MATH336 Project - Group 29

Lorenzo Colpani (34736662), Megan Harries (34733388)
and Hongsheng Lin (37527355)

December 2020

First meeting summary

Collectively, we organised our first virtual meeting on 20th November 2020, which all of the group members attended. During the meeting we discussed how we planned to approach the group project; what everyone's roles would be and what we would aim to accomplish individually in preparation for the following meeting. Table 1 outlines which initial tasks we decided to allocate to whom. Alongside these individual tasks we also planned to all initiate our own research on k-means clustering and its use as an image compression tool.

Group Member	Responsibility
Lorenzo	Run the given code (found in the Appendix) in `R Studio` and Section 3
Megan	LaTeX document, pseudo-code and Section 2 & 4 of the report
Hongsheng	Section 1 and Section 5 of the report

Table 1: Allocation of Tasks

1 Introduction

This report will investigate the clustering problem and how it differs from classification, focusing specifically on k-means clustering and its application as an image compression tool. Within the field of machine learning, clustering is a classical example of an unsupervised learning task, which is to identify meaningful patterns in the data from an unlabeled data set. In other words, the model receives no hints of how to categorise each piece of data and must infer its own rules for doing so. The details of the clustering problem will be discussed in in the following section. Clustering encompasses a broad set of techniques for finding subgroups of observations within a data set, and in the case of k-means clustering, splits a dataset into a set of k distinct groups. The k-means clustering algorithm will be discussed further in Section 3. One of the most wide-spread applications of clustering is its use as an image compression tool, which is the focus of Section 4, we used `R Studio` to run the k-means clustering algorithm to compress images which can be found in the Appendix.

2 The Clustering Problem

Clustering is an example of an unsupervised learning task, in contrast to classification which is supervised. In unsupervised learning no label sets are assigned to the learning algorithm, therefore the algorithm has to decide how to group the training data $\mathcal{D}$ itself based only on the domain set $\mathcal{X}$, as a label set $\mathcal{Y}$ is not given within the training data. Therefore there is no distinction between the data used to train the algorithm and the test data. The aim of using a clustering algorithm is to discover which (if any) groups emerge from within the data having similar traits and therefore are clustered together. A cluster is therefore a collection of data points which have common similarities between them and are dissimilar in some way to objects belonging to other clusters [6]. A clustering example we will explore in Section 4 is using an unsupervised learning algorithm to find and group similar colours together within an image.

One of the challenges when using clustering is deciding how the data will be grouped. There are many different ways to cluster data, one of them being k-means, which we will explore further in Section 3 as well as its applications to solving real problems. In general, algorithms decide how best cluster data based on two types of distances: minimising the distance between two data points in the same cluster (internal distance) and maximising the distance between two data points in different clusters (external distance) [6]. The user ultimately decides how to prioritise these distances to suit the needs of their project. There are multiple ways of finding the distance between two pieces of data in a dataset. If our data can be plotted as points on a graph, such as in Figure 2 then we can use the Euclidean distance between the points to determine the internal/external distances. However if our data is a vector, not plot-able on the plane, then we would use the cosine distance instead. In the rest of this report we will focus on using the Euclidean distance between two points, as this is applicable for k-means clustering.

One problem with using unsupervised learning is that the conclusions we draw from the data may not be meaningful as no true labels are available to compare against. For example in Figure 2 without knowing which points correspond to which type of flower, there is no way of knowing if the data has been clustered correctly in comparison to the true values of the data, therefore it isn't useful to use a method of clustering for this particular dataset.

3 The k-means Clustering Algorithm

k-means is an exhaustive clustering algorithm which works iteratively to find a certain number of k clusters in a dataset, where k is variable, this method of clustering can be applied to data that is numeric, has few dimensions and is continuous. When processing the data the k-means algorithm works by randomly selected a chosen amount of k centroids from the data, therefore producing an initial randomised set of clusters. It will then keep repeating iterative calculations to optimise the positions of the centroids until all the centroids are stable. Once all the centroids are stable it will return the k optimised clusters.

A pseudo-code of k-means is given by:

```
1. Obtain the dataset (x_1, x_2, . . . , x_n) and choose a number of
   clusters k
2. Choose the centroids c_1, c_2, . . . , c_k randomly amongst the data
3. For each data point x_i (i in 1 to n):
     - Find the centroid that is nearest
     - Assign x_i to that centroid's cluster
4. For each cluster j in 1 to k
     - A new centroid is produced = mean of all points in that cluster
5. Repeat 3. and 4. until a given number of iterations or until the clusters
   converge and stabilise
```

A practical use of k-means clustering is grouping viewers of a streaming service such as Netflix into clusters based on similar viewing behavior. We know that there are clusters, which represent the viewing groups however we do not know exactly what those clusters are. Linear models are of little help with these sorts of issues, whilst k-means is applicable [7].

Looking at supervised learning instead, for comparison, one method is classification. In classification models, the algorithm will recognise and group objects into preset labels from $\mathcal{Y}$ (which it will learn in the training stage). The data has to be divided into three sets: the first and biggest set of the data is the training set $\mathcal{D}$ which will be used by an algorithm to learn how to classify data. The second subset is the validation set, with this data the user can see how well its algorithm can recognise the labels and repeat the training stage if necessary until it performs with a reasonable accuracy. When the user is satisfied with the algorithm performance, it will test it with the last data set, the test data. If classification is performed with a high accuracy compared to the true labels, the algorithm can then be used to classify new unlabelled data.

Comparing k-means algorithm to classification, one of the most prominent advantages is that training data does not exist and the whole dataset is used instead, therefore you can implement k-means with real unlabelled data from the beginning. Another advantage is that the algorithm will teach itself and doesn't rely on the user to decide the label set $\mathcal{Y}$ as is the case with supervised learning data, this means that it could find new ways to categorise the data which might never have occurred to the user. Another advantage for k-means over classification is that it doesn't have the problem of overfitting. Overfitting occurs when a model is created (in this case the classification algorithm) that matches the training data so closely that the model fails to make correct predictions on new data. However, k-means does have certain drawbacks, one of these is that the user might not understand how the algorithm chooses to group each cluster and therefore can't interpret its results. Another disadvantage is choosing a value for k. There will be an optimal k but it is sometimes difficult to find, therefore using just slightly different values of k could give very different results and misplace data that would naturally belong to another set. A final drawback is with outliers, these could drastically impact the centroid of each new cluster, therefore also misplacing data.

4 Application to Image Compression

Clustering algorithms have many real-world uses within data analysis ranging from marketing to crime prevention, examples being: customer segmentation, so marketers can better identify their customers and work on target areas and k-means clustering can be used to detect fraud by recognising patterns with past fraudulent insurance claims. Another one of its useful applications is within image compression, which is what we will focus on in this section. Image compression reduces the size of an image file by finding pixels with similar colours and returning all of these pixels with their common colour (found at the cluster's centroid), with the aim to reduce the amount of data in the image and therefore decrease the file size resulting in less memory usage. The compressed image is visually similar to the original if it uses a higher number of k clusters since more colours are then being returned. When using image compression, we need to find a minimum k such that the file size is compressed adequately whilst the quality of the image is not compromised.

To reduce the colour count of an image we can run the R code in the Appendix, this converts the RGB (Red-Green-Blue) values of each pixel in the image into a large data frame. Choosing $k = 16$ for example we can use the `R` function `kmeans` to find 16 clusters and therefore 16 centre points. Each of these individual clusters will then be re-coloured with the RGB value of their respective centre point, reducing the colour count of the image to just 16 colours. An example of k-means clustering when $k = 16$ is shown in Figure 1.

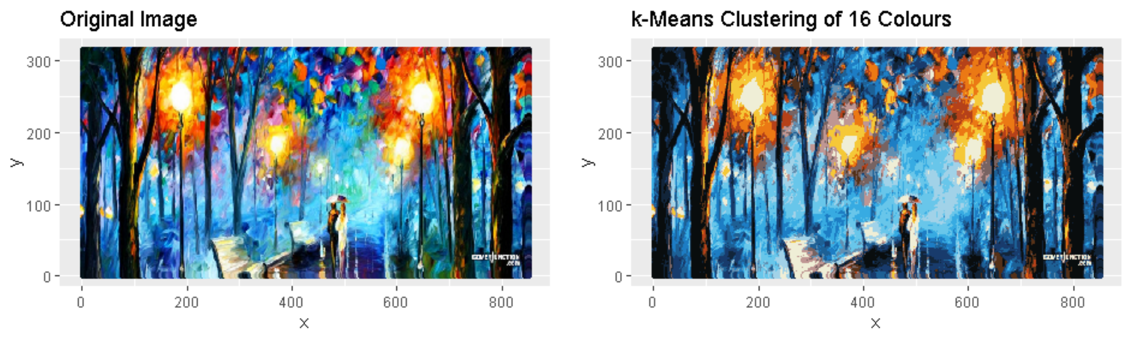

Figure 1: Plot of the original compared to the compressed image with 16 colours using k-means.

As shown by the images, using $k = 16$ hasn't reduced the quality of the image visibly, the only downside is using a k value this small on this particular image means that the compressed image is less vibrant as some of the brighter colours have been replaced by others. The compression quality of the image changes with k, as k is increased and tends towards the colour count of the original image the image compression quality is also higher. Conversely, as k is decreased the colours in the image are simplified as shown with $k = 3$ in Figure 3, therefore the compression quality also decreases, this means that k-means is an example of lossy compression (some original data from the image is lost when it is compressed). Although using a higher value of k means that the image is of better quality, the file size of the image will also increase, thus finding a k which optimises colour count whilst minimising the file size is important for image compression.

5 Conclusion

k-means clustering has lots of useful applications, one of these being as a tool for image compression as we saw in Section 4. However there are also some limitations to this method of clustering. Since unsupervised learning techniques have no given labels, the algorithm needs a suitable data set to be applied to. Three main problems may occur when the datasets are: different sizes, different densities or the data set have non-globular distributions. Examples of these data types are shown in Figures 4, 5 and 6 in the Appendix. As mentioned in Section 3, the centroids can be dragged by outliers, or outliers might be assigned their own individual cluster instead of being removed before analysis. Also, the clusters that are discovered are always dependent on k, therefore even small changes such as from k to $k + 1$ could cause different outcomes. In the application of image compression using different values of k will form very different images as shown with $k = 3$ in Figure 3. In conclusion, the application of k-means clustering removes a large amount of redundant information from an image, in order to reduce the file size, therefore improving storage and data transmission, however only retains the quality of the image when an optimal k is found.

Appendix

R code for compressing an image into k-means clusters when $k = 16$:

```
image.path<-paste(getwd(),"/Image.jpg",sep="")
img <- readJPEG(image.path) # Read the image

imgDm <- dim(img)

imgRGB <- data.frame(
  x = rep(1:imgDm[2], each = imgDm[1]),
  y = rep(imgDm[1]:1, imgDm[2]),
  R = as.vector(img[,,1]),
  G = as.vector(img[,,2]),
  B = as.vector(img[,,3])
)

plot1<-ggplot(data = imgRGB, aes(x = x, y = y))
        + geom_point(colour = rgb(imgRGB[c("R", "G", "B")]))
        + labs(title = "Original Image")

k <- 16
kMeans <- kmeans(imgRGB[, c("R", "G", "B")], centers = k)
num.of.colours <- rgb(kMeans$centers[kMeans$cluster,])

plot2<-ggplot(data = imgRGB, aes(x = x, y = y))
        + geom_point(colour = num.of.colours)
        + labs(title = paste("k-Means Clustering of", k, "Colours"))

plot1 # Plots the original image

plot2 # Plot of the compressed image
```

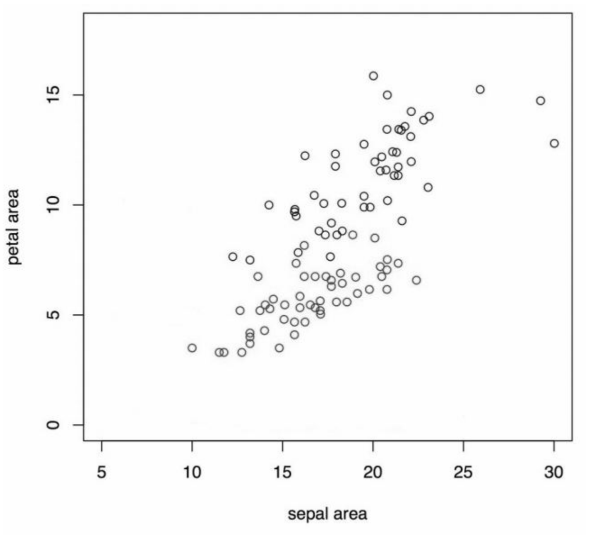

Figure 2: Scatter plot of data from two types of flowers [3]

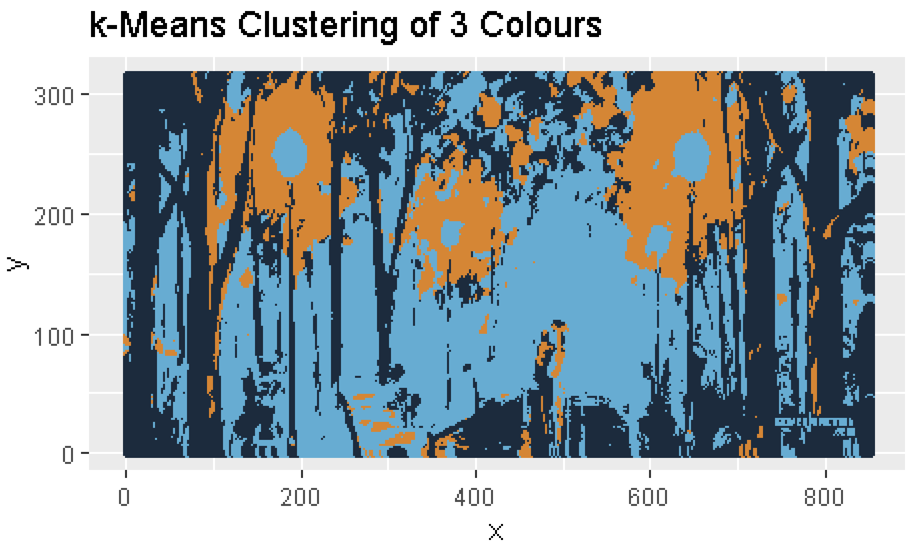

Figure 3: Plot of image compression of the original image from Figure 1 with 3 colours using k-means clustering.

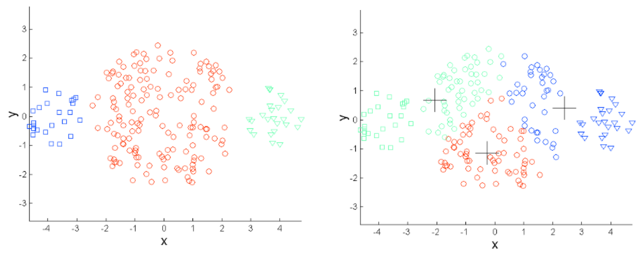

Figure 4: Different data set sizes.

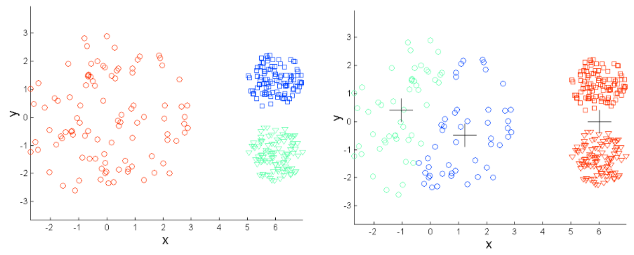

Figure 5: Different data set densities.

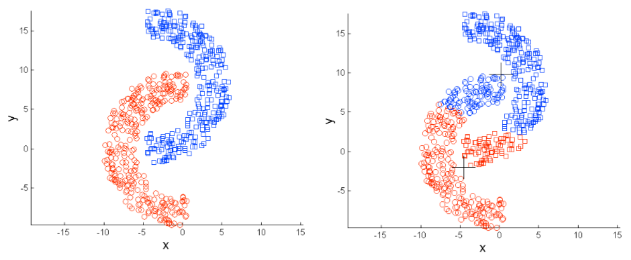

Figure 6: Non- globular shapes.

References

[1] Google (2020a). *k-Means Advantages and Disadvantages.* Google Developers.

[2] Google (2020b). *Training and Test Sets: Splitting Data.* Google Developers.

[3] Khaleghi, A. (2020). *Multivariate Statistics in Machine Learning.* Lancaster University. 6

[4] Kumar, S. (2020a). *Image Compression using K-Means Clustering.* Towards Data Science.

[5] Kumar, S. (2020b). *Understanding K-Means, K-Means++ and, K-Mediods Clustering Algorithms.* Towards Data Science.

[6] Mishra, S. (2017). *Unsupervised Learning and Data Clustering.* Towards Data Science. 1, 2

[7] Sears-Collins, A. (2019). *Advantages of K-Means Clustering.* Automatic Addison. 3