K-Means Clustering Method

Mary Manz\*

MATH637 - Math Techniques in Data Science

University of Delaware

Newark, DE 19716

April 30, 2021

1. Introduction

To help answer the question about how happiness levels for people in different countries have trended over the last 5 years, I first used a heatmap. A heatmap can be used to “show relationships between two variables, one plotted on each axis” and “by observing how cell colors change across each axis, you can observe if there are any patterns in value for one or both variables” [1]. The heatmap in my report did not show any obvious patterns, indicating that there were not many countries with significant increases or decreases in their happiness scores over the last five years. To get a better sense of whether this was the case, I decided to separate countries with similarly trending happiness scores, using the k-means clustering method, which is available in python’s sci-kit learn package. In this whitepaper, I discuss the algorithm of the k-means clustering method in Section 2; the elbow method associated with the k-means clustering method in Section 3; and the result from using these methods in Section 4.

1. Algorithm

The k-means clustering method is an iterative process for grouping data objects with similar features. Each group of data objects is referred to as a “cluster.” In my report, the objects are the countries, and the features are the happiness levels for each year between 2015 and 2020. In the simplest form of the k-means method, each object is a single data point for 2-dimensional data. However, in my report, each object is represented by a time series graph. The following steps are used in the k-means clustering method:

K-means algorithm [2]:

1. Select ‘k’ initial cluster centers. These will be the center graphs for each group.
2. Assign object graph to nearest cluster center.
3. Reassign center graphs to be the calculated mean value for each cluster.
4. Reassign object graphs to nearest cluster center.
5. Repeat steps 2-4 until object graphs stay in the same cluster.
   1. Step 1 of algorithm

I used the Elbow Method, discussed in Section 3, to determine the number of ‘k’ cluster centers to use in step 1 of the k-means algorithm. The rest of Step 1, the selection of initial cluster centers, is done randomly. It is possible that many different initial cluster centers will converge (when running the algorithm) to the same cluster centers; however, it is possible to select a weak initial cluster center, leading to less-than-optimal final cluster centers. The best performing final cluster centers have the lowest sum of squares error (SSE). The SSE is computed as “the sum of the squared Euclidean distances of each point to its closest centroid” [2], for which ‘point’ in my analysis is the object graph and ‘centroid’ is the cluster center. To help avoid selecting a weak initial cluster center, the k-means method in the scikit-learn python package tests ten different initial clusters by default. I did not change the default for my analysis.

* 1. Steps 2 through 5 of algorithm

After initial cluster centers are determined, the distance of each object graph to each cluster center is measured. Object graphs are assigned to cluster centers that they have the shortest distance to. The mean of each assigned group of object graphs is then calculated, and these means become the new cluster centers. Then the object graphs are once again assigned to the cluster center that they are closest too. If the resulting groups contain different object graphs than before, than the calculated mean of each group will differ from before, and there will be new cluster centers. Once the resulting groups do not contain different object graphs, or a maximum number of iterations of steps 2 through 5 has been reached, the last calculated cluster centers are the ones used for visualization. The default maximum number of iterations used in the scikit-learn python package is 300. I did not change the default for my analysis.

1. Elbow method

The elbow method tests out several different values for ‘k’, comparing the performance or quality of the resulting final cluster centers. As mentioned in Section 2.1, the performance of clusters is measured in SSE; this can be referred to as “inertia.” The quality of cluster centers improves as the value of ‘k’ increases; however, getting the best quality cluster centers, with a value of zero inertia, means that there is a cluster center for each object graph. This is not ideal since the objective of the k-means method is to group together similar object graphs. On the other hand, having a single cluster does not separate the object graphs at all. So, what is the best number of clusters or ‘k’? If one plots a range of ‘k’ values as a function of inertia “it turns out that the point that indicates the balance between greater homogeneity within the cluster and the greater difference between clusters is the point in the curve that is most distant of a line drawn between” the first and last points on the plot. This point is also where one might imagine an elbow would be if the graph were a person’s arm; or where the graph begins to decrease linearly.

I applied the k-means method using a range of ‘k’ values, calculating the inertia for each value of ‘k,’ and plotted the results.

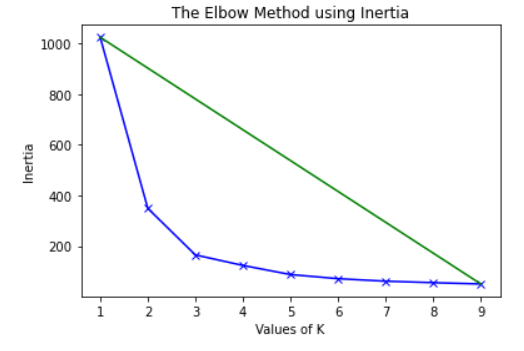


Figure 1 – The blue line represents the values of ‘k’ plotted as a function of inertia. The optimal value of ‘k’ is the x-value of a point on the blue line with the greatest distance to the green line.

Looking at Figure 1, it appears the “elbow” of the graph is at the point where ‘k’ equals three. I calculated the distances of each point on the blue line from the green line, confirming that the optimal value for ‘k’ is in fact three.

1. Summary

In conclusion, separating countries with similarly trending happiness scores is something I was able to do using the k-means clustering method from python’s sci-kit learn package. And I determined the number of clusters to use in the method via the elbow method, which gave me a result of three clusters.

References

[1] Yi, M. (2019, November 11). *A Complete Guide to Heatmaps*. Chartio.

https://chartio.com/learn/charts/heatmap-complete-guide/

[2] Jordan, J. (2017, October 18). Grouping data points with k-means clustering. Jeremy Jordan.

https://www.jeremyjordan.me/grouping-data-points-with-k-means-clustering/

[3] GeeksforGeeks. (2021, February 9). Elbow Method for optimal value of k in KMeans.

https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/

[4] How to define the optimal number of clusters for KMeans. (2019, April 13). Jessica

Temporal. https://jtemporal.com/kmeans-and-elbow-method/