Abstract

The world today is overflowing with information, and *big data* has become an important topic both in research and organisational terms. The amount of data in the world is growing exponentially by the minute, yet 99.5% of the collected data never gets used or analysed [Stack, 2018]. Predictive analytics are becoming more and more crucial for commercial success and it's estimated that companies that harness big data's full power could increase their operational margins by as much as 60%. [Court, 2015] A crucial part to understanding the underlying structures of the data lies in utilising the right tools. This paper looks into the k-means clustering algorithm, its theoretical foundation, current applications and improvements to the method. k-means has grown to become one of the most popular methods for unsupervised learning [Fränti and Sieranoja, 2019]. The algorithm is simple, easy to understand and, depending on the data, easy to implement. But, even though the algorithm has several use-cases, there are clear limitations to what one may expect from the algorithm. As illustrated by this paper, the algorithm yields varying results and often require modifications.

Table of Contents

List of Figures

1 Introduction

Because of the recent, rapid growth in computing power and data availability, machine learning has become a powerful tool for helping humans understand the world. Sadly, there is no one-size-fits-all solution that solves all our data problems because of the huge differences in user needs and the available data. Machine learning is categorised into *unsupervised*, *reinforcement* and *supervised* learning.

Supervised learning and reinforcement learning both require some sort of pre-existing knowledge about the data such as human-labeled data or a reward function for giving feedback about an agent's performance. However, pre-existing knowledge about the data can be expensive or difficult (sometimes even impossible) to obtain. Thus, the rise in popularity for unsupervised learning and the k-means algorithm - a technique free of labelling and cumulative awards. Along with clustering, unsupervised learning is divided into another method known as dimensionality reduction. Whereas dimensionality reduction aim to reduce the number of features in our data set, clustering aim to "reduce" the number of data points by grouping them into clusters based on their similarity. The strength of the k-means algorithm, being part of the clustering branch, lies in the ability to find previously unseen patterns and structures in our data, with minimum human supervision.

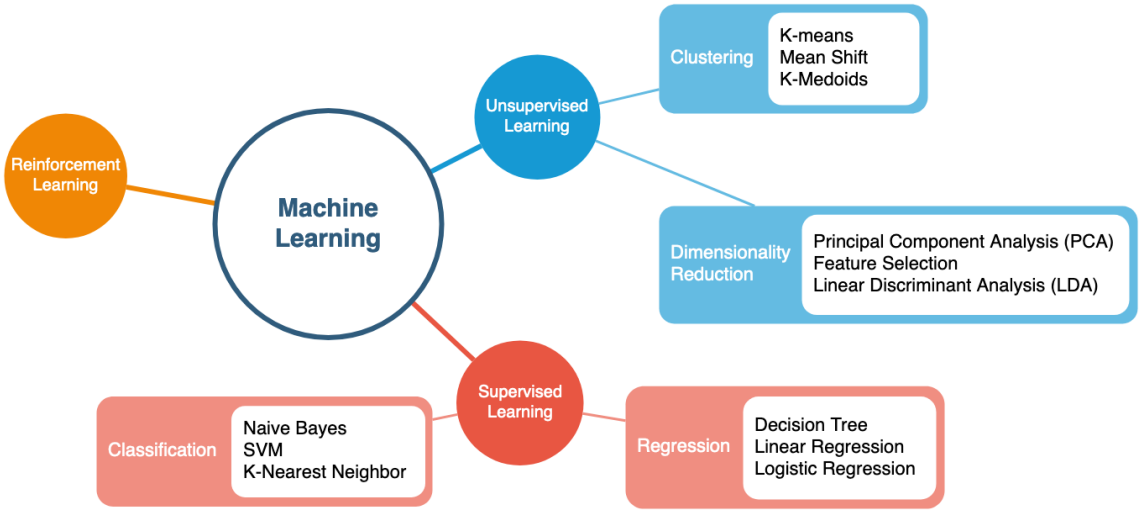


Figure 1: Roadmap to Machine Learning

The most common way of solving the clustering problem today is by using the k-means algorithm. Its popularity can be linked to its simplicity and ability to efficiently partition a data set into clusters. The algorithm was originally a method of vector quantization used for signal processing in electrical engineering, but has later proved to be applicable for both cluster analysis in data mining and machine learning (Michal Aharon [2006]).

2 Foundations

2.1 Core idea

The k-means clustering method is an iterative way of minimizing dissimilarity when partitioning n data points into k sets, where n is greater than k. The iterative process consists of creating clusters by reducing the distance between a cluster's centroid and an element. When the element is added to the cluster, the centroid needs to be recomputed to represent the updated cluster. Often is the average of a cluster's elements used as the centroid. This process is done iteratively until the composition of the clusters converges.

2.2 Theoretical foundation

The critical issue when clustering data points is the measure of dissimilarity. MacQueen defined this in 1967 as the sum of distances between all data points in the cluster and the centroid (Slonim et al. [2013]):

$$D(C) = \frac{1}{n} \sum_{c} \sum_{x \in c} d(v_x, v_c), \qquad v_c = \frac{1}{n(c)} \sum_{x \in c} v_x, \qquad v_c, v_x \in R^m$$

Here it is important to note that MacQueen looked at all the data points as vectors of the same dimension m (v_x) and that the centroid (v_c) is the average of all the elements in a cluster.

MacQueen's way of measuring dissimilarity when clustering objects forms the basis of the k-means methods. This is because he defined dissimilarity in relation to each clusters' centroid, and also how the centroids should be computed. All of the k-means methods uses this concept of centroids and only differ in their implementation of updating clusters and centroids.

2.3 Implementation

2.3.1 Lloyds' Algorithm

The classical implementation of the k-means method is Lloyds' algorithm, first presented in 1982 (Slonim et al. [2013]). Lloyds' version is a two step process where all the all the clusters and centroids are created intermittently. One way to initialize the method is by randomly choosing starting centroids (Figure 2 , b). Then each element is assigned to the closest centroid (Figure 2, c). The centroids are thereafter updated based upon the average of the elements in the cluster (Figure 2, d). These steps are repeated until the clusters' composition does not change (Figure 2, f).

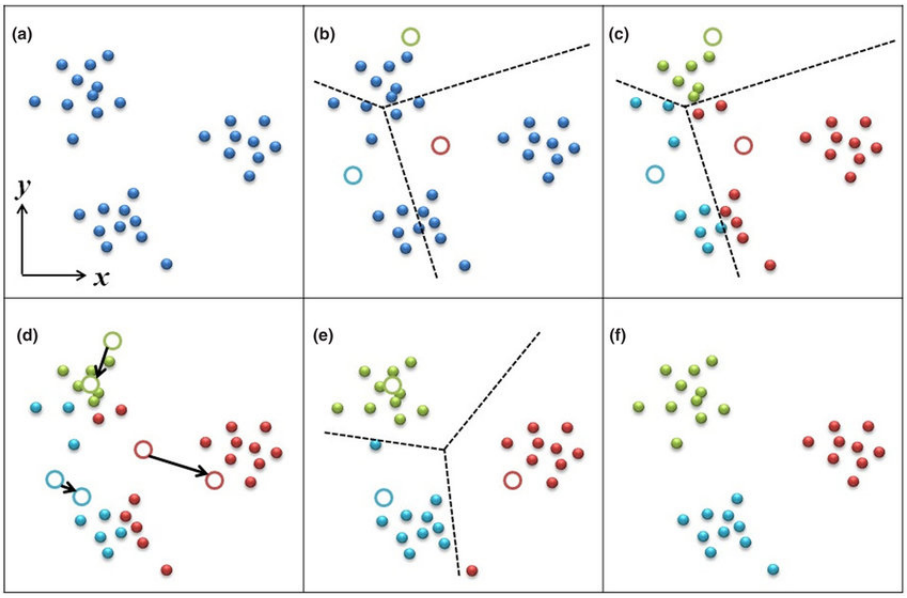


Figure 2: Visualization of Lloyds' algorithm, illustration from Chen and Lai [2018]

Input:
D= {t1, t2, …. Tn } // Set of elements
K // Number of desired clusters
Output:
K // Set of clusters
K-Means algorithm:
Assign initial values for *m1, m2,…. mk*
repeat
assign each item tj to the clusters which has the closest mean;
calculate new mean for each cluster;
until convergence criteria is met;

Figure 3: Pseudo code for Lloyds' algorithm, illustration from Mohd et al. [2012]

2.3.2 Hartigan's Algorithm

Lloyds' algorithm is often used for its simplicity. However, it has a lower probability of finding the optimal solution compared to Hartigan's. According to Slonim et al. [2013] the local minima of Hartigan's is a subset of Lloyds' algorithm and is possibly more robust to different initial partitionings of the data set.

The method starts with a random partitioning of the elements. Then, a data point is removed from a cluster and all the centroids of the clusters are computed. The object is assigned to the cluster that minimizes addition of dissimilarity. This includes the distance from the added element (x) to the new centroid (v_{c^+}) and the change in distance from all existing elements (x') to the new centroid.

$$\Delta D(x,c) = \frac{1}{n} d(v_x, v_{c^+}) + \frac{1}{n} \sum_{x' \in c} (d(v_{x'}, v_{c^+}) - d(v_x, v_c))$$

That means that the object can be assigned to a cluster without the nearest centroid. This process is done on all objects in the data set until no more optimization is possible (Figure 4).

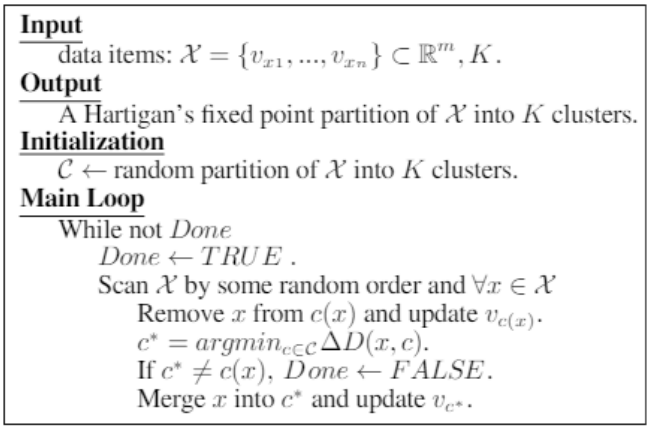


Figure 4: Pseudo code for Hartigan's algorithm, illustration from Slonim et al. [2013]

2.4 Bias

Lloyds' algorithm introduces a bias that Hartigan's does not. The centroids are computed in batch and are the averages of all elements in the clusters. When a data point is reassigned to a centroid, it has already affected the position of the closest centroid from the previous iteration. This creates a bias where all the elements have a tendency to stick to the "same" centroid. In Hartigan's algorithm this is avoided by computing all centroids without the current object. Then, there is no bias for it to choose the same centroid as before the reassigning.

3 Improvements to the Method

The standard k-means method is one of the most, if not *the* most, used clustering algorithm for discovering patterns in data sets. The algorithm has been around since the 1970s and has been surpassed by newer, more specialised methods. So why is k-means still being used and researched today? Fränti and Sieranoja said it best in their 2019 paper:

> There are other algorithms that are known, in many situations, to provide better clustering results than k-means. However, k-means is popular for good reasons. First, it is simple to implement. Second, people often prefer to use an extensively studied algorithm whose limitations are known rather than a potentially better, but less studied, algorithm that might have unknown or hidden limitations.
> —Fränti and Sieranoja [2019, sec. 1]

The rest of this section focuses on the shortcomings of k-means, and techniques for overcoming these.

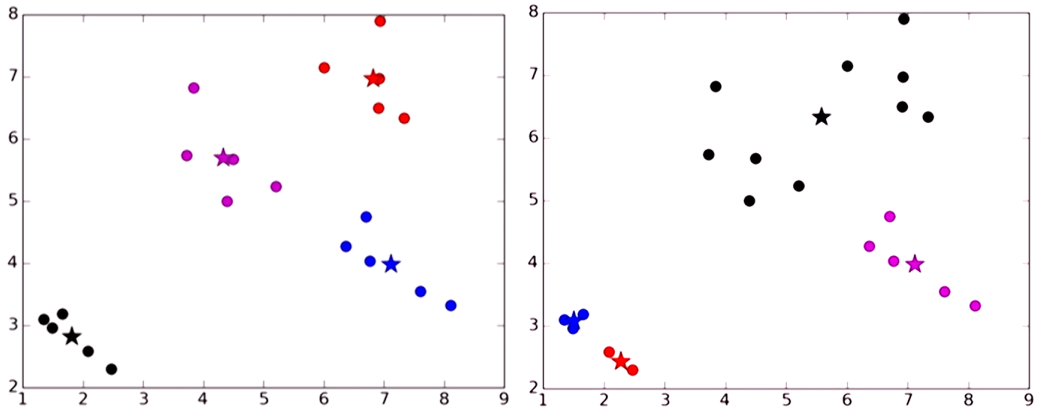
3.1 Shortcomings of k-means

When reading about k-means there are usually two main problems which present themselves:

- An unlucky initialization will converge at a sub-optimal solution
- Choosing the right "k" is not always trivial

The reason why we use the term "unlucky initialization" here stems from the fact that k-means is a non-deterministic, greedy algorithm which can get stuck in local optima. Figure 5 illustrate

this problem, with Figure 5 a) starting with a lucky initialization and Figure 5 b) ending up with a bad convergence due to its unlucky initialization.



a) Convergence with a lucky initialization b) Convergence with an unlucky initialization

Figure 5: The difference between a lucky and an unlucky initialization, illustrated by Guttag [2016]

Choosing how many clusters k-means should look for in the data set, often referred to as just "k", is essential for good results as k is sometimes the only hyper-parameter. Figure 6 shows an example where a k-value of 4 is chosen when a value of 3 might have fit the data set better.

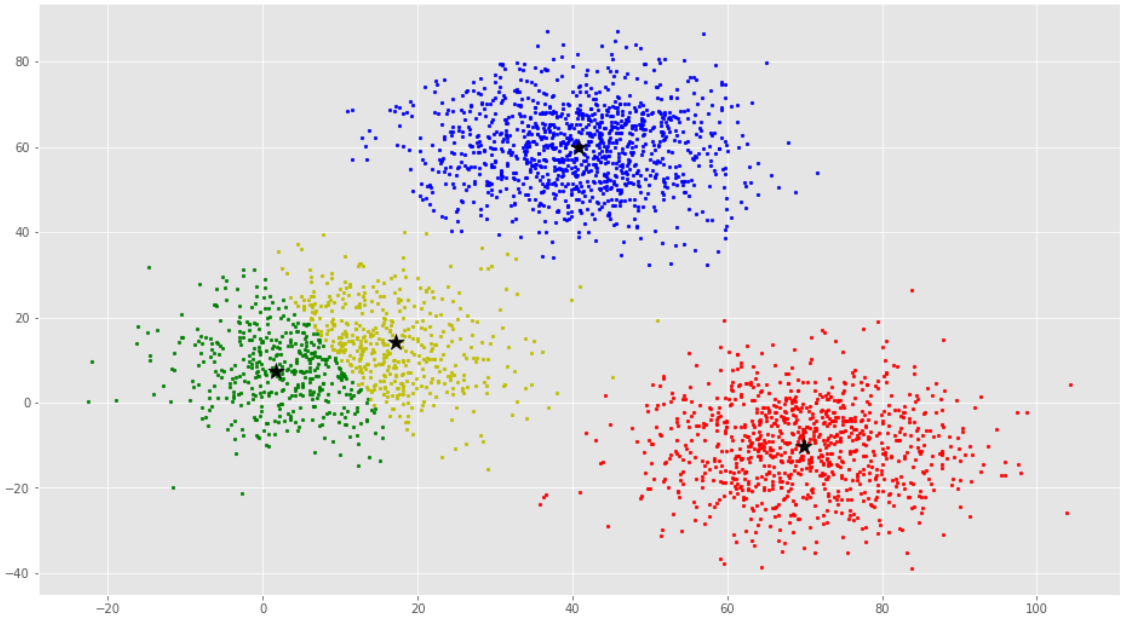


Figure 6: Setting k to 4 when there are are apparently 3 clusters, illustrated by Mubaris [2017]

3.2 Overcoming unlucky initializations

When dealing with the problem of unlucky initializations there are two main approaches one can make use of. The first one is to simply to re-run the algorithm several times and pick the clustering with the best results based on some metric. In an experiment done by Fränti and Sieranoja [2019], they managed to lower the error rate from 15% to 6% by doing 100 repeats. However, if you are dealing with a really large data set or lack the computational resources required, then this approach might be easier said than done.

The second approach is to use a smarter initialization technique than the standard k-means offers. There are many alternative initializations available for k-means which all have their strengths and weaknesses, but according to Fränti and Sieranoja [2019] Maxmin and k-means++ had the best overall performance. k-means++, invented by Arthur and Vassilvitskii [2007], is a heuristic-based seeding algorithm. Instead of selecting all the initial centroids all at once, k-means++ instead chooses the first centroid at random, then it chooses the next one randomly with uneven probabilities based on the distance squared to the nearest centroid. By doing it this way, the initial centroids are going to be more spread out which in turn lower the error rate. The price you pay is, of course the extra time it takes to compute the initial centroids.

3.3 Choosing the right "k"

Determining how many clusters you should use when running k-means algorithm can either be quite trivial or it can be bit more challenging. Choosing the hyperparameter k turns out to be very easy when we have a priori knowledge about our data (i.e. we know what k should be). An example of this can be found in image compression when we want to reduce the colour space by only using a small, fixed amount of colours for our image. Sometimes we might also be able to plot our data to "see" how many clusters there should be. The example in Figure 6 illustrates this approach well since the data can be plotted in 2D.

On the other hand, if we have no clue what the best value for k should be, we must find it. There is a popular method called *"The Elbow Method"* which uses a scree plot to find the k [Thorndike, 1953]. If you were to plot the total sum of squares in your data over the number of clusters used, you should see a figure resembling an elbow. The theory says that you have found the right amount of clusters when adding another cluster would not give a noticeably better modelling of the data - this usually happens at the elbow point.

The problem with the elbow method is that it is not always obvious where the elbow point is, and sometimes there might even be more than one elbow point. This is why the silhouette coefficient method [Rousseeuw, 1987] is sometimes used instead.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \qquad \bar{s}_k = \frac{1}{n}\sum_{i=1}^{n} s(i)$$

$s(i)$ denotes the silhouette score for a particular data point based on a combination of its mean intra-cluster distance, $a(i)$, and the shortest distance to another cluster, $b(i)$. $\bar{s}_k$ is the silhouette coefficient, sometimes referred to as *the overall silhouette width*, for a given value of k. We can use $\bar{s}_k$ as an objective function and maximise it with respect to k. The difference between the elbow method and the silhouette coefficient method is shown in Figure 7.

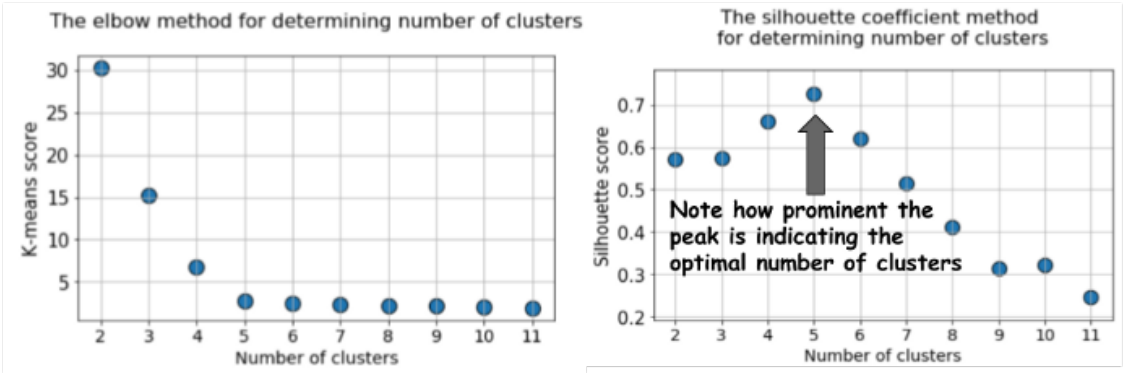


Figure 7: The elbow method and the silhouette coefficient method side by side with the same data set, illustrated by Sarkar [2019]

4 Current Applications

Machine learning has gone from a field of study, to a technology applicable in most industries, such as Oil and Gas, Financial Services, Health Care, Retail and Transport. During this transition, the commercial world has become more and more competitive, as organisations have to satisfy the needs of customers and attract new ones in order to enhance their business [Premkanth, 2012]. Today's commercial edge lies in understanding customer needs and the ability to correctly segment customers based on similar market behaviour and characteristics.

In an study published by the International Journal of Advanced Research in Artificial Intelligence (IJARAI), the k-means algorithm was used as part of a research on customer segmentation [Chinedu Pascal Ezenkwu, 2015]. The aim of the study was to test the validity of the algorithm by applying it on a data set collected from a mega retail business outfit in Akwa Ibom state, Nigeria. The data set consisted of 2 attributes and 100 tuples, representing 100 selected customers. The two attributes included average number of customer visits per month and average amount of goods purchased by the customer per month. In realising an accurate result, z-scaling was applied as a normalization technique and each data point converted to the range of -2 to +2 before running the k-means algorithm. Four cluster centres were then selected at initialisation using the *Forgy method*. The algorithm converged after 100 iterations. The study concluded the algorithm was able to cluster almost the entire data set correctly. The results are shown in figure 8.

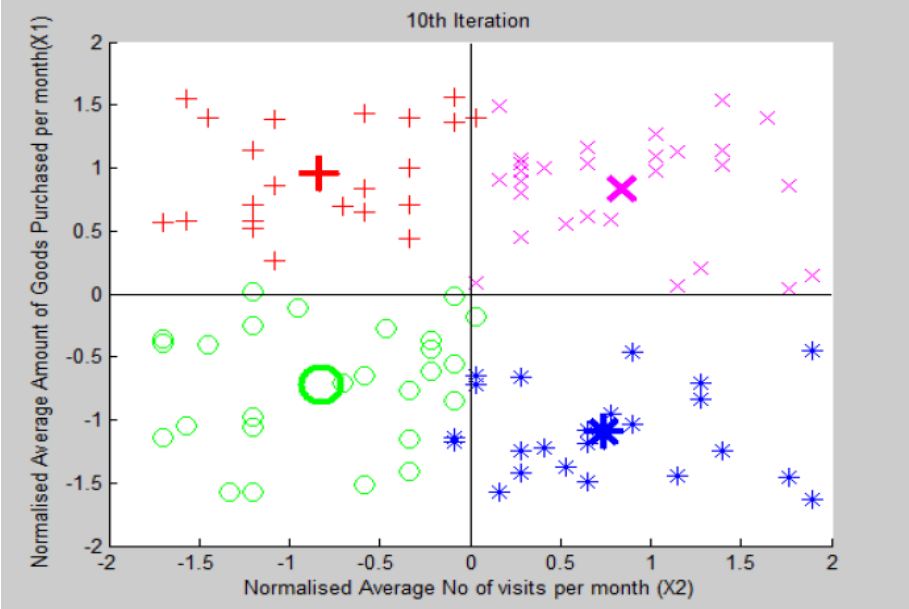


Figure 8: The centroids converge after 100 iterations, illustrated by Chinedu Pascal Ezenkwu [2015]

The performance evaluation yielded promising results for the algorithm, with a purity measure of 0.95 for the four clusters. This is in line with the expected error rate of using of the k-means clustering algorithm. As presented in a study by Fränti and Sieranoja [2019] one may expect an average purity of 0.94 when running 100 iterations of the algorithm, giving an expected average error rate of 0.06.

Given the clustering problem, segmentation might seem like a natural use case on which to apply the k-means clustering algorithm. However, there are several less obvious use-cases for the algorithm that have yielded promising results. In a study conducted by the Leshan Vocational and Technical College, Sichuan [Wan, 2019], the algorithm was tested for image compression.

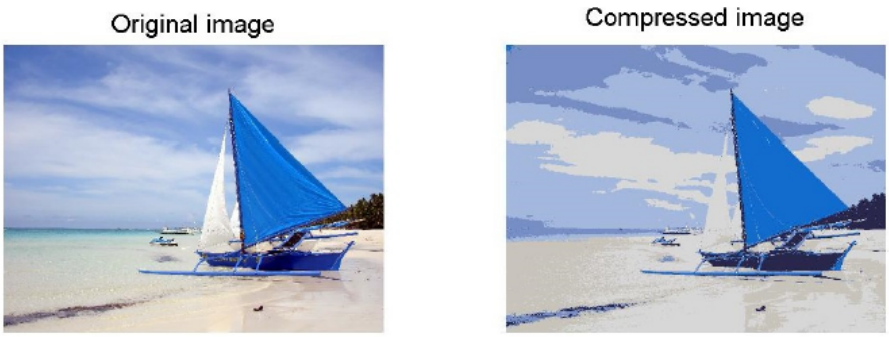


Figure 9: Image compressed with k-means algorithm, illustrated by Wan [2019]

The study concluded that the k-means clustering algorithm can be used for image compression, but the compression is a loss compression. This is clearly shown in figure 9 where the compressed image lack substantial details compared to the original image. The study also concluded that the compression ratio is proportional to the size of k, and that a higher number of iterations yield a better compression for images with large pixels.

5 Conclusions and Further Work

Firstly, viewing k-means as one single algorithm might be somewhat misleading. The k-means algorithm is more of a concept or a cluster of algorithms than a specific algorithm. Hartigan's and Lloyds' implementations differ in the process of updating centroids and clusters, but are both based on the use of centroids when minimizing dissimilarity.

Secondly, it's important to emphasise that the different implementations of the algorithm have both their strengths and weaknesses. Hartigan's algorithm have a higher chance of reaching the global minimum. Yet, it is harder to understand and implement than Lloyd's version. Therefore one should understand that even though the initial implementation of the algorithm depend somewhat on the goals of the user, every implementation will have its trade-offs.

The k-means method is a terrific method because of its simplicity, its steps are few and easy to understand. Yet, it handles the complex problem of clustering objects into sets of similar elements. The algorithm is also versatile, and as shown, the use cases are many, making it applicable for segmentation as well as for image compression. Still, it should be pointed out that even though k-means can be applied in several cases, it does not guarantee optimal results. As shown in figure 9, using the algorithm for image compression might work, but the quality loss is quite substantial.

There are several, well known challenges when it comes to initial centroids and choosing the right k. However, these are deeply studied and it is easy to find several ways of extending the basic k-means method to solve them.

Bibliography

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. volume 8, pages 1027–1035, 01 2007. doi: 10.1145/1283383.1283494.

Yu-Zhong Chen and Ying-Cheng Lai. Sparse dynamical boltzmann machine for reconstructing complex networks with binary dynamics. *Physical Review E*, 97, 03 2018. doi: 10.1103/PhysRevE.97.032317.

Constance kalu Chinedu Pascal Ezenkwu, Simeon Ozuomba. Application of k-means algorithm for efficient customer segmentation: A strategy for targeted customer services. *International Journal of Advanced Research in Artificial Intelligence*, 4, 2015.

David Court. Getting big impact from big data. *McKinsey Quarterly*, Jan 2015. URL https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/getting-big-impact-from-big-data.

Pasi Fränti and Sami Sieranoja. How much can k-means be improved by using better initialization and repeats? *Pattern Recognition*, 93:95 – 112, 2019. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.04.014. URL http://www.sciencedirect.com/science/article/pii/S0031320319301608.

John Guttag. 12. clustering, 2016. URL https://youtu.be/esmzYhuFnds?t=1591.

Alfred Bruckstein Michal Aharon, Michael Elad. K-svd: An algorithm for designing overcomplete-dictionaries for sparse representation. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 54, november 2006.

Wan Mohd, Abul Beg, Tutut Herawan, and Khandakar Rabbi. *MaxD k-Means: A Clustering Algorithm for Auto-generation of Centroids and Distance of Data Points in Clusters*, volume 316, pages 192–199. 01 2012. ISBN 978-3-642-34288-2. doi: 10.1007/978-3-642-34289-9_22.

Mubaris. k-means clustering in python. 2017. URL https://mubaris.com/posts/kmeans-clustering/.

Puwanenthiren Premkanth. Market segmentation and its impact on customer satisfaction with especial reference to commercial bank of ceylon plc. *Global Journal of Management and Business Research Publisher: Global Journals Inc. (USA)*, 12, 2012. ISSN 0975-5853.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL http://www.sciencedirect.com/science/article/pii/0377042787901257.

Tirthajyoti Sarkar. Clustering metrics better than the elbow-method. *towards data science*, Sep 2019. URL https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6.

Noam Slonim, Ehud Aharoni, and Koby Crammer. Hartigan's k-means versus lloyd's k-means: is it time for a change? pages 1677–1684, 08 2013.

Tim Stack. Internet of things (iot) data continues to explode exponentially. who is using that data and how? *Cisco Blogs*, Feb 2018. URL https://blogs.cisco.com/datacenter/internet-of-things-iot-data-continues-to-explode-exponentially-who-is-using-that-data-and-how.

Robert L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, Dec 1953. ISSN 1860-0980. doi: 10.1007/BF02289263. URL https://doi.org/10.1007/BF02289263.

Xing Wan. Application of k-means algorithm in image compression. *IOP Conf. Series: Materials Science and Engineering*, 2019.