# Application of K-means Algorithm in Image Compression

To cite this article: Xing Wan 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **563** 052042

# Application of K-means Algorithm in Image Compression

**Xing Wan**

Leshan Vocational and Technical College, Leshan, Sichuan, 614000, China

krantson@163.com

**Abstract.** In machine learning problems, there are two main algorithms: supervised learning and unsupervised learning. Supervised learning algorithms can be used to classify data for tagged data; non-supervised learning algorithms can be used to cluster data for unlabeled data. This paper discusses the basic principles of clustering algorithm and selection of key parameters of clustering algorithm. The application of clustering algorithm in image compression is also analyzed. This paper also emphasizes the problems that should be paid attention to when using clustering. Finally, a practical case of image compression with K-means is given.

## 1. Introduction

Clustering is a type of unsupervised learning that can be used to probe data structures. Clustering is the process of dividing data into multiple clusters, each of which consists of one or more similar data. The clustering algorithm requires the greatest similarity between the data of the same cluster, while the data of different clusters has the smallest similarity between the clusters. Unlike the classification learning, the clustering algorithm is an unsupervised learning method. The clustering algorithm does not need to label the categories of the samples, but divides the data set into several clusters according to the similarity of the samples. Therefore, the clusters of data are not predefined, but are defined according to the similarity of the characteristics of the samples. Therefore, the input cluster data does not need to be pre-marked.

## 2. K-means algorithm

The K-means[2][3] algorithm solves the clustering problem of data in multidimensional space. Supposing there is a data set $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}\}$, it is an n-dimensional data random variable $\boldsymbol{x}$ with $N$ samples. Since the clustering algorithm is unsupervised learning, there is no target attribute y. Assuming that the K value is given, the goal of the K-means clustering algorithm is to divide the data into K clusters. Since the distance between data in the same cluster should be smaller than the distance with the data outside the cluster, $\mu_k, k = 1,2, \dots, K$ is introduced as a vector, which can be considered the center of the Kth cluster, also known as the centroid.

The goal of the K-means algorithm is to find such a group $\{\boldsymbol{\mu}_k\}$, assign each data to the cluster with the centroid closest to itself, and find minimum accumulation distances of each data point with the cluster center $\mu_k$. For each data $\boldsymbol{x}^{(i)}$, introducing a set of corresponding binary variables $r_{ik} \in \{0, 1\}, i = 1,2, \dots, N; , k = 1,2, \dots, K$. if $\boldsymbol{x}^{(i)}$ belonging to cluster k, $r_{ik}$=1, and for any other cluster j ≠ k, $r_{ij}$=0. Cost function can be defined as following:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \ || \ \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k ||^2 \tag{1}$$

The $J$ of the above formula represents sum of the distances between each data $\boldsymbol{x}^{(i)}$ and cluster center $\boldsymbol{\mu}_k$. The goal of optimization is to find the best $\{\mathrm{r}_{\mathrm{ik}}\}$ with $\{\boldsymbol{\mu}_k\}$, so that $J$ reaches a minimum. First step to do should be randomly selected the initial value of $\boldsymbol{\mu}_k$. In the first phase, keeping $\boldsymbol{\mu}_k$Fixed, $\mathrm{r}_{\mathrm{ik}}$ is adjusted to optimize $J$. In the second phase, keeping $\mathrm{r}_{\mathrm{ik}}$ fixed, $\boldsymbol{\mu}_k$is adjusted to optimize $J$. Repeating these two phases to optimize until convergence. Due to adjustment $\mathrm{r}_{\mathrm{ik}}$with $\boldsymbol{\mu}_k$, these two stages correspond to the E-step and the M-step are called EM algorithm, so the E-step and the M-step can also be used.

First considering M-step, the function $J$ is a quadratic function of $\boldsymbol{\mu}_k$when fixing $\mathrm{r}_{\mathrm{ik}}$, let the derivative of $J$ about $\boldsymbol{\mu}_k$is equal to 0, and the minimum value can be obtained:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = \sum_{i=1}^{N} \mathrm{r}_{\mathrm{ik}}(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k) = 0 \tag{2}$$

solving out $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} r_{ik}x^{(i)}}{\sum_{i=1}^{N} r_{ik}} \tag{3}$$

The denominator of the above formula is the number of sample belonging to cluster $K$, and the numerator is the sum of the data, so $\boldsymbol{\mu}_k$ is the average of all data points belonging to cluster $K$.

Considering M-step, $\mathrm{r}_{\mathrm{ik}}$ is independent of each other when $i$ is different and $\boldsymbol{\mu}_k$ is fixed, so each data $\boldsymbol{x}^{(i)}$ can be optimized separately. As long as the norm is the smallest, and $\mathrm{r}_{\mathrm{ik}}$ can be set to 1, such as follow:

$$\mathrm{r}_{\mathrm{ik}} = \begin{cases} 1, k = argmin_j||\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j|| \\ 0, \qquad\qquad other \end{cases} \tag{4}$$

The above E-step and M-step are iteratively performed until the centroid is no longer changed or the number of iterations exceeds the predetermined maximum number of iterations. Since the value of the cost function $J$ is reduced at each stage, the convergence of the algorithm is guaranteed. However, the algorithm may converge to some local optimal value of $J$. Changing the initial value of $\boldsymbol{\mu}_k$ and running program multiple times is commonly used method to solve this problem. K-means algorithm is as follows:

- Get the initial data set.
- Determine the number $K$ of clusters and randomly generate the centroid of the cluster.
- Loop execution of iterative algorithms (E-step and M-step).
- E-step: get the index of the cluster for each data.
- M-step: calculate the centroid of $K$ clusters.

The K-means algorithm is used to classify the two types of randomly generated data, 2 centroids were randomly selected, 200 Gauss distribution data is generated randomly. The simulation results are as follows:

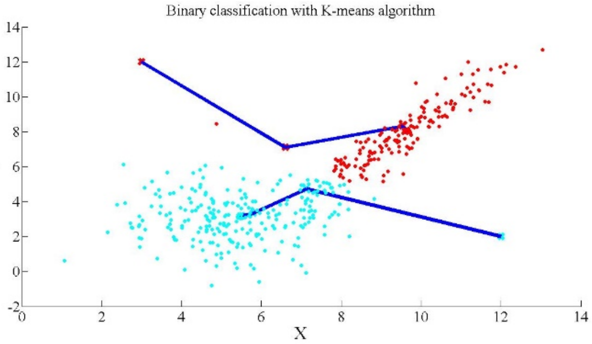

Figure 1. Binary classification with K-means algorithm

The K-means algorithm must randomly initialize all centroid positions before running, with the following considerations:

- The number $N$ of data must be less than the number $K$ of cluster centroids.
- Choose centroids randomly.
- It is possible to get a local optimal solution, depending on value the centroid is initialized
- It is usually necessary to run the K-means algorithm by multiple times while re-initializing the centroid position each time, and finally selecting the centroid with the lowest cost function according to the result.

Another difficulty in using the K-means algorithm is the choice of K. In theory, there is no standard selection formula. The $K$ value is manually selected by according to different problems. Taking different $K$ values, the value of the cost function $J$ is different. As the value of $K$ increases, the $J$ value decreases accordingly. When the number $K$ of clusters is equal to the number of data points, $J$ is reduced to zero. In practice, the $K$ value can be selected according to the curves of $K$ and $J$. The KJ curve is similar to the elbow of a person. When the $K$ is equal to 1, the value of $J$ is large. When $K$ drops to the inflection point, the $J$ rapidly drops and reaches the elbow position. Thereafter, as the $K$ value increases, the $J$ value decreases very slowly. This inflection point is the optimal choice of the number of clusters $K$. The figure below is a two-class classification KJ trace:

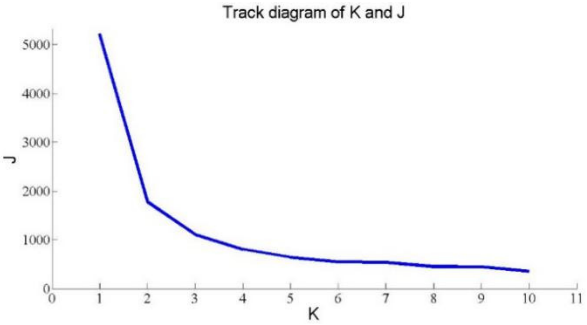

Figure 2. Track diagram of $K$ and $J$

## 3. Image Compression

The K-means algorithm can be used to compress the image. Unlike lossless compression, K-means uses lossy compression, so it is not possible to recover the original image from the compressed image. The larger the compression ratio, the larger the difference between the compressed image and the original image. The principle of K-means clustering algorithm for compressing images is as follow:

- Preferred number of selected clusters $K$ is very import, $K$ must be less than the number of image pixels $N$.
- Using each pixel of the image as a data point, clustering it with the K-means algorithm to obtain the centroid $\boldsymbol{\mu}_k$.
- Storing the centroid and the index of the centroid of each pixel, so it not need to keep all the original data.

It is assumed that the original image has $N$ pixels, each pixel adopts the RGB three-color mode. each value of the RGB mode needs 8 bits, and $24N$ bits are required to directly store the original image. If K-means cluster compression is performed and the number of elements is $K$ with cluster center vector $\boldsymbol{\mu}_k$, then the index of each pixel needs $\log_2 K$ bits to store. The K-means totally needs $\mathrm{N}\log_2 K$ bits. In addition, you need to store k centroids, which need 24k bits, so you need a total of $24\mathrm{K} + \mathrm{N}\log_2 K$ bits. The following example uses the K-means algorithm to compress the image. The size of the original image is 1536*2048, the number of selected clusters is K=6, and the number of iterations is 12.

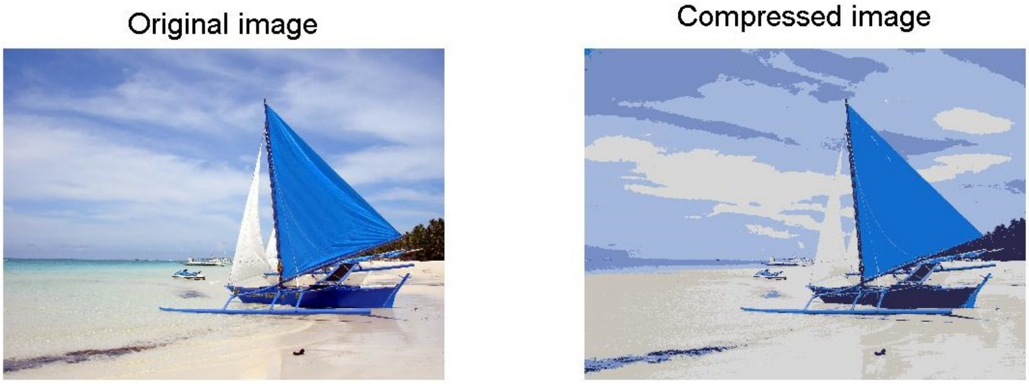

Figure 3. Compressed image with K-means algorithm

## 4. Conclusion

Based on the results and discussions presented above, the conclusions are obtained as below:

(1) K-means algorithm can be used to compress images.

(2) Compressing an image using the K-means algorithm is a loss compression

(3)When K is larger , the compression ratio become larger ; when K is lower, compression ratio become lower.

(4) When the image pixels are large, you should set a larger number of iterations for better compression.

**Acknowledgments**

This work was financially supported by the Applied Research Project of Education Department of Sichuan Province (18ZB0280).

## References

[1] Van der Geer, J., Hanraads, J.A.J., Lupton, R.A. (2010) The art of writing a scientific article. J. Sci. Commun., 163: 51–59.

[2] Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: a k-means clustering algorithm. Journal of the Royal Statistical Society, 28(1), 100-108.

[3] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: analysis and implementation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 24(7), 881-892.

[4] Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining & Knowledge Discovery, 2(3), 283-304.

[5] Wagstaff, Cardie, Claire, Rogers, Seth, & Stefan. (2001). Constrained k-means clustering with background knowledge. ICML-2001.

[6] Ding, C., & He, X. (2004). K-means clustering via principal component analysis. International Conference on Machine Learning.

[7] Lozano, J. A. (1999). An empirical comparison of four initialization methods for the K-Means algorithm. An empirical comparison of four initialization methods for the $K$-means algorithm.

[8] Modha, D. S., & Spangler, W. S. (2003). Feature weighting in k-means clustering. Machine Learning, 52(3), 217-237.

[9] Chang, D. X., Zhang, X. D., & Zheng, C. W. (2009). A genetic algorithm with gene rearrangement for k-means clustering. Pattern Recognition, 42(7), 1210-1222.

[10] Zhang, R., & Rudnicky, A. I. (2002). A large scale clustering scheme for kernel k-means. Computer Science Department, 4, 289-292 vol.4.