UPPSALA
UNIVERSITET

Comparison of initialization methods of K-means clustering for small data

Liam Tabibzadeh

Bachelor's thesis in Statistics

Advisor
Philip Fowler

2022

Abstract

Clustering of observations into groups arises as a fundamental challenge both in academia and industry. Many clustering algorithms exist, and the most widely used clustering algorithm, the K-means, notably suffers from sensitivity to initial allocation of cluster centers. Many heuristics and algorithms have been developed to find the best initial allocation, and this experimental study compares methods of initialization by measuring how well the initialization methods perform on simulated, small datasets, through various performance criterion. The results, with a strong margin, show that using the output clusters of a Hierarchical clustering is the best initialization method. Moreover, the most popular methods, Random partitioning and KMeans++, perform poorly. Although the experimental setup may favour some initialization methods over others, the applied researchers are recommended to perform a Hierarchical clustering as an initialization of the K-means algorithm.

Acknowledgements

I thank everyone, including my parents and brother, for the support I have received throughout my life. I thank my advisor Philip Fowler for the initial idea and the subsequent guidance, making this work possible.

Contents

1. Introduction

Cluster analysis refers to the task of grouping a set of observations into a set of clusters, where the observations in the same cluster are similar to each other, and dissimilar to observations in other clusters. Such grouping is an important aspect of data analysis, as it enables the scientist to gain insight into data and groups of observations in the data, while also being widely applied in industry and academia.

Many approaches have been proposed for solving such task, and these approaches can be broadly classified into partitional clustering, hierarchical clustering, and density based clustering. For a thorough explanation of the different clustering methods, see [8]. Within the partitional approaches, one finds the hard-partitioning methods called the K-means algorithms. Most frequently, Lloyd's algorithm [8] is used for hard partitioning of clusters. Therefore, this method has been called "the" K-means algorithm.

More formally, the K-means algorithm has the goal to partition data $\mathbb{X} \in \mathbb{R}^{\mathbb{V}}$, where V denotes the number of variables, into $\mathbb{S} = \{S_1, S_2, .., S_K\}$ clusters, where $\cup_{i=1}^{K} S_i = \mathbb{X}$, $S_i \cap S_j = \emptyset$ for $1 \leq i \neq j \leq K$. Hence, there are K clusters. This is achieved by minimizing the sum of squared distances between every observation and its nearest cluster center. This can be formalized as the minimization of the loss function defined by:

$$\sum_{i=1}^{K} \sum_{x_j \in S_i} ||x_j - c_i||^2 \tag{1.1}$$

Where $c_i = 1/|S_i| \sum_{x_j \in S_i} x_j$ is the center of cluster S_i, $|S_i|$ is the cardinality of S_i, and $||.||$ is the euclidean norm.

The K-means clustering algorithm requires K number of clusters to be specified beforehand, with the centers of each cluster initialized by some heuristic. The algorithm then iteratively reassigns each observation $x_1, x_2, ..x_N$ to a center $c_1, c_2, .., c_K$ where the cluster assignment of observation x_i to center c_j is chosen such that $d(x_i, c_j) < d(x_i, c_m)$, for all $1 \leq j \neq m \leq K$, where $d(a, b)$ denotes the euclidean distance between points a and b. Consequently, the centers are recalculated as the mean of all points that are allocated to the center. These steps are iterated until no observations change allocation to center, or until some predefined criteria has been met, such as a limited number of iterations.

The K-means class of algorithms are popular as they are easy to understand, implemented in most data analysis software, and can easily be modified with regards to initialization of centers, use of distance function and criteria for termination. Therefore, they are the most widely used algorithms for solving clustering tasks [6]. However there are disadvantages; using the euclidean norm as the distance metric, the shapes of the final clusters are hyperspherical, while real data seldom is. Moreover, this distance metric is sensitive to outliers. Other distance functions such as mahalanobis distance can detect hyperellipsoidal clusters, or City-block to neglect outliers [1]. Moreover, due to the non-convex property of equation 1.1, the algorithm converges to local minima, based on the initial partition of cluster centers. Moreover, the choice of the cluster centers is important, since a cluster center may remain stationary if its distance to observations is too large [3]. Furthermore, a poor choice may result in near-empty clusters, long convergence time, and poor local minima [1]. On the other hand, correct initial partitioning leads to fast convergence by reassigning cluster membership of observations more often, which ultimately leads to good final clusters. Therefore, the choice of the initial centers is an important aspect of the K-means algorithm.

There have been extensive research into the K-means algorithm, and many heuristics and algorithms have been developed for the initial allocation of centers. Previous research has attempted to systematically describe and compare initialization methods for the K-means algorithm, while restricting the comparison to initialization methods with linear-time complexity with respect to number of observations. Celebi et al.

[1] have described extensively many initialization methods available in the literature and have performed experiments to compare a subset of the initialization methods described, evaluating each method on both real data and simulated data. The data varied with respect to complexity of clusters, size, number of variables, and number of clusters. They used many performance criterion to compare the initialization methods and found that some methods are superior with respect to the type of data. Additionally, they categorized the initialization methods based on computational time complexity, and chose to include eight initialization methods in their study based on such criteria. They argued that since the K-means algorithm has linear time-complexity with respect to the number of observations, the initialization method should have no higher time complexity.

Fränti and Sieranoja [3] have similarly described and compared several initialization methods, restricting themselves to initialization methods with at most log-linear time complexity in observations. They evaluated the initialization methods on real data sets, with various sizes, number of variables, number of clusters, and overlap of clusters. They noted that for high cluster overlap, the choice of initialization method is less important. As the previous research has been restricted to initialization methods of linear time-complexity in number of observations, it is needed to include in the analysis initialization methods with higher time-complexity, as much of scientific research, including the medical and social sciences, are driven by data of smaller sizes, due to ethical concerns, and hence it is important to investigate the performance of those methods. The higher time complexity methods are appropriate for smaller data, as the time complexity may not be important when the data is small. The researcher will be able to execute the K-means algorithm initialized with those methods in a similar time compared to the K-means algorithm initialized with lower time complexity methods, if the data is small.

The aim of this simulation study is to compare initialization methods of higher computational time complexity, on data of small size: $N \leq 1000$, $V \leq 9$, $K \leq 8$, along with the default initialization methods in K-means clustering algorithm in the statistical software. N denotes the sample size.

2. Theory

2.1 Initialization methods

The following are the initialization methods compared in this study. Moreover, figure 2.1 illustrates how each initialization method could partition the centers, for a certain data.

Random partitioning [8]: This method chooses, among the N observations, K observations randomly as the centers. This is the default initialization method of the K-means algorithm in the function *kmeans* in R. This method may produce different initial centers on the same data, and therefore is less reliable.

Hierarchical clustering method [7]: Hierarchical clustering methods approach the task of grouping observations by two different ways; either agglomerative, where each observation starts in its own cluster and clusters are merged on the basis of distance metrics and linkage functions, until all observations are in the same cluster, or they strive to cluster observations by so called divisive methods, where all observations start in one cluster, and splits occur until each observation is in its own cluster. Both methods provide a dendrogram that allows one to see every merging or splitting step of the algorithm, and finally decide upon a fixed number of clusters. In this study of comparison of initialization methods, the location of the cluster centers of an agglomerative hierarchical clustering algorithm, with K number of clusters, is used for the initial centers of the K-means algorithm.

Mean method [4]: This method sorts the observations according to the distance of each observation to the mean of the observations, $\bar{\mathbb{X}}$, and chooses the i'th, $i \in$

$\{1, 2, .., K\}$, center to be the $(1 + (i - 1)N/K)$ ordered observation.

Variance method [2]: This method sorts the observations according to the value in the variable with the highest variance, then partitions the observations into K disjoint intervals, by the order of the observations, and chooses the median of each interval as a cluster center. This method is effective if one of the variables has high variance relative to the variance of the other variables.

K-means++ [13]: This method chooses one of the observations randomly as the first cluster center, then chooses the i'th $i \in \{2, 3, , .., K\}$ center to be $x' \in \mathbb{X}$ with probability $md(x')/\sum_{j=1}^{N} md(x_j)$ where $md(x)$ denotes the minimum distance from a point x to a previously selected center. Hence, the probability of choosing an observation as the next cluster center is higher the further away the previous cluster centers are to that observation. This method is the default initialization method in the K-means clustering algorithm in the Scikit-learn package in Python as well as in Matlab function *Kmeans*. This method may produce different initial centers on the same data, and therefore is less reliable.

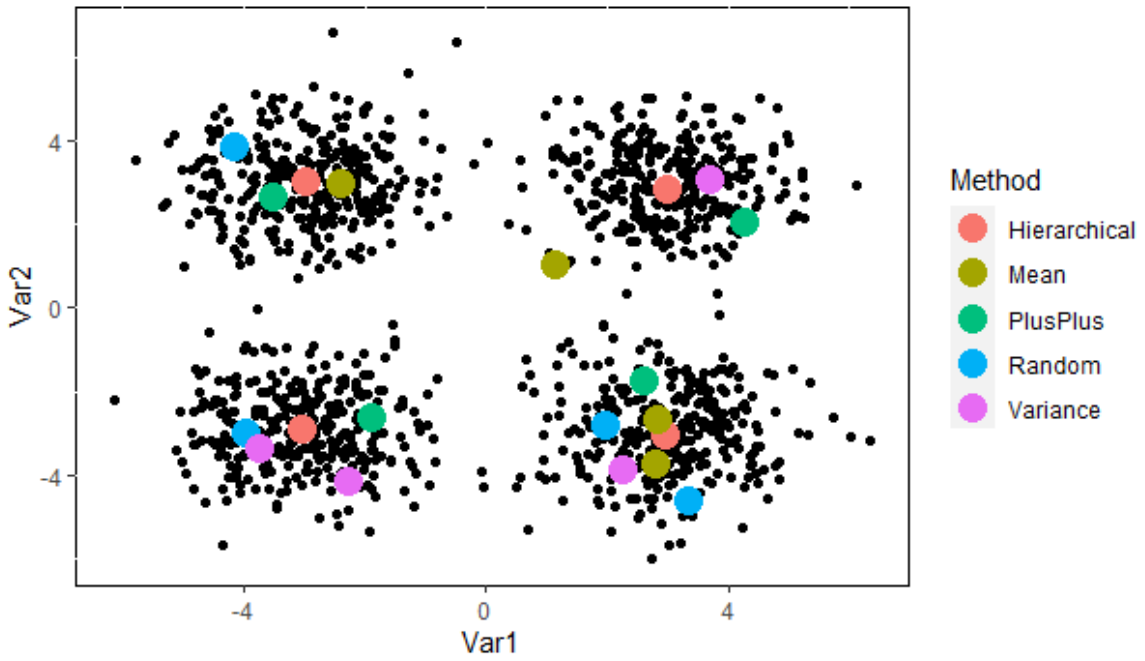


Figure 2.1: Initialization methods

2.2 Performance criterion

In this section, we describe the performance criterion used to compare the initialization methods. Each performance criteria is applied to each initialization method, and figure 2.2 illustrates this procedure.

SSE after allocation: The values of the loss function in Equation 1.1, after the centers have been allocated by each initialization method, are compared in order to obtain a measure of how initialization methods perform on their own as a clustering method, without the K-means algorithm. This criteria is denoted as *SSE1* in figure 2.2.

SSE after K-means algorithm: The value of the loss function in Equation 1.1, after convergence of k-means initialized with each method, is compared. Such a comparison will indicate how the initialization methods directly alter the performance of the K-means algorithm, which is arguably the most important purpose of an initialization method. This criteria is denoted as *SSE2* in figure 2.2.

Adj.Rand Index: The Adjusted Rand Index [5] provides a comparison of two cluster structures, where Adj.Rand Index $\in [0, 1]$. Since true cluster structure is known through simulation, this criteria compares the true cluster structure with the output of the K-Means algorithm. More precisely, the Rand Index measures the pairwise cluster membership of observations, and the adjusted Rand Index is adjusted for the randomness. For example, if two observations, truly belonging to the same cluster, are clustered into two different clusters, then the value of this index decreases.

Iterations: the number of iterations of the K-means algorithm initialized with the different methods are investigated.

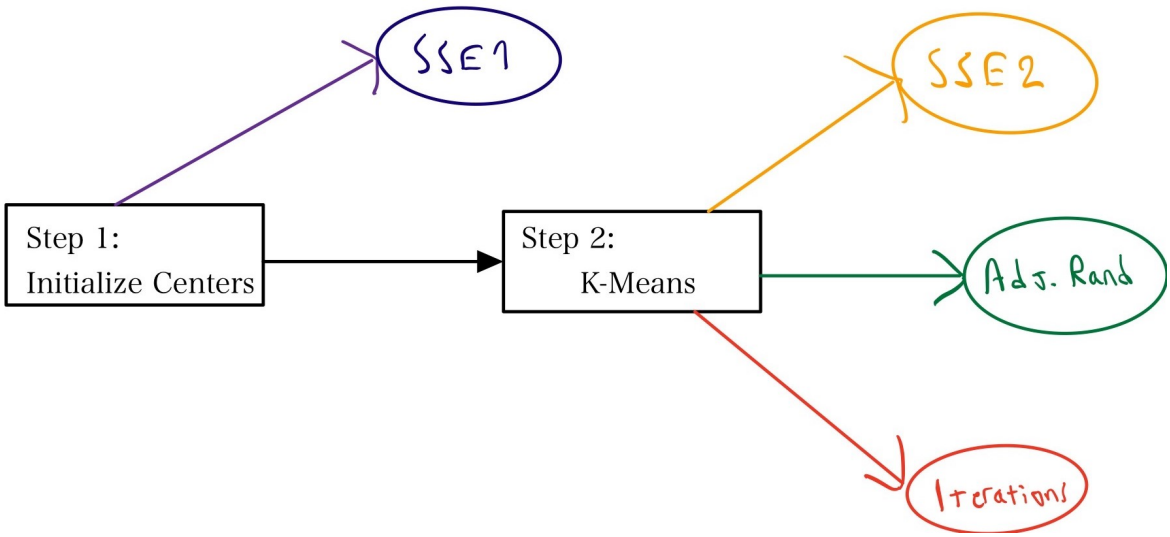


Figure 2.2: Experimental procedure

3. Experiments

3.1 Data simulation

In order to compare the performance of different initialization methods, the performance criterion is investigated on varying data. Consider m data-sets of varying size, number of dimensions, complexity, and number of clusters. Then let $\mathbb{X}_i$ denote the $i'th$ data-set $i = 1, 2, .., m$ sampled from multivariate normal distributions, where each distribution corresponds to one of the K_i clusters. Hence, the $j'th$ distribution of the $i'th$ data has parameters mean vector μ_{ij} and covariance matrix Σ_{ij}. Furthermore, let the number of variables of each distribution of the $i'th$ data-set be V_i, and the number of observations sampled from each distribution of the $i'th$ data-set be N_i/K_i, then $\mathbb{X}_i$ is sampled from K_i V_i-variate normal distributions with mean vectors $\{\mu_{i1}, \mu_{i2}, .., \mu_{iK_i}\}$ and covariance matrices $\{\Sigma_{i1}, \Sigma_{i2}, .., \Sigma_{iK_i}\}$. Hence, $\mathbb{X}_i$ is V_i-dimensional, with K_i clusters, and N_i observations.

The partitioning of $\mathbb{X}_i \in \mathbb{R}^{\mathbb{V}}$ into clusters $\mathbb{X}_i = \{S_1, S_2, .., S_K\}$ implies that the clusters are sampled from multivariate normal distributions with different parameters. The sampling procedure is described in tables 3.1 and 3.2, since the simulation is run under varying settings, the number of clusters and number of variables varies. The sampling of the data is done by sampling from K multivariate normal distributions, where table 3.1 indicates that the data is sampled from 8 distributions, and table 3.2 indicates that the data is sampled from 4 distributions. For different number of dimensions, the mean parameter of each distribution varies. N_i/K samples are drawn from the distributions of each cluster.

Table 3.1: Sampling from 8 distributions, with dimensions $V = \{3, 6, 9\}$

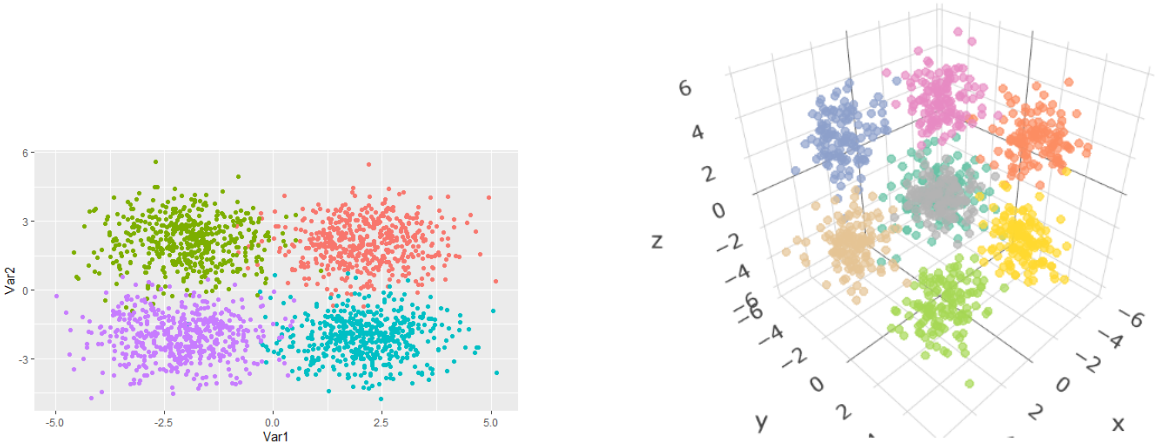
$\mathbb{X}_i = \{S_1, S_2, .., S_8\}$	$V = 9$	$V = 6$	$V = 3$
$S_1 \sim \mathcal{N}(c \cdot \mu_1, I)$	$\mu_1 = (1, 1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_1 = (1, 1, 1, 1, 1, 1)$	$\mu_1 = (1, 1, 1)$
$S_2 \sim \mathcal{N}(c \cdot \mu_2, I)$	$\mu_2 = (-1, 1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_2 = (-1, 1, 1, 1, 1, 1)$	$\mu_2 = (-1, 1, 1)$
$S_3 \sim \mathcal{N}(c \cdot \mu_3, I)$	$\mu_3 = (1, -1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_3 = (1, -1, 1, 1, 1, 1)$	$\mu_3 = (1, -1, 1)$
$S_4 \sim \mathcal{N}(c \cdot \mu_4, I)$	$\mu_4 = (-1, -1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_4 = (-1, -1, 1, 1, 1, 1)$	$\mu_4 = (-1, -1, 1)$
$S_5 \sim \mathcal{N}(c \cdot \mu_5, I)$	$\mu_5 = (1, 1, -1, 1, 1, 1, 1, 1, 1)$	$\mu_5 = (1, 1, -1, 1, 1, 1)$	$\mu_5 = (1, 1, -1)$
$S_6 \sim \mathcal{N}(c \cdot \mu_6, I)$	$\mu_6 = (-1, 1, -1, 1, 1, 1, 1, 1, 1)$	$\mu_6 = (-1, 1, -1, 1, 1, 1)$	$\mu_6 = (-1, 1, -1)$
$S_7 \sim \mathcal{N}(c \cdot \mu_7, I)$	$\mu_7 = (1, -, 1, -1, 1, 1, 1, 1, 1)$	$\mu_7 = (1, -, 1, -1, 1, 1)$	$\mu_7 = (1, -1, -1)$
$S_8 \sim \mathcal{N}(c \cdot \mu_8, I)$	$\mu_8 = (-1, -1, -1, 1, 1, 1, 1, 1, 1)$	$\mu_8 = (-1, -1, -1, 1, 1, 1)$	$\mu_8 = (-1, -1, -1)$

Table 3.2: Sampling from 4 distributions, with dimensions $V = \{3, 6, 9\}$

$\mathbb{X}_i = \{S_1, S_2, S_3, S_4\}$	$V = 9$	$V = 6$	$V = 3$
$S_1 \sim \mathcal{N}(c \cdot \mu_1, I)$	$\mu_1 = (1, 1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_1 = (1, 1, 1, 1, 1, 1)$	$\mu_1 = (1, 1, 1)$
$S_2 \sim \mathcal{N}(c \cdot \mu_2, I)$	$\mu_2 = (-1, 1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_2 = (-1, 1, 1, 1, 1, 1)$	$\mu_2 = (-1, 1, 1)$
$S_3 \sim \mathcal{N}(c \cdot \mu_3, I)$	$\mu_3 = (1, -1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_3 = (1, -1, 1, 1, 1, 1)$	$\mu_3 = (1, -1, 1)$
$S_4 \sim \mathcal{N}(c \cdot \mu_4, I)$	$\mu_4 = (-1, -1, 1, 1, 1, 1, 1, 1, 1)$	$\mu_4 = (-1, -1, 1, 1, 1, 1)$	$\mu_4 = (-1, -1, 1)$

The variation of parameters of each distribution, more precisely the variation of the scaling term c of the mean parameters, corresponds to the variation in overlap of the distributions, and thereof to the variation in overlap of clusters. Therefore, by varying c, one can sample clusters that overlap to an arbitrary degree. Such a measure is interesting since different initialization methods may perform better than others for different degrees of overlap, and real-life datasets vary in overlap. One can see, in table 3.1, that the parameters corresponding to the distribution of each cluster is chosen such that there is a systematic partitioning of the clusters, which is achieved by sampling each cluster from a multivariate normal distribution centered at a unique orthant (an orthant is the generalization of a two dimensional quadrant, or three dimensional octant). More precisely, if there are V variables, then the data is V-dimensional, and there exists 2^V orthants. Denote each orthant $\{\epsilon_1, \epsilon_2, .., \epsilon_{2^V}\}$. Then one can sample observations for the $j'th$ cluster in such a way that the observations lie in the $j'th$ orthant, $1 \leq j \leq 2^V$, by sensibly choosing each pair of mean vector and covariance matrix $\{\mu_{ij}, \Sigma_{ij}\}$ where $j = 1, ..K,\ i = 1, ..m$ of the $j'th$ multivariate normal distribution. Moreover, one can vary the distance of the observations to the origin

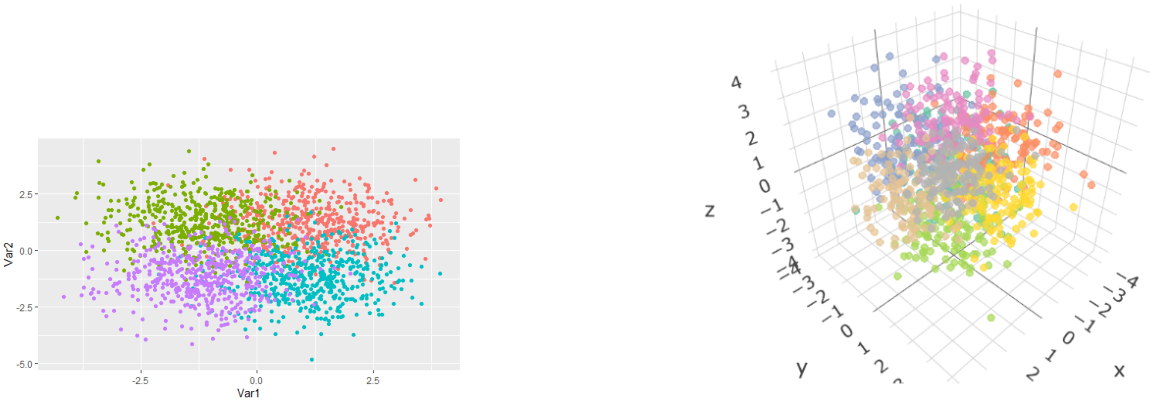
by varying the magnitude of the parameters, by varying the constant c. Using this procedure, along with the value of the adjusted rand index of K-means initialized with the true cluster centers ($\phi \in [0, 1]$), one can achieve a categorization of the clustering complexity of data $\mathbb{X}_i$, which we denote $\omega = \{low, high\}$, as it is interesting to compare the performance of the initialization methods for varying cluster complexity, as data in real-life has varying clustering complexity. Using these two heuristics described above, we say that a data has *low* cluster complexity if $c \geq 1.7$ *and* $\phi \geq 0.6$, and *high* cluster complexity if $c \leq 1.3$ *and* $\phi < 0.6$. Thus the cluster Complexity statistic ω measures the level of overlap in the clusters and the capacity of the K-means algorithm to detect the overlap. Figures 3.3 and 3.4 show two data-sets where the cluster complexity is *low* and *high* respectively.



(a) 4 Clusters, 2 variables

(b) 8 Clusters, 3 variables

Figure 3.3: Low complexity



(a) 4 Clusters, 2 variables

(b) 8 Clusters, 3 variables

Figure 3.4: High complexity

To mimic the variation in real-world data, we vary the sample size $N = \{160, 504, 1000\}$, the number of variables $V = \{3, 6, 9\}$, the number of clusters $K = \{4, 8\}$ and cluster complexity $\omega = \{low, high\}$, the latter is achieved by choosing $c = 1.2$ for $\omega = high$ and $c = 2$ for $\omega = low$. The K-means algorithm, with every initialization method, is applied to the data, and performance criterion are computed. Additionally, in order to account for sampling variation, each data $\mathbb{X}_i$ is sampled 1000 times, and the mean of those 1000 data $\mathbb{X}_i$ is used to rank the initialization methods.

3.2 Technical details

Computer simulations are executed on a PC with Windows 10 Pro 64 bytes operating system, using software R, version 4.1.2 (2021-11-01), and Rstudio version 1.4.1717. Function kmeans() from Base R Package [11] was used for the K-means clustering algorithm, using the K-means algorithm of Hartigan and Wong [4], an alternative to Lloyd's algorithm. The hierarchical clustering initialization is performed using the hclust() function in R, and all other initialization methods were programmed in R for this simulation study. The performance criterion *SSE after K-means* and *Iterations* are used from the fitted method in the Kmeans() function in R, while the performance criterion *SSE after allocation* and *Adj.Rand index* were written in R as functions. All code used in this experiment is published on github [12].

4. Results

In the tables below, the initialization methods are denoted by acronyms, where H denotes the Hierarchical clustering initialization method, R denotes the random partitioning method, M denotes the Mean method, X denotes the variance method, and finally P denotes the Kmeans++ method. The results are reported in the form $ABCDE$, where the order is the ranking of the methods. In this example, method A performs better than method B, which performs better than method C, and so on.

Table 4.3 shows the ranking of the initialization methods based on 1000 simulations for data $\mathbb{X}_i$, where each entry corresponds to a unique set of levels of the sample size, the number of variables, the number of clusters, and the cluster complexity. Table 4.4 presents the mean and standard deviation, in parenthesis, of the initialization methods, based on 1000 simulations of data where $N = 1000,\ V = 6,\ K = 8, \omega =$ *high*, for each performance criteria. The choice of this specific sample size, number of variables, number of clusters, and cluster complexity, is done by the values of the means, standard deviations, and rankings being representative of values for other levels of those factors. In this table, one may obtain the magnitude of the differences in means of the initialization methods with respect to a specific performance criteria. Moreover, the performance criterion of *SSE after allocation* is denoted as *SSE1* and *SSE after K-means algorithm* is denoted as *SSE2*.

Table 4.3: Rankings based on 1000 simulations, with varying N, V, K, ω

$\omega = low$									
		$K = 4$				$K = 8$			
		SSE1	SSE2	Adj.Rand	Iterations	SSE1	SSE2	Adj.Rand	Iterations
N=160	$V = 3$	$HPXMR$	$HXPMR$	$HXPMR$	$HXPRM$	$HPXMR$	$HXPMR$	$HXPMR$	$HPXRM$
	$V = 6$	$HMXPR$	$HXPRM$	$HXPRM$	$HXPRM$	$HPMXR$	$HXRPM$	$HXRPM$	$HPXRM$
	$V = 9$	$HMXPR$	$HXPMR$	$HXPMR$	$HXPRM$	$HMXPR$	$HXMRP$	$HXMRP$	$HXPMR$
		$K = 4$				$K = 8$			
		SSE1	SSE2	Adj.Rand	Iterations	SSE1	SSE2	Adj.Rand	Iterations
N=504	$V = 3$	$HPXMR$	$HPRXM$	$HPRXM$	$HPXRM$	$HPXMR$	$HXPMR$	$HXPMR$	$HPXMR$
	$V = 6$	$HMXPR$	$XHPRM$	$XHPRM$	$HXPRM$	$HPMXR$	$HXPMR$	$HXPMR$	$HXPRM$
	$V = 9$	$HMXPR$	$XHMRP$	$XHMRP$	$HXPRM$	$HMXPR$	$HXPMR$	$HXPMR$	$HXPRM$
		$K = 4$				$K = 8$			
		SSE1	SSE2	Adj.Rand	Iterations	SSE1	SSE2	Adj.Rand	Iterations
N=1000	$V = 3$	$HPXMR$	$RXHMP$	$MHRXP$	$HPXRM$	$HPXMR$	$HXPMR$	$HXPMR$	$HPXMR$
	$V = 6$	$HMXPR$	$HXRMP$	$RHXMP$	$HXPRM$	$HMPXR$	$HXMRP$	$HXMRP$	$HXPRM$
	$V = 9$	$HMXPR$	$RMHXP$	$HMRXP$	$HXPRM$	$HMXPR$	$HXPRM$	$HXPRM$	$HXPRM$
$\omega = high$									
		$K = 4$				$K = 8$			
		SSE1	SSE2	Adj.Rand	Iterations	SSE1	SSE2	Adj.Rand	Iterations
N=160	$V = 3$	$HMXPR$	$HMXPR$	$HXMRP$	$HXPRM$	$HPXMR$	$HMXPR$	$HRMXP$	$HPXRM$
	$V = 6$	$HMXPR$	$HXRMP$	$XHMRP$	$HXPMR$	$HMXPR$	$HXMPR$	$HPMXR$	$HPRXM$
	$V = 9$	$HMXRP$	$XHPRM$	$XPHRM$	$HXPRM$	$HMXRP$	$XHMRP$	$XMHRP$	$HXPRM$
		$K = 4$				$K = 8$			
		SSE1	SSE2	Adj.Rand	Iterations	SSE1	SSE2	Adj.Rand	Iterations
N=504	$V = 3$	$HMXPR$	$HXMPR$	$HXMPR$	$HXPRM$	$HPXMR$	$HXRPM$	$HRPXM$	$HPXRM$
	$V = 6$	$HMXPR$	$XPMHR$	$XPMHR$	$HXPRM$	$HMXRP$	$HXPMR$	$HXPMR$	$HXMPR$
	$V = 9$	$HMXRP$	$XHRPM$	$XPHRM$	$HXPRM$	$HMXRP$	$HMXPR$	$XPHRM$	$HXPRM$
		$K = 4$				$K = 8$			
		SSE1	SSE2	Adj.Rand	Iterations	SSE1	SSE2	Adj.Rand	Iterations
N=1000	$V = 3$	$HMXPR$	$MHXRP$	$HMRXP$	$HXPMR$	$HPXMR$	$HPXRM$	$HXPMR$	$HPXRM$
	$V = 6$	$HMXRP$	$PXHRM$	$XPMRH$	$HXPMR$	$HMXRP$	$HXRMP$	$HXMRP$	$HPXMR$
	$V = 9$	$HMXRP$	$XRHPM$	$XRHMP$	$HXPMR$	$HMXRP$	$HXPRM$	$HPXMR$	$HXPRM$

Table 4.4: Mean and std. deviation of 1000 simulations

$N = 1000,\ V = 6,\ K = 8,\ \omega = high$				
Method	SSE1	SSE2	Adj.Rand	Iterations
R	8040.0(759.1)	5093.0(99.0)	0.3729(0.03)	5.51(1.16)
H	5467.0(144.5)	5088.3(98.6)	0.3767(0.04)	4.53(1.06)
M	7482.5(391.8)	5093.1(96.0)	0.3730(0.04)	5.50(1.13)
V	7828.2(707.6)	5090.9(96.5)	0.3739(0.04)	5.42(1.13)
P	8040.2(688.0)	5093.7(99.5)	0.3716(0.04)	5.40(1.11)

Overall, one may see that the rankings, averaged over all data and over all performance criterion, indicate that the hierarchical method outperforms all other methods. Moreover, the Random partitioning method and the KMeans++ method perform poorly.

5. Discussion

The aim of this study is to compare initialization methods for the K-means clustering algorithm for small data. The initialization methods compared are Random partitioning, Hierarchical clustering, Mean, Variance, and K-means++ methods, in a computer simulation study, simulating multiple datasets with the aim of mimicking the variation in real-world data. This simulation study may be replicated with the help of the code used in this thesis available on github [12]. The results can be provided upon request to the author. The performance criterion used are SSE after allocation of cluster centers *(SSE1)*, SSE after K-means algorithm (*SSE2*), the Adjusted Rand Index *(Adj.Rand)*, and finally the number of iterations of the K-means algorithm *(Iterations)*,

The overall, average, trend for all performance criterion indicates that the Hierarchical clustering initialization method outperforms the other methods, and the Random and KMeans++ perform poorly. Furthermore, one may investigate the rankings averaged over all data for a particular performance criteria, as each performance criteria intends to measure a particular aspect of the performance of an initialization method. Focusing on the average rankings for the performance criteria SSE after allocation of cluster centers, then similarly to the overall picture, the Hierarchical method performs the best. One may deduce such a ranking for this criteria, since the Hierarchical method allocates the cluster centers through the output of a Hierarchical clustering algorithm, while the other initialization methods may be viewed as heuristics rather than algorithms, and are not as developed with this respect. Moreover, the magnitude of the difference in the initialization methods for this performance criteria can be seen in Table 4.4, indicating that the Hierarchical method strongly outperforms the other methods. Next, the rankings for the performance criteria of SSE after K-means

algorithm also indicate that the Hierarchical method performs best, followed closely by the Variance method. However, the magnitude of the differences indicate that the initialization methods for this particular criteria differ only to a small extent. This performance criteria is indicative of the extent to which an initialization method affects the final partitioning, and may arguably be the most useful since the purpose of initialization methods is precisely that. Next, recall that the Adjusted Rand index compares the final pairwise cluster membership with the true pairwise cluster membership, for all pairs of observations. Such a measure is important since the researcher conducts clustering in order to find true underlying clusters in the data. The rankings for this criteria suggest, similarly to the previous criterion, that the Hierarchical method performs the best, followed by the Variance method, with large differences in magnitude between the Hierarchical method and all other methods, and also large differences between the worst performing method, Kmeans++, and all other methods. Lastly, the rankings for the performance criteria of number of iterations until convergence of the K-means algorithm also provides the same picture as the previous criteria, namely the superiority of the Hierarchical method, followed by the Variance method. The criteria of number of iterations may be of greater interest to researchers with larger data, as the computational time of each iteration depends on the size of the data. Moreover, one can note that the rankings do not change when varying the levels of only one of sample size, number of variables, number of clusters, or cluster complexity. In summary, based on these observations, an initial recommendation to the applied researcher is to use the Hierarchical clustering initialization method, and preferably avoid the methods Random and KMeans++.

Previous studies [1][3][10] have also conducted experiments of similar nature, with the goal of comparing initialization methods for larger data-sets than the data in this study. Each of these experimental studies should be considered unique, since the set of initialization methods compared are different. As far as our knowledge goes, this is the first study to include the three initialization methods of Hierarchical, Variance, and Mean. And although the other two popular methods, the Random and KMeans++ methods, have been included in previous research, it is the first time that their performance are compared, and ranked, with the other three methods in this study. As

it is the relative performance of methods that is important, the findings of the previous research regarding the Random and KMeans++ methods may be limited to their studies. Nonetheless, the previous studies seem to agree in that the popular Random method, which is the default initialization method in the function *Kmeans* in the base R package, along with the popular KMeans++ method, which is the default initialization method in the Scikit-learn [9] package in Python as well as in Matlab function *Kmeans*, both perform moderate to poorly.

As the aim of this simulation study has been to compare initialization methods specifically for small data-sets, the number of observations has been chosen to be ≤ 1000, and the number of variables ≤ 9. Moreover, as some initialization methods have computational time-complexity, in number of observations, of $\mathcal{O}(n^2)$ [1], the results of this study may be of limited use for medium to large data. More precisely, the Hierarchical clustering algorithms have a time complexity of $\mathcal{O}(n^2)$ or higher, the Mean and Variance initialization methods have $\mathcal{O}(n \cdot log(n))$ time complexity, and lastly, the Random and KMeans++ methods have $\mathcal{O}(n)$ time complexity [1]. Additionally, the performance of the initialization methods may vary significantly for medium to large data. It is also worth mentioning that, from a practical point of view, some methods are to be favoured over others; the Hierarchical method requires the applied researcher to conduct an agglomerative Hierarchical clustering, select K clusters, find the cluster centers, and finally use those as the initial centers for K-means. Meanwhile, the Random and Kmeans++ methods are both implemented as default or optional initialization methods in functions in popular data analysis software. Lastly, the methods Variance and Mean are, to the best of our knowledge, not implemented in those functions, and have to be implemented by the applied researcher. Consideration of these limitations of some initialization methods should be taken by the applied researcher.

In a simulation study, the evaluation of initialization methods is most comprehensive when the data varies across as many factors as possible, so that the strength and weaknesses of the initialization methods are highlighted and displayed. Therefore, the data generating process is important. In this simulation study, data has been sampled from K equally sized clusters each from a multivariate normal distribution.

This procedure has its advantages; this accurately reflects the statistical distribution of a large number of data in the real world, and the clusters may overlap to an arbitrary degree by the choice of parameters of each distribution. Nevertheless, there exists major flaws with this choice of data generation; as the natural clusters in the real world seldom are of equal sizes. Moreover, the choice of sampling from normal distributions implies that there is a lack of outliers in the data. This is also a limitation, as the real world data is more likely to include outliers, and the initialization methods that are sensitive to outliers may therefore have over-performed in this study. Lastly, there has been a further restriction by setting the correlation between the variables to zero. This results in spherically shaped clusters. This favours clustering algorithm that minimizes the loss function in 1.1 [3], and may be beneficial for some initialization methods. In future studies, these flaws may be remedied by including data with unequal cluster sizes, generating outliers in the data, and including non-spherically shaped clusters.

The results of this simulation study further raise an interesting question; is it more favorable to perform cluster analysis with K-means using the Hierarchical clustering method as initialization or to perform cluster analysis though the Hierarchical clustering method on its own? And is it worth it? A hint to the first question may be obtained in this study, since the value of the loss function in equation 1.1 is less after the two clustering algorithm are applied together than when only Hierarchical clustering is applied (*SSE2* compared to *SSE1*). By including the Adj. Rand Index directly after the Hierarchical method, future studies can investigate this question further.

In conclusion, the Hierarchical clustering method performs well as an initialization method, while the Random and KMeans++ methods perform poorly. Moreover, practical consideration of implementation time need to be taken into account by the applied researcher. This simulation study, whilst employing an experimental design, has several shortcomings; the cluster shapes are limited to spherical, the cluster sizes are equal, and the non-existence of outliers. Despite these, due to the strong performance of the Hierarchical clustering method as an initialization method, the recommendation to applied researchers is to initialize the cluster centers through the Hierarchical clustering method.

References

[1] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. "A comparative study of efficient initialization methods for the k-means clustering algorithm". *Expert systems with applications* 40.1 (2013), pp. 200–210.

[2] Moth'D Belal Al-Daoud. "A new algorithm for cluster initialization". *WEC'05: The Second World Enformatika Conference*. 2005.

[3] Pasi Fränti and Sami Sieranoja. "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognition* 93 (2019), pp. 95–112.

[4] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm". *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[5] Lawrence Hubert and Phipps Arabie. "Comparing partitions". *Journal of classification* 2.1 (1985), pp. 193–218.

[6] Anil K Jain. "Data clustering: 50 years beyond K-means". *Pattern recognition letters* 31.8 (2010), pp. 651–666.

[7] Godfrey N Lance and William Thomas Williams. "A general theory of classificatory sorting strategies: II. Clustering systems". *The computer journal* 10.3 (1967), pp. 271–277.

[8] S. Lloyd. "Least squares quantization in PCM". *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[9] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[10] José M Pena, Jose Antonio Lozano, and Pedro Larranaga. "An empirical comparison of four initialization methods for the k-means algorithm". *Pattern recognition letters* 20.10 (1999), pp. 1027–1040.

[11] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: https://www.R-project.org/.

[12] Liam Tabibzadeh. *Comparison of initialization methods for K-means clustering for small data*. URL: https://github.com/lita4579/Initialization-methods-K-Means-clustering.git.

[13] Sergei Vassilvitskii and David Arthur. "k-means++: The advantages of careful seeding". *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 2006, pp. 1027–1035.