Optimal k-means clustering

Mattia Ravasio, Hamza Zerhouni

December 9, 2022

Problem statement

K-means clustering is often the go-to choice for clustering problems. The main idea is to find k centroids ($C_1, \ldots, C_k$) by minimizing the sum over each cluster of the sum of the square of the distance between the point and its centroid.

$$C_1, C_2, ..., C_k = \arg\min(\sum_{i=1}^{k} \sum_{x \in S_i} \|x - C_i\|)$$

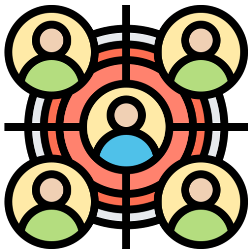
The random nature of the algorithm results in two big issues:

1. The algorithm has an **issue of stability**, and often different random seeds lead to different solutions
2. The algorithm has **no proof of optimality** when converged

We propose an MIO (mixed integer optimization) approach that focus on solving these two issues.

Why do we care?

Clustering has many applications:



Customer segmentation

Genetic data

Financial markets

Political campaigns

We need a **reliable method** that is guaranteed to **find a stable and provable optimal solution.**

The effect of such a method can impact all the different applications of clustering.

Proposed formulations

Manhattan Distance

$$
\begin{aligned}
\min_{x,\gamma,z,r,\mu,y} \quad & \sum_j \gamma_j \\
\text{s.t.} \quad & \sum_{d=1}^{D} y_{ijd} \leq r_{ij} && \forall i,j \\
& y_{ijd} \geq x_i^d - p_i^d && \forall i,j,d \\
& y_{ijd} \geq -(x_i^d - p_i^d) && \forall i,j,d \\
& \gamma_j \geq r_{ij} - \mu_{ij} && \forall i,j \\
& M(1 - z_{ij}) \geq \mu_{ij} && \forall i,j \\
& \sum_i z_{ij} = 1 && \forall j \\
& \mu, r, x, y \geq 0, z_{ij} \in \{0,1\}
\end{aligned}
$$

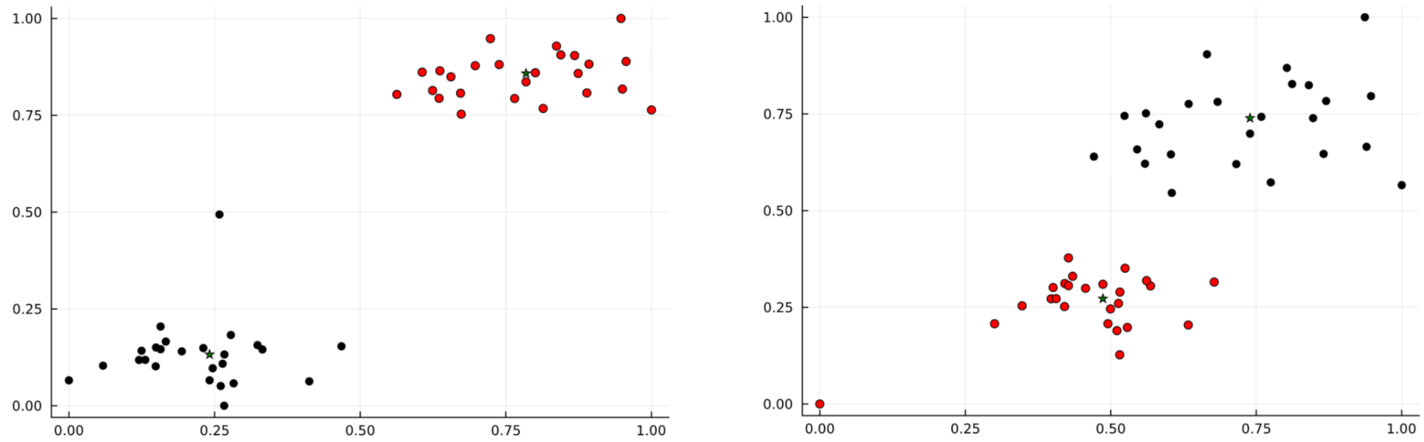
Euclidean Distance

$$
\begin{aligned}
\min_{x,\gamma,z,r,\mu,y} \quad & \sum_j \gamma_j \\
\text{s.t.} \quad & \sum_{d=1}^{D} y_{ijd} \leq r_{ij} && \forall i,j \\
& \|x_i^d - p_i^d\|_2 \leq y_{ijd} && \forall i,j,d \\
& \gamma_j \geq r_{ij} - \mu_{ij} && \forall i,j \\
& M(1 - z_{ij}) \geq \mu_{ij} && \forall i,j \\
& \sum_i z_{ij} = 1 && \forall j \\
& \mu, r, x, y \geq 0, z_{ij} \in \{0,1\}
\end{aligned}
$$

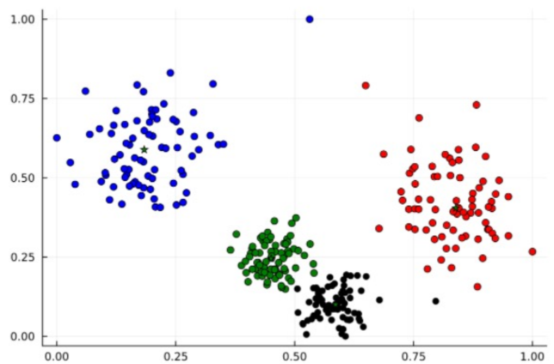
We also implemented a **Warm Start** solution that leverages k-means and an **outlier detection** system

Convergence issues

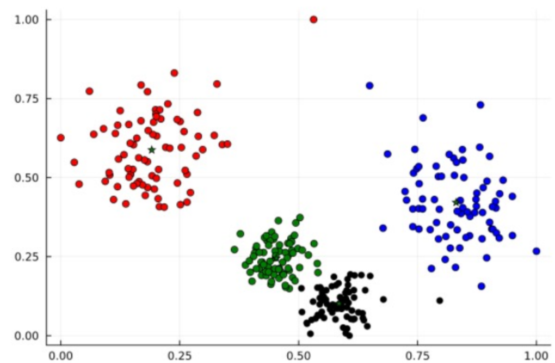
“Easier” vs “Harder” problem



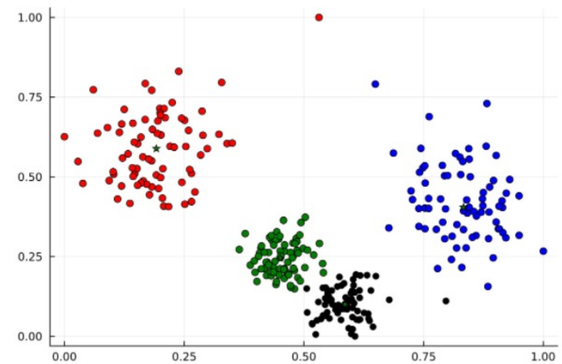
Solutions found under 5 minutes



Cold-Start Manhattan distance



K-means with random restart



Warm-start Manhattan distance

Results

Using **synthetic data** we evaluated the proposed formulations (using a **5 minutes limit** for the run time) and compared them to k-means with random restarts. As evaluation metric we used the **Silhouette score.**

Warm-start Euclidean distance

Centroids	Points 20	50	100	300	500
2	0.7492435	0.8124023	0.822535	0.8271316	0.8318493
3	0.7493559	0.7404216	0.7705543	0.7774862	0.7635197
4	0.6992221	0.6271979	0.6645417	0.6987365	0.6948316
5	0.4869698	0.5969384	0.5968344	0.6530551	0.6445701

Warm-start Manhattan distance

Centroids	Points 20	50	100	300	500
2	0.7560766	0.8124023	0.822535	0.8271316	0.8318493
3	0.7493559	0.7404216	0.7705543	0.7774862	0.7635197
4	0.6992221	0.6271979	0.6645417	0.6987365	0.6948316
5	0.4869698	0.5969384	0.5968344	0.6530551	0.6445701

Traditional k-means

Centroids	Points 20	50	100	300	500
2	0.7492435	0.8124023	0.822535	0.8271316	0.8318493
3	0.7493559	0.7404216	0.7705543	0.7774862	0.7635197
4	0.6992221	0.7525949	0.6645417	0.6987365	0.6376002
5	0.5236181	0.5475841	0.6163493	0.6535013	0.6527689

Even with a limit on the run time the **models tend to outperform** the traditional approach

We also evaluated the performance of models in **High dimensionality** problems. When in high dimension (1000) our models perform as k-means, but when the **dimensionality is reduced**, with an autoencoder, our models **outperform** k-means.

Dimensionality reduction

	K-means	Euclidean distance	Manhattan distance
Before dimensionality reduction	0.2388746	0.2388746	0.2388746
After dimensionality reduction	0.3954584	0.4036549	0.3996829

Dataset with 10,000 in dimension 1,000