Advanced Density Peak and K-means Clustering on Image Segmentation

Yumiao Hui
School of Data and Computer Science, Sun Yat-sen University, Guangzhou, P. R. China.
Email: 18940211160@163.com

***Abstract*—K-means is a classical and widely-used data clustering algorithm. Despite its effectiveness, the drawbacks are obvious that it needs to know k value previously and not suitable for complex situations. Density Peak clustering can practice on irregular data sets with a higher accuracy and better performance than K-means and doesn't need to get prior knowledge. However, few concentrated on their performances on image segmentation. In this paper, we propose novel image segmentation approaches based on K-means and Density Peak clustering which greatly reduce running time. Compared with current methods, our methods have improved aspects as following: 1) The methods could have much shorter run time performance than other current normal methods. 2) Unlike other current image segmentation methods, our method could save the original colors of the pictures and provide a rather real image segments. Experiments on test data will testify the validity of the methods and a detailed description based on empirical results will be provided as conclusions.**

***Index Terms*—Density Peak clustering, Image segmentation, K-means clustering**

I. Introduction

Image segmentation [1] is a progress which means to divide the image into several parts based on their similar features like colors. Clustering is an approach of classifying data into categories based on their features too. Thus these two concepts can be closely associated together. The data belonging to the same group means their similarities between in themselves are higher than others. When applying clustering algorithms on images, people need to give the cluster number so that can access the features of each segments in the images. K-means clustering is considered as one of the most classical unsupervised machine learning algorithms, which has been widely used for a long time since it was first invented in 1967 [2]. In the last decade, with the advent of strong computer power and larger data sets, the power and usefulness of this algorithm became more evident. Density Peak clustering was proposed in 2014 [3], which could determine the cluster numbers and cluster centers based on the decision graph. Experiment results based on a variety of data sets is presented in the following parts. It shows that our proposed K-means and Density Peak clustering can realize its function on pure data sets to give reasonable results and especially the last one is better adapt to non-convex data sets. Then we draw our attention back to image segmentation field. We modify the original algorithms to let them deal with pixels, however the processing time is rather long. Thus, we furthermore to improve the original algorithms by using the concept of dictionaries, which uses a hash-key [4] to access data for a particular key (or index). In addition, rather than working with individual pixels, we have divided the RGB cube (with length 256) [5] into several sub-cubes of equal length (e.g. cubes with length of the geometric numbers of 2. For each mini-cube within the overall RGB cube, we have then counted the number of pixels that sit within the mini-cube, and assigned this number to the pixel in the center of the mini-cube. Completing this process for an image with 100,000+ pixels allowed us to compress the number of distinct data points (pixels) below 1,000 in most cases. An illustrative example will be given in Figure 2. It demonstrates the features of advanced K-means clustering and Density Peak clustering on the images. We describe the characteristics of each clustering algorithms and compare their advantages and drawbacks as conclusions. The rest of this paper is organized as follows. We will present our new image segmentation algorithms in details in Section II. Section III will present the experiments. Section IV will analyze the experiment results and evaluate them. Finally, our conclusions will be presented in Section V.

II. The Proposed Algorithms

In this section, we describe the preparation work and selected parameters for image segmentation. We demonstrate the image pre-possessing progress. Then we propose our novel clustering algorithms on image segmentation on the base of previous algorithms.

A. The Pre-Process of Images

In the first step, we started to slice the RGB Pixel space (256x256x256) into several mini-cubes with equal length. To achieve equally sized mini cubes, we had to choose geometric numbers of 2 (e.g. 1, 2, 4, 8, up to 256). In the next step we have counted the number of image pixels in each of the mini-cubes and assigned their numbers to an artificial point at the center of the cube: this pre-processing step is very fast (of the order O(n))which allow us to drastically reduce the number of pixels that had to be processed for the K-means and Density Peak clustering algorithms. And finally we choose the mini-cube size for reasonable image quality in the pre-processed image as result. With granularity = 1, there is no difference between the original and the preprocessed image. With granularity = 256, only one large cube remains in the RGB space. Its centre is (127,127,127) and this is exactly the colour that is shown for the pre-processed image(grey). It is remarkable that cube size 128 produces a total of 8 clusters, which in itself could be

considered as a reasonable segmentation of the initial image with a fixed number of 8 segments. We achieve reasonable image quality in the pre-processed image with a mini-cube size (granularity) of 16. The distance metric is relevant for the K-means and Density Peak clustering algorithms. We select Euclidean as default distance metric parameter.

B. The Improved K-means clustering on Image Segmentation

The improved K-means clustering is verisimilar with the original algorithm, only with some slight modifications, which is caused by the different structure of the input data. Each input record (a center of a mini-cube) has been assigned a weight, or number of pixels it represents. Based on this weight the calculation for the distance between such a point and the centroid is multiplied by this weight. We input parameter k, which is the number of clusters and we need to transfer pixels to lists then we add the values of two lists for each dimension and calculate the mean value of the lists across all dimensions. We pick k random points to start with the process and assign points to groups accordingly The rule is assigning groups accordingly to the center they are closest to, after adding new data, we should recalculate the k centroids, once we don't get any changes means the clustering progress is done. With the generic K-means algorithm, based on partitioned mini-cubes, we can now derive original K-means for the clustering of pixels in an image.

C. The Improved Density Peak clustering on Image Segmentation

The average distance depends on the distance metric: We calculate the density for a given point in the list of points for points inside the same cube using the average distance between two random points in a cube. All pixels in a mini-cube are now compressed into the center of its cube. This means that the algorithm will calculate a distance of 0 for all data points in the same mini-cube. However, this is not correct and we use the estimated average distance between two points in a cube. We have decided to run a simple simulation to estimate the mean differences for Euclidean distances. n is the total amount of points in the mini-cube. The value e are then used in the density calculation for all pixels in the Density Peak clustering algorithm.i, j are dependent parameters iterating from 1 to n.

$$sq = \sum_{i,j=1}^{i,j=n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (1)$$

$$e = \frac{sq}{n} \quad (2)$$

For the implementation of the Density Peak clustering algorithm, we use the following average distance measure for the points in a mini-cube, the length of mini-cube is l:

$$d = e * l \quad (3)$$

We use the Gaussian kernel function to sum up calculated densities.Now calculate the density for a given point i in the list of points, ρ_i means the density of point i, d_{ij} represents the distance between the points i and j.

$$\rho_i = \sum_j \exp^{-\frac{d_{ij}^2}{(dc)^2}} \quad (4)$$

Now we use this function to calculate the density of each point in the list. First we should estimate the dc parameter. Some papers [1] advised to set it as 0.5%, however we should rethink this idea here. Based on observations, we find that for a cube length of 16, the value of 3% of the maximum distance between all pixels produces good results. s is a configurable parameter with the setting s = 400.0. n represents number of pixels in the image. $\max_{i,j}(x_i, x_j)$ means the maximum distance between any two pixels in the image.

$$dc = \frac{\max_{i,j}(x_i, x_j)}{s} + log(n) \quad (5)$$

Now we consider density threshold for outliers detection. We achieved good results with a density threshold of 5% of density range: only pixels above this density threshold are considered as candidates for the cluster centroids. The final densities for each of the points have been scaled in the range from 0 to 100. Next we calculate a distance for each pixel, based on the closest point with higher density than the point itself. For each point we calculate the minimum distance of the point to another points with higher or equal density:

$$\begin{cases} \delta_i = \min_j(d_{ij}), & \rho_i < \rho_j. \\ \delta_i = \max_j(d_{ij}), & \rho_i \quad is \quad max. \end{cases} \quad (6)$$

We model the distribution of the distance (y-axis) via the exponential distribution, test several settings and achieve good results with the following equation, dst represents the distance threshold, n represents number of all pixels in the image.

$$dst = -log(po) * d + log(n) \quad (7)$$

po is a configurable hyper-parameter and works well with the setting 0.20. The term $log(n)$ has been added as a correction number for very large images: in those circumstances the chances increase that too many pixels are above the distance threshold, and this is tackled by pushing the distance threshold slightly up when the number of pixels in an image is very large. In the next step, the data needs outlier detection based on the decision graph, we convert the data to a decision graph, with density (ρ) as x- axis and distance (δ) as y-axis. Points in the area where coordinate are larger than both density threshold and distance threshold are classified as outliers and marked as the cluster centers. Assign the remaining points, by building up the dictionary assigned groups in recursive calls. Finally, we use those outliers and assign the remaining points to their appropriate groups. According to paper, a point x_i is assigned to the same group as x_j, if it meets the following two conditions:

$$\begin{cases} \rho_i < \rho_j \\ d_{ij} = \min_{i \neq l}(d_{il}) \end{cases} \quad (8)$$

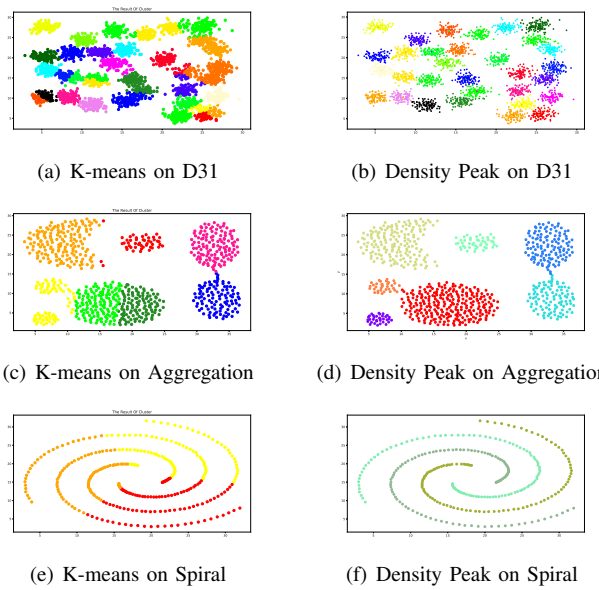
The recursive function assigning points has very high recursive depth in the beginning of the process, when only very few pixels have been assigned to groups. However, as the process gradually works through all the image pixels, less and less recursions will be required. It is remarkable that this function can assign all remaining points in order close to O(n) and causes no performance issues.

III. Experiments on Data Sets and Images

In this section, we verify K-means and Density Peak clustering method as a sanity check for the implements and both can get reasonable results on given pure data sets. Then, comparison experiments are conducted to show the effectiveness by comparing the cost time of original and proposed K-means clustering. Finally, we apply the novel clustering methods to a series of images and analyze their characteristics separately.

A. Experiments on given data sets

Based on the improved K-means and Density Peak clustering, we practice these algorithms on data sets named D31, Aggregation, Spiral provided from paper [3]. From the Figure 1 we can confirm that both the new clustering algorithms produce reasonable results, and we notice that Density Peak Clustering performs better at non-convex data sets than K-means. The results are believable and now we are in a position to test the same code on actual images.



(a) K-means on D31 (b) Density Peak on D31

(c) K-means on Aggregation (d) Density Peak on Aggregation

(e) K-means on Spiral (f) Density Peak on Spiral

Fig. 1: Clustering results generated by the original clustering methods.

B. Comparison of Cost Time on Images

Compared with normal and novel K-means clustering on image segmentation, we calculate their average running time either. We know that the number of clusters need to be given before K-means clustering progress, and the run time has a close connection with k parameter. The Density Peak clustering only presents the same determinative results on an image. So we only represent the results of K-means clustering in the Table I. Then we compare the average running time of K-means with Density Peak clustering algorithm with the same k value. Table I presents the comparison results generated by

TABLE I: Comparison on Cost Time with Different k.

k value	Original K-means	Advanced K-means
k=2	0.73	1.03
k=3	0.78	1.65
k=5	1.22	7.5
k=10	1.25	25.12
k=20	1.41	56.19
k=50	2.1	87.21
k=100	4.96	123.86

the proposed K-means and counterpart clustering methods. The first column lists the k parameter of each iteration. The second column lists the cost time of the improved K-means and the last columns are results generated by the traditional K-means. From the table, we can see that, our mentioned algorithm is much faster than the old one. The results testify our method. Because the modified parts of both K-means and Density Peak clustering algorithm are similar, we believe that Density Peak clustering also improve the performance of image segmentation. To prove our assumption, we did the experiments on the same image and we get the result of k=3, average cost time = 5.73s, which is better than 175s by normal Density Peak clustering. Our main steps in the clustering algorithm are as follows.

Step 1. First of all, we will choose the Euclidean metric as standard and choose the appropriate granularity and then pre-process the input image and transfer pixels to data stream in list forms.

Step 2. Next, we start to run all steps of the K-means and Density Peak clustering algorithm in the single function and measure the running time, the steps of the algorithm has been described in Section II.

Step 3. Finally, after many times of experiments, we come to the conclusion that our proposed algorithms do have more effective performances than previous ones.

Furthermore, it is necessary to analyze the results of the algorithms especially.

IV. Analysis and Evaluations on Clustering Results

It is an interesting observation that the K-means clustering algorithm runs generally faster than Density Peak clustering. In this section, analysis and comparison experiments are conducted on several real images in Figure 1. Evaluations based on these images will be given and we select some aspects deserved to present and discuss in this section. As Figure 2 shows, we can see that there are no obvious difference between the original and preprocessed images, whose pixels have been compressed. So it is efficient to pre-process the input image to get expected results. Then based on the preprocessed images, we further practise the K-means algorithm and Density Peak Clustering algorithm on the given images. From those groups of images,

(a) Original

(b) Preprocessed

(c) Density Peak Clustered

(d) K-means Clustered

(e) Original

(f) Preprocessed

(g) Density Peak Clustered

(h) K-means Clustered

we find that Density Peak clustering would like to present the contradiction of the colors in a macro view, while K-means would like to balance the whole image in micro prospective. We also find that the Density Peak clustering algorithm is very sensitive to the light. Its results are mainly affected by the light condition. It tends to expose the brightest or darkest segments of the picture. This can explain why the results of the pictures and especially for Density Peak clustering, it is interesting to find that the algorithm is sensitive to the outliers of segments in images,for examples like (g),(k),(o). We can find the logo and sign of the images, which is useful and time-saving for

(i) Original

(j) Preprocessed

(k) Density Peak Clustered

(l) K-means Clustered

(m) Original

(n) Preprocessed

(o) Density Peak Clustered

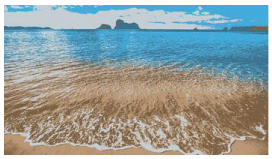
(p) K-means Clustered

Fig. 2: Original and Clustered Images

recognizing words while there are a plenty of images.

(a) Original

(b) K = 2

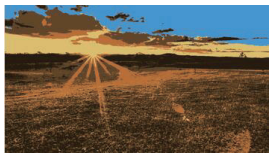


(c) K = 5

(d) K = 10

(e) Original

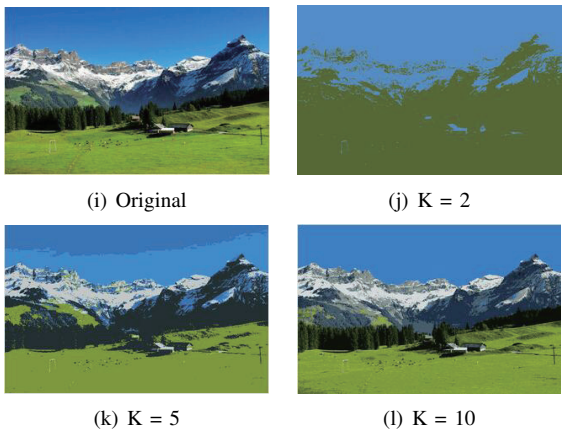
(f) K = 2



(g) K = 5

(h) K = 10

In Figure 3 we find K-means clustering is rather good at catching the details of the pictures, and unlike Density Peak clustering automatically detect the centroid of an image, K-means can realize this function by changing the k parameter manually. We find the image results are mainly affected by k value and different k values generate different results. As the increasing of k, the results are more detailed, which we can see it is more close to the original images. Besides we observe that when applying K-means method to the image, it is interesting to find that the gradual change in the image layer



(i) Original (j) K = 2

(k) K = 5 (l) K = 10

Fig. 3: K-means Clustered Images with Different K Value

is rather distinctive and our method could easily catch out this feature and clearly present this phenomenon as halo like (h),(k). In general, Density Peak clustering tends to produce starker contrasts, while K-means clustering produces more balanced colour schemes which are more accepted by human eyes. In fact there is no winner between these two algorithms and the performances are based on the situations and conditions that the input images provide and decide.

V. Conclusions and Future Works

In this paper, we have proposed improved clustering algorithms based on image segmentation. For general, image segmentation can be useful to test the performance of generic clustering processes. It might be difficult to get hold of large volumes of real data(e.g. twitters or other Internet personal data). It is easy to use photographs with really huge amount of pixels and test prototypes of clustering algorithms against the pixels in these images. The proposed algorithms both have their unique advantages.

A. Density Peak Clustering Pros

Density Peak clustering is better at identifying some unusual outliers in colour patterns (such as the logo in the city night views) just like the above results show. K is not an essential input parameter in Density Peak clustering. This is helpful in situations where a program needs to process hundreds of images in bulk and automatically set an optimised cluster number k for each picture as part of the process. It is suitable for the cases which cannot afford accurate k values.

B. K-means clustering Pros

Overall, K-means produced more balanced colour schemes that were somehow easier for the human eyes to interpret. Running time of K-means is generally better than of Density Peak clustering algorithm. Running time of K-means were generally better than of Density Peak Clustering algorithm; this could be an argument to work with K-means when runtime performance is critical. K-means can run and complete clustering of images on the raw pixels of an image. In contrast, Density Peak clustering could not operate and complete the process on the raw pixel data in reasonable time, and pre-processing of images is a necessity in Density Peak segmentation. Besides the results of K-means are more close to the balanced color themes for human eyes.

C. Future Works

Further work is required to refine the calculation of the dc parameter in the Density Peak clustering algorithm. In particular, it would be useful to develop a formula that provides meaningful clustering for a range of different input. In my view, the application of the Euclidean metric was assumed as must in most of the cases. However, considering the computationally less complex calculations of the Manhattan and Supremum metrics, they could be viable alternatives; their use in image clustering should be further investigated, as they also produced excellent clustering results. In summary, this would include the development of more generic algorithms to determine outliers in a decision graph and with more alternative metrics.

References

[1] C. Zhensong *et al.*, "Image segmentation via improving clustering algorithms with density and distance," *Procedia Computer Science*, vol. 55, pp. 1015–1022, 2015.

[2] J. MacQueen *et al.*, *Some methods for classification and analysis of multivariate observations*. University of California Press, 1967.

[3] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[4] V. Moen, H. Raddum, and K. J. Hole, "Weaknesses in the temporal key hash of wpa," *Acm Sigmobile Mobile Computing Communications Review*, vol. 8, pp. 76–83, 2004.

[5] S. M. Aqilburney and H. Tariq, "K-means cluster analysis for image segmentation," *International Journal of Computer Applications*, vol. 96, pp. 1–8, 2014.