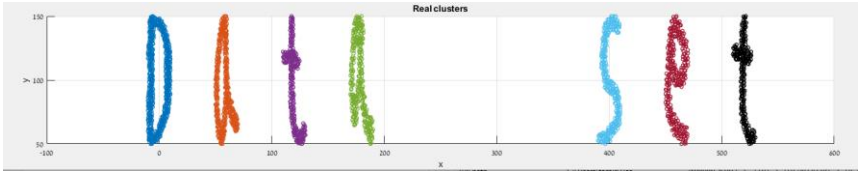
1.Introduction

Unsupervised learning is a set of statistical tools for scenarios in which there is only a set of features and no labels. Therefore, we cannot make predictions since there are no associated responses to each observation. Instead, we are interested in finding way to visualize data or in discovering subgroups of similar observations.

Unsupervised learning tends to be more challenging because there is no clear objective for the analysis, and it is often subjective. Additionally, it is hard to assess if the obtained results are good, since there is no accepted mechanism for performing cross-validation or validating results on an independent dataset, because we do not know the true answer.

Clustering refers to a broad set of techniques for finding subgroups or clusters in a dataset. This helps us partition observations into distinct groups so that each group contains observations that are like each other. There are many clustering methods and we will Introduced several of them. My goal is to shift from theoretical point of view to a practical manner by implementing and examining several algorithms.



2. Exercise '0'-creating the data set

2.1 The Data set features

I use different Data Sets for different simulations:

(1.) 4 Gaussian without an overlap In 3D.

(2.) 4 Gaussian with partial overlap between 2 Gaussian In 3D.

(3.) 4 Gaussian with partial overlap 3D.

(4.) 5 Gaussian without an overlap In 5D.

(5.) 5 Gaussian with large covariances In 5D.

(6.) 5 Gaussian without large covariances and few samples In 10D..

(7.) 3 Spheres In sphere without an overlap In 3D.

(8.) Sphere In sphere In sphere without an overlap and close centers In 3D.

(9.) 3 Spheres without an overlap In 3D.

(10.) 3 Shapes without an overlap In 3D.

(11.) "Smile" shape without an overlap In 3D.

(12.) 3 Coils without an overlap and same centers In 3D.

(13.) 3 Coils with an overlap and same centers In 3D.

(14.) 3 "Capacitors" dataset In 3D.

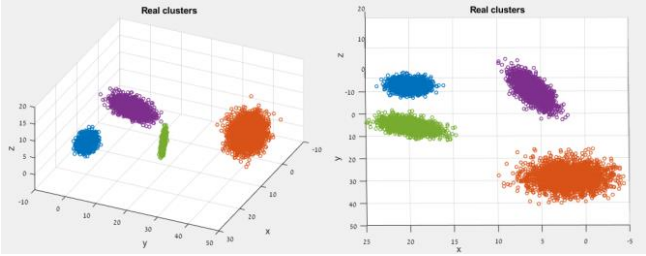
(15.) 7 letters with linear distances without overlap In 3D.

(16.) 7 letters with circular shape without overlap In 3D.

2.2 Data Presentation

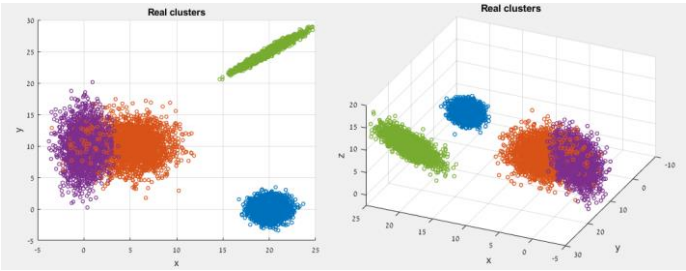
2.2.1. *10,000 samples from 4 Gaussian without an overlap In 3D*

X	*10000x3 double*
Labels	*10000x1 double*
n	10000
d	3
c	4
cluster_samples	[3000,2800,2200,2000]



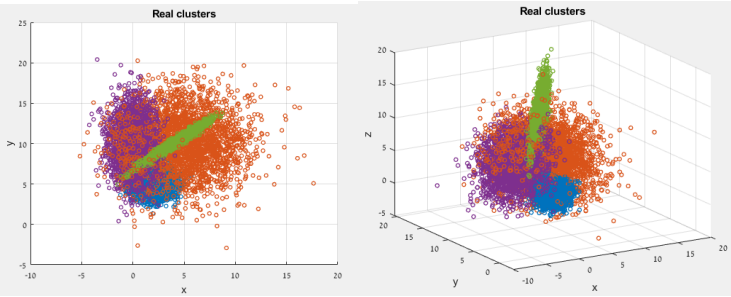
2.2.2 *10,000 samples from 4 Gaussian with an overlap between 2 Gaussians In 3D*

X	*10000x3 double*
Labels	*10000x1 double*
n	10000
d	3
c	4
cluster_samples	[3000,2800,2200,2000]



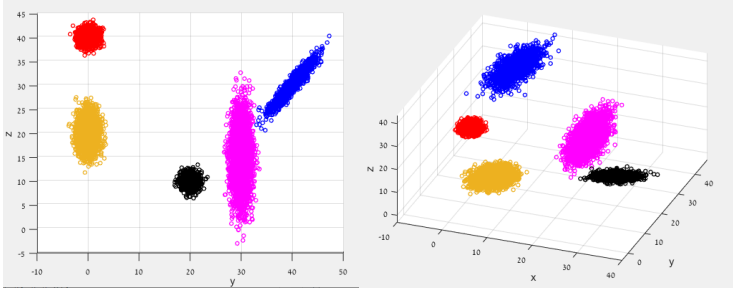
2.2.3 *10,000 samples from 4 Gaussian with an overlap between all Gaussians In 3D*

X	*10000x3 double*
Labels	*10000x1 double*
n	10000
d	3
c	4
cluster_samples	[3000,2800,2200,2000]



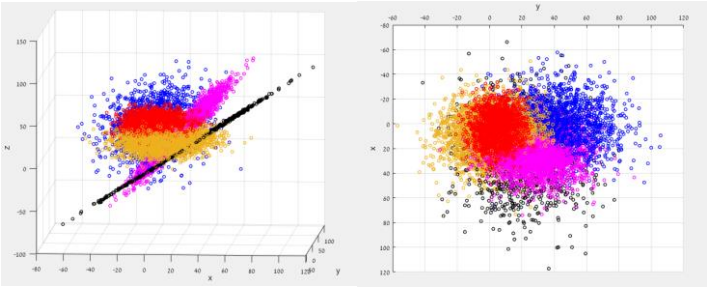
2.2.4 *12,500 samples from 5 Gaussian without an overlap In 5D*

X	*12500x5 double*
Labels	*12500x1 double*
n	12500
d	5
c	5
cluster_samples	[2500,1000,3500,4000,...



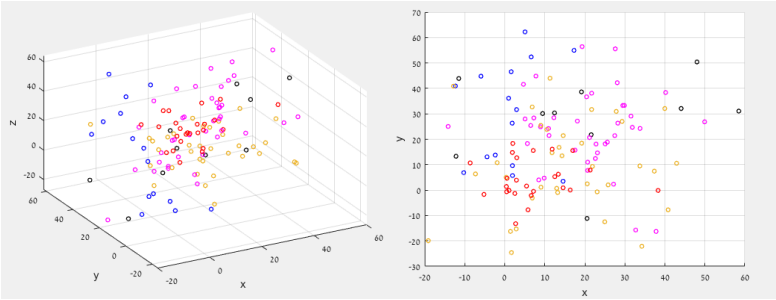
2.2.5 *12,500 samples from 5 Gaussian with large variance In 5D*

X	*12500x5 double*
Labels	*12500x1 double*
n	12500
d	5
c	5
cluster_samples	[2500,1000,3500,4000,...



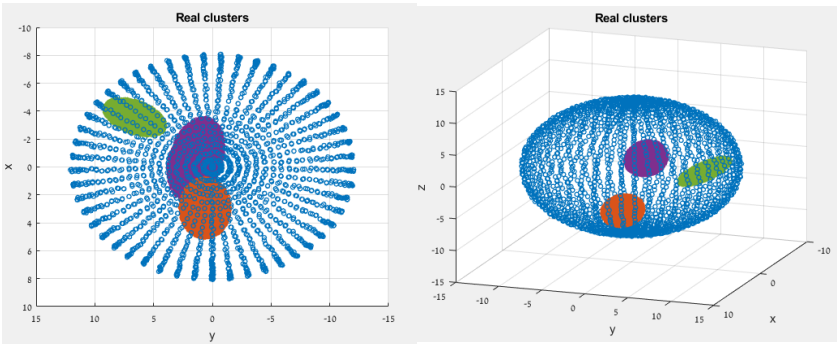
2.2.6 *125 samples from 5 Gaussian without an overlap and large covariance In 10D*

X	*125x10 double*
Labels	*125x1 double*
n	125
d	10
c	5
cluster_samples	[25,10,35,40,15]



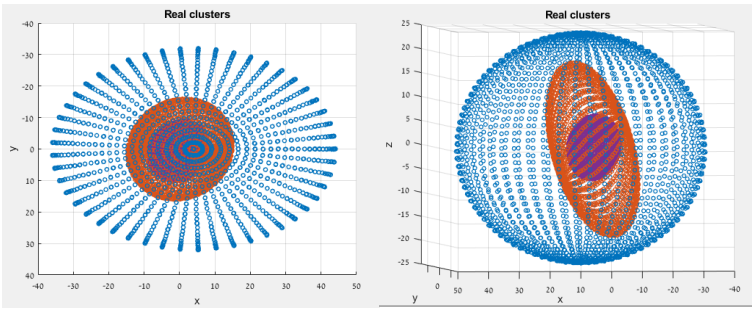
2.2.7 *10,000 samples from 3 Spheres in big sphere without an overlap In 3D*

X	*10000x3 double*
Labels	*10000x1 double*
n	10000
d	3
c	4
cluster_samples	[2500,2500,2500,2500]



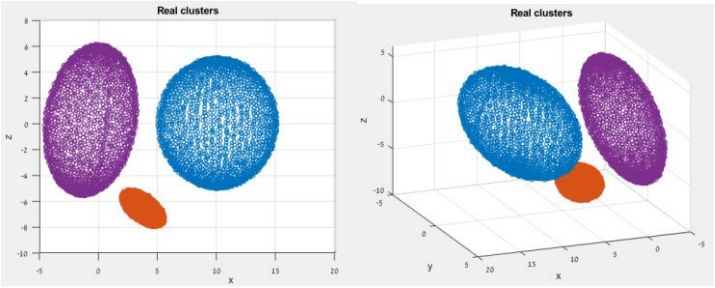
2.2.8 *7,500 samples from 3 spheres without an overlap and close centers In 3D*

X	*7500x3 double*
Labels	*7500x1 double*
n	7500
d	3
c	3
cluster_samples	[2500,2500,2500]



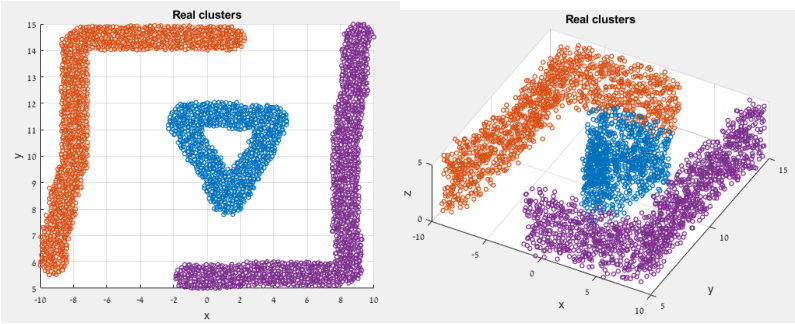
2.2.9 *7,500 samples from 3 Spheres without an overlap In 3D*

X	*7500x3 double*
Labels	*7500x1 double*
n	7500
d	3
c	3
cluster_samples	[2500,2500,2500]



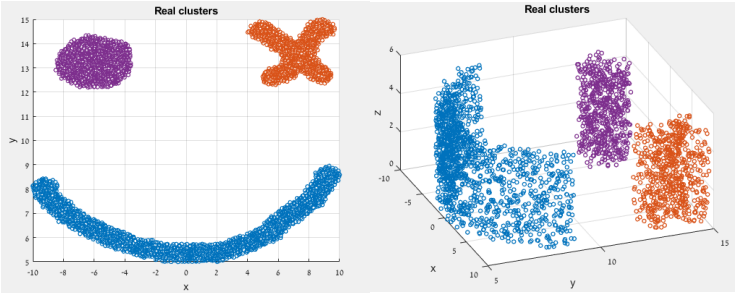
2.2.10 *3,504 samples from 3 shapes without an overlap In 3D*

X	*3504x3 double*
Labels	*3504x1 double*
n	3504
d	3
c	3



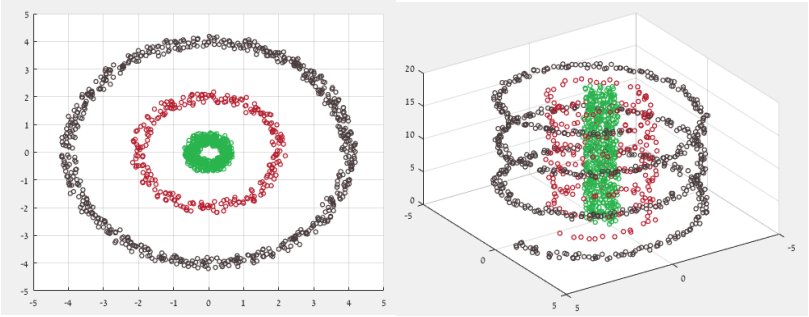
2.2.11 *2,521 samples from "smile" shape without an overlap In 3D*

X	*2521x3 double*
Labels	*2521x1 double*
n	2521
d	3
c	3



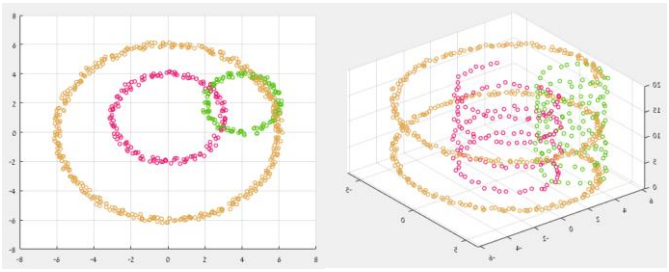
2.2.12 *1,763 samples from 3 coils without an overlap and same centers In 3D*

X	*1763x3 double*
Labels	*1763x1 double*
n	1763
d	3
c	3



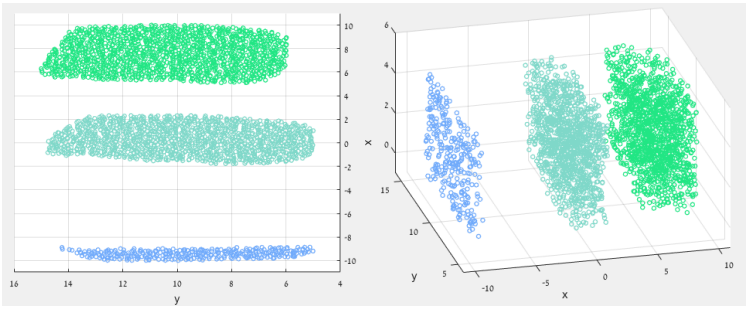
2.2.13 *1,363 samples from 3 coils with an overlap In 3D*

X	*1363x3 double*
Labels	*1363x1 double*
n	1363
d	3
c	3



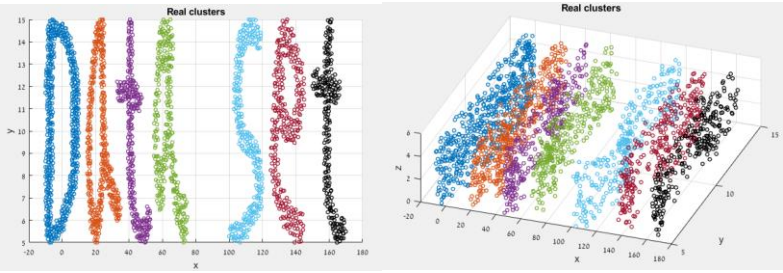
2.2.14 *2,874 samples from "Capacitor" shape without an overlap In 3D*

X	*2874x3 double*
Labels	*2874x1 double*
n	2874
d	3
c	3



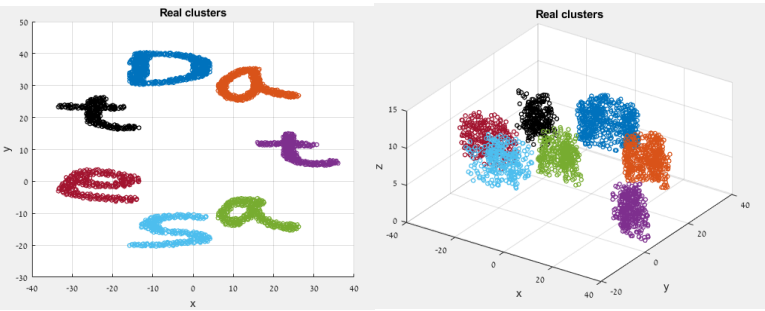
2.2.15 *2,494 samples from letters without an overlap In 3D*

X	*2494x3 double*
Labels	*2494x1 double*
n	2494
d	3
c	7



2.2.16 *2,494 samples from letters In circular shape without an overlap In 3D*

X	*2494x3 double*
Labels	*2494x1 double*
n	2494
d	3
c	7



3. Exercise '1'-Maximum Likelihood Estimation Algorithm

3.1 The primary objective of the algorithm

The MLE Algorithm objective Is to find the parameters of a Gaussian Mixture Model. In this algorithm there are 4 Pre-defined parameters:

1. *The number of Gaussians-c.*
2. *The mean of each Gaussian- μ_i.*
3. *The Covariance of each Gaussian-$\sum_i$.*
4. *The prior probability (or mixing parameter) of each Gaussian- $P(w_i)$ (i=1,2,3,…c).*

There are 3 types of MLE problem:

Case	μ_i	Σ_i	$P(\omega_i)$	c
1	?	×	×	×
2	?	?	?	×
3	?	?	?	?

1. *μ_i- is unknown.*
2. *μ_i, $\sum_i$ -are unknown.*
3. *μ_i, $\sum_i$, $P(w_i)$- are unknown.*
4. *μ_i, $\sum_i$, $P(w_i)$,c- are unknown.*

I will focused on NO.3 problem i.e. the number of Gaussians, groups and probabilitys Is known.

3.2 Algorithm description

3.2.1 The initialization part:

I initialize the algorithm in two different ways:

1. *Randomized Initialization:*
   a. *Randomly choosing the mean as random sample choice.*
   b. *Choosing the covariance matrix as an Identity matrix.*
   c. *Choosing all the prior probabilities as 1 over c.*

1.2 Notation for Randomized Initialization

Assume we have k clusters named $C\epsilon\{C_1, C_2, \dots \dots C_k\}$ and we choose exactly k points $U\epsilon\{x_1, x_2, \dots \dots x_k\}$,assume the probability that we choose sample from cluster C_j is : $P(x = C_j) = P_j$ for j=1,2,…k. thus : $\sum_{j=1}^{k} P_j = 1$, our goal is that the initial sample set U contain sample for each different clusters, thus $\bigcup_{j=1}^{k} C(x_j) = C$ where $C(x_j)$ is the cluster C_{m_j} that belong to x_j – $C(x_j) = C_{m_j}$ because we have k choices so we have k! permutations and the probability the pick the exactly permutation is $\prod_{j=1}^{k} p_j$ thus, the probability to choose sample from each cluser:

$$P\left(\bigcup_{j=1}^{k} C(x_j) = C\right) = \sum_{n=1}^{k!}\prod_{j=1}^{k} p_j = k!\prod_{j=1}^{k} p_j$$

To maximize this probability, we need to choose uniform distribution i.e.: $p_1 = p_2 = \cdots = p_k = \frac{1}{k}$ *and in this case:*

$$P\left(\bigcup_{j=1}^{k} C(x_j) = C\right) \le k!\prod_{j=1}^{k}\frac{1}{k} = \frac{k!}{k^k}$$

And if we assume uniform distribution the probability that the sample set U contain n_j $identical\ values\ j\ for$ *j=1,2,…k is :*

$$P\left(U\epsilon \bigcup_{j=1}^{k} n_j\ identical\ C_j\right) = \frac{k!}{k^k}\left(\prod_{j=1}^{k}(n_j)!\right)^{-1} \quad , \quad \sum_{j=1}^{k} n_j = k$$

For example : the worst case happens if all the samples are in the same cluster C_j *i.e.:* $n_1 = n_2 = \cdots = n_{k-1} = 0 \ , \ n_k = k$

$$P\left(U\epsilon \bigcup_{j=1}^{k} n_j\ identical\ C_j\right) = \sum_{m=1}^{k}\frac{k!}{k^k}\left(\prod_{j=1}^{k}(n_j)!\right)^{-1} = \frac{1}{k^{k-1}}$$

2. Pre-Partitioning Initialization - initialize via C-means Clustering Algorithm.

<u>K-means algorithm:</u>

K-means algorithm Is a method of vector quantization that aims to partition n observations (samples) Into k cluster. Each observation belongs to the cluster with the nearest mean (cluster centers). Suppose we have k cluster centers $\{\mu_1, \mu_2, \dots \dots \mu_k\}$*, for i=1, 2,…,n sample* x_i *belong to the closest center, if we define the cluster mapping function* $C_{km}(x) : R^n \rightarrow \{1,2,\dots,k\}$ *then :*

$$C_{km}(x_i; \mu_1, \mu_2, \dots \dots \mu_k) = \underset{j=1,2,\dots,k}{argmin}\, d(x_i, \mu_j)$$

Where d() is some metric distance, in my case I choose Squared Euclidian distance $d(x_i, \mu_j) = |x_i - \mu_j|^2$

K-means algorithm minimize the objective function:

$$J(X, \mu_1, \mu_2, \dots \dots \mu_k) = \sum_{j=1}^{k}\sum_{x_i \in group\ i}^{-} d(x_i, \mu_j) = \sum_{j=1}^{k}\sum_{x_i \in group\ i}^{-} |x_i - \mu_j|^2$$

To solve this optimization problem, we used the algorithm in above:

<u>Input:</u>

1. *Data Set*
2. *Number of groups*
3. *Randomized group means choice.*

<u>Step 1</u>: Each data point is classified to a certain group by the minimal Euclidian distance.

Step 2: *Calculate the groups centers as the mean of the data points classified to it.*
Step 3: *Repeat Steps 1-2 until:*

a. Convergence condition is reached.
b. Maximum allowed iterations is reached.

Output: Clustered groups means.

- *In my program I defined the convergence condition (tolerance) to be :* $\|\boldsymbol{\mu}_{\boldsymbol{correnT}} - \boldsymbol{\mu}_{\boldsymbol{previous}}\| = 10^{-6}$.
- *The max Iteration is 1000.*

$\hat{\theta} = \underset{\theta \epsilon \Theta}{argmax}\, f(x_1, x_2, \dots, x_n; \theta)$

Where $f(x_1, x_2, \dots, x_n; \theta)$ *is the joint distribution function with set of parameters* θ.
We don't know the prior probabilities so for initialization we assume uniform distribution of the samples over k groups, then we need to estimate also the prior probability and find the group that maximize the probability, those:

$C_{mle}(x_i; \theta, P(w_1), P(w_2), \dots, P(x_k)) = \underset{j=1,2,\dots,k}{argmax}\, P(w_j) f(x_i, \theta_j)$

Where $P(w_j)$ *is the probability of sample* x_i *belong to group j (for all i)*
$f(x_i, \theta_j)$ *is the density function with set of parameters that belong to cluster j (e.g. mean and covariance matrix of normal distribution) in point* x_i.
In my case I choose normal distribution so the set of parameters θ *is the mean and covariance matrix and the prior probability off each group*
$\theta = \{\mu_1, \Sigma_1, P(w_1), \mu_2, \Sigma_2, P(w_2), \dots, \mu_k, \Sigma_k, P(w_k)\}$
To solve this optimization problem, we used the algorithm in above:

MLE algorithm:

MLE (Maximum Likelihood Estimation) is a method that estimating the parameters of probability distribution by maximizing the likelihood function. Our goal is maximizing the objective function:

Input:

1. *Data Set*
2. *Number of groups*
3. *Initial group means, covariance matrices, prior probabilities.*

Step 1: *Calculate the posterior probability for every data point via bays' formula:*

$$P(w_i|x_j) = \frac{P(x_j|w_i)p(w_i)}{\sum_{i=1}^{c} P(x_j|w_i)p(w_i)}$$

**(i=1,2,3…c , j=1,2,3…N , N-number of Data Set)*

$\boldsymbol{p(w_i)}$*-prior probability*

$\boldsymbol{P(x_j|w_i)}$*-conditional probability (or* ***likelihood probability****)- the gaussian probability of each group, calculated via the* μ_i *&* $\sum_i$ *of each group.*
$\boldsymbol{P(w_i|x_j)}$*- the posterior probability*

<u>*Step 2:*</u> *Calculate the groups centers (or* μ_i*) (for next iteration):*

$$\mu_i = \sum_{j=1}^{N} \frac{P(w_i|x_j)x_j}{\sum_{j=1}^{N} P(w_i|x_j)}$$

<u>*Step 3:*</u> *Calculate the groups covariance matrices (for next iteration):*

$$\Sigma_i = \sum_{j=1}^{N} \frac{P(w_i|x_j)x_j}{\sum_{j=1}^{N} P(w_i|x_j)} * (x_j - \mu_i)(x_j - \mu_i)^T$$

<u>*Step 4:*</u> *Calculate prior probabilities (for next iteration):*

$$P(w_i) = \frac{number\ of\ sample\ belong\ to\ group\ i}{n}$$

<u>*Step 5:*</u> *Repeat Steps 1-4 until:*

a. Convergence condition is reached.
b. Maximum allowed iterations is reached.

<u>*Output:*</u> *Estimated group means, Estimated group covariance matrices, Estimated prior probability, the Estimated posterior probability.*

- *In my program I defined the convergence condition (tolerance) to be :* $\|\boldsymbol{u_{mle}}(i) - \boldsymbol{u_{mle}}(i-1)\|_F = 10^{-5}$.
- *The max Iteration is 100.*

3.3 Results

3.3.1 NMI Normalized Mutual Information Test

NMI is a popular information theory measure used to evaluate the clustering quality. If we have two random variables X and Y, the NMI is computed as follow:

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}}$$

Where,

X, Y=Random variables that represent the cluster labels.

H-Entropy

$$H(X) = -\sum_{x \in X} p_X(x) \log\left(p_X(x)\right)$$

I-Mutual Information

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x) p_Y(y)}\right)$$

** NMI between X and itself is 1.*

** The larger the NMI Indicates similarity between the random variables*

** The worst case : X and Y are In depended, thus* $p_{(X,Y)}(x,y) = p_X(x) p_Y(y)$ *and :*

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p_{(X,Y)}(x,y) \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x) p_Y(y)}\right) = \sum_{x \in X} \sum_{y \in Y} p_{(X,Y)}(x,y) \log(1) = 0$$

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} = \frac{0}{\sqrt{H(X)H(Y)}} = 0$$

We assume that X, Y not constant, I.e.: $H(X) > 0$ *and* $H(Y) > 0$

** The worst case: the clusters are exactly the same, thus : X=Y and :*

$$I(X,X) = \sum_{x \in X} p_X(x) \log\left(\frac{p_X(x)}{p_X(x) p_X(x)}\right) = -\sum_{x \in X} p_X(x) \log\big(p_X(x)\big) = H(X)$$

$$NMI(X,Y) = \frac{I(X,X)}{\sqrt{H(X)H(X)}} = \frac{H(X)}{\sqrt{H(X)H(X)}} = 1$$

3.3.2 RI-Rand Index Test

Given a set of n elements $S = \{o_1, \ldots \ldots o_n\}$ *and two partition of S to compare,* $X = \{X_1, \ldots \ldots X_r\}$ *, a partition of S Into r subsets, and* $Y = \{Y_1, \ldots . Y_s\}$*, a partition of S Into s subsets, define the following:*

- ***a****, the number of pairs of elements In S that are in the <u>same</u> subset In X and In the <u>same</u> subset In Y. (true positive)*
- ***b****, the number of pairs of elements In S that are in the <u>different</u> subset In X and In the <u>different</u> subset In Y. (true negative)*
- ***c****, the number of pairs of elements In S that are in the <u>same</u> subset In X and In the <u>different</u> subset In Y. (false positive)*
- ***d****, the number of pairs of elements In S that are in the <u>different</u> subset In X and In the <u>same</u> subset In Y. (false negative)*

The Rand Index-R Is:

$$R = \frac{a+b}{a+b+c+d}$$

3.3.3 Relative Silhouette coefficients

Assume the data have been clustered via any technique into k cluster named $\{C_1, C_2, \dots \dots C_k\}$ *, for data point* x_i *in cluster* $C_m = C(x_i)$ *when* $m\epsilon\{C_1, C_2, \dots \dots C_k\}$ *the value a(*x_i*) that represent the similarity for the other point in his cluster define as:*

$$a(x_i) = \frac{1}{|C_m| - 1} \sum_{\substack{j \in C_m \\ j \neq i}} d(x_i, x_j)$$

When $|C_m|$ *is the number of samples that clustered int* C_m *and* $d(x_i, x_j)$ *is some metrical distance between* x_i *and* x_j *, I choose the Squared Euclidian distance :*

$$d(x_i, x_j) = \left|\left|x_i - x_j\right|\right|^2 = \sum_{l=1}^{d} (x_{il} - x_{jl})^2$$

When $x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \in R^d$

And we defined the value b(x_i*) that represent the dissimilarity of point* x_i *into all other cluster C/*C_m *(all the cluster except the cluster than contain* x_i*)*

And, we choose the same Squared Euclidian distance for $d(x_i, x_j)$

The Silhouette coefficients $s(x_i)$ *defined as:*

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max\{b(x_i), a(x_i)\}} = \begin{cases} 1 - \dfrac{a(x_i)}{b(x_i)} & , a(x_i) < b(x_i) \\ 0 & , a(x_i) = b(x_i) \\ \dfrac{b(x_i)}{a(x_i)} - 1 & , a(x_i) > b(x_i) \end{cases}$$

- *if* $|C(x_i)| = 1$ *,* $(i.e\ the\ sample\ x_i\ is\ the\ only\ sample\ in\ cluser\ C_m = C(x_i)$ *thus* $s(x_i) = 0$.
- *From the above definition It Is clear that* $-1 \leq s(x_i) \leq 1$
- *If* $s(x_i) \cong -1$ *, thus we can say that the sample* x_i *does not belong to the right cluster* $C_m = C(x_i)$ *because In the case* $a(x_i) \gg b(x_i)$ *and the mean distance between the sample* x_i *from Its own cluster Is bigger than the mean distance from other clusters.*
- *If* $s(x_i) \cong 1$ *, thus we can say that the sample* x_i *belong to the right cluster* $C_m = C(x_i)$ *because In the case* $a(x_i) \ll b(x_i)$ *and the mean distance between the sample* x_i *from Its own cluster Is less than the mean distance from other clusters.*
- *If* $s(x_i) \cong 0$ *, thus we can say that Is difficult to decide If sample* x_i *belong to the right cluster* $C_m = C(x_i)$ *because In the case* $a(x_i) \cong b(x_i)$ *and the mean distance between the sample* x_i *from Its own cluster Is equal than the mean distance from other clusters*

Now If we assume that we have the real labels we can calculate the relative Silhouette coefficients as follow : calculate the Silhouette coefficients with our real label $S_R(x_i)$ *, then calculate the coefficients with our cluster with some algorithm* $S_{est}(x_i)$ *and subtract those coefficients* $S_{relative}(x_i) = S_{est}(x_i) - S_R(x_i)$

- *The greater the coefficient* $S_R(x_i)$ *It can say that Is easier to cluster the sample* x_i *correctly and vice versa.*
- *From the above definition It Is clear that* $-2 \leq S_{relative}(x_i) \leq 2$
- *if* $S_{relative}(x_i) \cong 0$ *that mean* $S_{est}(x_i) \cong S_R(x_i)$ *and we can say that our cluster Is better and look exactly like the real cluster*

- *if $S_{relative}(x_i) > 0$ that mean $S_{est}(x_i) > S_R(x_i)$ and we can say that we cluster the sample x_i closer that those sample should be*
- *if $S_{relative}(x_i) \cong 0$ that mean $S_{est}(x_i) \cong S_R(x_i)$ and we can say that our cluster Is better and look exactly like the real cluster.*

Now we can investigate some statistics for the R.V $Y = S_{relative} \in R^n$ where n is the number of sample of our data set X .

- *First we calculate the median, mean and variance of X as follow:*

$$med(X) = \{a : |Y < a| = |Y > a|\} = \{a : P(Y < a) = P(Y > a)\}$$

$$\mu_y = E[Y] = \frac{1}{n}\sum_{i=1}^{n} S_{relative}(x_i)$$

$$\sigma_y^2 = Var[Y] = \frac{1}{n-1}\sum_{i=1}^{n}(S_{relative}(x_i) - E[Y])^2$$

As those values closer to 0, then $Y = S_{relative}$ more deterministic and closer to 0.

- *Then in more general case we can calculate the central moment function as follow*

$$M_y(t) = E\left[\left(Y - \mu_y\right)^t\right] \quad , \quad t \geq 1$$

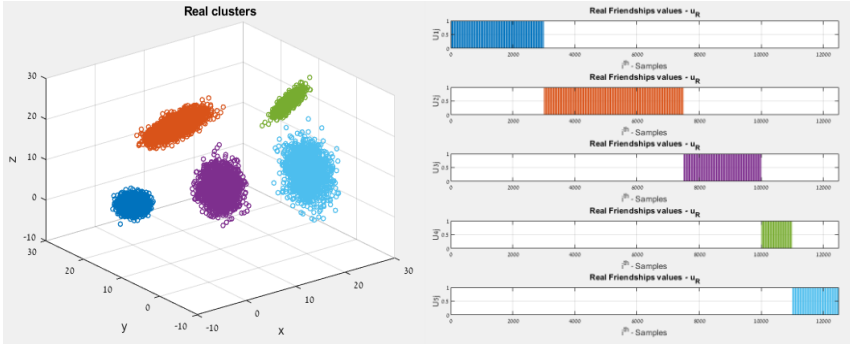
More the $M_y(t)$ Is closer to 0, the R.V Y=$S_{relative}$ Is more deterministic and closer to his mean μ_y (that should be closer to 0)

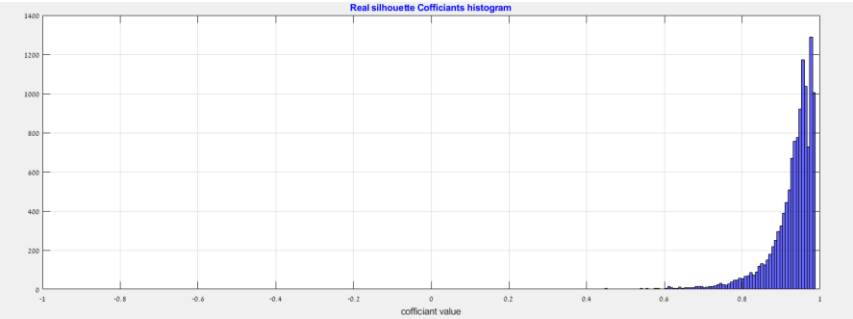
- *In my simulations the Inputs for the NMI and the RI functions are the labels of the Data Sets and the Labels after the MLE or MLE+KM.*

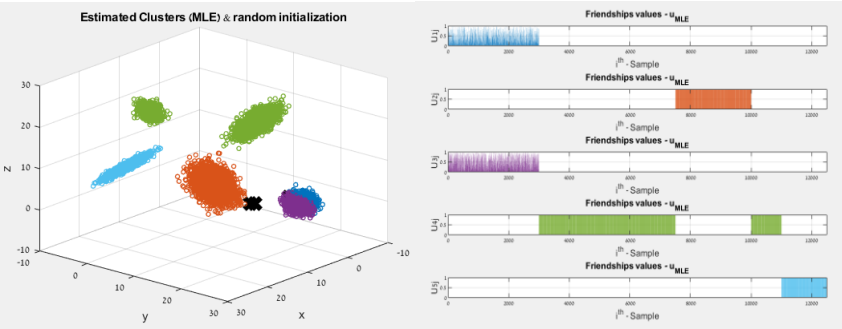
3.3.4 Result presentation

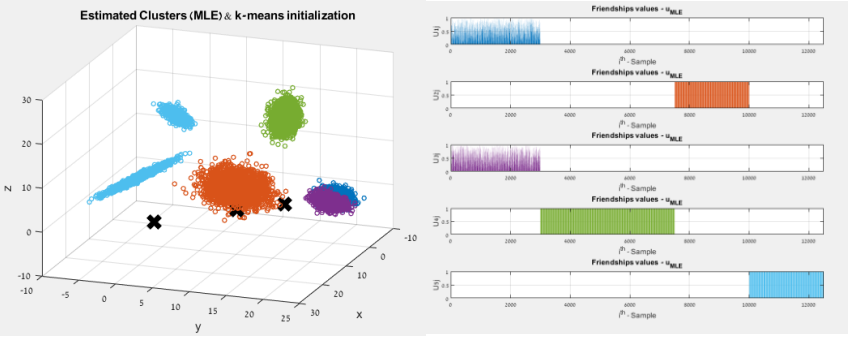
3.3.4.1 <u>4 Gaussian without an overlap in 3D</u>

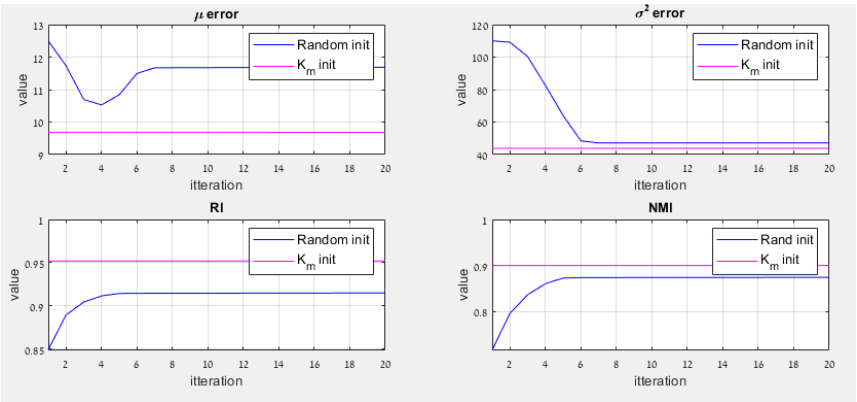
X	12500x3 double
Labels	12500x1 double
n	12500
d	3
c	5

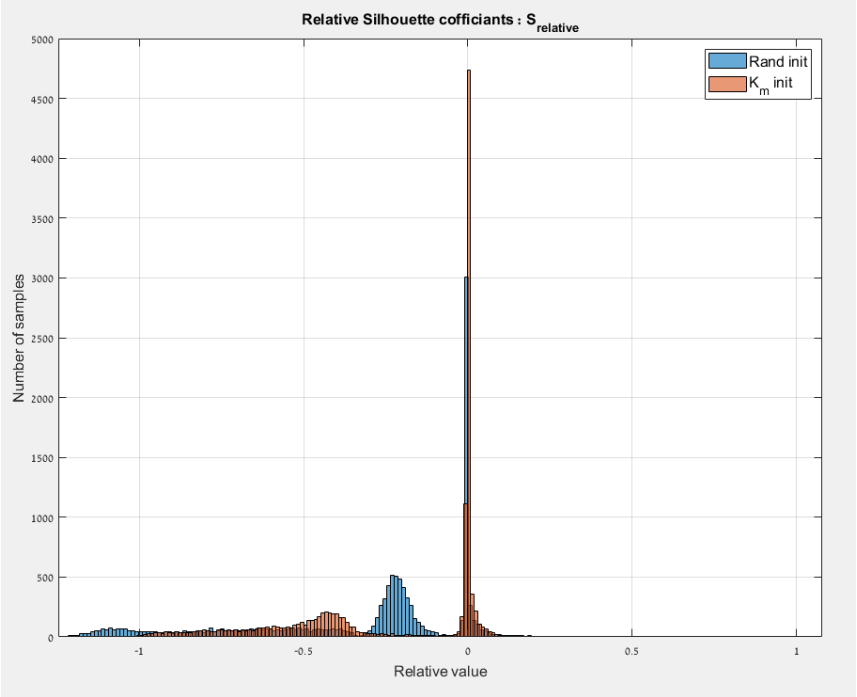


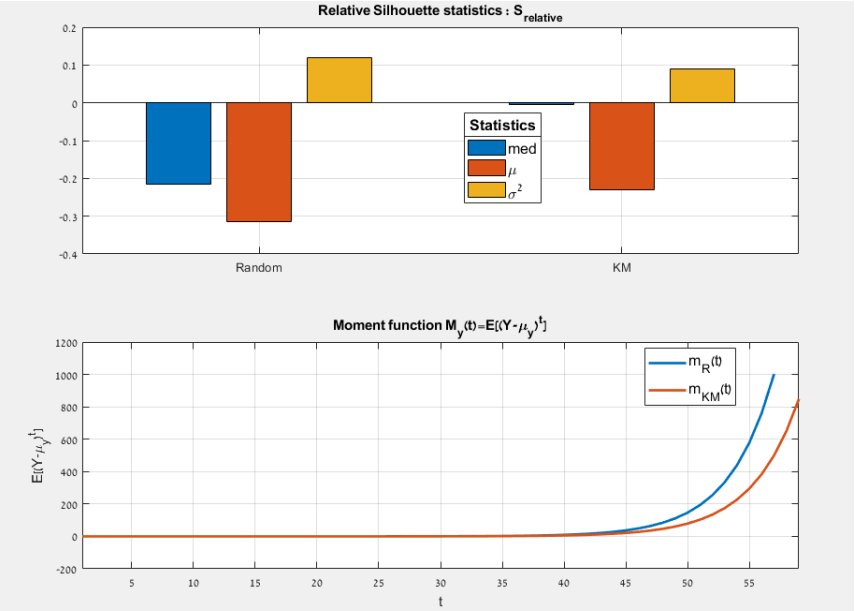

Real silhouette Cofficiants histogram
cofficiant value


Estimated Clusters (MLE) & random initialization
Friendships values - u MLE
i th - Sample


Estimated Clusters (MLE) & k-means initialization
Friendships values - u MLE
i th - Sample

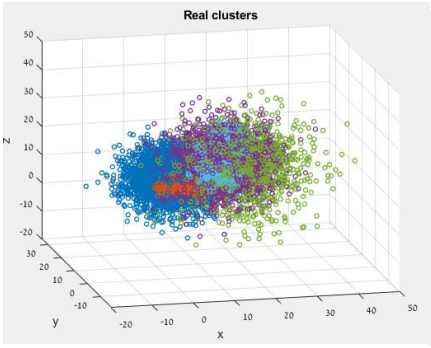

μ error
Random init
K_m init
value
itteration
σ^2 error
Random init
K_m init
value
itteration
RI
Random init
K_m init
value
itteration
NMI
Rand init
K_m init
value
itteration

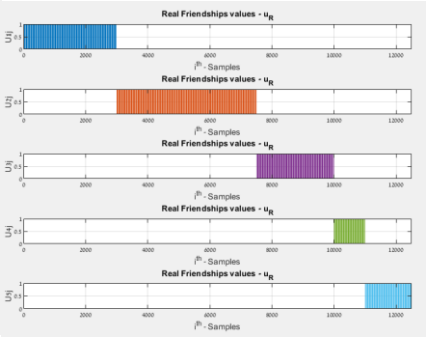

Relative Silhouette cofficiants : $S_{relative}$
Rand init
K_m init
Number of samples
Relative value

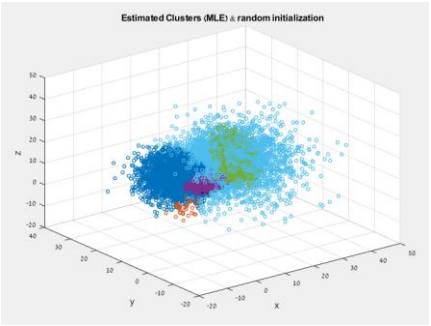


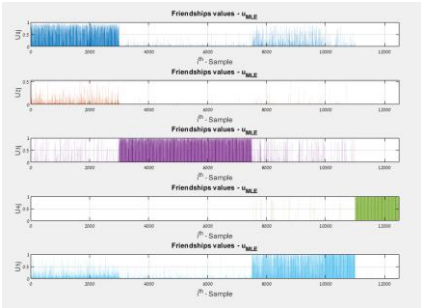
- *Mu_error define as the vector norm of the difference between DS means (centroid) and the means after the MLE algorithm.*
- *Cov_error define as the sum of matrices norm of the difference between DS covariances and the covariances after the MLE algorithm.*
- *We can see that the results with KM Initialization are slightly better than the randomized Initialization.*
- *With KM Initialization converge faster then the random Initialization. we can see It In the almost constants graph of the RI, NMI In KM (the pink graphs).*
- *We also can see that the relative silhouette coefficients of KM are closer to 0, the median, mean, variance and moment function are smaller from random Init.*

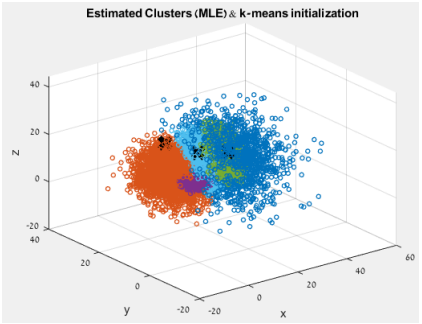
3.3.4.2 <u>5 Gaussian with partial overlap in 3D</u>

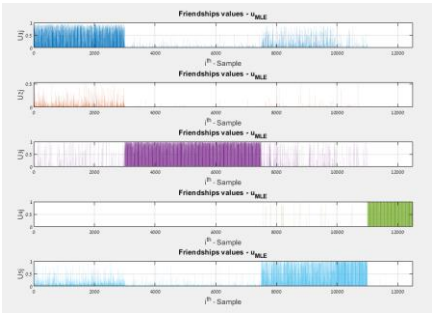

Real clusters
z
y
x

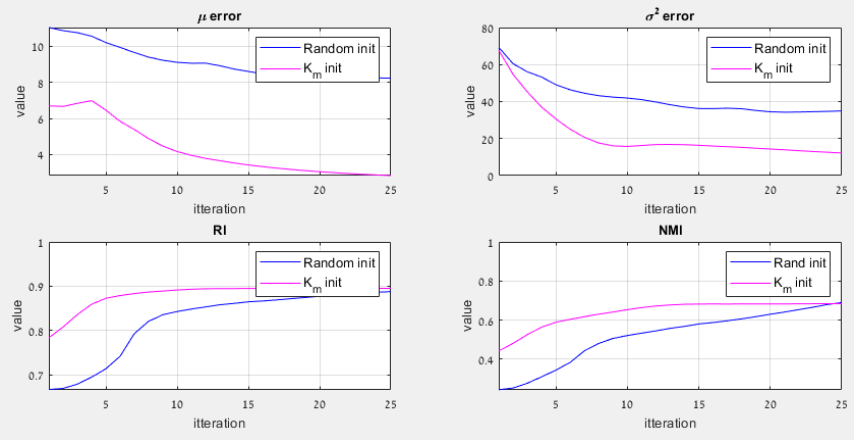

Real Friendships values - u_R
Real Friendships values - u_R
Real Friendships values - u_R
Real Friendships values - u_R
Real Friendships values - u_R
i^th - Samples

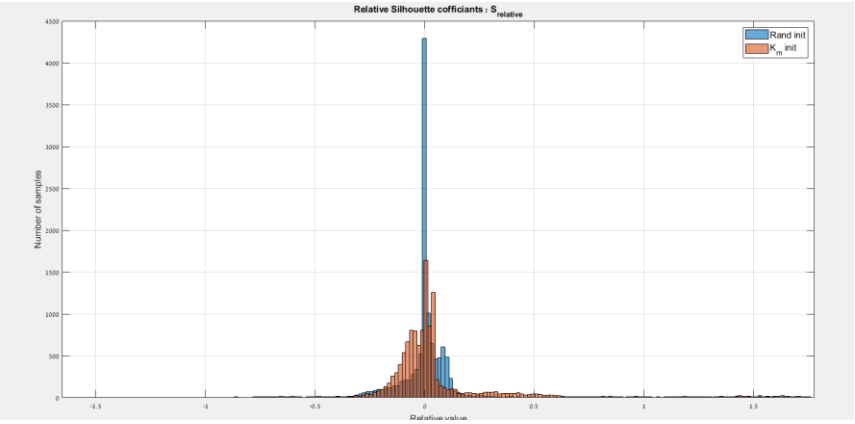

Estimated Clusters (MLE) & random initialization
z
y
x

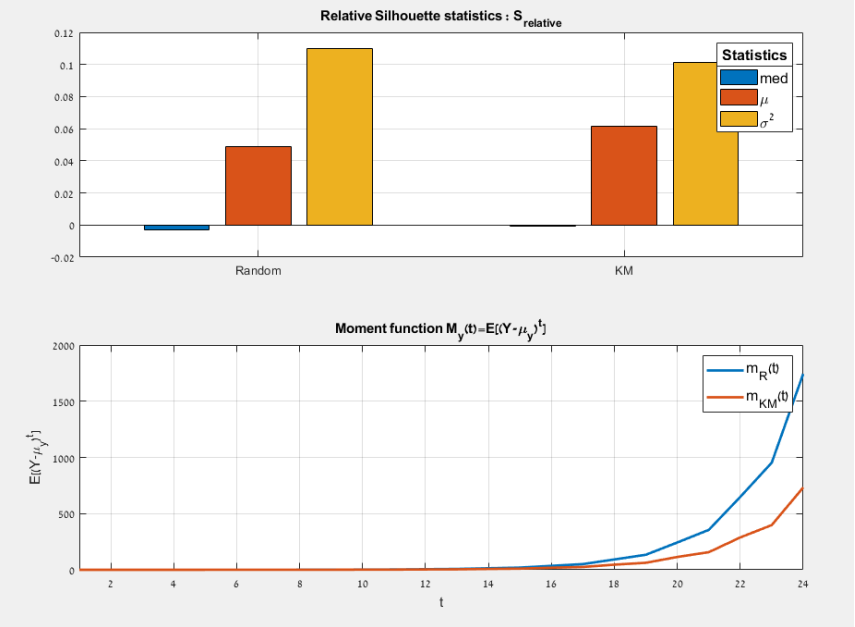

Friendships values - u_MLE
Friendships values - u_MLE
Friendships values - u_MLE
Friendships values - u_MLE
Friendships values - u_MLE
i^th - Sample


Estimated Clusters (MLE) & k-means initialization
z
y
x


Friendships values - u_MLE
Friendships values - u_MLE
Friendships values - u_MLE
Friendships values - u_MLE
Friendships values - u_MLE
i^th - Sample

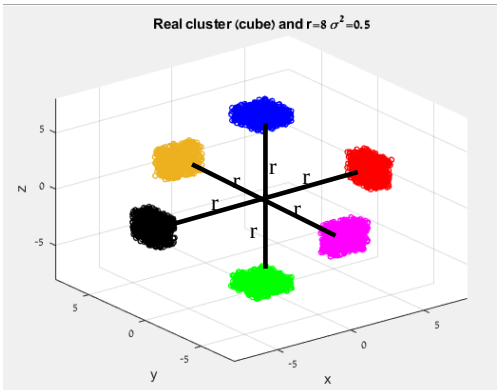

μ error
Random init
K_m init
value
itteration
σ^2 error
Random init
K_m init
value
itteration
RI
Random init
K_m init
value
itteration
NMI
Rand init
K_m init
value
itteration

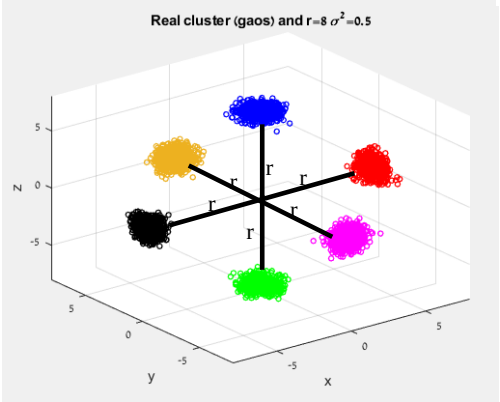

Relative Silhouette cofficiants : $S_{relative}$
Rand init
K_m init
Number of samples
Relative value

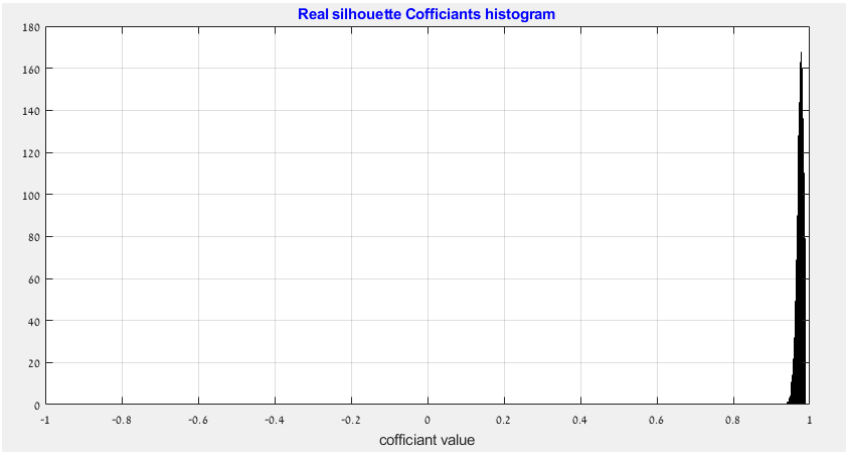


- *In the first graphs (p_{error} , μ_{error} , σ^2_{errror} , RI , NMI) we can se that the KM Initialization have also better results but more slightly.*
- *In this case the KM has taken more Iterations to converge unlike the first case (gaussians without an overlap).*
- *In the relative silhouette coefficients graph, the result for the random and KM Initialization are closet to each other (In the median and mean and variance values) but on the other hand there Is more "super negative" samples (I.e. : more samples that their coefficients obtain $s(x_i) < -0.5$ and we know In this case that those sample are not belong to her clusters In high probability.*

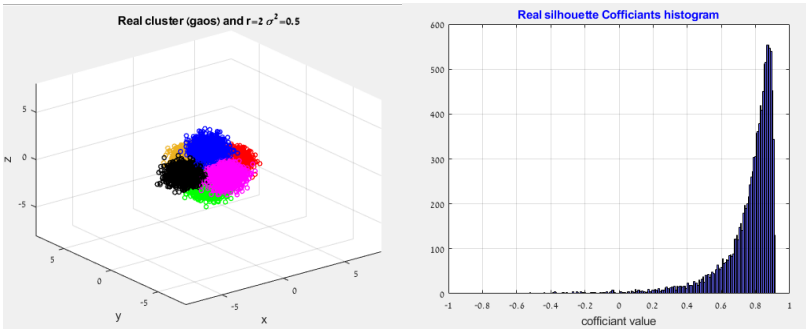
3.3.4.3 6 Gaussian and cubes with variable distance



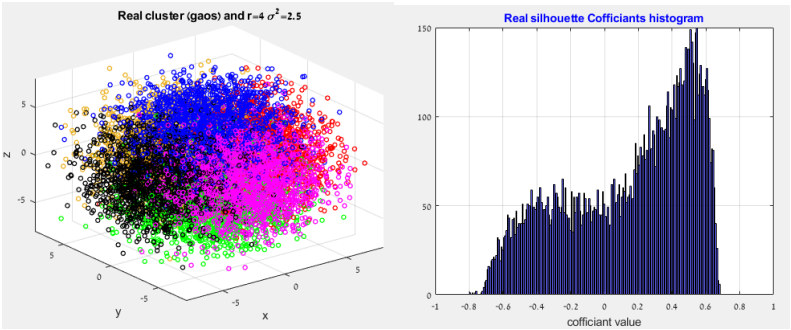




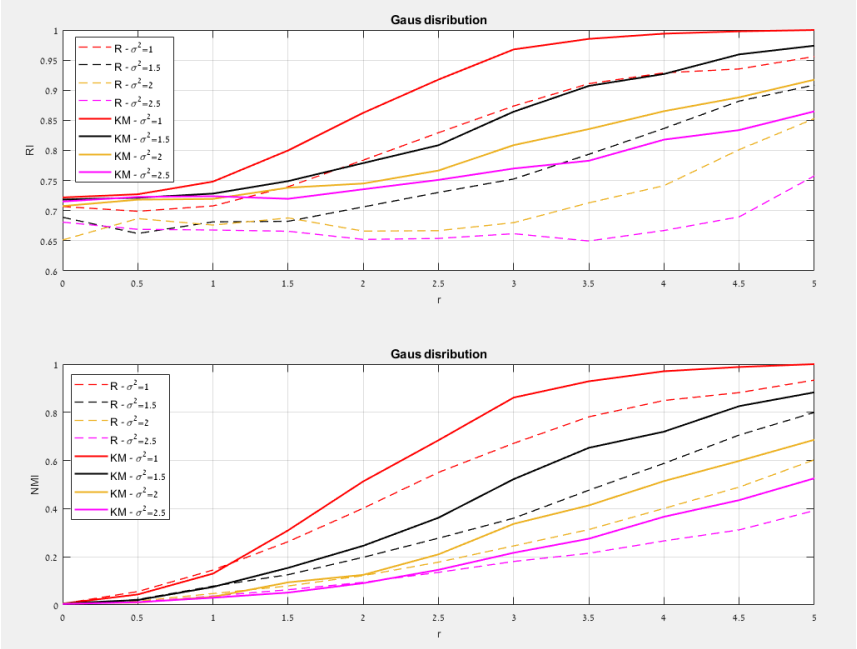
For example : In this case the silhouette coefficients closer to 1

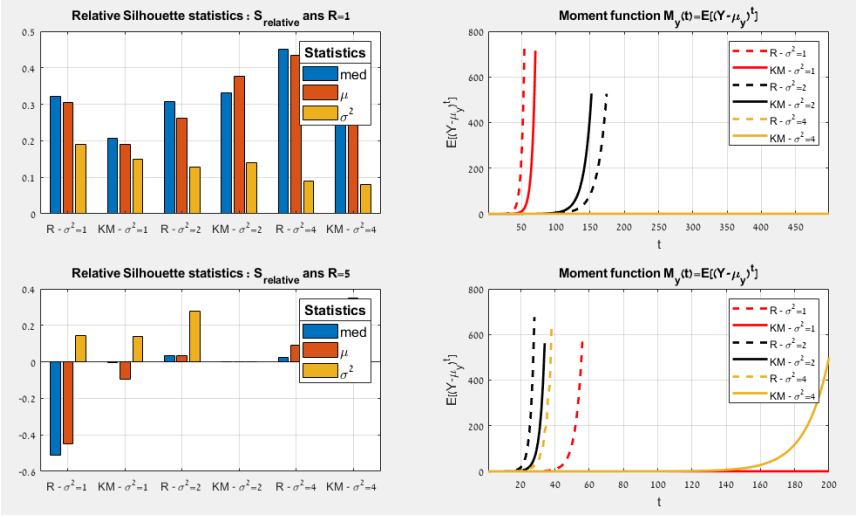


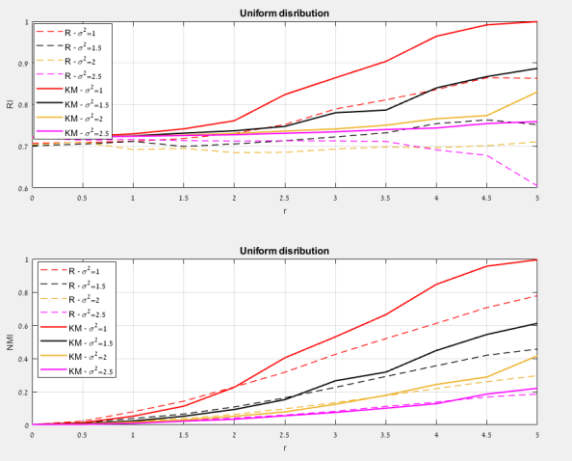
Now In this case the silhouette coefficients closer to 0

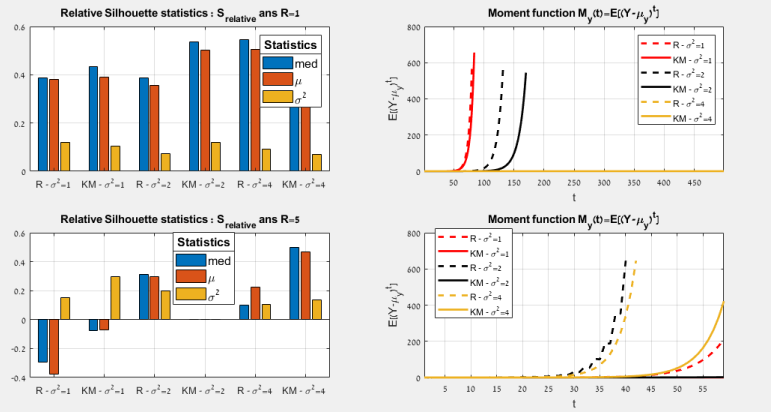


- *As long the radius r Is smaller or the variance σ^2 Is bigger so the real silhouette coefficients get smaller because In this case the samples are far from their centers and get closer to other cluster that not belong to the cluster of the specific sample, (and vice versa).*
- *As long the real silhouette coefficients close to 1, It can be easy to classified this sample correctly and vice versa*


Gaus disribution
RI
r
R - σ²=1
R - σ²=1.5
R - σ²=2
R - σ²=2.5
KM - σ²=1
KM - σ²=1.5
KM - σ²=2
KM - σ²=2.5
Gaus disribution
NMI
r

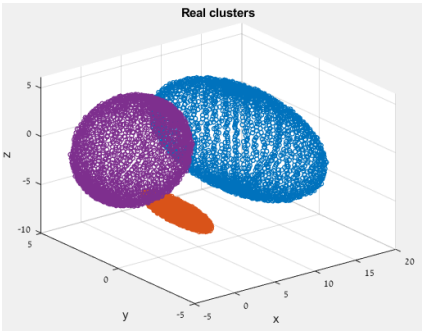

Relative Silhouette statistics : S relative ans R=1
Statistics
med
μ
σ²
R - σ²=1
KM - σ²=1
R - σ²=2
KM - σ²=2
R - σ²=4
KM - σ²=4
Moment function M_y(t)=E[(Y-μ_y)^t]
E[(Y-μ_y)^t]
t
Relative Silhouette statistics : S relative ans R=5
Moment function M_y(t)=E[(Y-μ_y)^t]

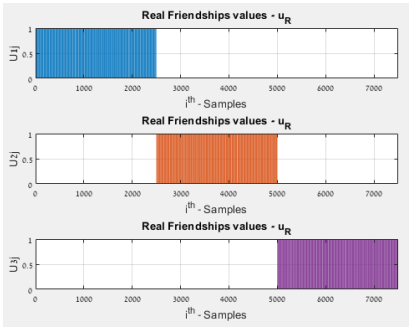


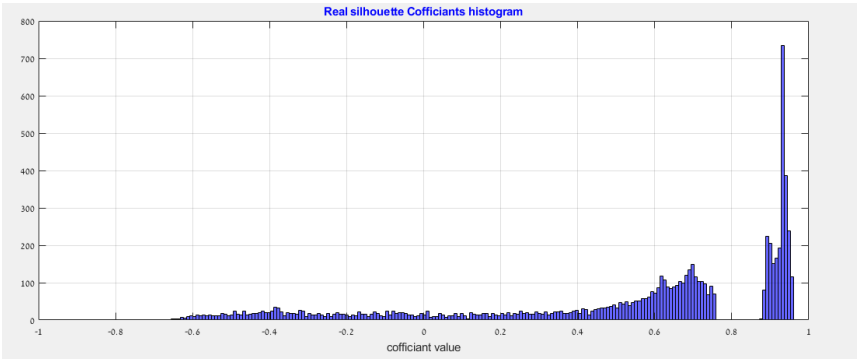


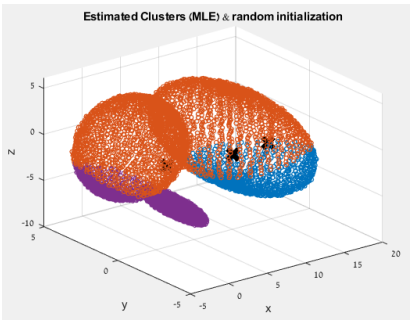
- *We can se In RI and MNI graphs that the KM Initialization always better rather than random Initialization*
- *As long the variance Is bigger, the difference between the 2 Initialization (random and KM) Is smaller (the KM still better but more slightly).*
- *As long the distance between cluster is smaller (r) the RI, NMI also ger smaller with any initialization and any variance (little changes).*
- *As long the distance between cluster is bigger (r) the randomness of the relative silhouette coefficients also get smaller (more moment are equal to 0), that mean - the KM Is more efficient In the case*

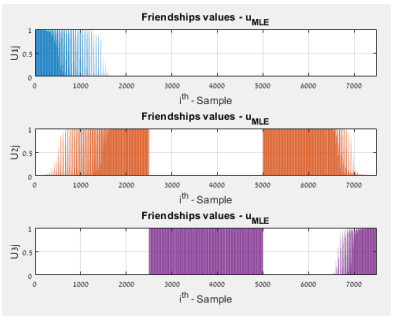
3.3.4.4 4 Spheres in 3D

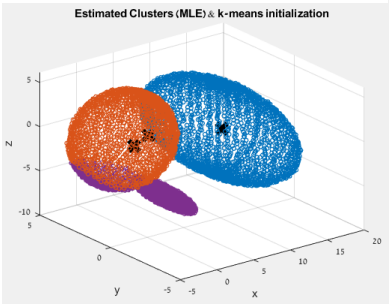


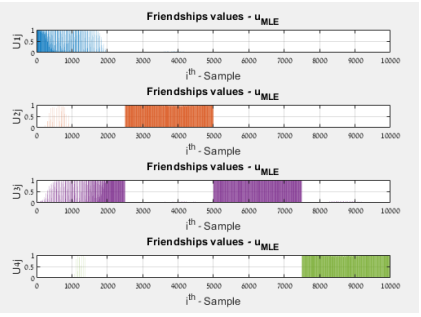


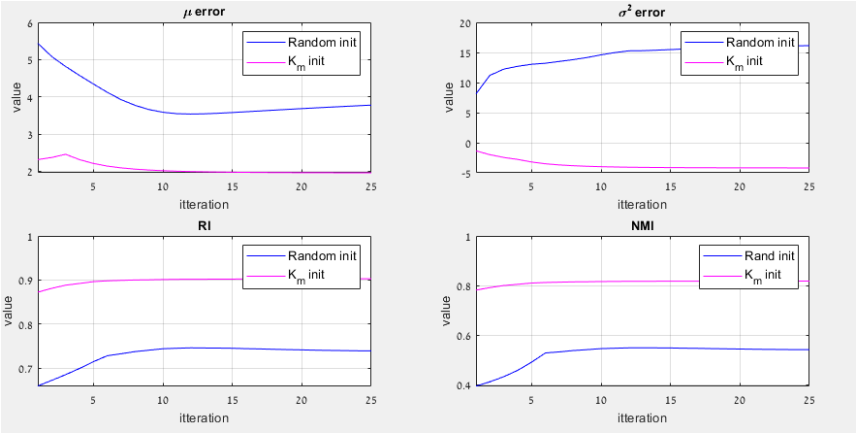


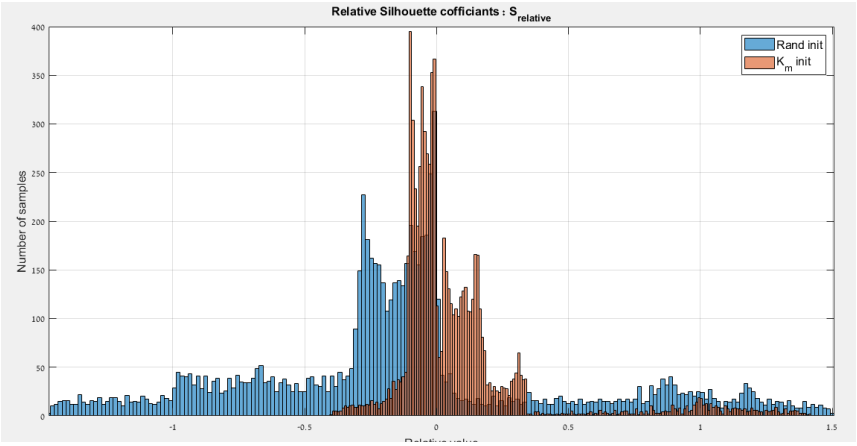


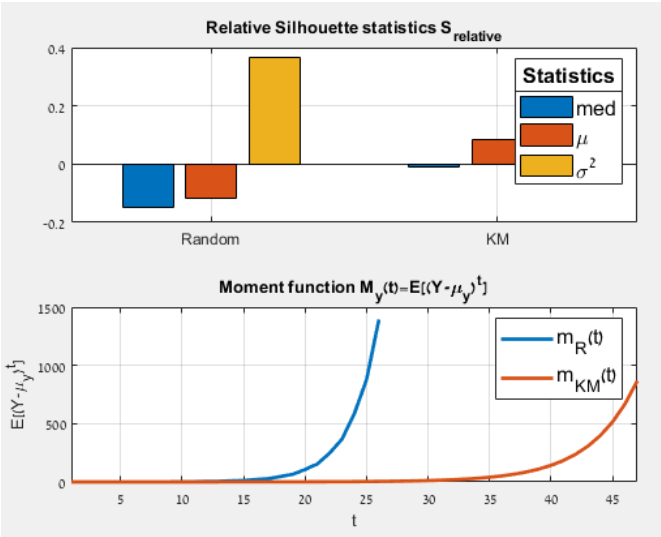






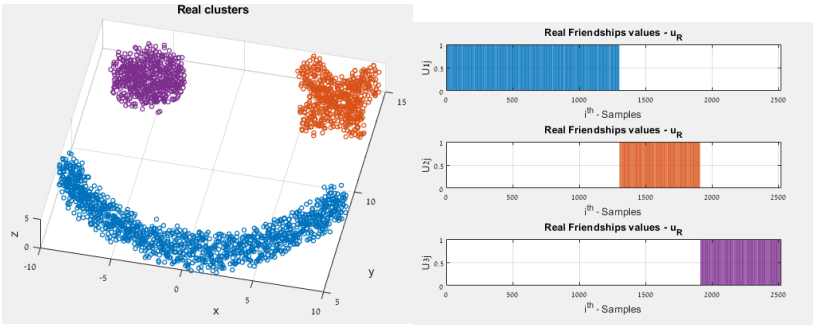

μ error
Random init
K_m init
value
itteration
σ² error
Random init
K_m init
value
itteration
RI
Random init
K_m init
value
itteration
NMI
Rand init
K_m init
value
itteration

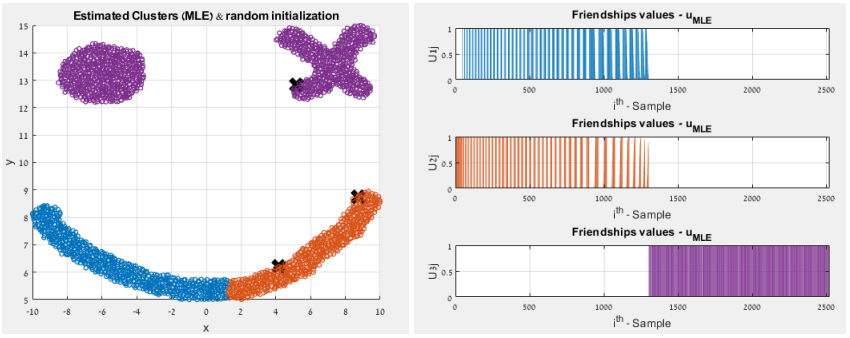

Relative Silhouette cofficiants : S_relative
Rand init
K_m init
Number of samples

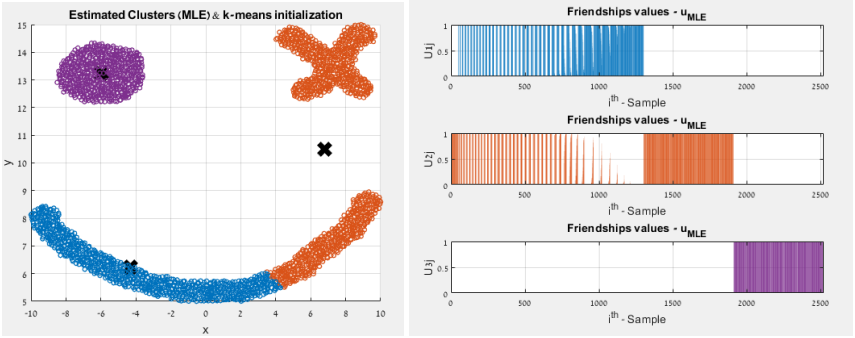

Relative Silhouette statistics S_relative
Statistics
med
μ
σ²
Random
KM
Moment function M_y(t)=E[(Y-μ_y)^t]
m_R(t)
m_KM(t)
E[(Y-μ_y)^t]
t

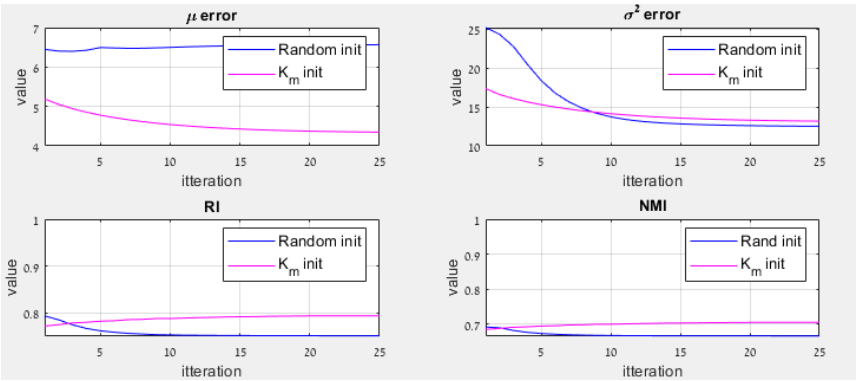
- *Beacause the DS Isn't gauss, the RI, NMI graphs are smaller than the gauss case (for example in this case RI=0.9 and In the gauss case RI=0.96).*
- *If the DS Isn't gauss, is cat take more Iterations In KM to converge.*
- *The KM Initialization have better results, the silhouette coefficients are more matched and so there are more moments that equal to 0.*

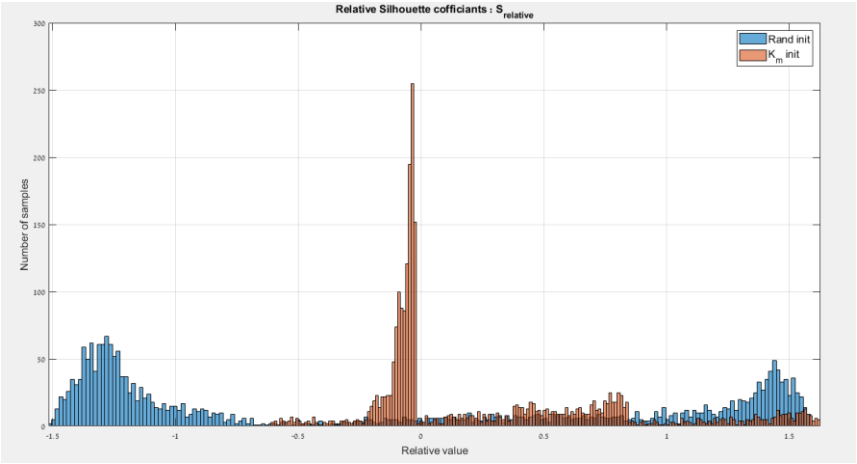
3.3.4.5 *Volume structure In 3D*

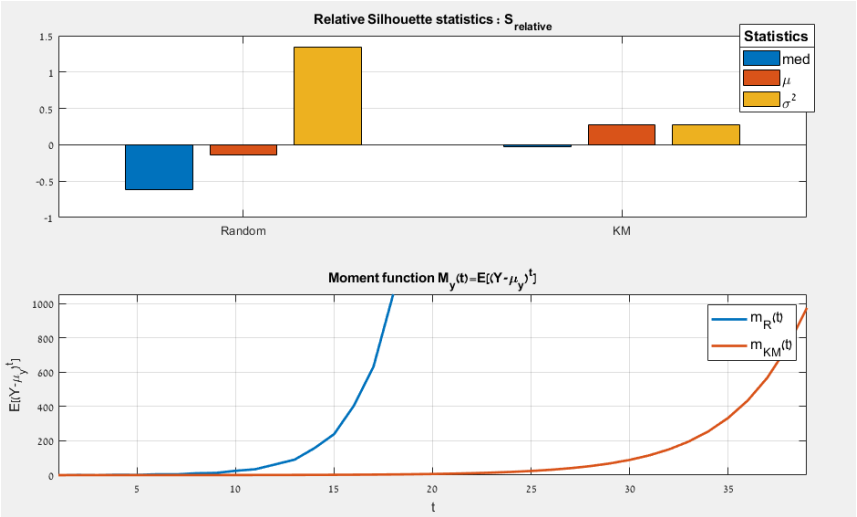







μ error
Random init
K_m init
value
itteration
σ² error
Random init
K_m init
value
itteration
RI
Random init
K_m init
value
itteration
NMI
Rand init
K_m init
value
itteration


Relative Silhouette cofficiants : S_relative
Rand init
K_m init
Number of samples
Relative value


Relative Silhouette statistics : S_relative
Statistics
med
μ
σ²
Random
KM
Moment function M_y(t)=E[(Y-μ_y)^t]
m_R(t)
m_KM(t)
E[(Y-μ_y)^t]
t

- *Beacause the DS Isn't gauss, the RI, NMI graphs are smaller than the gauss case and even from the ellipse case (for example In this case RI=0.8 and In the gauss case RI=0.96).*
- *The converge Is fast because the cluster are far from each other.*
- *The KM Initialization have better results, the silhouette coefficients are more matched and so there are more moments that equal to 0.*

3.4 Conclusion

3.4.1. The KM initialization in the Gaussian example are useful according to my results.

3.4.2. KM initialization can help algorithms to gauss the first centroids and by doing so, we can prevent errors that come from stack in a local optimum.

3.4.3. Using KM in large DS is computational dereddening so we should take it in consideration.

3.4.4. MLE works better in spherical shape clusters and introduce the best solutions on Gaussian DS.

3.4.5. If the number of samples is different from each cluster, the accuracy of the random initialization can be reduced.

3.4.6. MLE preform great in high dimensional DS, but it should be emphasized that computational resources are more demanding.

3.4.7. MLE needs a lot of data sample in order to introduce a good result- i.e. to cluster the DS.

3.4.8. The KM initialization can reduce the number of iterations to converge.

4. Exercise '2'-UOFC

4.1The primary objective of the algorithm

Implement the UOFC Algorithm, using it to identify the number of clusters in addition to the fuzzy classification of every data point to all the clusters.

The UOFC algorithm uses the Fuzzy approach to cluster the data, where every data vector is assigned a membership value to every cluster as oppose to "hard" assignments from data vectors to clusters as in the standard C-mean Algorithm.

The UOFC algorithm uses an incremental approach to the number of clusters, converging and extracting 6 different criterion measures for clustering validity for each number of clusters.

Finally, at the end the algorithm chooses the optimal number of clusters. The UOFC algorithm is the solution to our previously unsolved issue with automatically finding the number of clusters, namely problem no. 3 from the lectures.

Case	μ_i	Σ_i	$P(\omega_i)$	c
1	?	×	×	×
2	?	?	?	×
3	?	?	?	?

4.2 Algorithm description

4.2.1 The initialization part:
Using the data set mean as the first and only mean.
Input:

1. *Data Set*
2. *Fuzziness Parameter-(q).*
3. *The Maximal possible Number of clusters*

Step 1: Cluster the data set using the current means vector as the initializer for the Fuzzy C-mean algorithm.

$$P = \frac{1}{N}\sum_{j=1}^{N} X_j$$

Step 2: Calculate:

1. *The Euclidean distance-**d***
2. *The Data points memberships-**U***
3. *The Centroid-**p***

Step 3: Calculate the 6 clustering validation criteria and save the results.

Step 4: *Add a cluster centroid placed at the distance of 5*norm(var(X)).*

Step 5: *Call the FCM algorithm.*
Input: *DS, **P**-centroid estimation, q-fuzziness parameter, d- distance, number of clusters.*
Output: ***U**-Data points memberships, P-centroid estimation, **d**- distance*

Step 6: *Call the FMLE algorithm.*
Input: *DS, **P**-centroid estimation, q-fuzziness parameter, d- Euclidean distance, number of clusters.*
Output: ***U**-Data points memberships, P-centroid estimation, F-fuzzy covariance matrix, and the **Exponential distance**.*

Step 7: *Calculate the 6 clustering validation criteria and save the results*

Step 8: *Repeat Steps 4 until reaching the maximal number of clusters*

Step 9: *Choosing (manually) the optimal number of cluster.*
.
Output: *Estimated group means, Data points memberships, Clustering validity criterions, number of clusters.*

- *step 6 (use FMLE) is utilized to refine the partition for normally distribution clusters with large variability of the covariance matrix.*
- *Other distance function can be used*
- *d- In the first Iteration Is Euclidian and after that It Is expansional distance.*

6-Clustering Validation Criterions:

As mentioned above, the UOFC utilizes 6 different clustering criterions in order to optimally choose the number of clusters.
The criterions are the following:

1. Hyper Planar Volume (HPV) – The sum of hyper volumes of each cluster in the data set. (min)

2. Partition Density (PD) – The density of each cluster determined by all the membership values in that cluster (central members-i.e. smaller than mahalanobis distance) [$\frac{sum\ (U)}{HPV}$]. (max)

3. Average Partition Density using Central members (APDC) – The density of each cluster determined by its mean calculation on the central members [$mean(\frac{sum\ (U)}{HPV})$] (max).

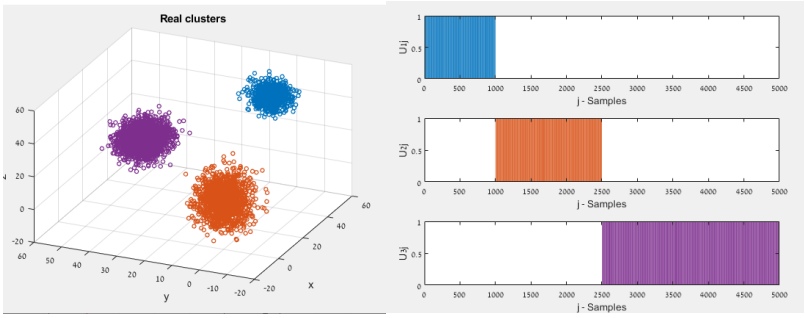
4. APD using Maximal membership members (APDM) – The density of each cluster determined by its maximal membership of each cluster. [$(\frac{sum\ (\max(U))}{HPV})$] (max).

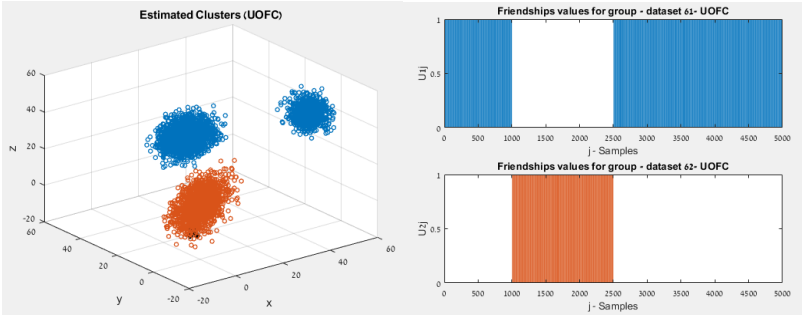
5. J Criterion (J) – the distance of the weighted fuzzy c-mean (min).

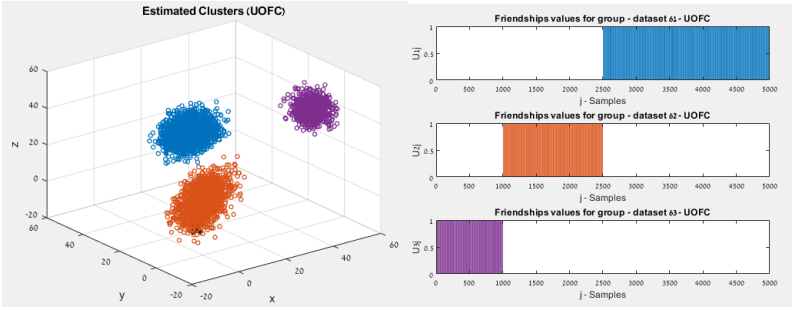
6. Invariant Criterion (INV) – The sum of the hyper volumes normalized by the hyper volume of the entire data set. (min).

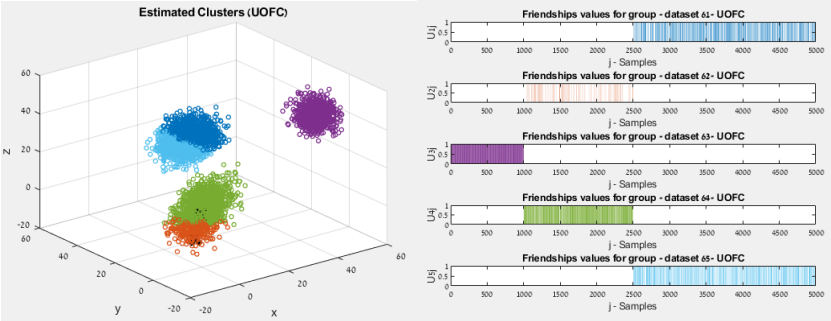
4.3 Results

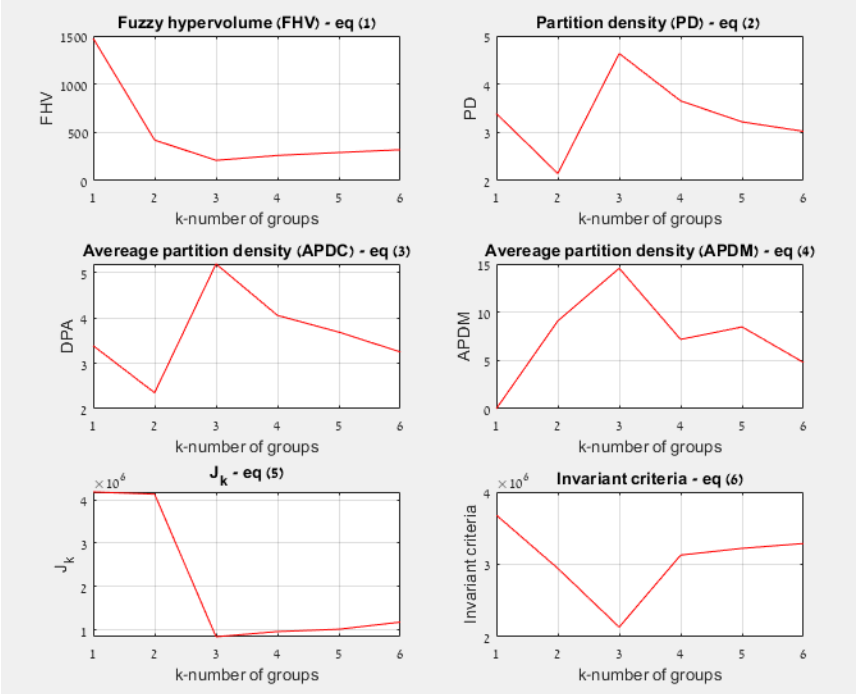
4.3.3.14 Gaussian without an overlap in 3D

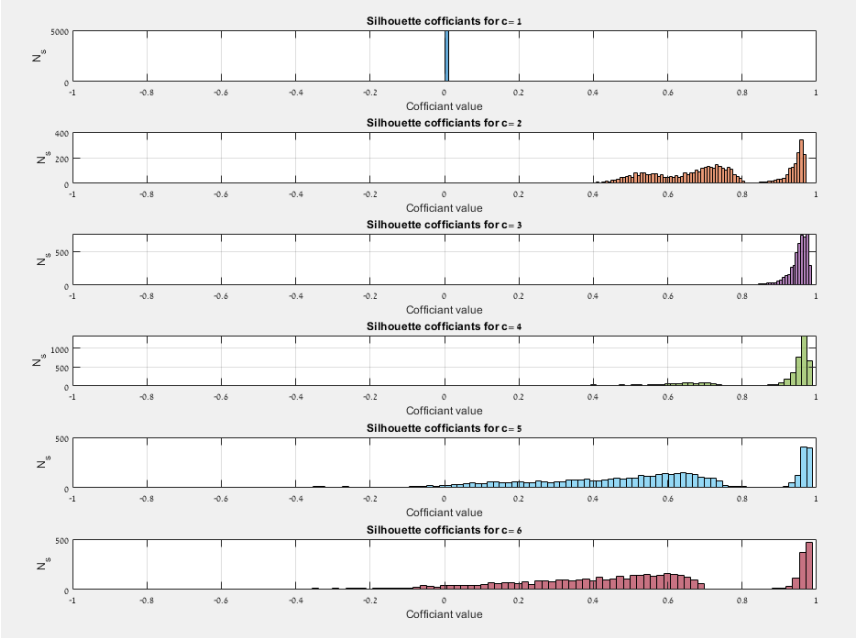


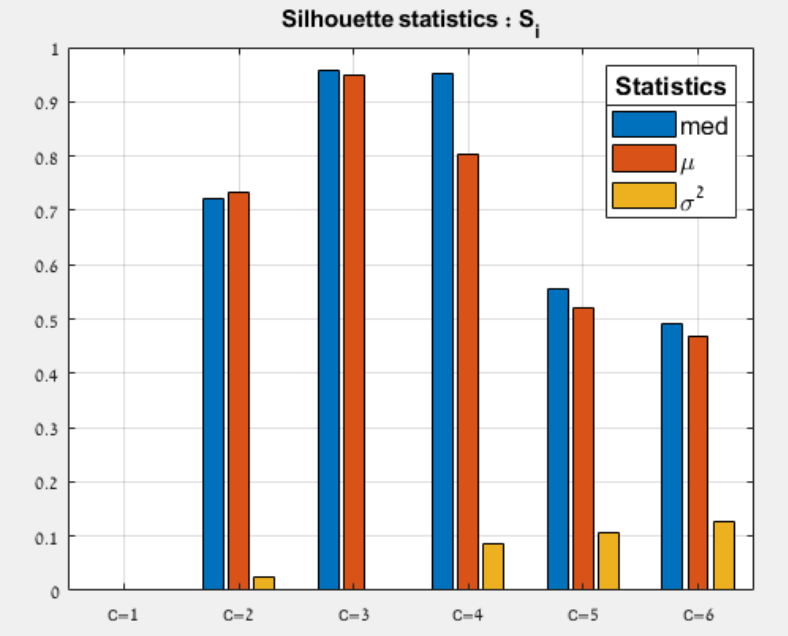





Estimated Clusters (UOFC)
x
y
z
Friendships values for group - dataset 61- UOFC
Friendships values for group - dataset 62- UOFC
Friendships values for group - dataset 63- UOFC
Friendships values for group - dataset 64- UOFC
Friendships values for group - dataset 65- UOFC
j - Samples


Fuzzy hypervolume (FHV) - eq (1)
FHV
Partition density (PD) - eq (2)
PD
Avereage partition density (APDC) - eq (3)
DPA
Avereage partition density (APDM) - eq (4)
APDM
J_k - eq (5)
J_k
Invariant criteria - eq (6)
Invariant criteria
k-number of groups

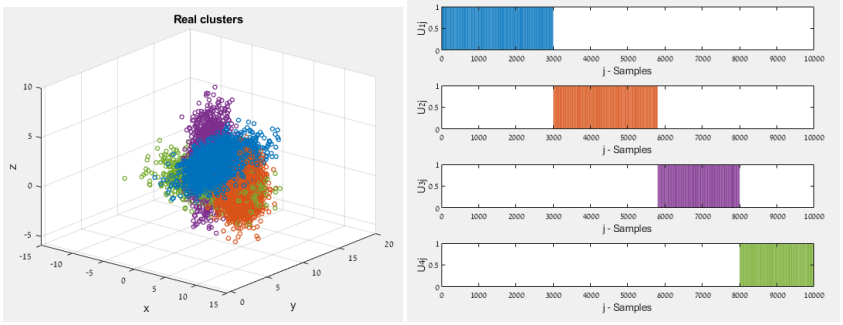


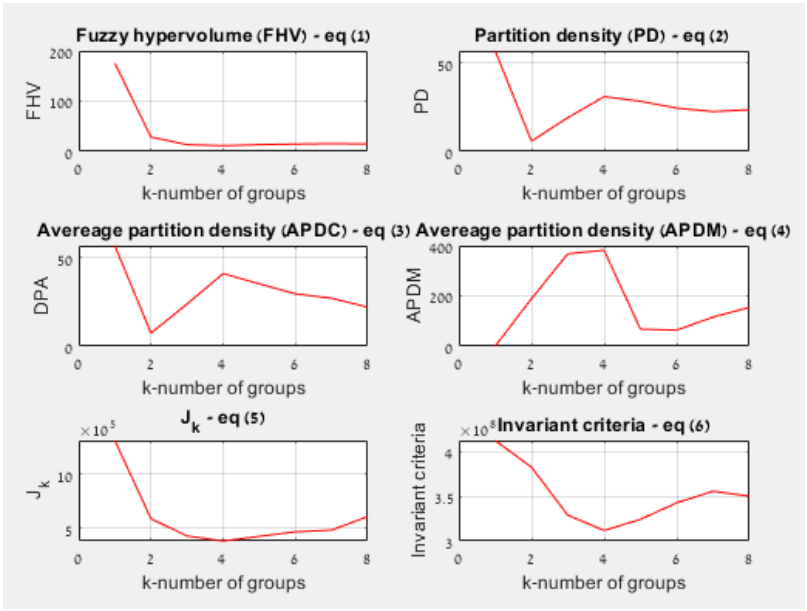


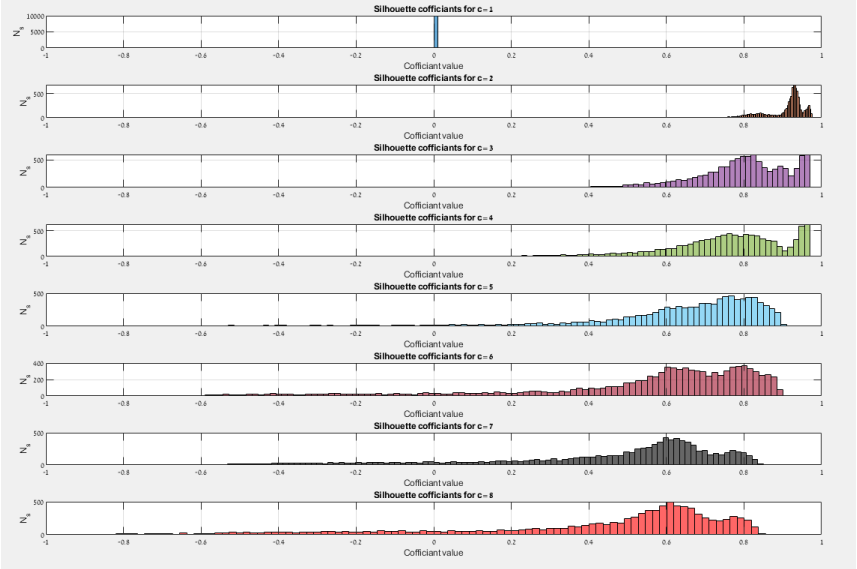
- *we can see that UOFC in the error table that the result is pretty good*
- *we can see in the 6 criteria graphs that every criteria show that the group of cluster are 3.*

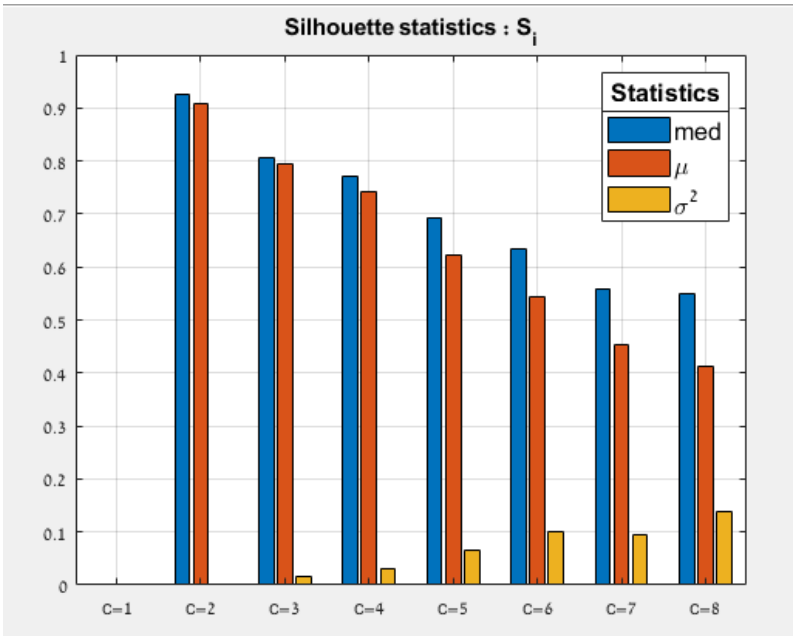
- *we can see in the silhouette coefficients graphs that the group of clusters are 3 because In this c, the coefficients closer to 1 (mu=1, sigma=0).*

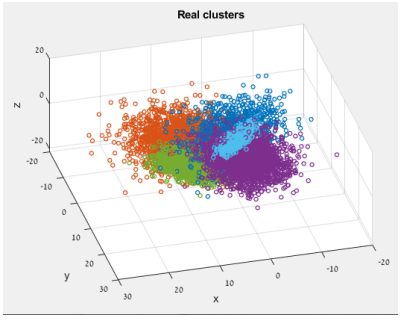
4.3.3.2 4 Gaussian with partial overlap in 3D

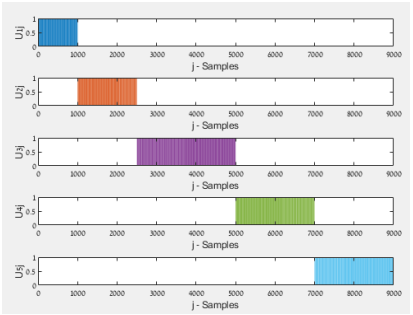


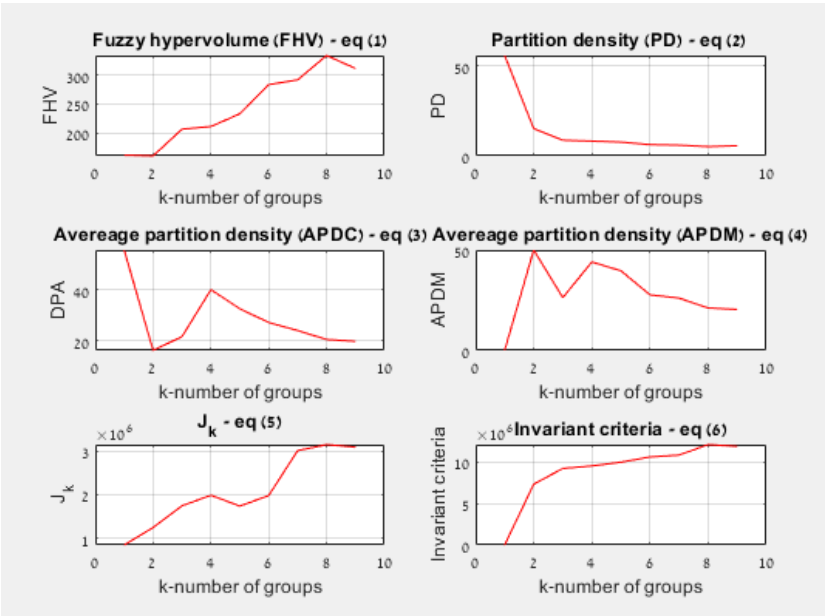



Silhouette cofficiants for c = 1
Silhouette cofficiants for c = 2
Silhouette cofficiants for c = 3
Silhouette cofficiants for c = 4
Silhouette cofficiants for c = 5
Silhouette cofficiants for c = 6
Silhouette cofficiants for c = 7
Silhouette cofficiants for c = 8
Cofficiant value


Silhouette statistics : S_i
Statistics
med
μ
σ²
C=1
C=2
C=3
C=4
C=5
C=6
C=7
C=8

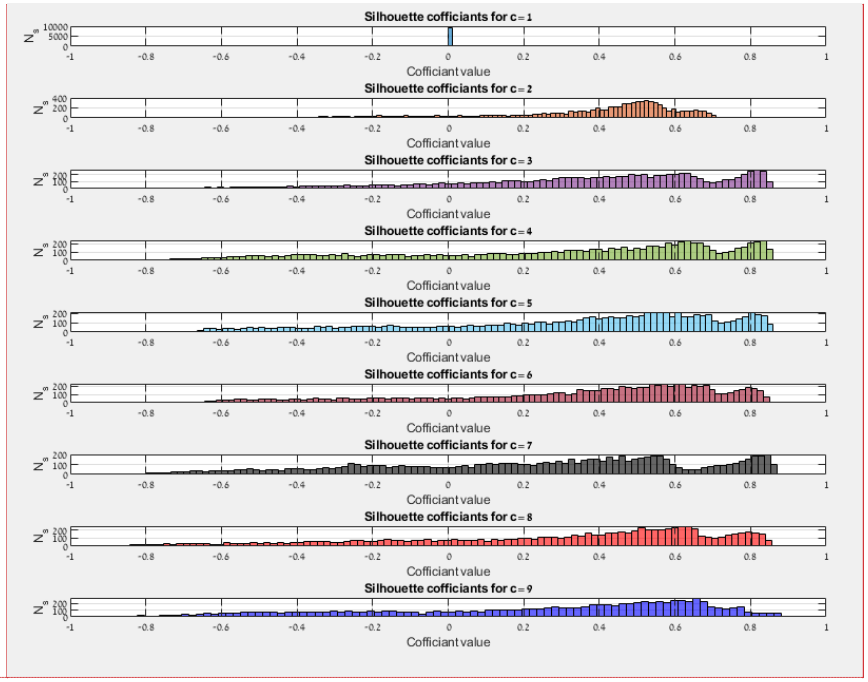

Real clusters
z
y
x

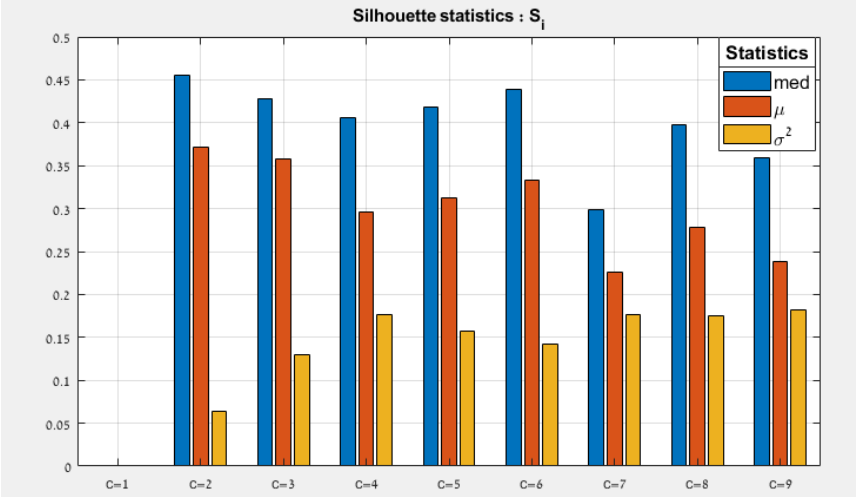

U1j
U2j
U3j
U4j
U5j
j - Samples


Fuzzy hypervolume (FHV) - eq (1)
FHV
k-number of groups
Partition density (PD) - eq (2)
PD
k-number of groups
Avereage partition density (APDC) - eq (3)
DPA
k-number of groups
Avereage partition density (APDM) - eq (4)
APDM
k-number of groups
J_k - eq (5)
J_k
k-number of groups
Invariant criteria - eq (6)
Invariant criteria
k-number of groups

[tg1] עם הערות:

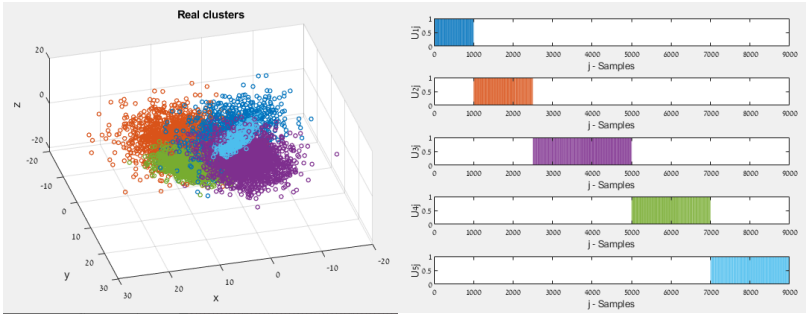
[tg2R1] עם הערות:

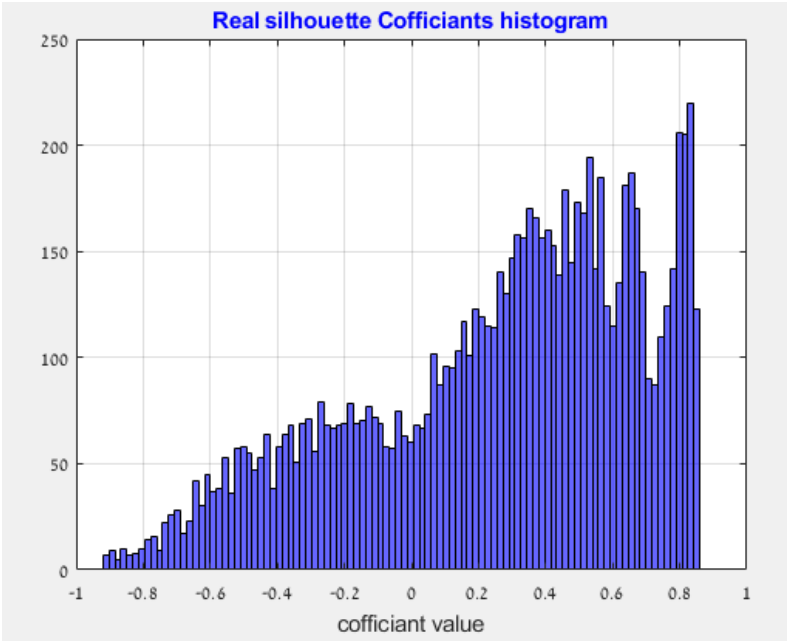


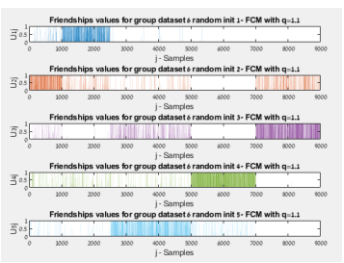


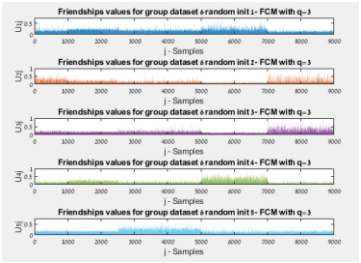
- *we can see that UOFC in the error table that the result is not pretty good because the overlap between clusters.*
- *we can see from the J_k graph that the number of groups are 5, but If more difficult to realize It from the other graphs (and also in the coefficients)*

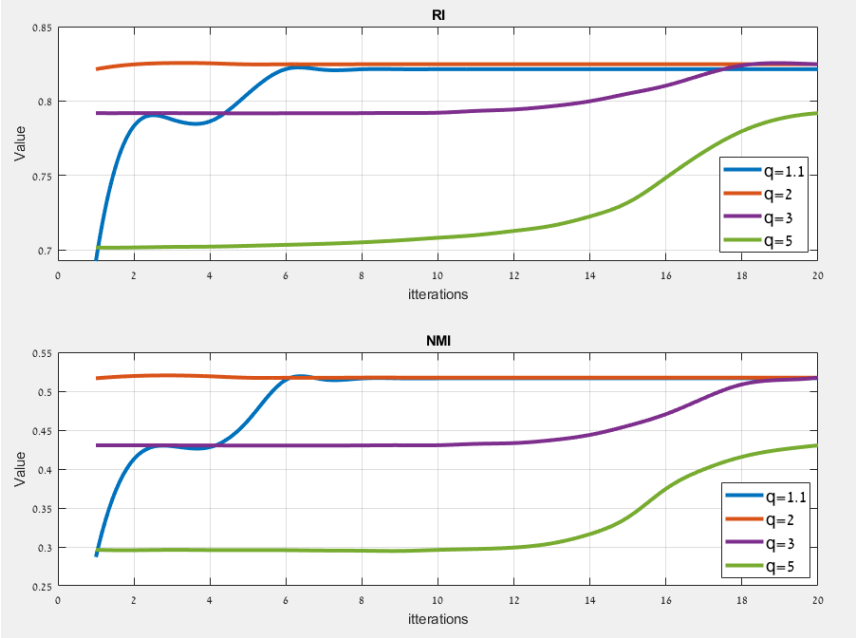
- *Now we assume that we know the number of groups and apply the fuzzy k-means algorithm and plot the results.*

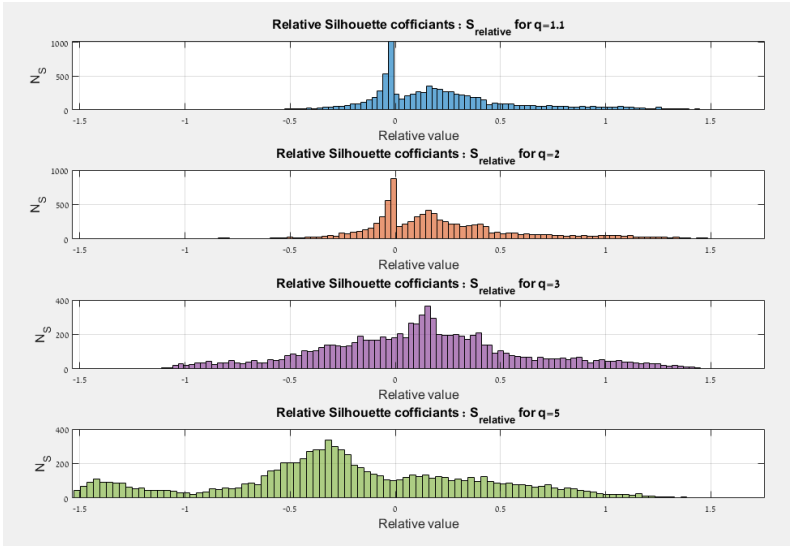


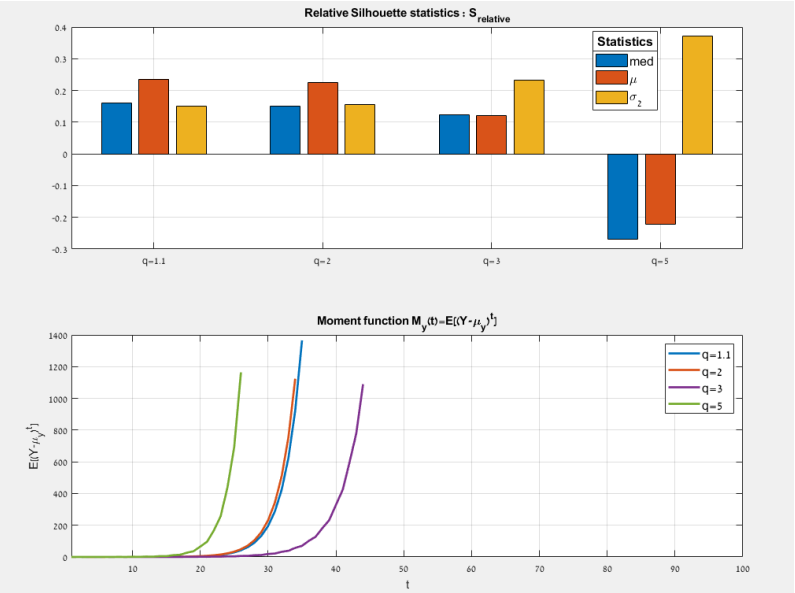



Friendships values for group dataset 6 random init 1- FCM with q=1.1
Friendships values for group dataset 6 random init 2- FCM with q=1.1
Friendships values for group dataset 6 random init 3- FCM with q=1.1
Friendships values for group dataset 6 random init 4- FCM with q=1.1
Friendships values for group dataset 6 random init 5- FCM with q=1.1
j - Samples


Friendships values for group dataset 6 random init 1- FCM with q=3
Friendships values for group dataset 6 random init 2- FCM with q=3
Friendships values for group dataset 6 random init 3- FCM with q=3
Friendships values for group dataset 6 random init 4- FCM with q=3
Friendships values for group dataset 6 random init 5- FCM with q=3
j - Samples

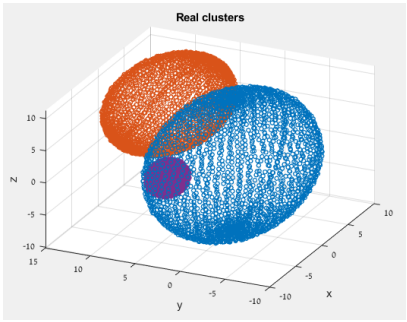

RI
Value
itterations
q=1.1
q=2
q=3
q=5
NMI
Value
itterations
q=1.1
q=2
q=3
q=5

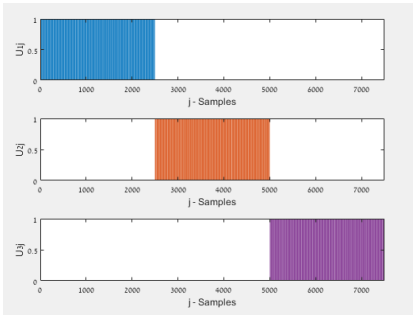


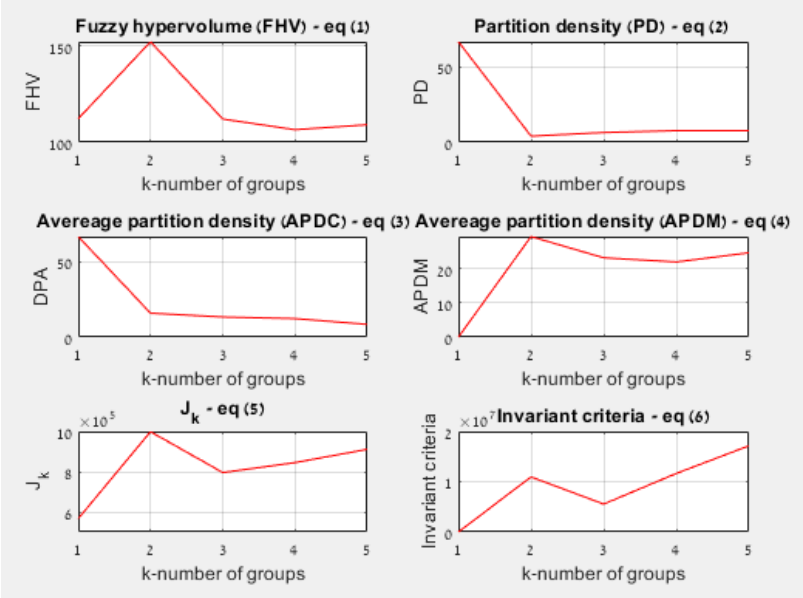


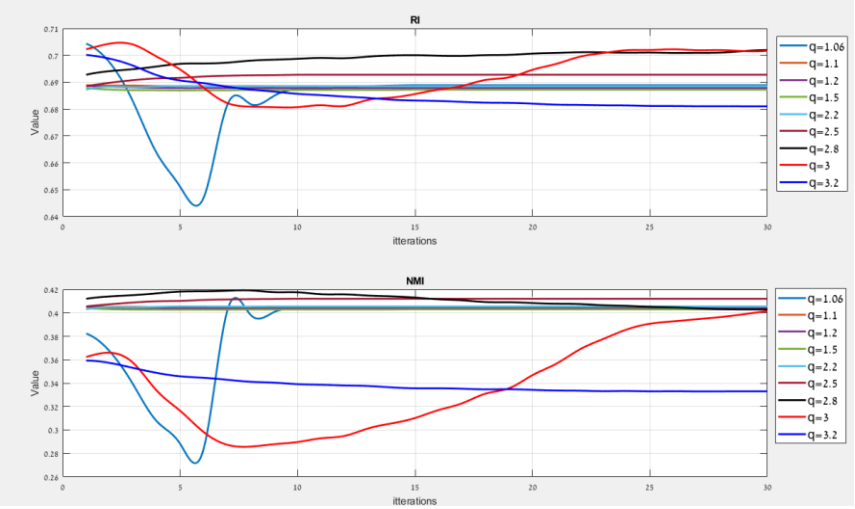
- *we can see that UOFC performance in the error table Is better when we take q (the fuzziness number) is equal to 2.*
- *In the q vs NMI&RI table we can see that there are some particular set of q that give us a better performance, and the trend of the graph as we took more large values of q, is worsen our result.*
- *Big q values blear the graph and then It's hard to allocate the optimum point.*

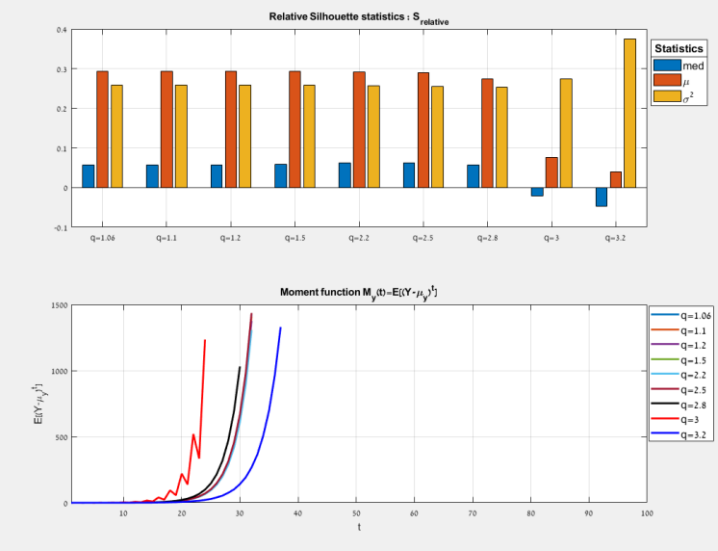
4.3.3.3 4 Spheres in 3D





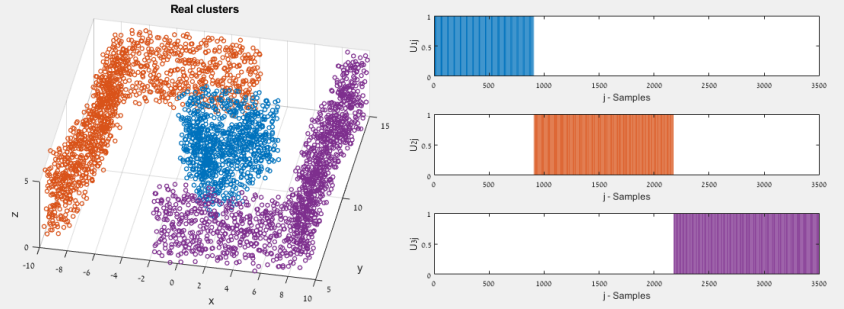


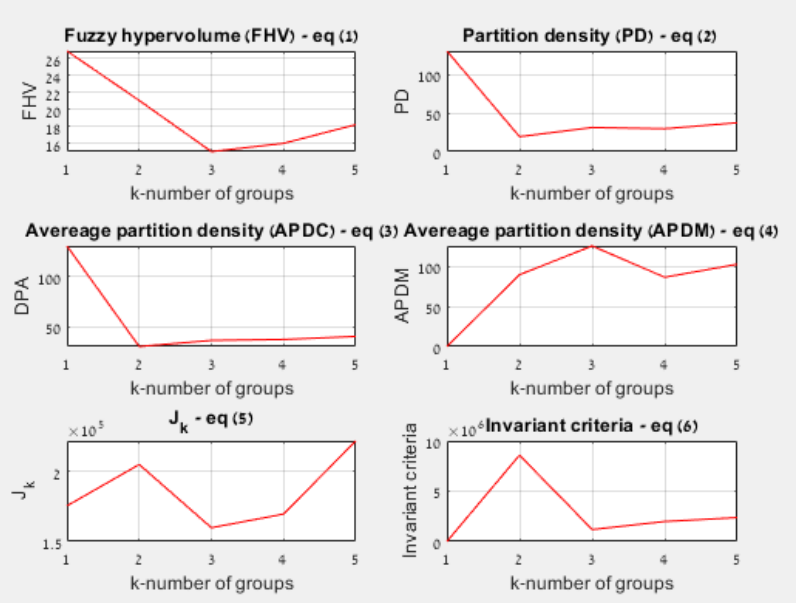


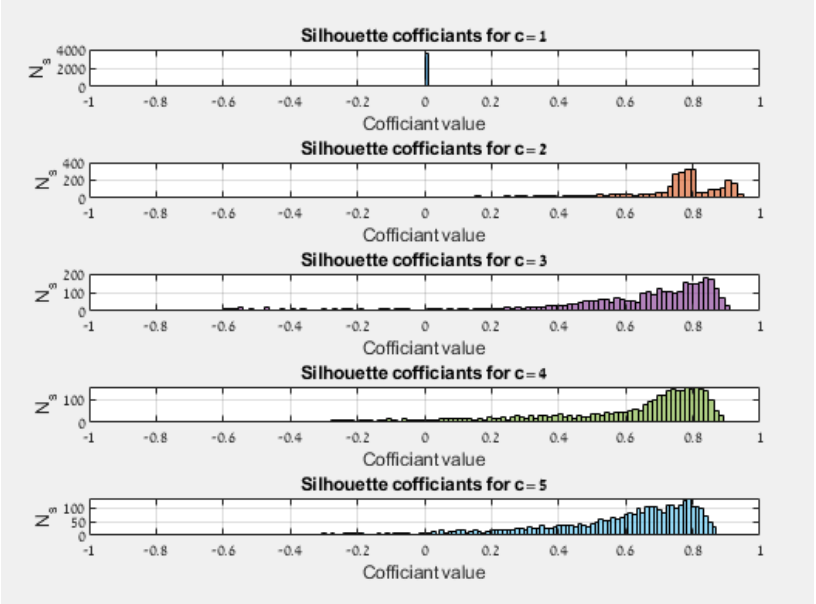


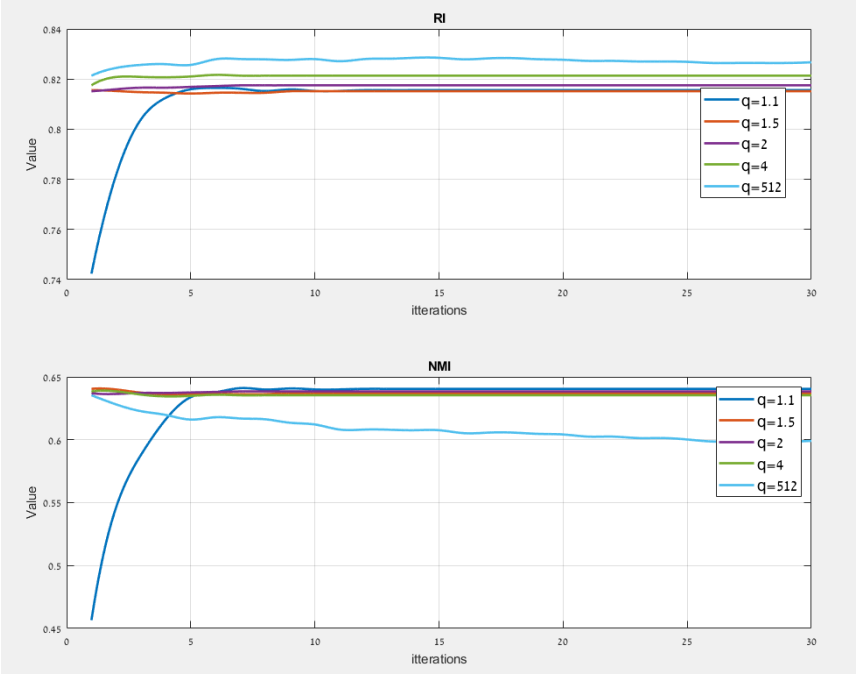
- *we can see that UOFC in the error table that the result is not so good-Its mean that the UOFC cannot perform on hollow sphere shape.*
- *In filled un sphere we can assume that the performance will be better*
- *we can see in the 6 criteria graphs that every criterion show that the group of cluster are 4-so we can say that the UOFC can predict the number of cluster even In non-Guessing look alike DS.*

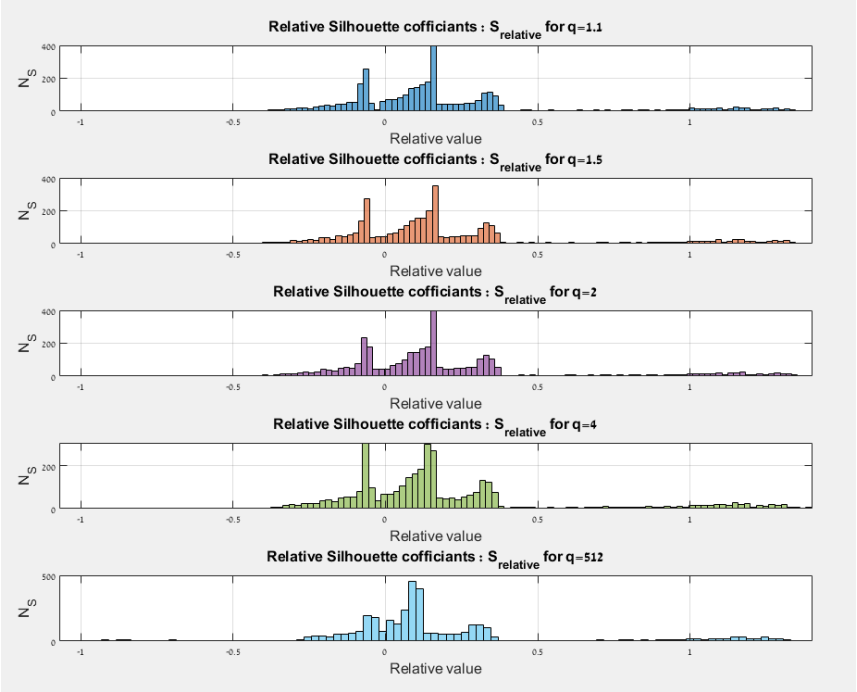
4.3.3.4 3D shape

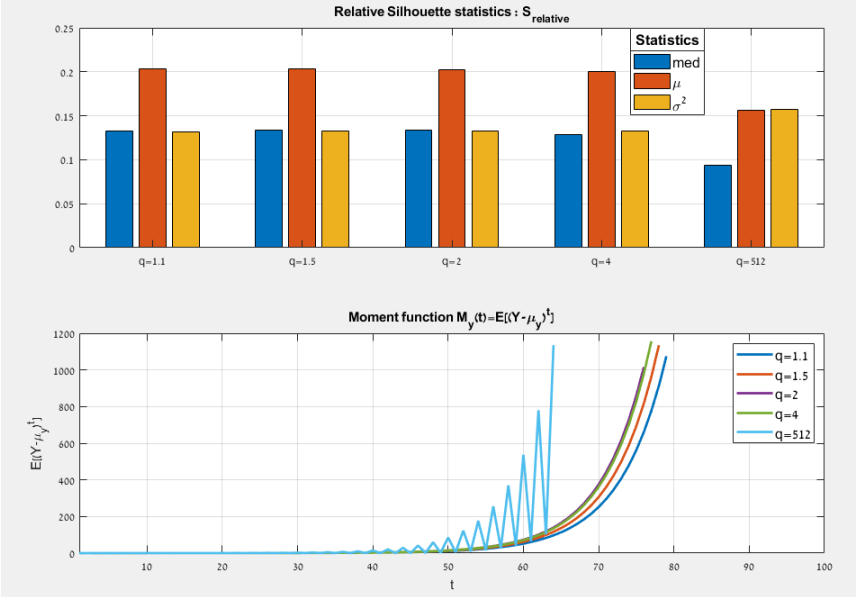





Silhouette cofficiants for c = 1
Silhouette cofficiants for c = 2
Silhouette cofficiants for c = 3
Silhouette cofficiants for c = 4
Silhouette cofficiants for c = 5
Cofficiant value
N_s


RI
NMI
Value
itterations
q=1.1
q=1.5
q=2
q=4
q=512





- *we can see that UOFC in the error table that the result is not bad-Its mean that*

the UOFC can't perform on non-trivial DS.

- *we can see in the 6 criteria graphs that every criterion show a different solution.*

4.4 Conclusion

4.4.1. The UOFC (with FMLE) preform very good on gaussian DS.

4.4.2. There is a pricy computational load when dealing with large data sets- the running time take me approximately one second for one iteration i.e. the calculation of U, d, p-centroid. for example: for 3,500 DS the program run for 3,500 sec (approximately 1 hour).

4.4.3. Taking a different value of q for each attempt (4.3.3.2) cluster we can clearly see the difference in performance. when q=1 an harder clustering has been made and where we took q = 2 the clustering becomes more fuzzy and we can more agile with the clustering technic in (4.3.3.2) example q=2 perform better.

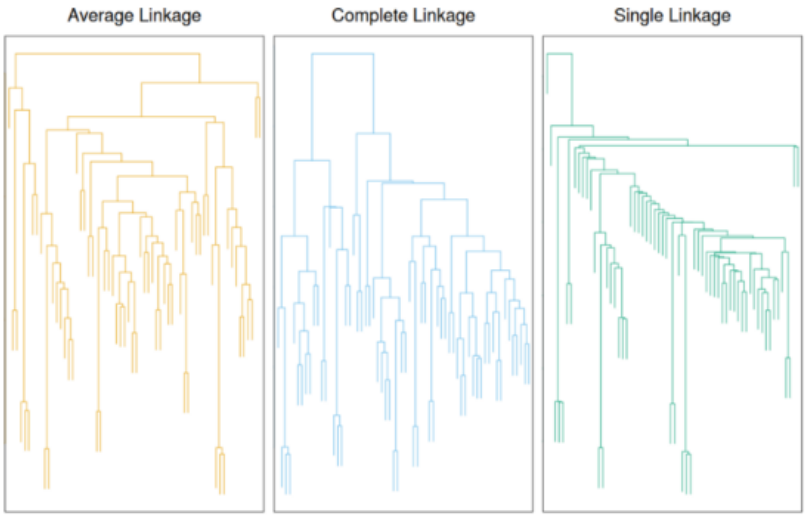
5. Exercise '3'-Agglomartive Clustering -AHC

Up to now, our methods have formed disjoint clusters—in engineering /computer science terminology, we would say that the data description is "flat."
However, there are many times when clusters have sub clusters, these have sub-sub clusters, and so on.

The procedures themselves can be divided to - agglomerative and divisive. Agglomerative is bottom-up (or clumping) procedures- start with n singleton clusters and form the sequence by successively merging clusters. Divisive is top-down (or splitting) procedures- start with all of the samples in one cluster and form the sequence by successively splitting cluster.

Hierarchical clustering does not require an initial number of clusters.

The most common type of hierarchical clustering is the bottom up approach, and we will demonstrate and practice this procedure. We can Illustrated the agglomerative clustering using dendrogram (figure below).



Agglomerative clustering starting from the leaves and combining clusters up to the trunk. It starts by defining a ***dissimilarity measure*** *between each pair of observations, like the Euclidean distance, then, it starts by assuming that each observation pertains to its own cluster ,then the two most similar clusters are fused, so that there are*

n-1 clusters. Afterwards, other two similar clusters are fused, resulting in n-2 clusters. The process is repeated iteratively until all observations are part of a single cluster. Although simple, something was not addressed.

How to define a dissimilarity measure between clusters? This is achieved with the concept of linkage. The five most common types of linkage are summarized in the table below:

$$
\begin{aligned}
d_{min}(\mathcal{D}_i, \mathcal{D}_j) &= \min_{\substack{\mathbf{x}\in\mathcal{D}_i \\ \mathbf{x}'\in\mathcal{D}_j}} \|\mathbf{x}-\mathbf{x}'\| \\
d_{max}(\mathcal{D}_i, \mathcal{D}_j) &= \max_{\substack{\mathbf{x}\in\mathcal{D}_i \\ \mathbf{x}'\in\mathcal{D}_j}} \|\mathbf{x}-\mathbf{x}'\| \\
d_{avg}(\mathcal{D}_i, \mathcal{D}_j) &= \frac{1}{n_i n_j} \sum_{\mathbf{x}\in\mathcal{D}_i} \sum_{\mathbf{x}'\in\mathcal{D}_j} \|\mathbf{x}-\mathbf{x}'\| \\
d_{mean}(\mathcal{D}_i, \mathcal{D}_j) &= \|\mathbf{m}_i - \mathbf{m}_j\|.
\end{aligned}
$$

$$d_e(\mathcal{D}_i, \mathcal{D}_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}} \|\mathbf{m}_i - \mathbf{m}_j\|$$

Complete, average, and centroid are the most popular types of linkage, because single linkage tends to yield unbalanced dendrograms.

1-d_{min}- used to measure the distance between clusters-take the Minimum distance from the point in the cluster to the other cluster.

2. d_{max}- used to measure the distance between cluster- take the Maximum distance from the point in the cluster to the other cluster. (still gather group with minimum distances).

3. d_{avg}- used to measure the distance between clusters-calculate the average distances from one cluster to all the point in the other cluster .

4. d_{mean}- used to measure the distance between clusters means.

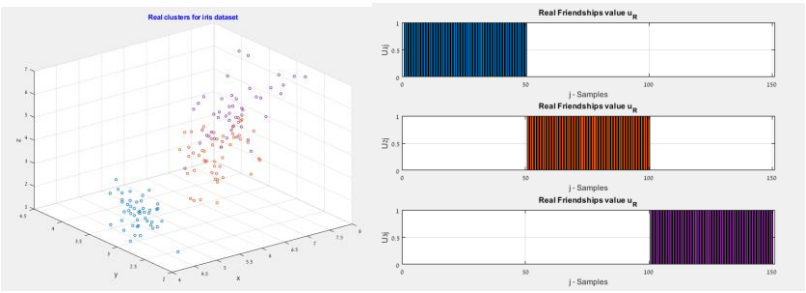
5. d_e-sum of square error criteria-we merge the cluster that have both small distance between cluster means and change in the most little chang In the number of the data sets of the two cluster.

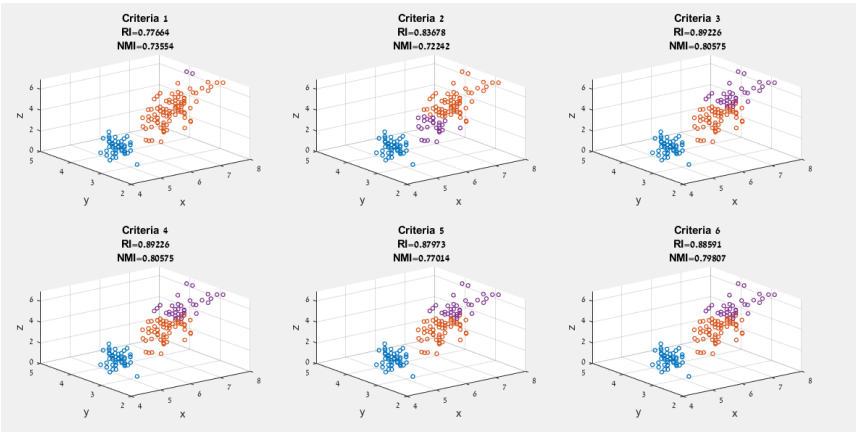
5.1 Algorithm description

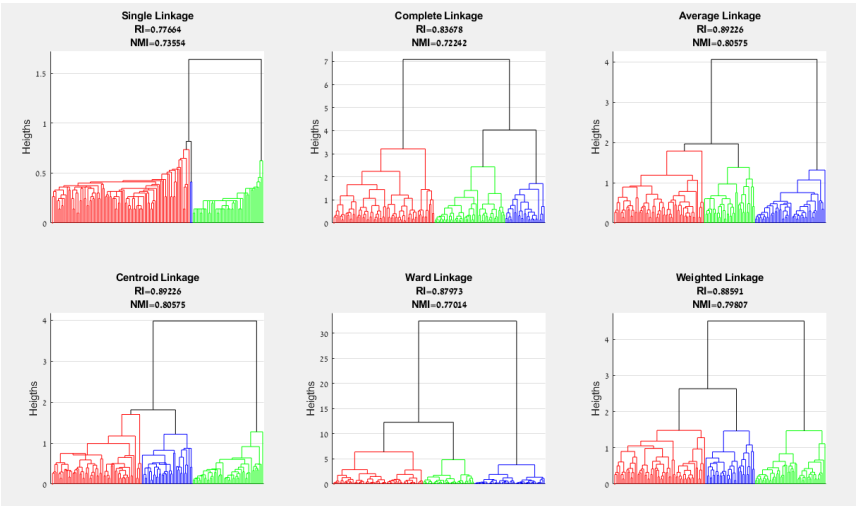
```
1 begin initialize c, ĉ ← n, D_i ← {x_i}, i = 1, ..., n
2         do ĉ ← ĉ − 1
3             Find nearest clusters, say, D_i and D_j
4             Merge D_i and D_j
5         until c = ĉ
6   return c clusters
7 end
```

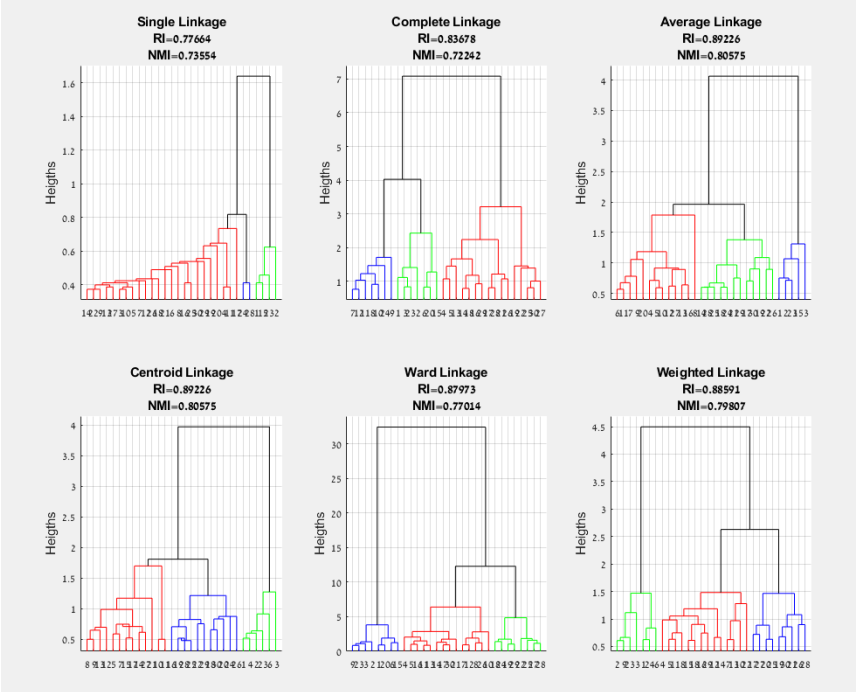
5.3 Results

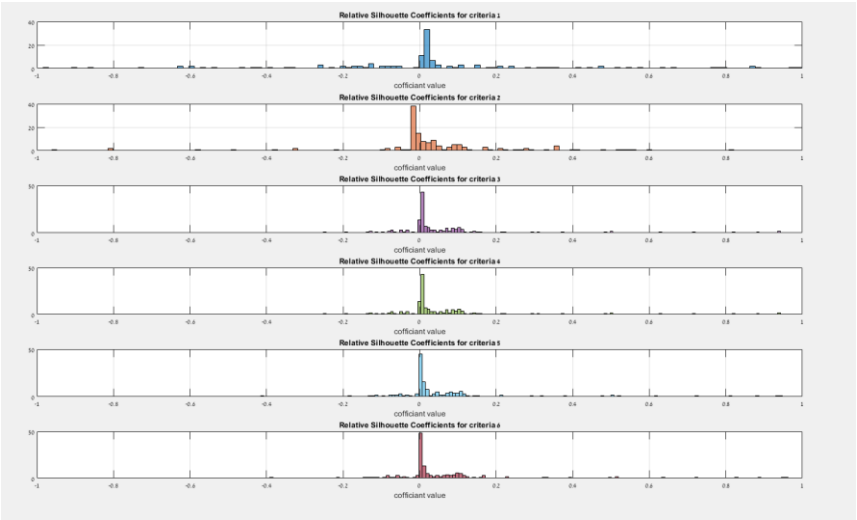
5.3.1 Fisher Iris dataset

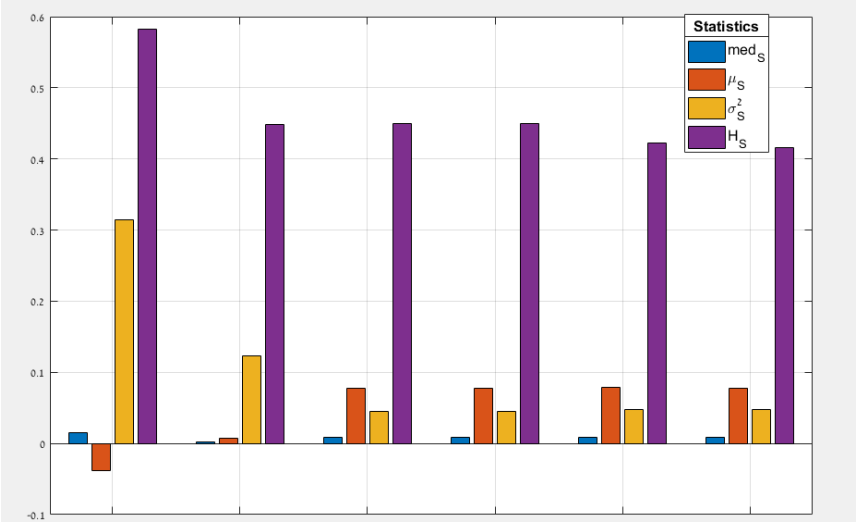






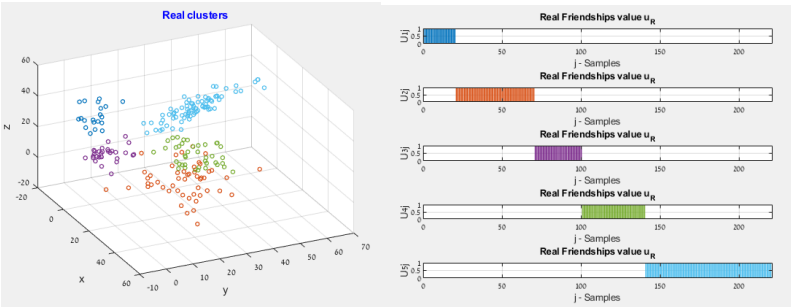

Single Linkage
RI=0.77664
NMI=0.73554
Complete Linkage
RI=0.83678
NMI=0.72242
Average Linkage
RI=0.89226
NMI=0.80575
Centroid Linkage
RI=0.89226
NMI=0.80575
Ward Linkage
RI=0.87973
NMI=0.77014
Weighted Linkage
RI=0.88591
NMI=0.79807
Heigths

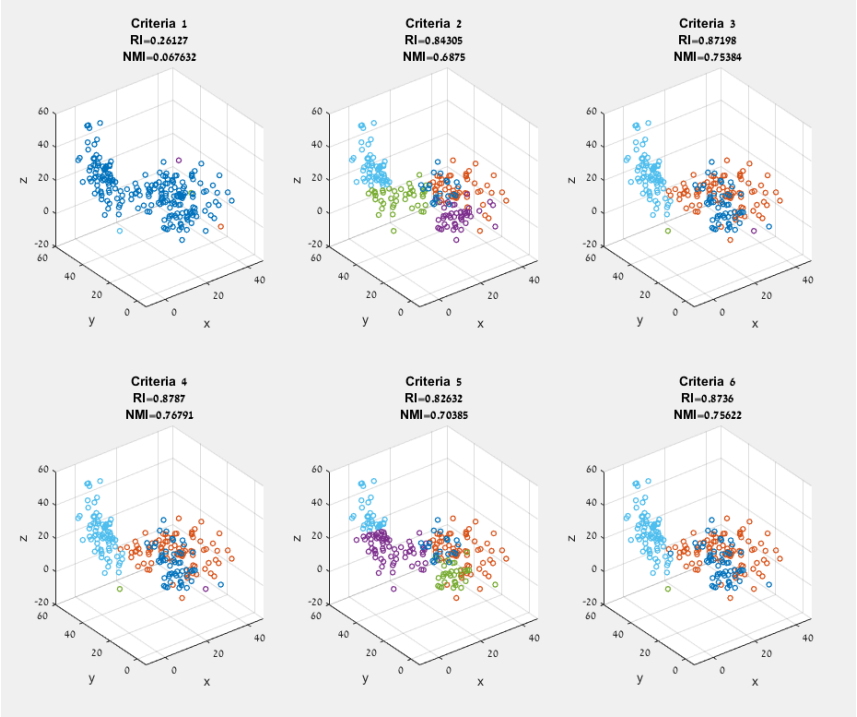

Relative Silhouette Coefficients for criteria 1
Relative Silhouette Coefficients for criteria 2
Relative Silhouette Coefficients for criteria 3
Relative Silhouette Coefficients for criteria 4
Relative Silhouette Coefficients for criteria 5
Relative Silhouette Coefficients for criteria 6
cofficiant value

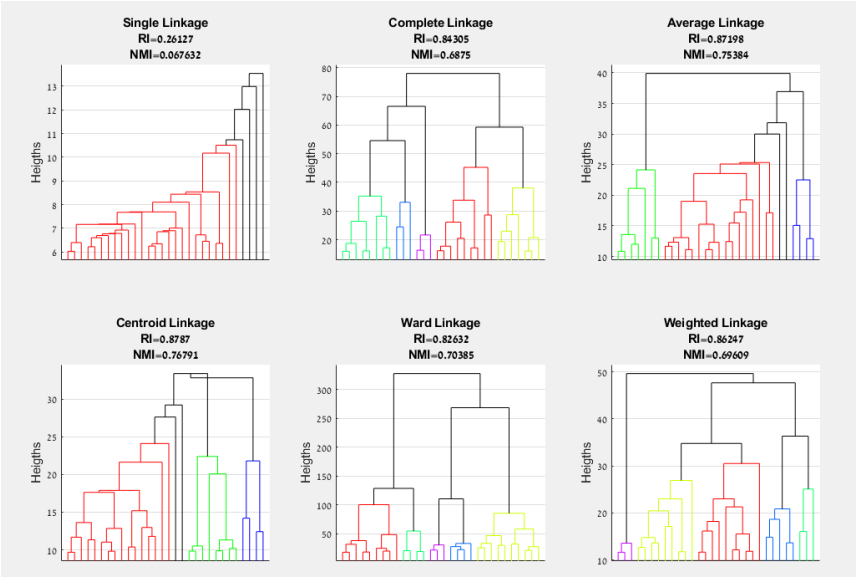


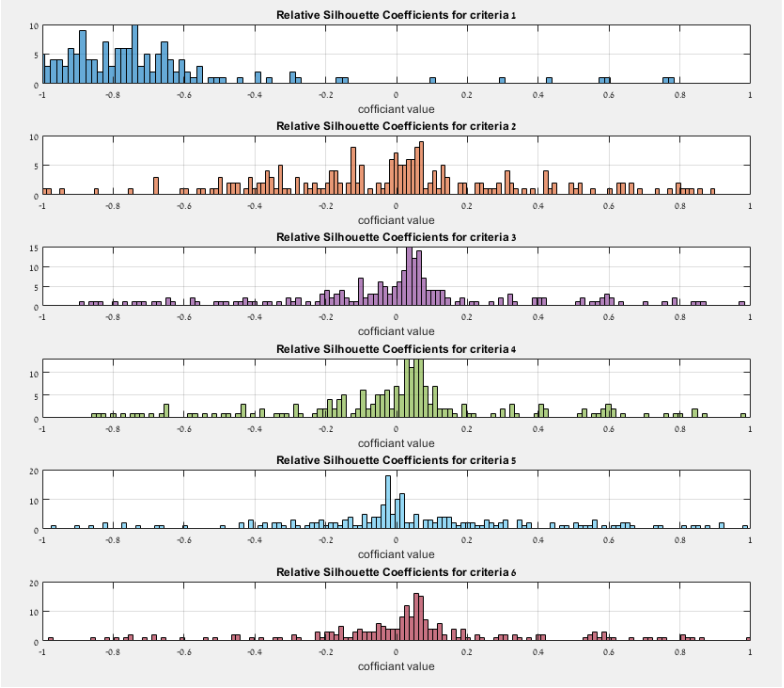
- *In this case the ward linkage (criteria 5) give the best results but the other results are very close, we can see It In RI, NMI values for each linkage and In the statistics of the relative silhouette coefficients.*
- *The simple linkage (criteria 1) Is bad for those data set because he depends on the minimum distance between clusters (i.e. : the closet samples for each clusters) and In Iris data set there Is 2 cluster that close to each other and In this case simple linkage will merge those cluster Into 1 cluster and reduce the accuracy.*

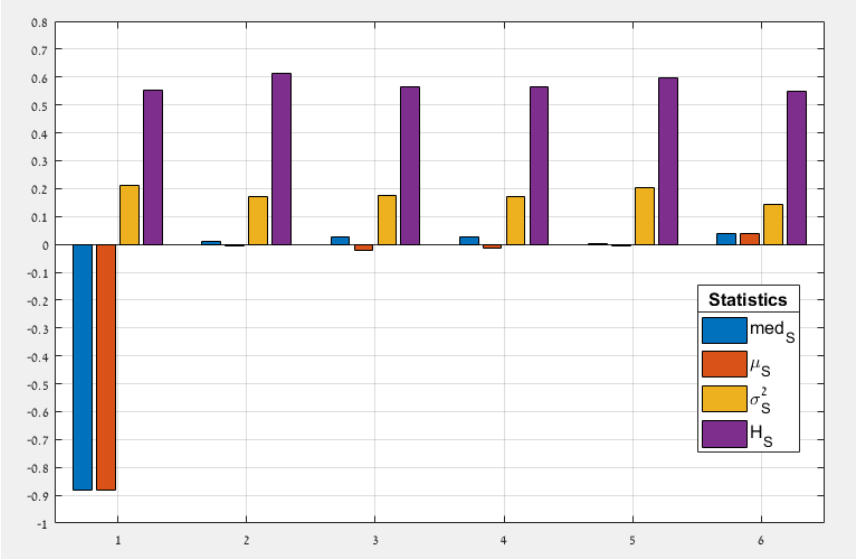
5.3.2 Gauss dataset




Criteria 1
RI=0.26127
NMI=0.067632
Criteria 2
RI=0.84305
NMI=0.6875
Criteria 3
RI=0.87198
NMI=0.75384
Criteria 4
RI=0.8787
NMI=0.76791
Criteria 5
RI=0.82632
NMI=0.70385
Criteria 6
RI=0.8736
NMI=0.75622
x
y
z

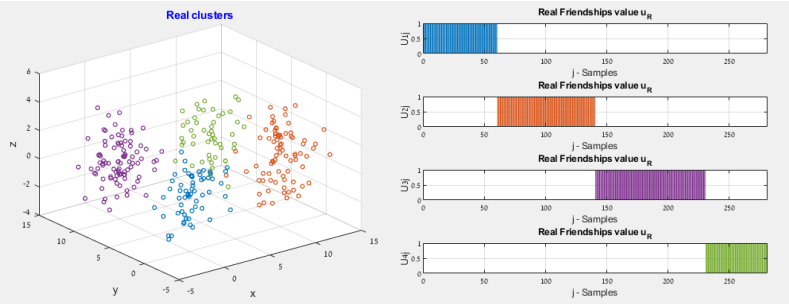

Single Linkage
RI=0.26127
NMI=0.067632
Complete Linkage
RI=0.84305
NMI=0.6875
Average Linkage
RI=0.87198
NMI=0.75384
Centroid Linkage
RI=0.8787
NMI=0.76791
Ward Linkage
RI=0.82632
NMI=0.70385
Weighted Linkage
RI=0.86247
NMI=0.69609
Heigths

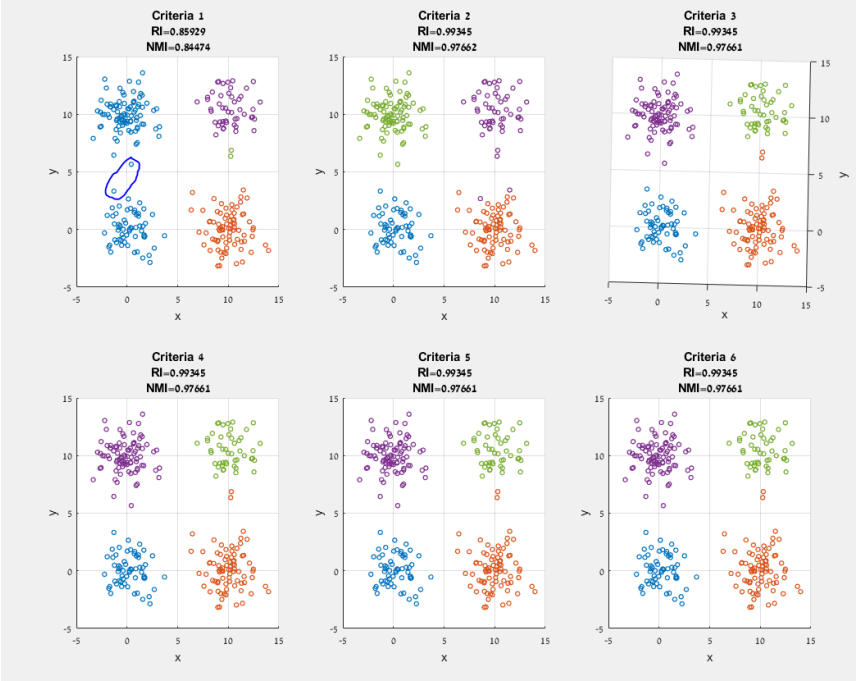

Relative Silhouette Coefficients for criteria 1
cofficiant value
Relative Silhouette Coefficients for criteria 2
cofficiant value
Relative Silhouette Coefficients for criteria 3
cofficiant value
Relative Silhouette Coefficients for criteria 4
cofficiant value
Relative Silhouette Coefficients for criteria 5
cofficiant value
Relative Silhouette Coefficients for criteria 6
cofficiant value

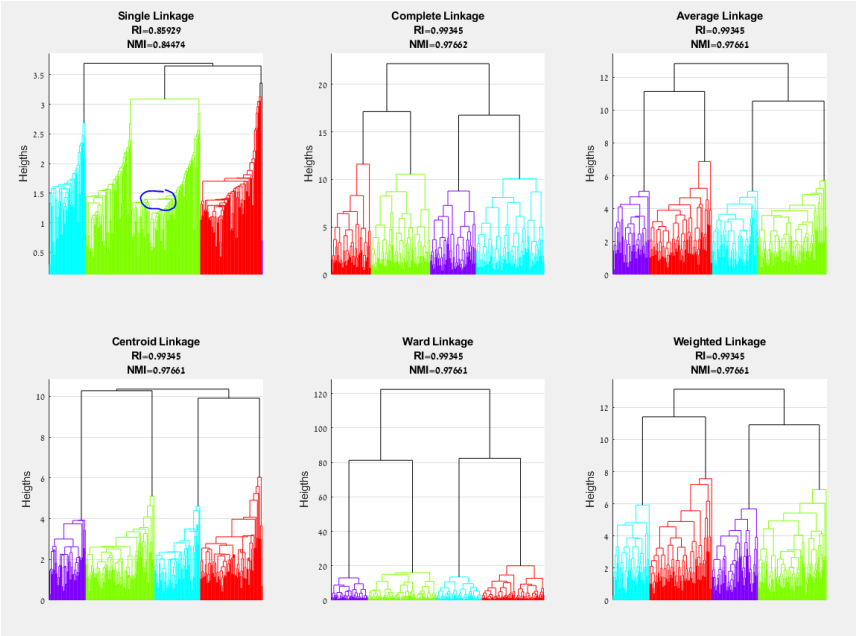

Statistics
med S
μS
σ²S
HS

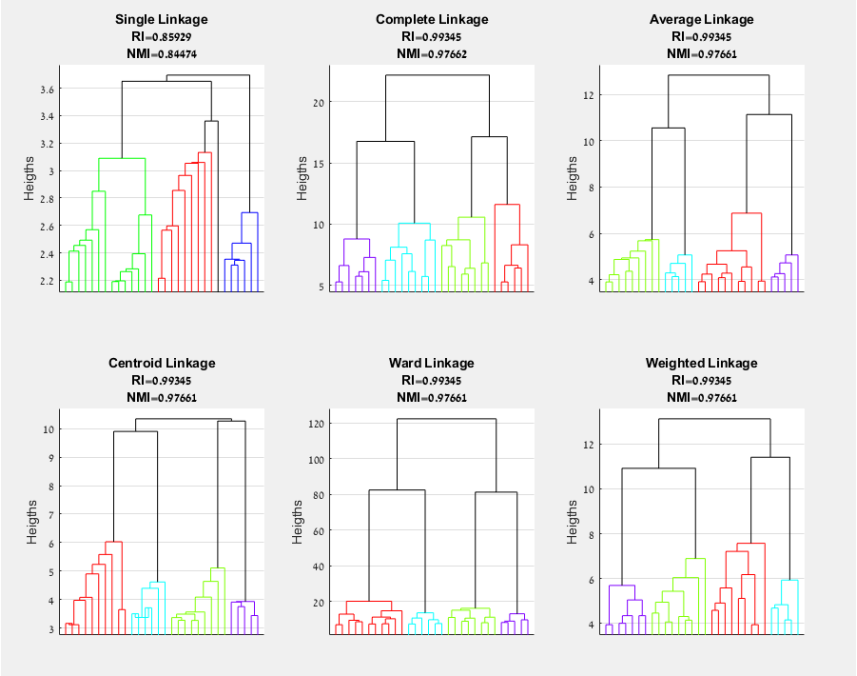
- *In this case the centroid linkage (criteria 4) give the best results because our dataset generate from gauss distribution with constant centers so the centroids can represent pretty good the clusters and maximize the accuracy but the other results are very close, we can see It In RI,NMI values for each linkage and In the statistics of the relative silhouette coefficients.*
- *In this case the variance Is big, so the centroid linkage merge between closet centers and we can se It In the unstable dendrogram graph.*
- *Also, the average linkage (criteria 3) give a good results because we choose a gauss distributions and the average distance Is closet to the variance of the clusters and In large number of sample It will be like a centroid linkage.*
- *The complete and ward linkage (criteria's 2,5) doesn't merge the cluster and the dendrogram graph Is stable because It depend on some weighted centroids (or maximum distance) and close samples don't be necessary In one cluster. but the accuracy Is low because those criteria do not represent the distribution that our cluster generate from.*

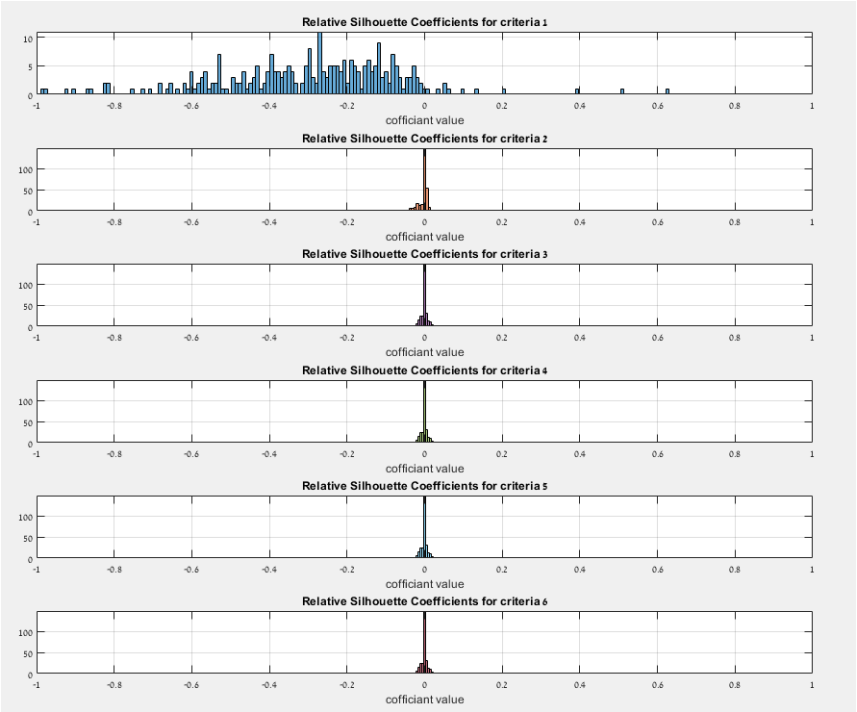
5.3.3 dense Gauss dataset

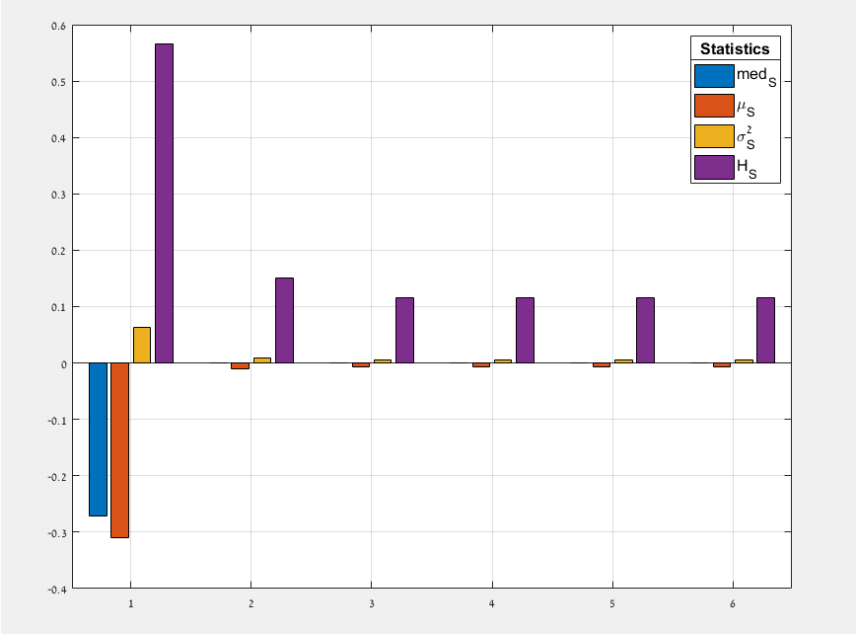



Criteria 1
RI=0.85929
NMI=0.84474
Criteria 2
RI=0.99345
NMI=0.97662
Criteria 3
RI=0.99345
NMI=0.97661
Criteria 4
RI=0.99345
NMI=0.97661
Criteria 5
RI=0.99345
NMI=0.97661
Criteria 6
RI=0.99345
NMI=0.97661
x
y


Single Linkage
RI=0.85929
NMI=0.84474
Complete Linkage
RI=0.99345
NMI=0.97662
Average Linkage
RI=0.99345
NMI=0.97661
Centroid Linkage
RI=0.99345
NMI=0.97661
Ward Linkage
RI=0.99345
NMI=0.97661
Weighted Linkage
RI=0.99345
NMI=0.97661
Heights

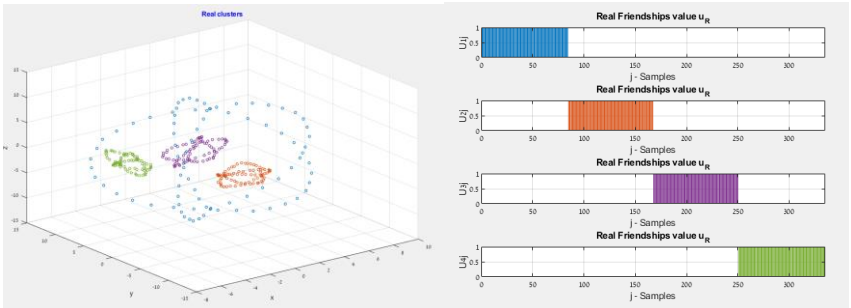

Single Linkage
RI=0.85929
NMI=0.84474
Complete Linkage
RI=0.99345
NMI=0.97662
Average Linkage
RI=0.99345
NMI=0.97661
Centroid Linkage
RI=0.99345
NMI=0.97661
Ward Linkage
RI=0.99345
NMI=0.97661
Weighted Linkage
RI=0.99345
NMI=0.97661
Heights

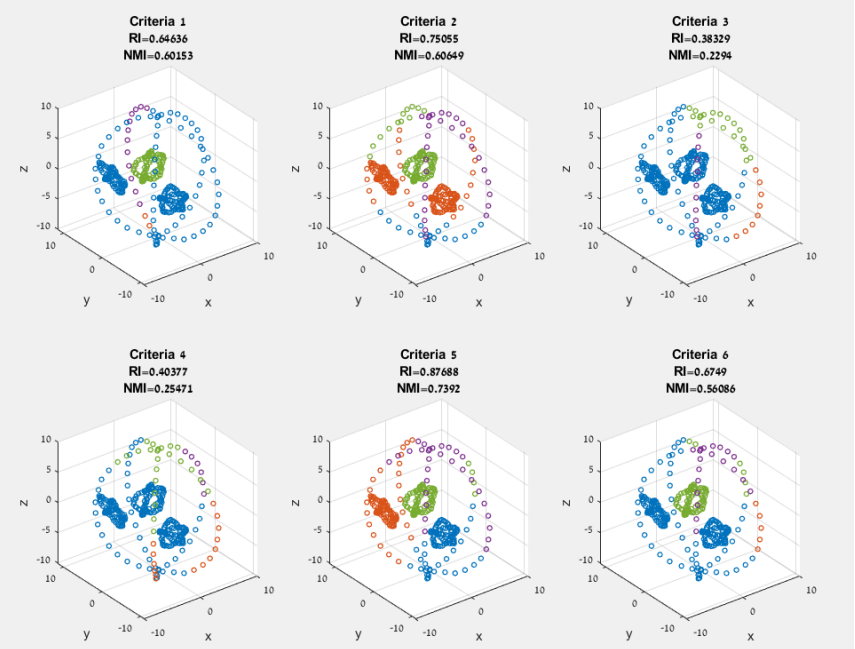

Relative Silhouette Coefficients for criteria 1
Relative Silhouette Coefficients for criteria 2
Relative Silhouette Coefficients for criteria 3
Relative Silhouette Coefficients for criteria 4
Relative Silhouette Coefficients for criteria 5
Relative Silhouette Coefficients for criteria 6
cofficiant value

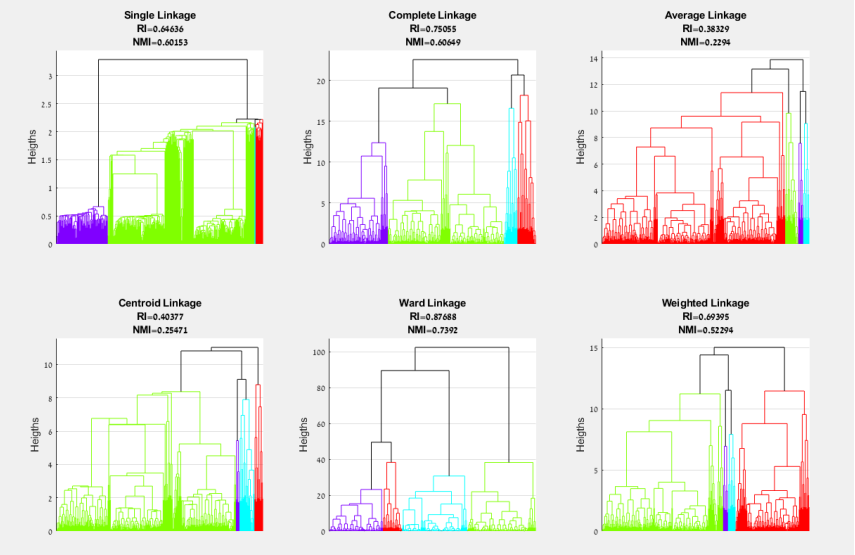


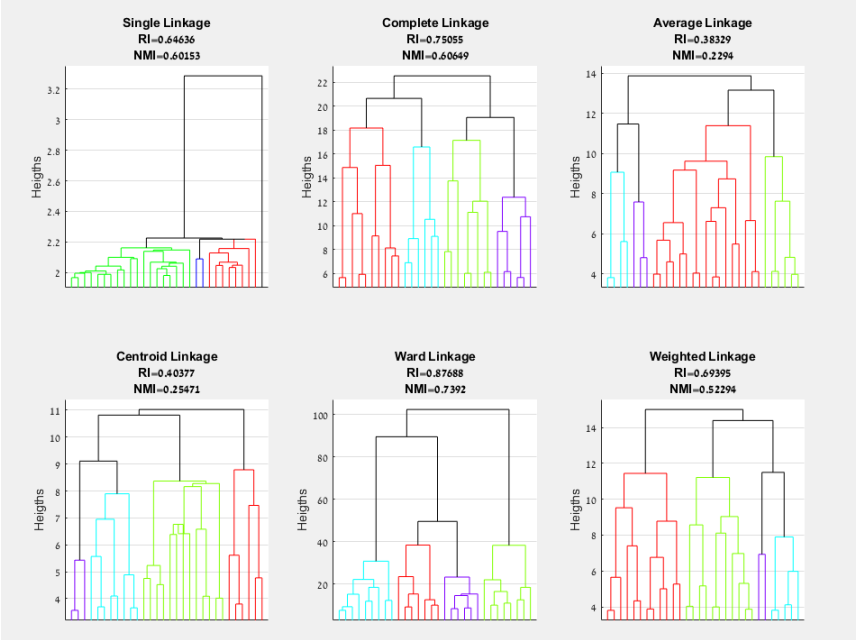
- *The 2 points in blue circle In simple linkage (criteria 1) cause the merge between 2 gaussians (the minimum distance Is low) and reduce the accuracy.*
- *The other linkages Is same all almost give a 100% accuracy.*

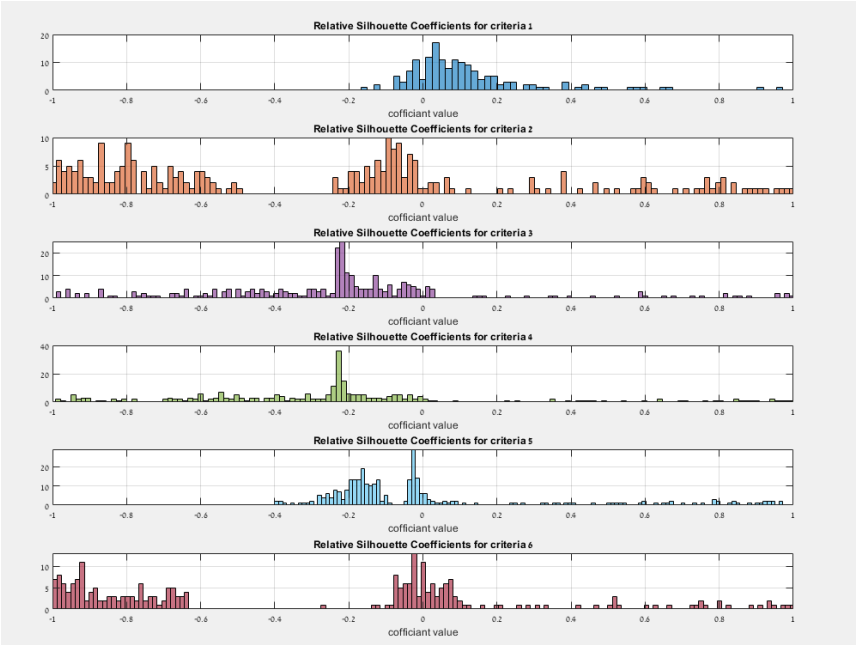
5.3.4 "Circle balls 3D"

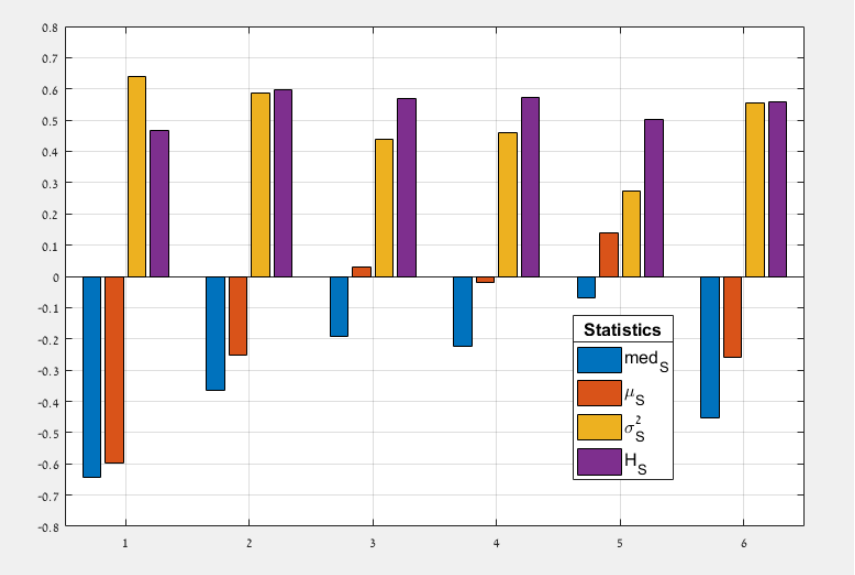



Criteria 1
RI=0.64636
NMI=0.60153
Criteria 2
RI=0.75055
NMI=0.60649
Criteria 3
RI=0.38329
NMI=0.2294
Criteria 4
RI=0.40377
NMI=0.25471
Criteria 5
RI=0.87688
NMI=0.7392
Criteria 6
RI=0.6749
NMI=0.56086


Single Linkage
RI=0.64636
NMI=0.60153
Complete Linkage
RI=0.75055
NMI=0.60649
Average Linkage
RI=0.38329
NMI=0.2294
Centroid Linkage
RI=0.40377
NMI=0.25471
Ward Linkage
RI=0.87688
NMI=0.7392
Weighted Linkage
RI=0.69395
NMI=0.52294
Heights

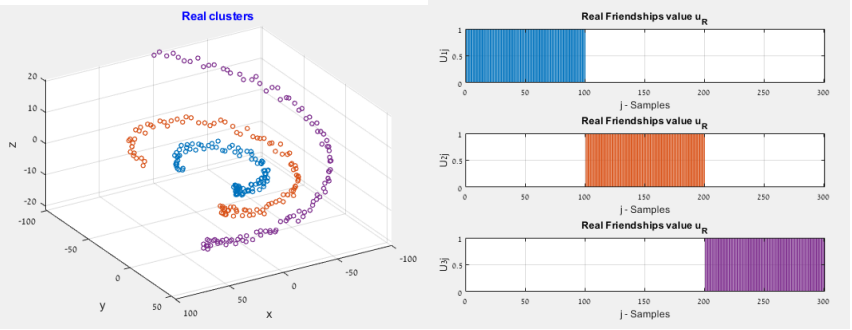

Single Linkage
RI=0.64636
NMI=0.60153
Complete Linkage
RI=0.75055
NMI=0.60649
Average Linkage
RI=0.38329
NMI=0.2294
Centroid Linkage
RI=0.40377
NMI=0.25471
Ward Linkage
RI=0.87688
NMI=0.7392
Weighted Linkage
RI=0.69395
NMI=0.52294
Heigths

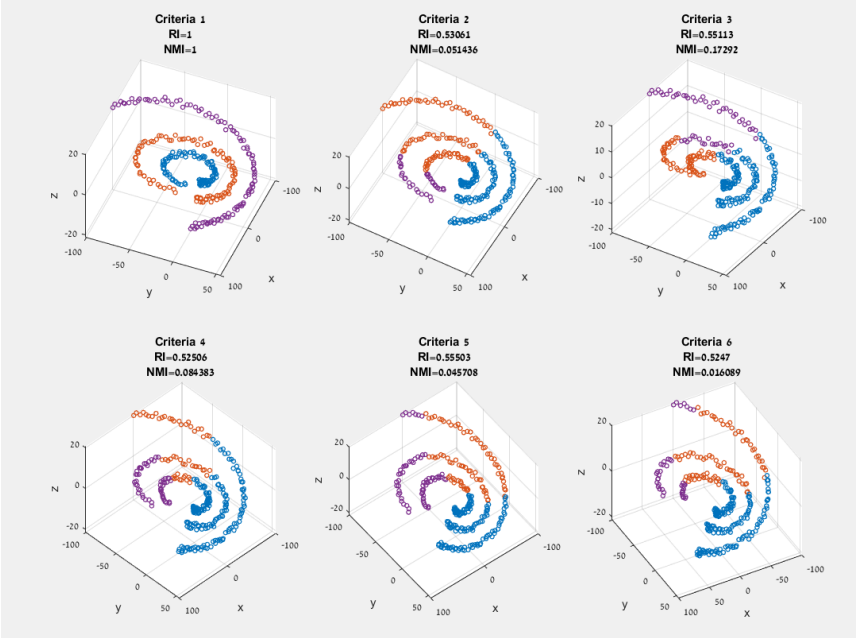

Relative Silhouette Coefficients for criteria 1
Relative Silhouette Coefficients for criteria 2
Relative Silhouette Coefficients for criteria 3
Relative Silhouette Coefficients for criteria 4
Relative Silhouette Coefficients for criteria 5
Relative Silhouette Coefficients for criteria 6
cofficiant value

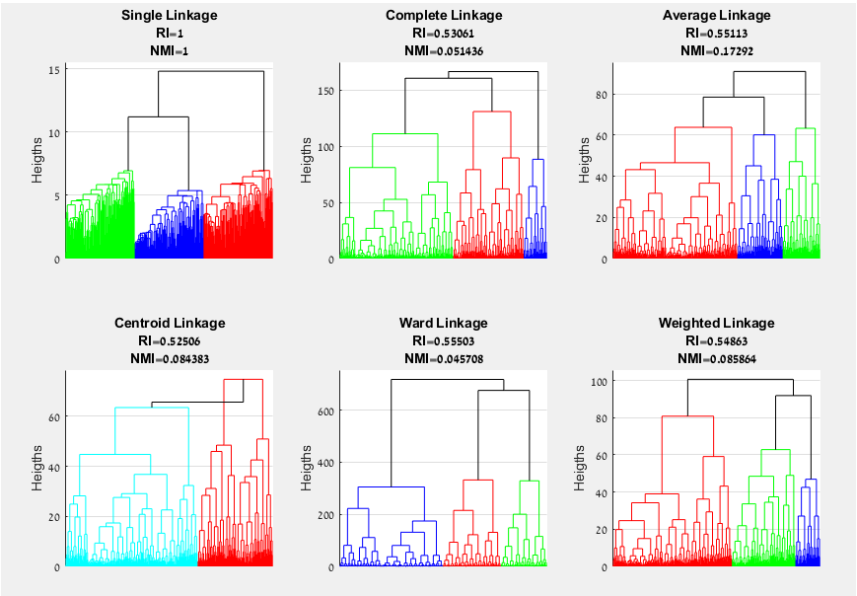


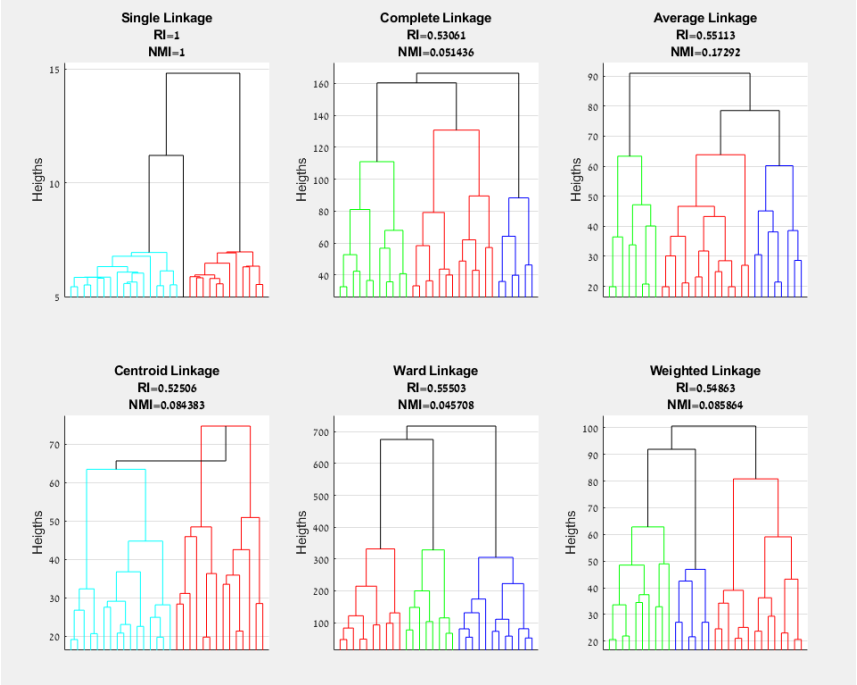
- *In this case the ward linkage (criteria 5) give the best results because our dataset generates from circles and the weighted centroid distances (or squared distances) are represent pretty good the generation of this dataset.*
- *Because the centers of our dataset close each other, the simple, average and centroid linkages give as bad results.*
- *The big circle Improve the complete and ward linkage because those samples (In the big circle) prevent the merge or samples from other cluster with close centers.*

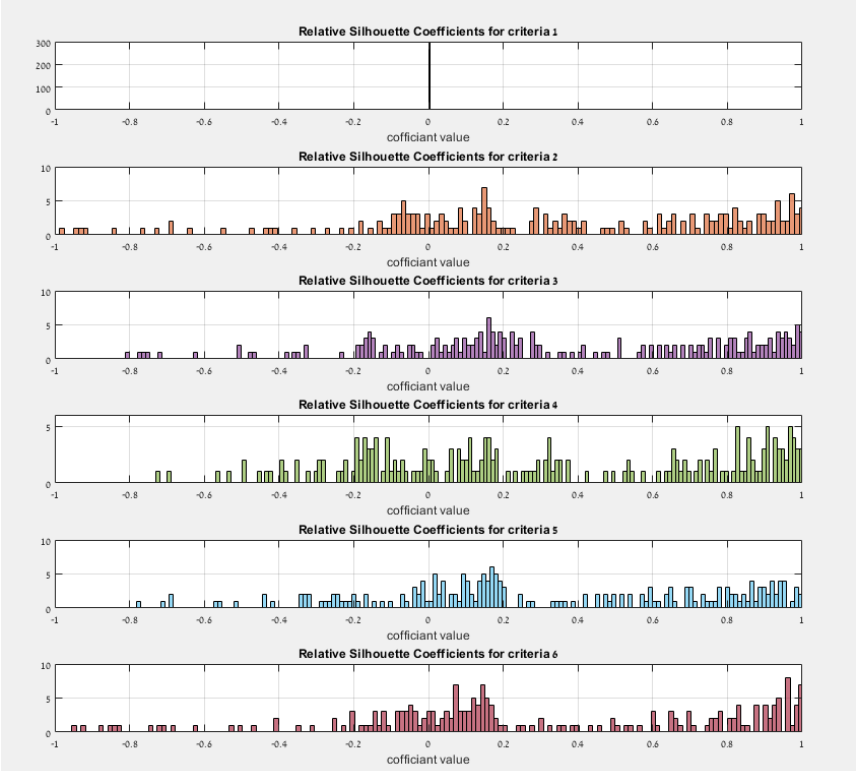
5.3.4 Spiral dataset

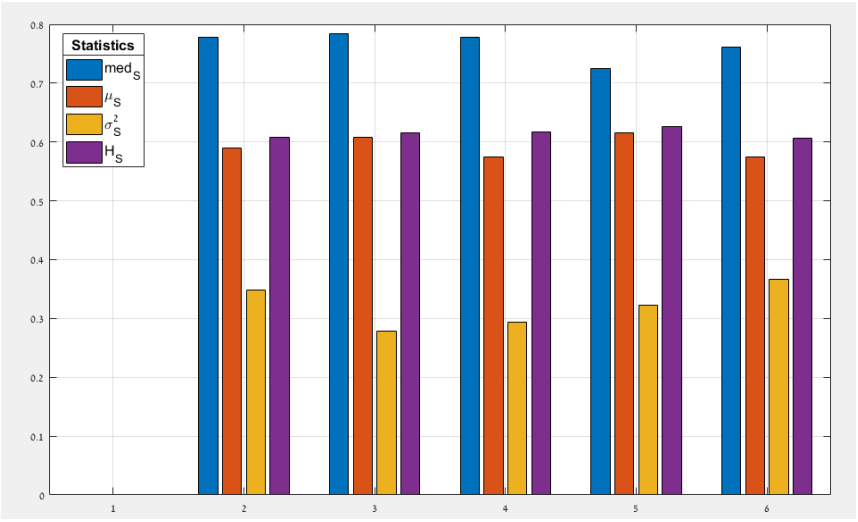



Criteria 1
RI=1
NMI=1
Criteria 2
RI=0.53061
NMI=0.051436
Criteria 3
RI=0.55113
NMI=0.17292
Criteria 4
RI=0.52506
NMI=0.084383
Criteria 5
RI=0.55503
NMI=0.045708
Criteria 6
RI=0.5247
NMI=0.016089


Single Linkage
RI=1
NMI=1
Complete Linkage
RI=0.53061
NMI=0.051436
Average Linkage
RI=0.55113
NMI=0.17292
Centroid Linkage
RI=0.52506
NMI=0.084383
Ward Linkage
RI=0.55503
NMI=0.045708
Weighted Linkage
RI=0.54863
NMI=0.085864
Heigths


Single Linkage
RI=1
NMI=1
Heigths
Complete Linkage
RI=0.53061
NMI=0.051436
Heigths
Average Linkage
RI=0.55113
NMI=0.17292
Heigths
Centroid Linkage
RI=0.52506
NMI=0.084383
Heigths
Ward Linkage
RI=0.55503
NMI=0.045708
Heigths
Weighted Linkage
RI=0.54863
NMI=0.085864
Heigths


Relative Silhouette Coefficients for criteria 1
Relative Silhouette Coefficients for criteria 2
Relative Silhouette Coefficients for criteria 3
Relative Silhouette Coefficients for criteria 4
Relative Silhouette Coefficients for criteria 5
Relative Silhouette Coefficients for criteria 6
cofficiant value


Statistics
med S
μ S
σ² S
H S

- *The simple linkage gives as the worst results.*
- *Because the centers of our dataset close each other, the simple, average and centroid linkages give as bad results.*
- *The big circle Improve the complete and ward linkage because those samples (In the big circle) prevent the merge or samples from other cluster with close centers.*

4.4 Conclusion

4.4.1. We can see that the best result in NMI&RI is for the first criteria (dmin).

4.4.2. The agglomerative process has very high computational expense.