Introduction to K-means Clustering

Mehmet Emre Özyurt[1]
Seminar: Data Visualization
ge75tis@in.tum.de
Supervisor: Fatemeh Farokhmanesh
[1] Technische Universität München

Abstract

K-means clustering is one of the most commonly utilized methods for data clustering. It is an unsupervised partitional clustering method, which aims to split the given data into k groups assigning each data point to a corresponding cluster. This paper aims to provide an overview of k-means clustering and its relevance including its applications in both data visualization and computer graphics. This paper also discusses the strengths and weaknesses of k-means clustering by comparing it to hierarchical clustering. The simplicity of the conventional algorithm and the potential to modify it is demonstrated through methods like K-means++ and Elbow method. A general awareness of K-means clustering is provided by describing the rationale of every principle. This is needed to further improve oneself on the vast subject of Data Clustering. The paper concludes with two solid examples of application which reinforces the importance of K-means clustering.

1. Introduction

As the importance of data in the modern world increases and a new age of analytics appears, data sizes continue to become more massive each day. As a result of this size increase, labeling data manually is eventually unattainable. Thus automatic data labeling has become necessary. Data clustering is one of the most popular data labeling approaches and a big part of data mining. [AR13]

Data clustering splits a set of objects into different groups, such that objects in the same group are similar to each other. But by what means can someone find structure within a collection of data points and propose a good partition? The solution that K-means clustering offers is very intuitive and efficient. The main idea is to iteratively work out where the ideal means of clusters should be.

K-means algorithm has many applications in the field of data visualization and computer graphics. It is a simple practice of exploring the internal structure of the data. Through labeling and highlighting the similarities, k-means clustering transforms big data into a useful and meaningful substance. It is usually applied to data that has a small number of dimensions. It is also frequently used in data compression. [AK17]

The aim of this paper is to introduce the conventional algorithm and present different factors that might affect the performance and results. In this context, particular methods of applying the algorithm will also be discussed. After providing some suggestions to improve on the shortcomings of the algorithm, the paper will do a comparison between k-means clustering and hierarchical clustering. Finally, two concrete examples of its applications will be presented.

2. K-means clustering algorithm

K-means algorithm is a partitional clustering method that aims to split the given data into k entirely distinct groups. It is an unsupervised learning mechanism indicating that the technique can work on its own to discover patterns and does not need human supervision.

The algorithm starts by choosing k starting points. Initialization of these starting points can be random or systematic depending on the approach. These k starting points are the initial means of the data. The following step is to assign each point to the mean that is closest to them. After all points are assigned and initial clusters are generated, the new mean of each cluster is calculated. Mean in this context is the representative of each cluster. The assignment of each data point is repeated based on the new means. If at any point the clustering does not change, suggesting that optimal means are reached, then the process is done. [AR13] A basic description of the algorithm is as follows:

Algorithm 1: K-means clustering

```
Select K points as initial means;
while any clusters change do
    Construct K clusters by assigning each point to its
      closest mean;
    Recalculate the new mean of each cluster;
end
```

To determine which cluster a data point belongs to, K-means algorithm can use various proximity measures. The most commonly

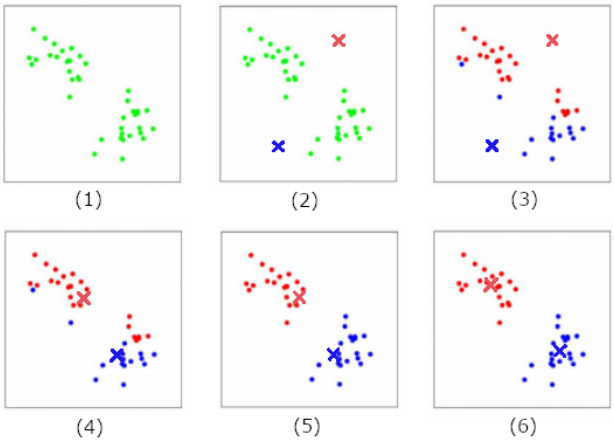

Figure 1: *A simple iteration of K-means clustering where x's represent means. (From Chris Piech, Stanford, 2013)*

used approach is the Euclidean distance. [AR13] When using Euclidean distance if the different dimension values are not acknowledged, it can lead to a situation where dimensions with greater values affect the clusters more. This could be avoided by scaling the dimensions to the same bounds. There are other proximity measuring methods however that take these issues into consideration such as Mahalanobis distance or Manhattan distance. Measuring preferences are only a small part of what determines the quality of clustering.

3. Performance and Quality of Clustering

How is the quality of a clustering assessed? What constitutes a good cluster? A high-quality cluster is where intra-cluster similarity is as high as and inter-cluster similarity is as low as possible. In other words, each cluster is as dense as possible while being away from other clusters.

The similarity in a cluster can be measured by calculating the sum of squared errors(SSE). The error is the distance of a point to its nearest cluster. Mathematically k-means clustering can be considered an optimization problem where the objective function is to minimize the SSE value which maximizes the similarity within each cluster. The formula for SSE is presented below where c_k is the mean of cluster C_k [AR13] :

$$SSE(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} (c_k - x_i)^2$$

$$c_k = \frac{\sum_{x_i \in C_k} x_i}{C_k}$$

But why is specifically the mean c_k used to update clusters? Let c_k be the mean of C_k and c_j the representative of C_j. The following equation shows that the mean c_k is the best representative for minimizing SSE. [TSKK19]:

$$\frac{\partial}{\partial c_j} SSE = \frac{\partial}{\partial c_j} \sum_{k=1}^{K} \sum_{x_i \in C_k} (c_k - x_i)^2$$

$$= \sum_{k=1}^{K} \sum_{x_i \in C_j} \frac{\partial}{\partial c_j} (c_j - x_i)^2$$

$$= \sum_{x_i \in C_j} 2 * (c_j - x_i) = 0$$

$$\Rightarrow |C_j| \cdot c_j = \sum_{x_i \in C_j} x_i \Rightarrow c_j = \frac{\sum_{x_i \in C_j} x_i}{|C_j|}$$

The performance and the quality of clustering depend on different factors. But two main aspects are:

- *Initialization of means*
- *Selection of number of clusters K*

3.1. Initialization methods

The results of k-means clustering depend heavily on the initialization of starting points. This section will review three primary techniques for initialization.

3.1.1. Random initialization

One of the easiest and most widely used methods is to choose starting means randomly. However, this is a highly volatile and inconsistent approach. Random selection by its very nature leads to different results for the same data because the starting means define in which direction the clusters will develop. It can also cause a situation where the means are not well distributed throughout the data. This flawed selection may potentially produce sub-optimal clusters. [CKV13] A simple example is shown in Figures 2 and 3.

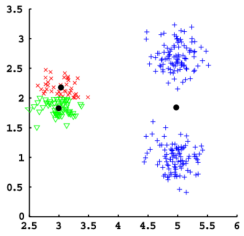

Figure 2: *Poor clustering where two initial means are adjacent to each other.*

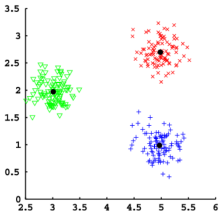

Figure 3: *Ideal clustering where initial means have balanced distribution. (Fig. 2 and 3 from Tzortzis and Likas, "The MinMax k-Means clustering algorithm", 2014)*

Inconsistencies in results could be solved through trial and error. By running the algorithm multiple times, different results can be observed and a decision can be met for the best result based on predefined criteria.

3.1.2. Nearest neighbor

This method which was also used in Hartigan and Wong's version of the algorithm selects the points that are disconnected from each other and that are in dense regions as initial means. [HW79] It utilizes the nearest neighbor algorithm to compare the densities of the potential initial points. The point with the highest density is the first mean. Following initial points are chosen by next highest density but with the condition that the distance to other initial means is equal to or greater than the average pair-wise distance of all points provided below as d_1 [AR13]:

$$d_1 = \frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1} N(x_i - x_j)$$

3.1.3. K-means++

As mentioned in 3.1.1, the more closed in the starting means are, the more cramped the clusters turn out to be. Therefore K-means++ tries to solve this underlying risk of unbalanced distribution by choosing initial points as far from each other as it can. This enhances the likelihood that starting means belong to different clusters thus improving the quality of clustering hugely.

The algorithm starts by choosing the first mean as random. The selection of other means is based on a weighted probability distribution. The probability of selecting a point x as a mean is proportional to the distance from x to the closest mean that is already selected. This increases the chance of selecting the point that is farthest from the closest mean. The selection goes on till K means are selected. [AV06]

Despite the huge improvement through minimizing randomization, running the algorithm multiple times with other starting points is recommended. Even though this initialization method requires more time at the start in comparison to random initialization, algorithm analysis shows that the total run-time of K-means++ is on average less and at worst equal to the random initialization. [AV06] Balanced distribution leads to optimal clusters forming quicker.

3.2. Selecting the number of clusters K

Estimating the optimum number of clusters k is a critical part of k-means algorithm. This selection obviously affects the quality of the clustering as well as the performance of the algorithm. For example, choosing k as 3 in a data that very clearly has 4 clusters is unreasonable as seen in Figure 4.

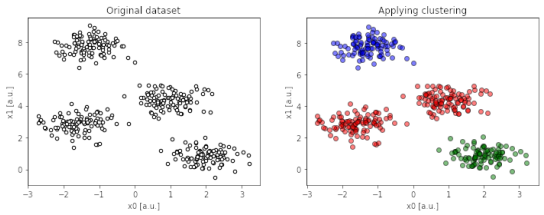

Figure 4: *An example of non-optimal k selection. (From Oleg Żero, "Mistakes with K-means clustering", 2019)*

A simple approach of trial and error would be to find the appropriate number of clusters by running the algorithm with different K values and identifying the best result based on different metrics such as total variance or silhouette value.

3.2.1. Elbow Method

When the total intra-cluster variance per value for K is plotted in a graph, generally a bend in the graph is observed. This elbow point after which the reduction in variance slows down rapidly illustrates the optimal number of clusters K. After this point increasing the number of clusters decreases the total variance very slightly, which makes it non-optimal to form an additional cluster. [KM13]. Figure 5 demonstrates the elbow point in a simple graph.

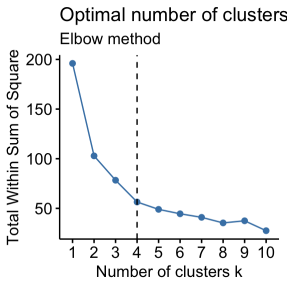

Figure 5: *A Graph showing the elbow point.*

The elbow method can also be used with inter-cluster variance where a higher value demonstrates higher unsimilarity between clusters. Then the elbow point is directed upwards.

However in practice, for data that is not distinctly clustered, this elbow point can not be precisely identified.

3.2.2. The Silhouette Method

The silhouette value determines how similar a point is to its own cluster compared to other clusters.

The value $a(i)$ is the average of the distances to all points in the same cluster. This value illustrates the intra-cluster similarity for each point. The value $b(i)$ is the average of the distances to all points in other clusters. This value is inversely proportionate to inter-cluster similarity

Using these two values, the silhouette value between –1 and +1 is found:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

A high silhouette value is desirable and indicates that the point is placed in the correct cluster, because the a-value is as small as possible meaning that it is very similar to the points in its cluster, and the b value is as large as possible meaning that it is very unsimilar to the points in other clusters. [Rou87]

We can look at the average of all the silhouette values for different K's. The average silhouette value will reach its global maximum when the optimal cluster is produced. [AR13]

$$S = \frac{\sum_{i=1}^{N} \frac{b_i - a_i}{max(a_i, b_i)}}{N}$$

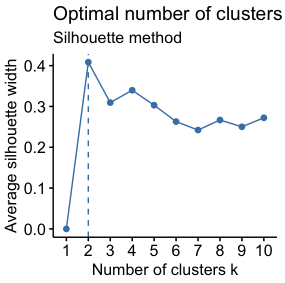

Figure 6: *A Graph showing the maximum average silhouette for different k values. (Fig. 5 and 6 from Kassambra, 2017, http://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/)*

3.3. Comparative breakdown

K-means algorithm is a very popular clustering method mostly because of its ease of implementation. However, there are many other clustering algorithms that differ in their strengths and weaknesses. This section will provide a brief comparative analysis of K-means algorithm and Hierarchical Clustering.

Even though K-means algorithm is fairly efficient, it still requires parameter tuning for the optimal number of clusters K, which is hard to predict. This becomes more challenging for highly dynamic data, which forces the algorithm to recalculate k after each adjustment. Besides k-means algorithm is non-deterministic in its essence as shown in Section 3.1. To improve on these shortcomings, Hierarchical Clustering offers a more deterministic solution that does not demand predefined parameters. [AR13] By merging small clusters or splitting big clusters, it presents a hierarchical relation between each possible sub-cluster which is called a dendrogram. And so the process of merging/splitting can be paused at any time if the result is satisfying. Any result is also reproducible. [Hal09] Figure 7 shows the 3-dimensional representation of the hierarchical tree for the cities of Europe.

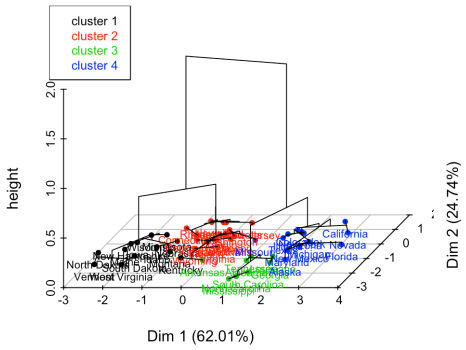

Figure 7: *Hierarchical clustering on the cities of Europe. (From Husson, 2010, Fig. 4)*

In terms of computational costs, K-means clustering is relatively fast. Its complexity is $\mathcal{O}(tkn)$ where t is the number of iterations, k is the number of clusters and n is the number of data points. As long as k and t are kept small, it behaves like a linear algorithm. [KM14] That is why one of the biggest advantages of K-means algorithm is that it can scale well to large data sets. In contrast, hierarchical clustering is more suitable for small data.

K-means algorithm assumes that clusters are spherical and equally sized. When this is the case it helps produce tighter clusters than hierarchical. But it can lead to bad results for data where clusters greatly differ in size and variance. [AR13] The implementation has to be generalized to adapt, for example by limiting the width of clusters depending on the data. Figure 8 illustrates the intuitive clustering and K-means without generalization.

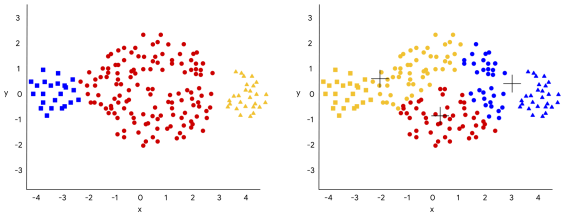

Figure 8: *Intuitive clustering(left) vs. K-means without generalization(right)*

Lastly, data points that are situated away from the main body can affect k-means clustering negatively. These outliers can pull means away from main clusters, or take a cluster on their own. They have to be eliminated before applying the algorithm which weakens the performance.

4. Applications of K-means clustering

In many fields collecting unlabeled data is often faster and cheaper than gathering labeled data. Therefore K-means clustering is a critical part of data analysis and machine learning to identify and label any database. It has a variety of applications ranging from Artificial Intelligence and Pattern Recognition to Data Compression. [AK17]

K-means clustering can be used on any number of dimensional data, although it is mostly used in low dimensional data. An introductory example is its use in personalization and ad targeting on the internet. By clustering people with similar interests together, it makes it easier to analyze the behavior of customers for different ads. Another regular utilization of k-means clustering is image compression, to which a detailed example is presented.

4.1. Image Compression by Clustering RGB-data

An RGB image consists of three channels that represent the colors red, green, and blue. Each data point/pixel in the image has a color value for each color. When compressing an image, the color values that each data point can take will be limited. Moving around in the color space for each color value, the K-means algorithm clusters the data into k dominant colors. This way the image is simplified and compressed. By defining k, the image is limited to k-color clusters. [Wan19] The same technique can be used for image segmentation, which helps locate objects and identify boundaries in images.

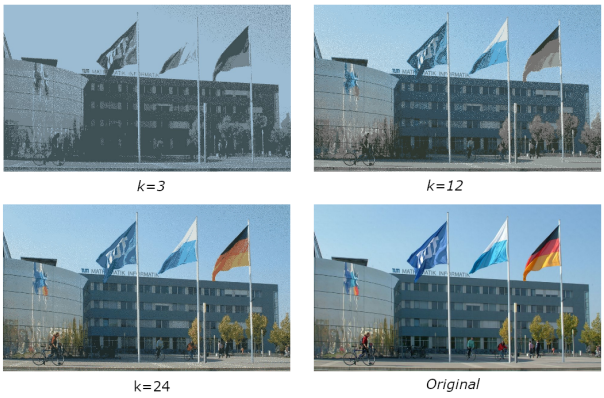

Figure 9: *Image compression through K-means clustering of pixels into k colors. Created on MATLAB*

4.2. Medical diagnosis and disease classification

Identifying patterns and classifying information in medical databases is really crucial to diagnose and treat illnesses. K-means clustering can analyze and extract information that experts might have a hard time detecting. It can also categorize patients into different treatment groups.

An example is its application on Alzheimer's Disease(AD) patients to explore the transition from mild to advanced stages. [AEHC*19] By using K-means clustering the characteristics of different stages and their progression is examined. Analyzing blood samples, MRI, and neurological data of cognitively normal(CN), mild cognitive impairment(MCI) and AD patients can reveal the pattern of transition between these stages. [EZI11] By looking at which group of CN people and MCI patients end up having AD, the likelihood of catching the disease for different genotypes and brain chemicals can be calculated. That way it can be detected before it gets worse, which leads to early treatment.

This approach of diagnosis can be applied not only in biological fields but also in human psychology and sociology to predict the changes in human behavior for groups and individuals.

5. Conclusion

This paper has thus far introduced the conventional k-means algorithm and described how the quality of clustering is assessed. It has shown that the best representative for a cluster is its arithmetical mean. Initialization methods and the selection of K as factors that affect the quality have been highlighted and primary approaches to those have been provided in detail. The strengths, as well as the shortcomings of the algorithm, have been examined with the help of comparative analysis with hierarchical clustering. Finally, its use in medical diagnosis and image compression has been demonstrated through concrete examples. Throughout the paper, the potential of modifying the K-means algorithm has been made quite evident. But no distinct variation of the K-means algorithm was described such as Fuzzy K-means or K-modes.

Although K-means algorithm has been around since the 1970s, it is still widely used in data classification and visualization. It is important to have a general understanding of its role in clustering to keep improving on the subject. This introductory explanation has added insight into different ways of interpreting data.

References

[AEHC*19] ALASHWAL H., EL HALABY M., CROUSE J. J., ABDALLA A., MOUSTAFA A. A.: The application of unsupervised clustering methods to alzheimer's disease. *Frontiers in computational neuroscience 13* (2019), 31. 5

[AK17] ALI H. H., KADHUM L. E.: K- means clustering algorithm applications in data mining and pattern recognition. 1, 5

[AR13] AGGARWAL C., REDDY C.: *DATA CLUSTERING Algorithms and Applications*. 08 2013. 1, 2, 3, 4

[AV06] ARTHUR D., VASSILVITSKII S.: *k-means++: The advantages of careful seeding*. Tech. rep., Stanford, 2006. 3

[CKV13] CELEBI M. E., KINGRAVI H. A., VELA P. A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert systems with applications 40*, 1 (2013), 200–210. 2

[EZI11] ESCUDERO J., ZAJICEK J. P., IFEACHOR E.: Early detection and characterization of alzheimer's disease in clinical scenarios using bioprofile concepts and k-means. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2011), IEEE, pp. 6470–6473. 5

[Hal09] HALKIDI M.: *Hierarchial Clustering*. Springer US, Boston, MA, 2009, pp. 1291–1294. URL: https://doi.org/10.1007/978-0-387-39940-9_604, doi:10.1007/978-0-387-39940-9_604. 4

[HW79] HARTIGAN J. A., WONG M. A.: Ak-means clustering algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 28*, 1 (1979), 100–108. 3

[KM13] KODINARIYA T. M., MAKWANA P. R.: Review on determining number of cluster in k-means clustering. *International Journal 1*, 6 (2013), 90–95. 3

[KM14] KAUSHIK M., MATHUR B.: Comparative study of k-means and hierarchical clustering techniques. *International journal of software and hardware research in engineering 2*, 6 (2014), 93–98. 4

[Rou87] ROUSSEEUW P. J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20* (1987), 53–65. 4

[TSKK19] TAN P., STEINBACH M., KARPATNE A., KUMAR V.: *Introduction to Data Mining*. What's New in Computer Science Series. Pearson, 2019. URL: https://books.google.de/books?id=_ZQ4MQEACAAJ. 2

[Wan19] WAN X.: Application of k-means algorithm in image compression. In *IOP Conference Series: Materials Science and Engineering* (2019), vol. 563, IOP Publishing, p. 052042. 5