K-Means Clustering
WUT IML Project

Mateusz Szymoński

November 18, 2020

1 Introduction

K-means clustering is a widely used, unsupervised machine learning algorithm. Its aim is to partition a set of N data point into K distinct, non-overlapping clusters so that the within-cluster variation is minimized.
Number of clusters K is predefined.
Centroid is an arithmetic mean of all the data points that belong to the cluster.
K-means is guaranteed to converge but the final cluster configuration depends on the initial centroid locations. Algorithm is very sensitive to outliers.

There are multiple k-means algorithm versions available.
For example: Hartigan-Wong, Lloyd, Forgy or MacQueen.

Important factor in k-means clustering is the distance measure used.
In most of the approaches Euclidean distance is used, however instead of it, for example, Manhattan distance can be used.

Clustering approaches can be divided based on how the points are assigned to clusters:

- Hard clustering: each object either belongs to a cluster or does not.
- Soft clustering: each object belongs to each cluster to a certain degree.

Main k-means clustering applications are:

- document classification
- delivery routes optimization
- market and customer segmentation
- image compression
- data preprocessing
- etc

2 Algorithm description

In this project I used Lloyd's approach invented by Stuart P. Lloyd in 1957. Lloyd's approach is a hard clustering method that originally uses squared Euclidean distance measure.

The way k-means algorithm works is as follows:

1. Specify number of clusters K
2. Initialize centroids by randomly picking K locations in the area occupied by data points
3. Assign each data point to the cluster with the closest centroid
4. For each cluster compute mean from all assigned data points and set it as new centroid
5. Keep repeating step 3 and step 4 until termination criterion has not been met

It is not possible to find an exact solution, which means that k-means clustering is NP-hard problem. However, because steps 3 and 4 take linear time, the practical (if iteration limit is used) run time of the algorithm is basically linear.

Termination criterion:

- There is no change in assignment of data points to clusters
- Iteration limit is exceeded

Other possible approaches to choose initial centroids:

- Randomly pick K data points (centroids are positioned where those points are located)
- Initialize ith centroid to the data point whose minimum distance to the preceding centroids is the largest (farthest heuristic)
- Density-based searches
- k-means++ approach

Within-cluster variation is defined as follows:

$$W(C_k) = \sum_{x_i \in C_k} (x_i + \mu_k)^2 \tag{1}$$

where:
x_i is data point belonging to the cluster C_k
μ_k is mean value of all points belonging to the cluster C_k

Total within-cluster variation is defined as follows:

$$\text{Total within-cluster variation} = \sum_{k=1} W(C_k) \tag{2}$$

where:
$W(C_k)$ is within-cluster variation of cluster C_k

Between-cluster variation is defined as follows:

$$\text{Between-cluster variation} = \sum_{k=1}^{K} \sum_{i=1, i \neq k}^{K} (C_k + C_i)^2 \tag{3}$$

where:
K is a number of clusters
C_k and C_i are centroids of clusters C_k and C_i

3 CustomKMeans.R function documentation

CustomKMeans.R performs k-means clustering on a dataframe.

Arguments

data	Dataframe where first column is x value of data point, second column is y value of data point and each row is one data point
cluster.number	Number of clusters to partition data to
iteration.limit	Maximal number of iterations to perform. If reached the algorithm will stop

Return value

CustomKMeans.R returns an object of class "customKMeansResult" which is a list with the following elements:

clusters	Vector of length N of integers indicating the cluster to which each data point is allocated
centers	Matrix of cluster centres.
size	Vector of number of data points assigned to each cluster, one element per cluster
radius	Vector of distances to the furthest assigned data point for each cluster from its centroid, one element per cluster
iterations	Number of iterations performed to get the result
wcv	Vector of within-cluster variation, one element per cluster
twcv	Total within-cluster variation
bcv	Between-cluster variation

4 Case studies

4.1 Synthesized data sets

Clusters below were generated by ClusteringSynthesizedData.R using CustomKMeans.R

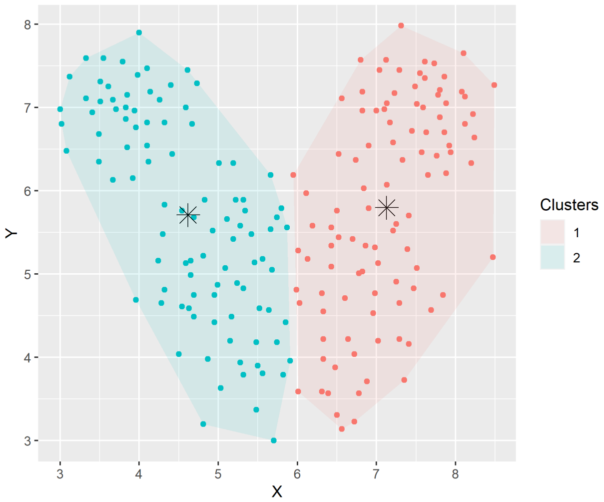

Figure 1: Result of clustering Mickey Mouse set (13 Iterations, N=200, K=2)

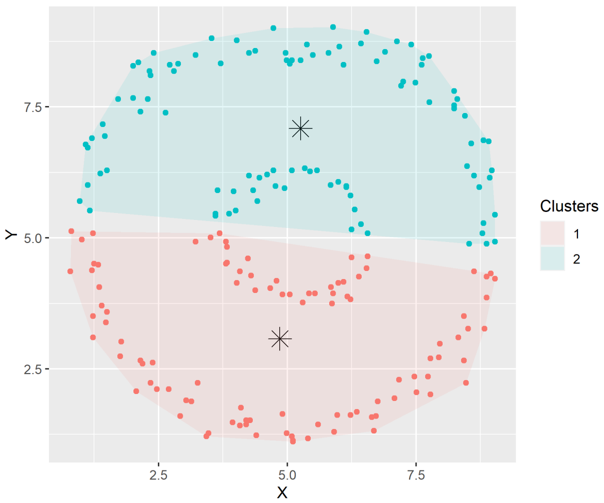

Figure 2: Result of clustering Circles set (3 Iterations, N=200, K=2)

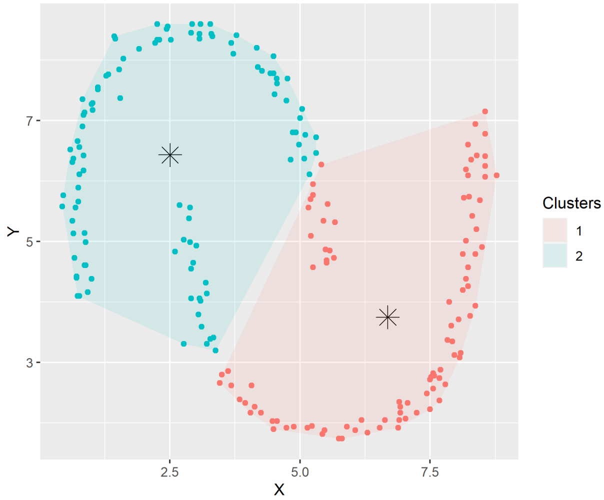

Figure 3: Result of clustering Crescents set (9 Iterations, N=200, K=2)

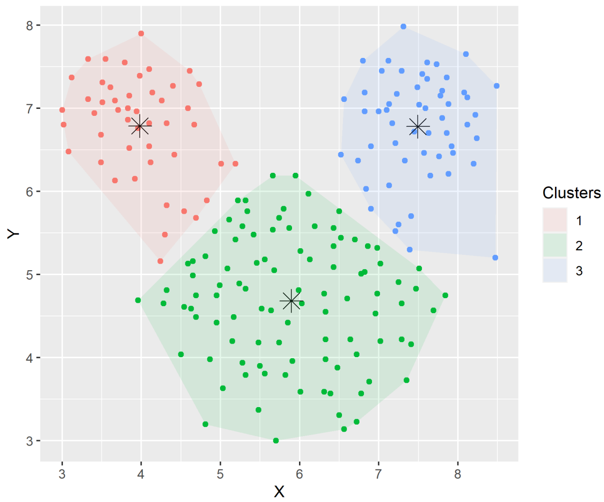

Figure 4: Result of clustering Mickey Mouse set (8 Iterations, N=200, K=3)

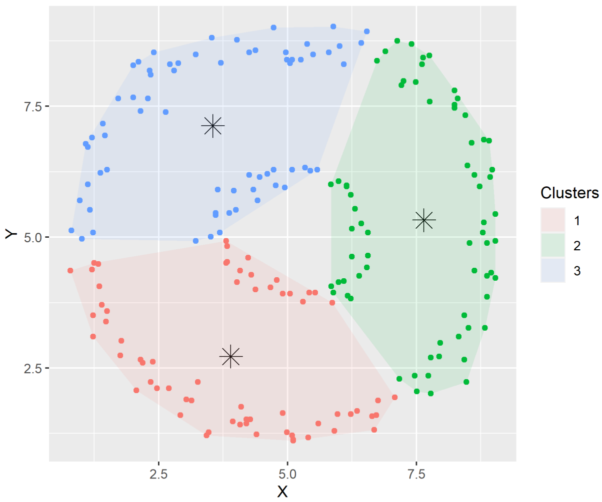

Figure 5: Result of clustering Circles set (11 Iterations, N=200, K=3)

Figure 6: Result of clustering Crescents set (9 Iterations, N=200, K=3)

4.2 Actual data sets

Clusters below were generated by ClusteringActualData.R using CustomKMeans.R
Data set of Population and Gini Market Rate for USA, Mexico, Chile, Turkey, Germany, Poland, Czechia and Sweden.
Each point represents one-year observation for each country from around 1960 to 2017.

Data comes from The Penn World Table version 9.1 and The Standardized World Income Inequality Database versions 8

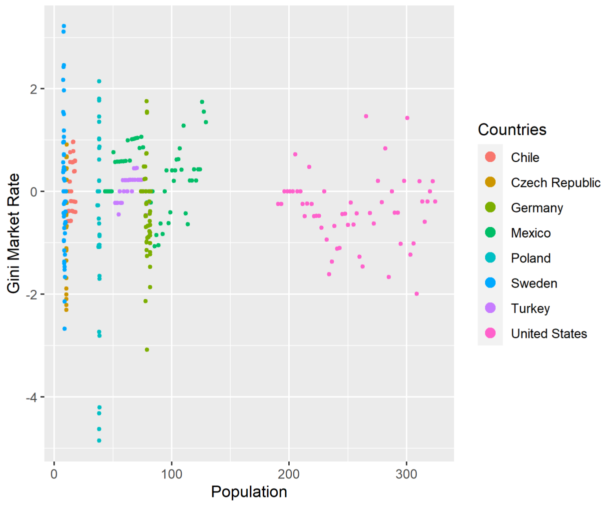

Figure 7: Scatterplot of Population and Gini Market Rate

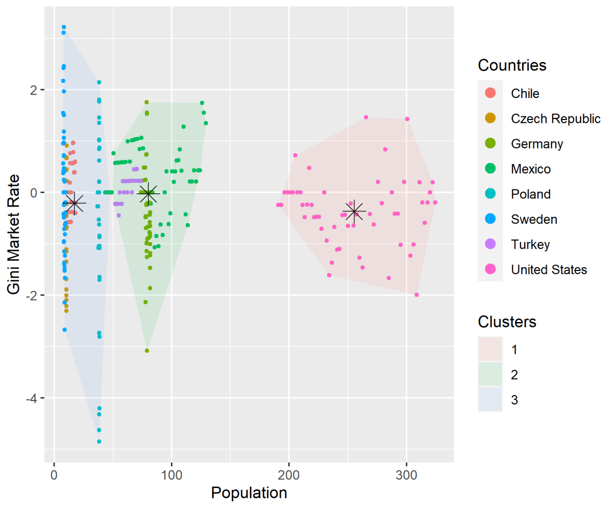

Figure 8: Result of clustering Population and Gini Market Rate set (4 Iterations, K=3)

References

[1] J. A. Hartigan, M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[2] David Arthur, Sergei Vassilvitskii. k-means++: The advantages of careful seeding. `http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf`, 2006. [Online; accessed 29-October-2020].

[3] Naftali Harris. Visualizing k-means clustering. `https://www.naftaliharris.com/blog/visualizing-k-means-clustering/`, 2014. [Online; accessed 26-October-2020].