Rough K-Means Clustering
WUT IML Project

Mateusz Szymoński

November 20, 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 sets called clusters, so that the within-cluster variation is minimized.
Number of clusters K is predefined.
Centroid is an arithmetic mean of 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.

Contrary to classic k-means clustering, rough k-means clustering operates on rough sets proposed by Zdzisław I. Pawlak in 1982. Rough set as an extension to set theory, is defined as a pair of classic sets which give the lower and the upper approximation of the original set. In rough set theory an element can belong to both approximations, to none or only to the upper approximation. This means that the lower approximations form non-overlapping clusters while upper approximations can overlap.
There are multiple rough k-means algorithm versions available.
For example: Lingras & West's, Peters', or PI.

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.

Classic k-means clustering is hard clustering while rough k-means clustering is soft clustering.

Main k-means clustering applications are:

- document classification
- delivery routes optimization
- image compression
- data preprocessing

2 Algorithm description

In this project I used soft rough k-means clustering algorithm with squared Euclidean distance measure.

The way in which rough k-means algorithm works is as follows:

1. Specify number of clusters K
2. Randomly assign each data point to the lower approximation of one cluster. By definition of rough sets, it also belongs to the upper approximation of the same cluster
3. For each cluster compute centroids in the following way:
   If $\underline{U}(K) \neq \emptyset \wedge \overline{U}(K) - \underline{U}(K) = \emptyset$
   $$C_j = \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|} \tag{1}$$
   Else if $\underline{U}(K) = \emptyset \wedge \overline{U}(K) - \underline{U}(K) \neq \emptyset$
   $$C_j = \sum_{x \in \overline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|} \tag{2}$$
   Else
   $$C_j = W_{lower} \times \sum_{x \in \underline{U}(K)} \frac{x_i}{|\underline{U}(K)|} + (1 - W_{lower}) \times \sum_{x \in \overline{U}(K) - \underline{U}(K)} \frac{x_i}{|\overline{U}(K) - \underline{U}(K)|} \tag{3}$$
4. Assign each data point to the lower approximation $\underline{U}(K)$ and to the upper approximation $\overline{U}(K)$ of cluster in the following way:
   Let $d(X, C_j)$ be distance between data point and centroid of cluster C_j
   $$\text{Distance To The Closest Centroid} = \min_{1 \leq j \leq K} d(X, C_j) \tag{4}$$
   $$\text{Assignment Factor} = \frac{d(X, C_{i \neq j})}{\min_{1 \leq j \leq K} d(X, C_j)} \tag{5}$$
   If Assignment Factor ≤ epsilon then x $\in \overline{U}(C_i)$ and x $\in \overline{U}(C_j)$ and x will not be a part of any lower approximation.

   If Assignment Factor > epsilon and already x $\notin \overline{U}(C_j)$, then x $\in \underline{U}(C_j)$ and x $\in \overline{U}(C_j)$ only.
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.

Possible termination criterion:

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

3 RoughKMeans.R function documentation

RoughKMeans.R performs rough 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
epsilon	Relative threshold in rough k-means algorithms (epsilon ≥ 1.0)
weight.lower	Weight of the lower approximation in rough k-means algorithms ($0 \leq$ weightLower ≤ 1)
iteration.limit	Maximal number of iterations to perform. If reached the algorithm will stop

Return value

RoughKMeans.R returns an object of class ”roughKMeansResult” which is a list with the following elements:

lowerApprox	Matrix of obtained lower approximations [data points × clusters]
upperApprox	Matrix of obtained upper approximations [data points × clusters]
centers	Matrix of cluster centers
iterations	Number of iterations performed to get the result

4 Case studies

4.1 Synthesized data sets

Clusters below were generated by RoughClusteringSynthesizedData.R using RoughKMeans.R

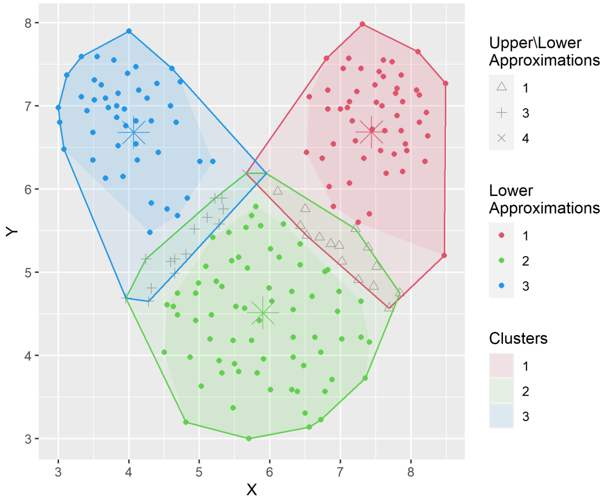

Figure 1: Result of clustering Mickey Mouse set
(N=200, K=3, epsilon=2.0, weight.lower=0.9, iterations=12)

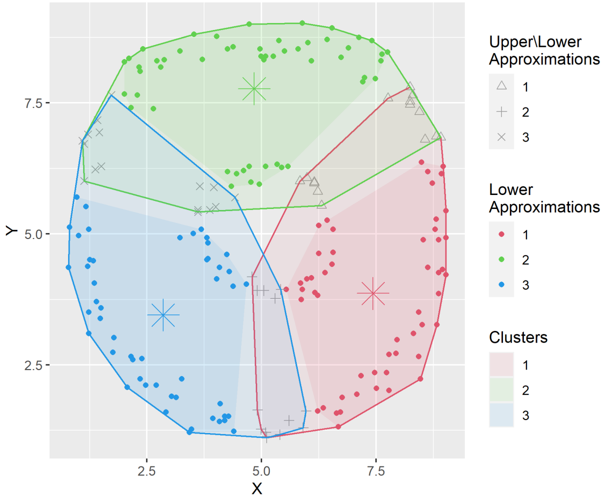

Figure 2: Result of clustering Circles set
(N=200, K=3, epsilon=2.0, weight.lower=0.9, iterations=7)

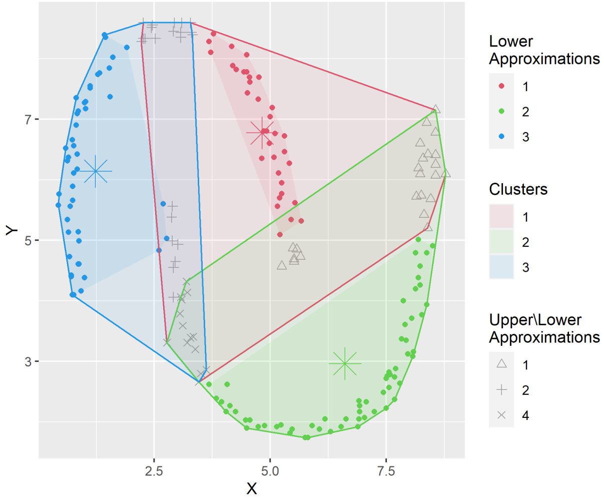

Figure 3: Result of clustering Crescents set
(N=200, K=3, epsilon=2.0, weight.lower=0.9, iterations=12)

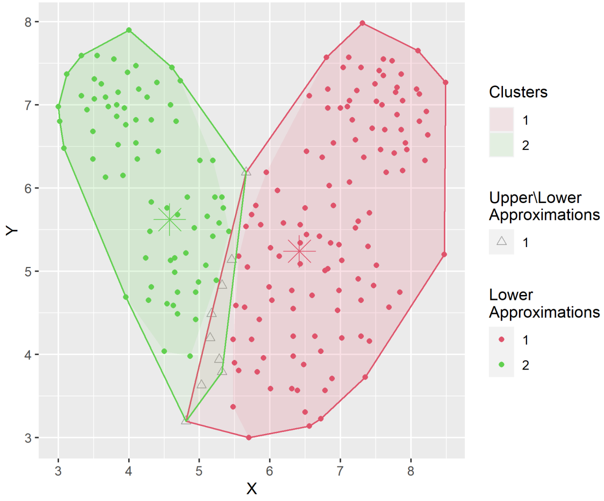

Figure 4: Result of clustering Mickey Mouse set
(N=200, K=2, epsilon=1.3, weight.lower=0.7, iterations=7)

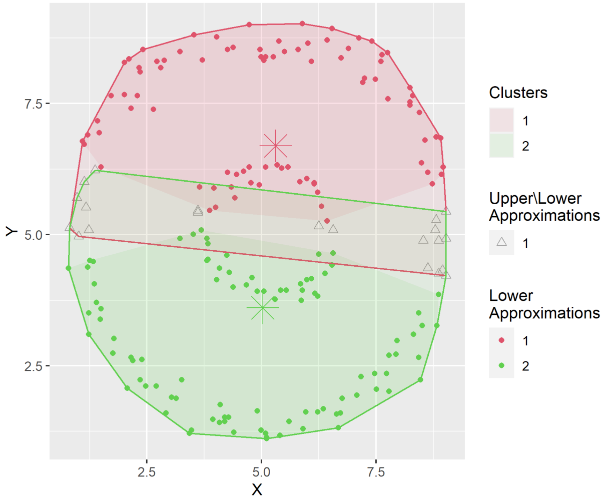

Figure 5: Result of clustering Circles set
(N=200, K=2, epsilon=1.3, weight.lower=0.7, iterations=8)

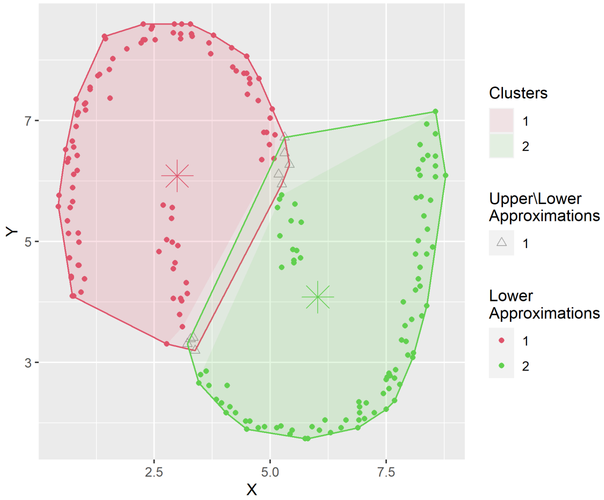

Figure 6: Result of clustering Crescents set
(N=200, K=2, epsilon=1.3, weight.lower=0.8, iterations=9)

4.2 Actual data sets

Clusters below were generated by RoughClusteringActualData.R using RoughKMeans.R Data set with different data 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

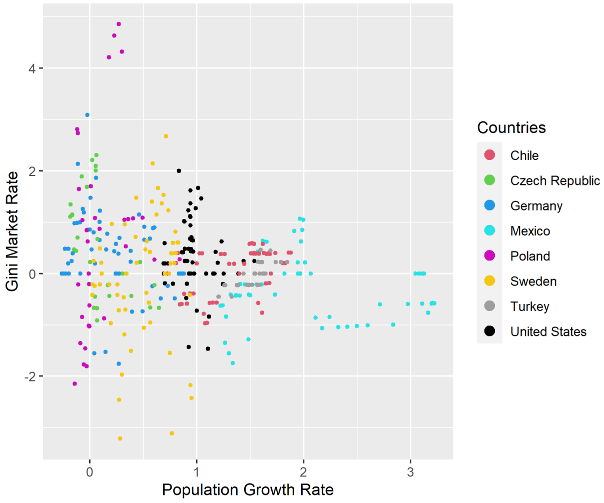

Figure 7: Scatterplot of Population Growth Rate and Gini Market Rate

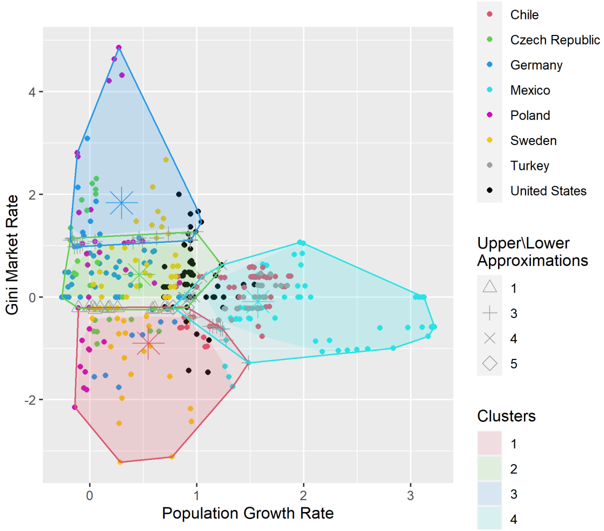

Figure 8: Result of clustering Population Growth Rate and Gini Market Rate set
(N=364, K=4, epsilon=1.5, weight.lower=0.7, iterations=18)

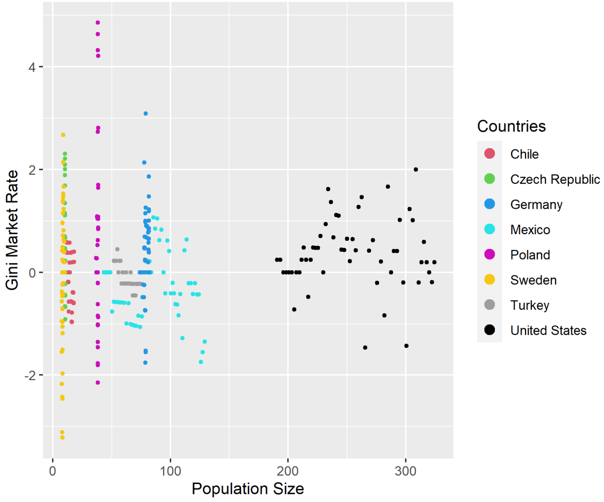

Figure 9: Scatterplot of Population Size and Gini Market Rate

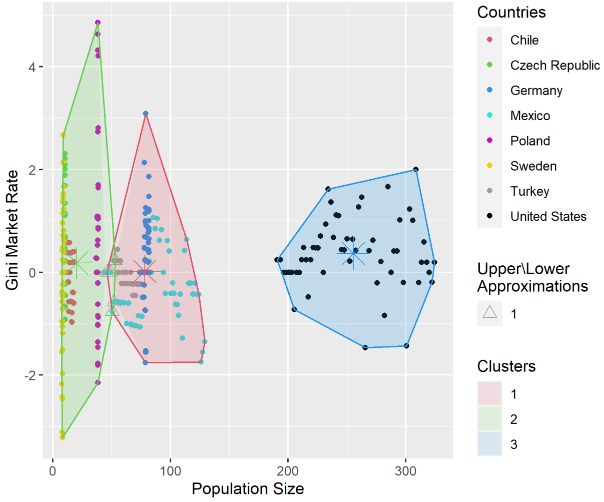

Figure 10: Result of clustering Population Size and Gini Market Rate set (N=364, K=3, epsilon=1.75, weight.lower=0.9, iterations=8)

References

[1] E. N. Sathishkumar. Rough k-means clustering algorithm. `https://www.slideshare.net/ENSathishkumar/rough-k-means-numerical-example`, 2018. [Online; accessed 19-November-2020].

[2] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356, 1982.

[3] A. Nowak-Brzezinska. Rough set theory in decision support systems. `http://zsi.tech.us.edu.pl/~nowak/bien/w2.pdf`. [Online; accessed 19-November-2020].