPb-owt-ucm) over the test set of the BSDS500.

3. Results

Results for the BSDS benchmark for the three methods are shown in Fig. 1. Numerical data regarding these curves is shown in Table 1. These results show that, from the methods that were developed here, the one with the best performance in this dataset is GMM with spatial and color information (La^*b^*+xy).

Naturally, the k parameter that is chosen to run the algorithm over the whole dataset thus having enormous impact on the actual performance of the method, since it will determine if each image is either well segmented or over/under-segmented. A rather big issue with this is that k depends on the contents on the image, that is, the type of scene it depicts, the perspective, the scale, and how much of the actual scene it is capturing. In this case, this parameter is just fixed (within a run of the algorithm) and is not a function of any information from the image, therefore, in most cases the segmentation will be far from perfect and, probably, will never achieve high performance.

Within a known dataset, the best choice for k would probably be the mode of the number of objects in the images, but that is information which, in theory, is inaccessible. Consequently, if the performance is to be enhanced, a method for at least estimating k should be developed, which is pretty much a whole different problem to be addressed. A simple approach for this would be to estimate the parameter for an extended-minima transform that results in the best match between the number of regions generated by a follow-up watershed transform and the actual number of objects in the image; this should be run over the training set for the estimation and then use the same parameter for testing the algorithm.

In this method, only the pixel's information is being taken into account, that is, the algorithm is not very sensible to either local nor global information. One of the methods, however, takes into consideration spatial information when the xy coordinates are used, nonetheless, it is actually affected by the position of the pixel among other pixels but not by the features of the surrounding pixels. Every pixel is on its own, as opposed to using its information relative to that of other pixels to produce a more descriptive feature space. This in turn leads to the production of heterogeneous regions or regions riddled with holes by the segmentation.

On the other hand, *gPb-owt-ucm* considers local and global information regarding color and texture and 'makes decisions' about the boundaries of the different objects, which leads to segmentations that have more meaning in the image; additionally, it creates a hierarchy of segmentations instead of just one, which gives some sense of the importance of a given boundary in the image. All of this produces a very robust output with segmentations that have distinctive features that make sense for natural images, like the fact that (i) spatial proximity is not a feature but rather a constraint for the definition of a region as an object, (ii) regions, in the feature space, are not assumed to be distributed according to a sphere (k-means) or a normal distribution (GMM), and (iii) more importantly, the concept of the regions being defined in a hierarchical manner, which is closer to what a human actually does when asked to segment an image.

When comparing the methods developed here with *gPb-owt-ucm*, it is rather evident that this algorithm is a way better procedure that makes use of a lot of information that the methods developed here just ignore. It uses the information in a very smart way by computing appropriate data and applying transforms with an approach that aims at extraction of regions that have a higher probability of meaning something to a human, instead of just trying to cluster pixels that look alike and assuming that they must represent regions that correspond to objects.

In the previous work [3], the fairly simple evaluation metric that was devised stated that the best algorithm among the two shown here was GMM (with information coming from a *lab+(xy)* representation). The current results here show again the same trend, by locating GMM's PR-curve a bit above that of k-means; consistently, GMM's area under

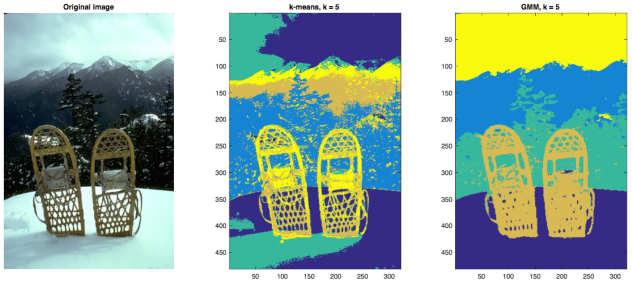

Figure 2. First example of segmentation of both algorithms over an image of the test set. The number of clusters was set to 5 for both methods.

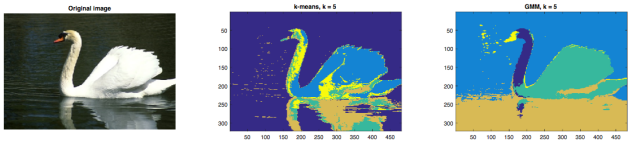

Figure 3. Second example of segmentation of both algorithms over an image of the test set. The number of clusters was set to 5 for both methods.

the curve and maximum F-measure are also higher than K-means'. Nevertheless, it should be stated clearly that it is hard to compare their performance, since there is no section in which they present the same precision or recall with the k's used to run the algorithms. This means that GMM always has better precision and that k-means always has better recall, but there is no overlap between them, so it would not be fair to fully compare them.

As to the superiority of GMM, the current hypothesis is that the clusters that GMM forms (normally distributed) are a better fit for the data than those formed by k-means (spherical). This allows the algorithm to be more flexible and, therefore, provide improved segmentations that do not require very 'clusterized' points in the feature space. On the other side, this means that, probably, the apparently larger recall inherent of k-means is because it is more sensible to *not* detecting outliers in the sense of some pixels clearly within a region that happen to have a different label than that of the region. Consequently, labels for a given region are more sparsely distributed across the whole image and have a higher probability of getting a 'hit' of the places that do correspond to a given object (see Figs. 2 and 3), rising the recall of the algorithm.

Even though GMM seems to be quite good (according to the maximum F-measure), by examining the segmentations it generates, it is evident that it has some very hard limitations. For instance, it is trying to segment according to proximity in the feature space and, since it uses spatial information, that is translated to actually segmenting by taking into account spatial proximity; however, for a human observer it is clear that spatial position is not really a feature that defines a region, but rather connectivity among pixels, so that spatial information/connectivity is more a constraint than a feature for segmentation problems. Apart from this, the algorithm does not take into account texture information, which is what can introduce huge amounts of noise in the segmentation, since every element of a texture in an object will probably be understood as an object itself instead of a property of the object. Finally, the ubiquitous problem of the selection of the k parameter represents a large problem with this method, since without a proper way of choosing it, it will always cause over- and under- segmentation. This generates a clear pattern in the segmentations, by breaking well-defined regions into smaller regions.

If the problem of segmentation in images still wants to be addressed with a clustering approach, a method more appropriate for this would be mean-shift, since, as mentioned earlier, it uses spatial information not so much as a feature of every pixel but as a constraint for segmenting regions, given that it generates the clusters by moving in a propagating manner. This algorithm will change the problem of choosing k by the problem of choosing the radius of the n-dimensional sphere that will be moving around the feature space. If the non-spatial features are normalized through z-score and the image is resized in a clever way by extracting some information about some object in the image (which could be hard as a matter of fact, but probably easier than actually estimating the number of objects in the image), the radius could be given an upper boundary and some optimization algorithm could be used as to iterate through different radii and look for some convergence on the number of clusters found. For tackling the problem of the radius of the sphere, one of the methods here developed could be run with different k values and some object whose existence is agreed for various ks could be used as a way of determining in what scale the objects of the image are. This approach will probably give the background of the image as the 'consenting' object across different runs of these methods, but still, by assuming this, some estimate about the scale could be given.

4. Conclusions

The BSDS500 Benchmark evaluation with its region based segmentation metrics, as well as the images and annotations it provides, are the basis for being able identify the superior segmentation method. This due to the fact that the comparison of the methods in the present work, GMM and K-means in a La^*b^*+xy and RGB feature space respectively, could be done under the same conditions. Consequently, the results obtained point towards GMM being superior to K-means in this context with a max. F-measure of 0.71 as opposed to the 0.61 of the K-means segmentation. This was corroborated by the area under the curve for each method, where GMM (0.13) was higher than K-means (0.04). The shortcomings of both of these methods

can be attributed to the pixel-wise segmentation that disregards the relationship amongst pixels (their affinity/ dissimilarity). The result is a multitude of heterogeneous regions and segmentations with discontinuities. This is one of the main contributions of the *gPb-owt-ucm* method, it takes into account the effect of global and local information on any given pixel. This is the next step towards further improving the methods here presented, but also, there could be an inclusion of other important types of information of the objects in an image such as the texture.

References

[1] E. Alpaydin, Introduction to Machine Learning, 2nd ed. London, England: The MIT Press, 2010.

[2] P. Arbeláez, Lecture 5: Clustering, Computer Vision, Universidad de los Andes, Colombia, 2017.

[3] J. Pérez, L. Bravo, Segmentation of Images. Computer Vision: Superpixels Laboratory. Universidad de Los Andes, 2017.

[4] T. Maintz, Digital and Medical Image Processing. Chapter 10: Segmentation. Universiteit Utrecht, Department of Information and Computing Sciences.

[5] D. Russakoff *et al*., Image Similarity Using Mutual Information of Regions. Lecture Notes in Computer Science. Computer Vision - ECCV 2004, p. 596-607

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