Clustering and Unsupervised Learning
Final Course Project

Submitted by: Dor Litvak

Ben-Gurion University of The Negev

The Clustering problem

The Clustering problem is the task of grouping a set of objects in such a way that objects in the same cluster are more similar to each other than to those in other clusters.
Formally, assuming N data points $x_1, ...x_N$ generated from K Gaussian's with $\theta_1 = (\mu_1, \Sigma_1), ..., \theta_K = (\mu_K, \Sigma_K)$. The prior probability $P(w_i)$ and the classes conditional densities $P(x|w_i, \theta_i)$
The assignment is to find the correct labels $y_1, ...y_N$, such that $\forall j \in (1, ..., K)$ and $\forall i$ when $x_i \sim N(\mu_i, \Sigma_i)$ the label of x_i would be: $y_i = j$.

There are different types of clustering problems and thou different solutions. The problems are differ from one another in the given information we have: Does the number of clusters known? θ_i known? The $P(w_i), P(x|w_i, \theta_i)$ known?

As you can see in Figure 1 Flowchart, if we know K - the number of classes the The prior probability $P(w_i)$ and the classes conditional densities $P(x|w_i, \theta_i)$ we can use simple Bayes Decision Theory Classification to find $y_1, ...y_N$ and $\theta_1, ..., \theta_K$ using the following formula:
$P(x|\theta) = \sum_{j=1}^{K} P(x|w_i, \theta_i)P(w_i)$ where $\theta = \theta_1, ..., \theta_K$, $x = x_1, ...x_N$
But, if we do not have the information above, we are getting to the left side of the flowchart, and assume we have an available data set.
Now we are moving to the next question in the flowchart, and the main question in the field of the clustering problem do we have Pre-classification.

- Supervised Learning, the data set is classified. We will not discuss farther about Supervised Learning the second case is more interasting.
- Unsupervised Learning, the data set is not classified. So we don't know the labels $y_1, ...y_N$.

Unsupervised Learning the next two questions will help us to decide which Unsupervised Clustering method to use.

1. Do we know the probability density function nature, meaning we know if the data came from Gaussian / Gamma / Cauchy / Beta / Poisson / etc distribution ?
2. Do we have a partition criteria?

If the answer to 1 is yes: We can use on of the Expectation Maximization / Maximum Likelihood Estimation algorithms, we will further discuss on both.

If the answer to 1 is no and the answer to 2 is yes: We can use Hard C-means / Fuzzy C-means / Deterministic Annealing , we will further discuss on them too.
If the answer to 1 is no and the answer to 2 is no: We will use graphical clustering such Hierarchic Clustering / Minimum distance Clustering / Density Estimation / Self-Organizing Feature Maps.

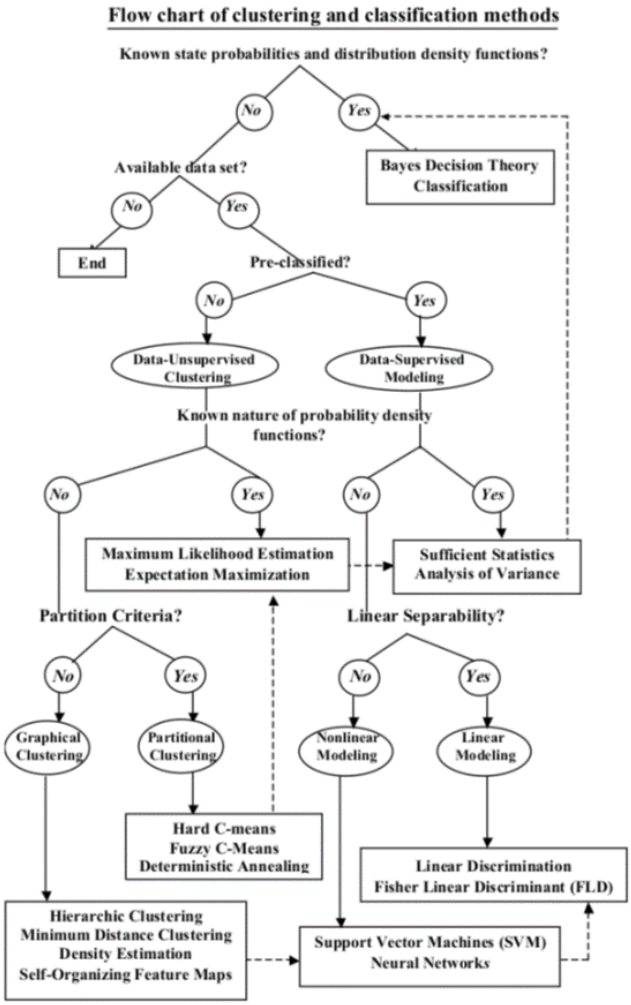

Fig. 1: Clustering Flowchart

Assignment 0 - Create the data

In the first assignment we were asked to create a diverse data to tests the clustering assignment over.

2D random data

Randomize data taken from c (constant number chosen) clusters with dots - number of data points in each cluster. the values of c, dots can easily change. The data point are draw from Gaussian distribution with random parameters mu and sigma different for each cluster.

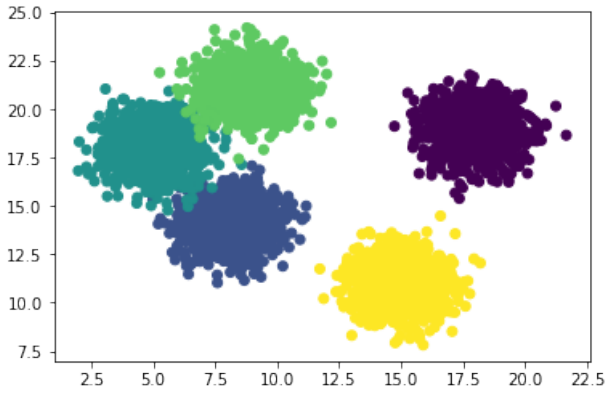

Fig. 2: 5 clusters with 1000 random data points each and axis scale 20

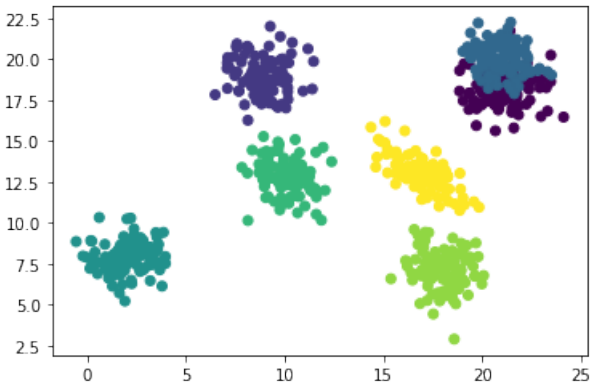

Fig. 3: 7 clusters with 100 random data points each and axis scale 20

3D random data

Randomize data taken from c (constant number chosen) clusters with dots - number of data points in each cluster. the values of c, dots can easily change. The data point are draw from Gaussian distribution with random parameters mu and sigma different for each cluster.

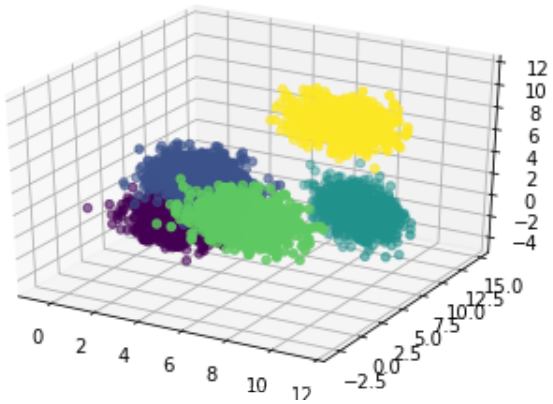

Fig. 4: 5 clusters with 1000 random data points each and axis scale 10

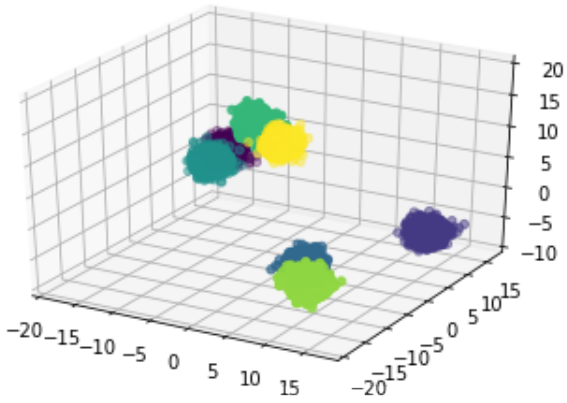

Fig. 5: 7 clusters with 1000 random data points each and axis scale 15

IRIS data set

The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

Based on the combination of these four features, we want to distinguish the species from each other.

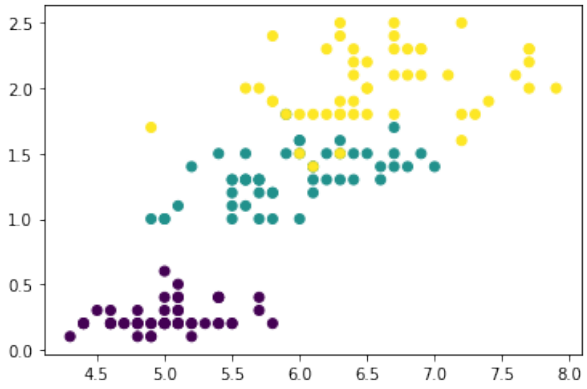

Fig. 6: Using the first and third features to distinguish between the iris sample

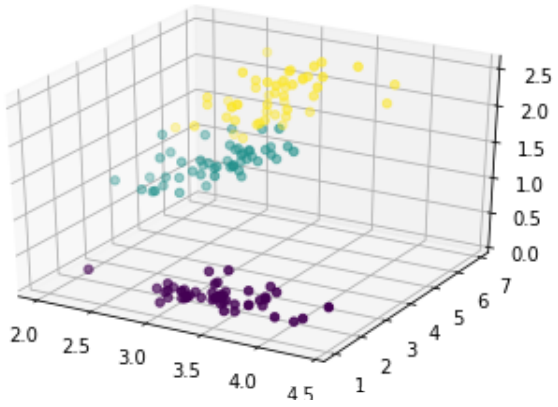

Fig. 7: Using the first, second and third features to distinguish between the iris sample

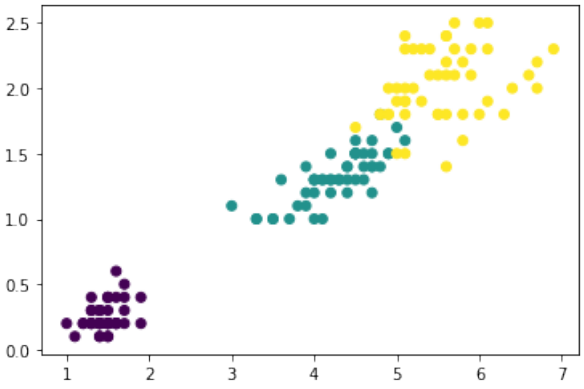

Fig. 8: Using the second and third features to distinguish between the iris sample

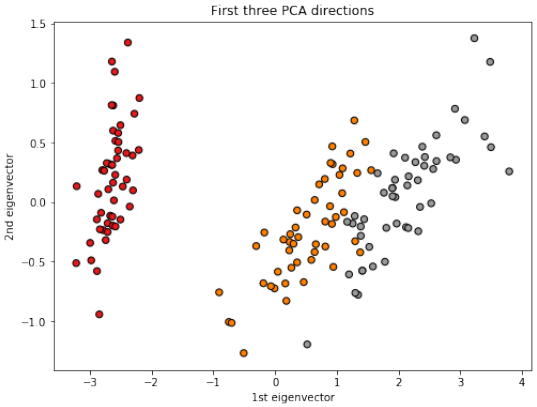

Fig. 9: Using PCA to dimensionality reduction and taking the 2 eigenvectors correspond with the 2 largest eigenvalues

Circular Data

Another interesting data is a circular data, with 2 or 3 rings (could be more, just for the example) in order to test how different methods will deal with this kind of data. In the figure circles with the same center (5,5), the first one with radius 1.5-1.7, the second with radius 1-1.2, the third with radius 0.2-0.5.

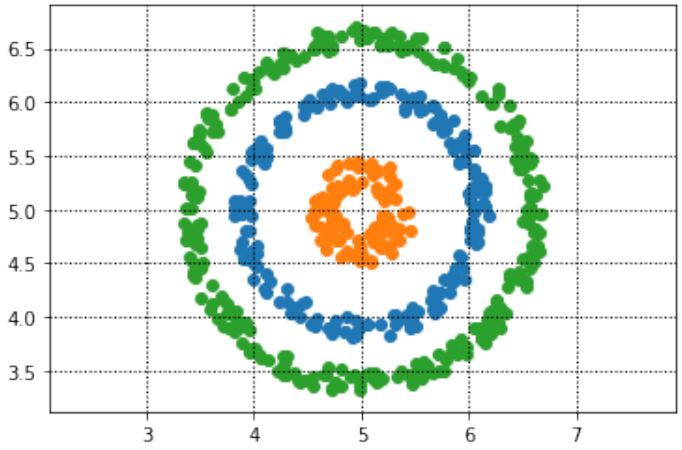

Fig. 10: Three random data taken from circles

Assignment 1 - Maximum likelihood estimation

Maximum likelihood estimation is a method of estimating the parameters of a probability distribution by maximizing the likelihood function, so that under the assumed statistical model the observed data is most probable.

Formally (we will use 2D for the explanation but it could be 3D as well), Given n data points $(x_1, y_1), ..., (x_n, y_n)$ we wish to classify those data points to c different clusters when c is fixed number as we choose.

For the local Maximum likelihood estimate $\hat{\mu}_i, \hat{\Sigma}_i, \hat{P}(w_i)$:

$$\hat{P}(w_i) = \frac{1}{n}\Sigma_{k=1}^{n}\hat{P}(w_i|x_k, \hat{\theta}) \tag{1}$$

$$\hat{\mu}_i = \frac{\Sigma_{k=1}^{n}\hat{P}(w_i|x_k, \hat{\theta})x_k}{\Sigma_{k=1}^{n}\hat{P}(w_i|x_k, \hat{\theta})} \tag{2}$$

$$\hat{\Sigma}_i = \frac{\Sigma_{k=1}^{n}\hat{P}(w_i|x_k, \hat{\theta})(x_k - \mu_i)(x_k - \mu_i)^t}{\Sigma_{k=1}^{n}\hat{P}(w_i|x_k, \hat{\theta})} \tag{3}$$

$$\hat{P}(w_i|x_k, \hat{\theta}) = \frac{p(x_k|w_i, \hat{\theta}_i)\hat{P}(w_i)}{\Sigma_{j=1}^{c}p(x_k|w_j, \hat{\theta}_j)\hat{P}(w_j)} \tag{4}$$

Algorithm 1 Maximum likelihood estimation

1: Choose centers randomly*: $\mu_i \leftarrow rand(x_i, y_i)$.
2: Initialize covariance matrix $\Sigma_i \leftarrow rand\_spd\_matrix(2, 2)$.
3: Initialize $\hat{P(w_i)} \leftarrow \frac{1}{c}$
4: **while** Run until convergence **do**
5: $\hat{P}(w_i|x_k, \hat{\theta}) \leftarrow \forall i, k$ calculate the probability data point k came from center i: Eq.4
6: Use $\hat{P}(w_i|x_k, \hat{\theta})$ to update $\hat{\mu}_i, \hat{\Sigma}_i, \hat{P}(w_i)$ using Eq.1,2,3

*choose c data points to be centers, choose random values in the data range, etc...

Results

Over 2D data points randomly drawn from 4 Gaussian's We initialize the MLE algorithm with randomly chosen from the data points centers .

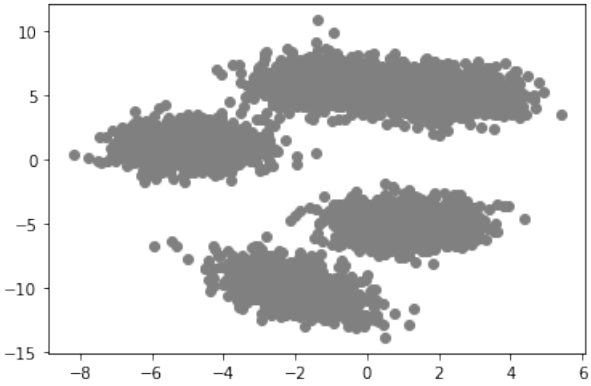

Fig. 11: Data taken from 5 different Gaussian's with 1000 data points taken from each one.

Fig. 12: The MLE algorithm after the 17 iteration.

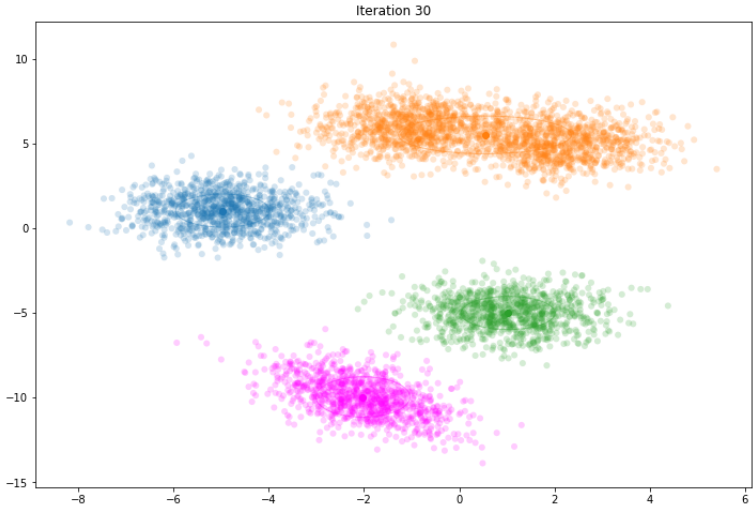

Fig. 13: The MLE algorithm in the 30 iteration, after convergence.

Over iris data set 3D projection:

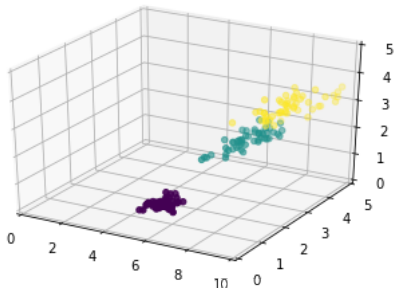

Fig. 14: The Original Iris 3D data set with first second and third features.

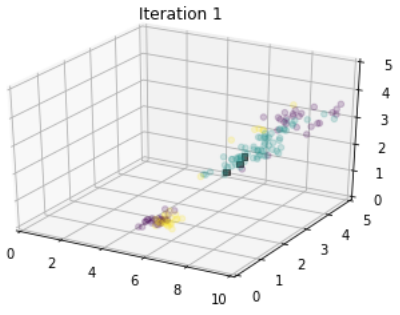

Fig. 15: The initialize centers in the Iris 3D data set for the MLE.

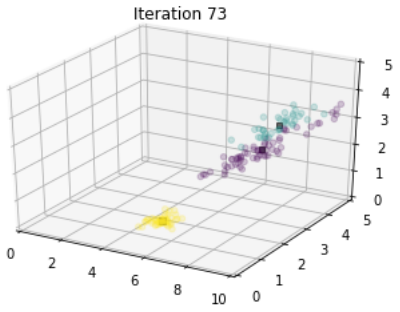

Fig. 16: The MLE after convergence at the 73 iter.

Over iris data set 2D projection using PCA:

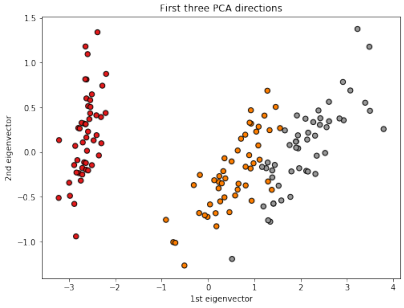

Fig. 17: The projection 2D using PCA of iris data set.

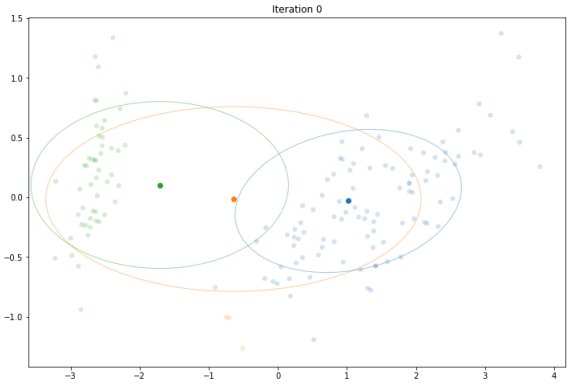

Fig. 18: The initialize centers in the Iris 2D data set for the MLE.

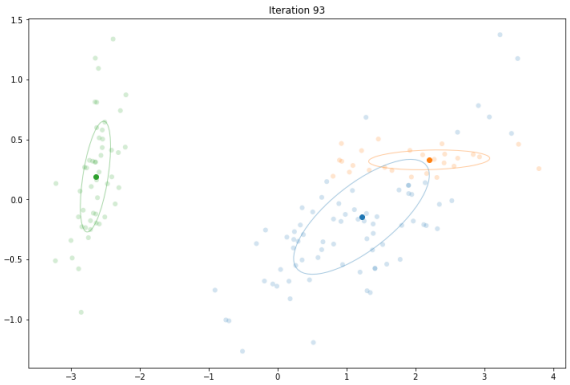

Fig. 19: The MLE after convergence at the 94 iter.

FAIL CASE - Circular data set:

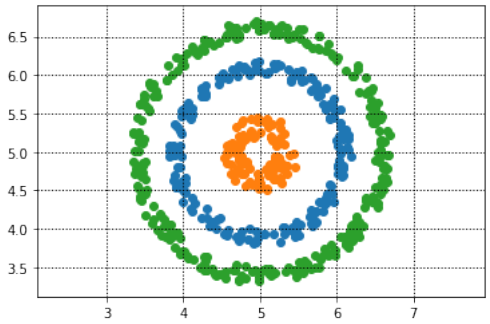

Fig. 20: Try to model 3 circles using MLE Gaussian model.

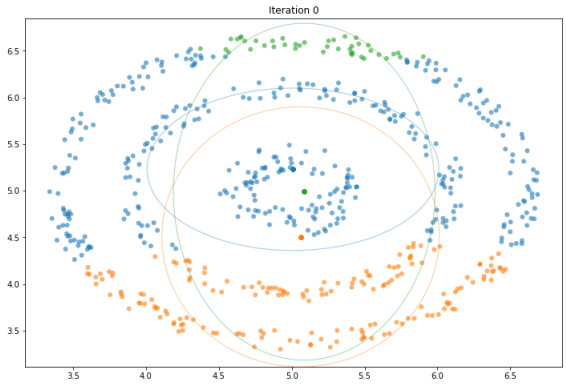

Fig. 21: The initialize centers in the circular data set for the MLE.

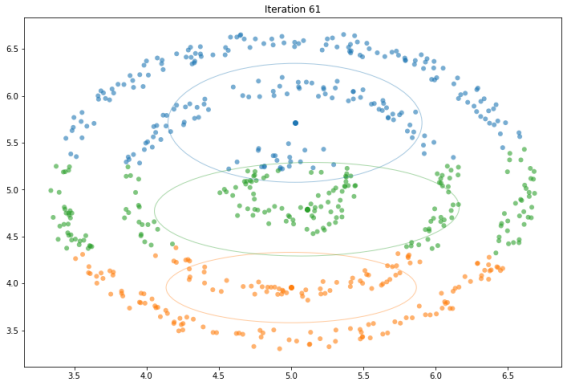

Fig. 22: The MLE after convergence at the 61 iter.

MLE Pros and Cons:

Pros:

- Very useful on large data sets.
- Good performances
- The best for known Gaussian distributed data.

Drowbacks:

- Limited to only gaussians, so cannot use for flexible data.
- Need to know ahead who is k.
- The statistical inference of the maximun likelihood needed large amount of data so the postirior distributions would be clouse to the data.
- Maximum likelihood estimates can be heavily biased for small samples. The optimality properties may not apply for small samples.
- Maximum likelihood can be sensitive to the choice of starting values.

Assignment 2 - Unsupervised Optimal Fuzzy Clustering

The UOFC [3] estimates both number of clusters and the fuzzy classification of data points to the clusters. In the classical set theory (aka: the MLE we saw before) an element either belongs or does not belong to the cluster.
Fuzzy set theory permits the gradual assessment of the membership of elements in a set/cluster. This is described with the aid of a membership function valued in the real unit interval $[0, 1]$.
In UOFC uses incremental approach in order to estimate number of clusters. In each step starting from k = 1 we evaluate the best assignment of the data to the k clusters. To measure how good the k clusters explain the data, We use 6 criterion measurements, the k with the best score is considered to be right k.
The Clustering Validation Criterion:

- HPV - Hyper Planar Volume:
  The sum of hyper volumes of each cluster in the data set, we wish to minimize it.
- PDC - Partition Density Central:
  The density of each cluster determined by the clusters centers, we wish to maximize it.

- PDM - Partition Density Maximal:
  The density of each cluster determined by the maximal membership values in each cluster, we wish to maximize it.

- APDC - Average Partition Density Central members:
  The Average PDC depend on number of clusters, we wish to maximize it just like PDC.

- APDM - Average Partition Density Maximal members:
  The Average PDM depend on number of clusters, we wish to maximize it just like PDM.

- NPIC - Normalized partition index criteria

- INV - Invariant Criteria:
  The sum of the hyper volumes normalized by the hyper volume of the data set. we would like to minimized it.

- J Criteria:
  The cost function of the fuzzy C-mean, Trade-off between sparse clusters Concentration and high density clusters, we want to minimized it.

Algorithm 2 Unsupervised Optimal Fuzzy Clustering

1: $k = 1$
2: q = 3 controls the fuzziness
3: Initialize the first center c_1 to be the mean of the data.
4: **while** Run until convergence **do**
5: **If** we are in the first iteration **do**
Initialize U_k the membership matrix given the canter c_1.
6: **Else** Add another center to the membership matrix U_k and
initialize it with equal distance from all the data points.
k = k + 1
7: centers, U_k = **calculate fuzzy K mean**(k, data, centers, q, U_k)
8: centers, U_k = **calculate fuzzy MLE**(k, data, centers, q, U_k)
9: Calculate fuzzy co-variance matrix F_k
10: Compute HPV, PDC, PDM, APDC, APDM, NPIC, INV, J_D using F_k.

Results

Over 2D data points randomly drawn from 5 Gaussian's:

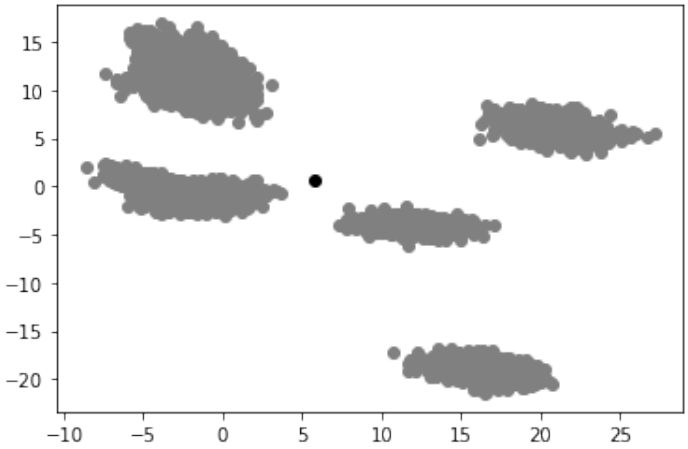

Fig. 23: The UOFC initialization.

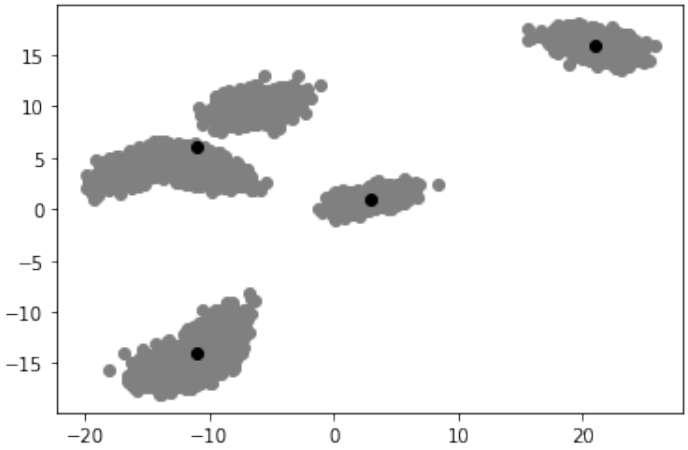

Fig. 24: The UOFC centers with k = 5 the best k in sense of the criteria.

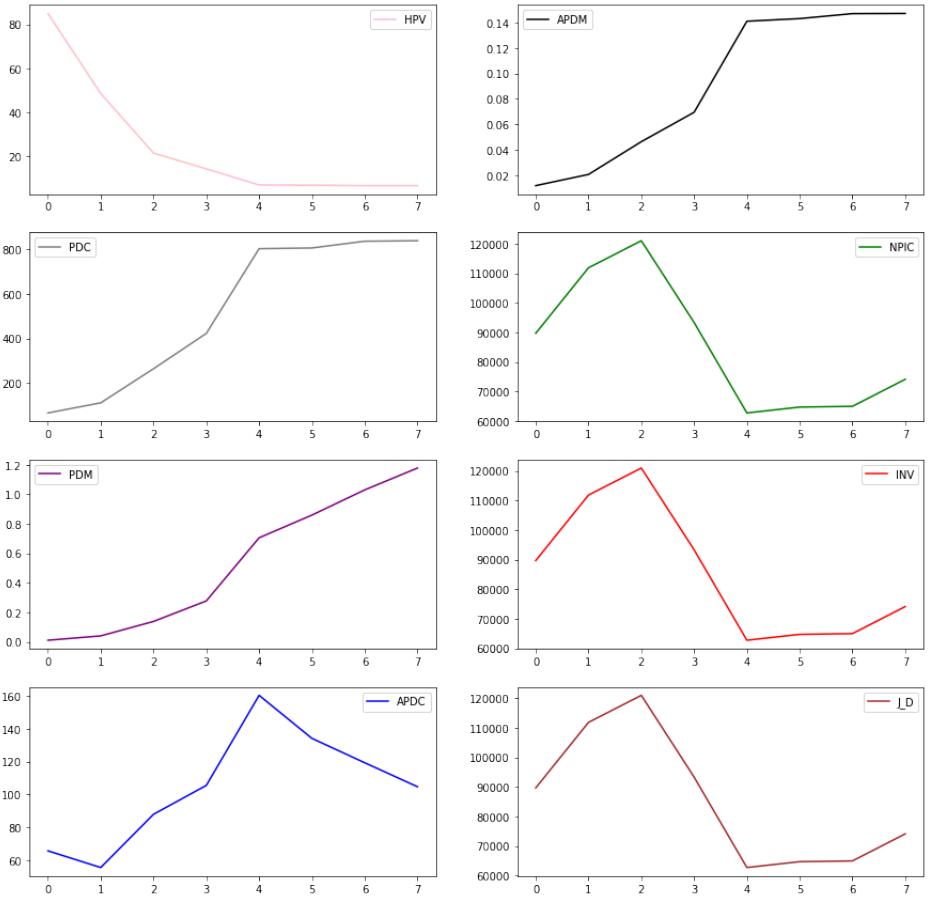

Fig. 25: All the criteria depending on k.

Over iris data set 2D projection using PCA:

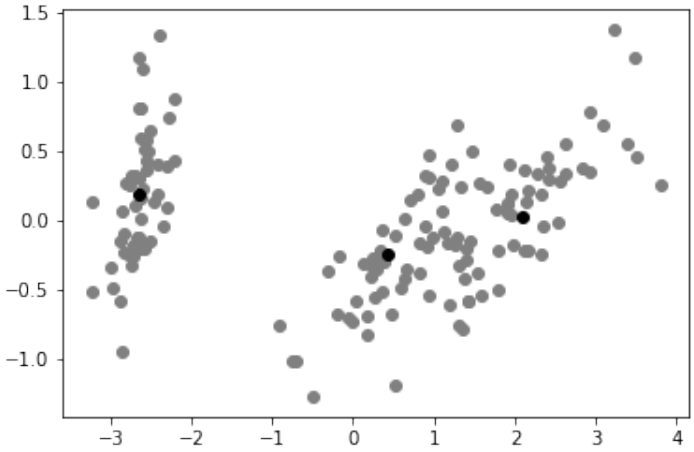

Fig. 26: The UOFC results using 2 centers, the best in sense of criteria.

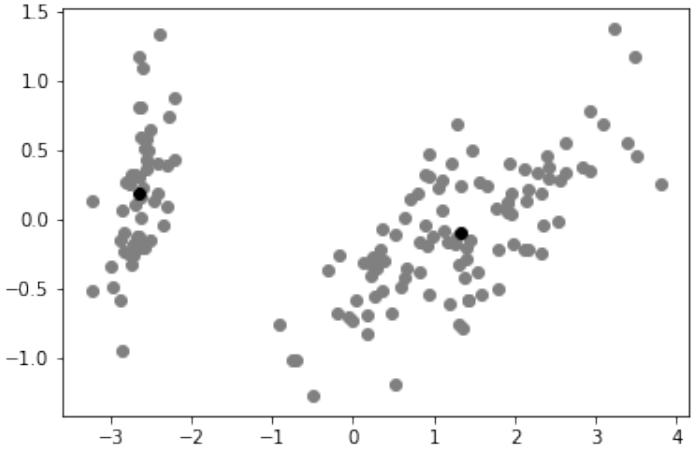

Fig. 27: The UOFC results using 3 centers.

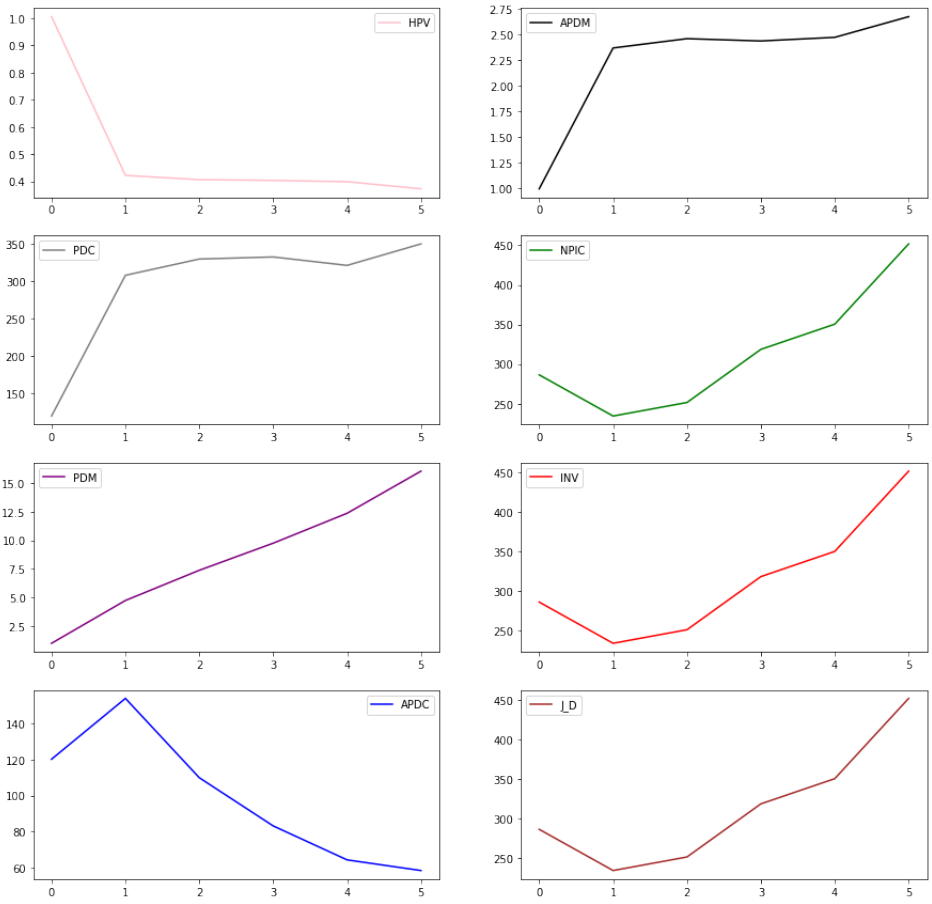

Fig. 28: All the criteria depending on k.

Iris projection over the first 2 features:

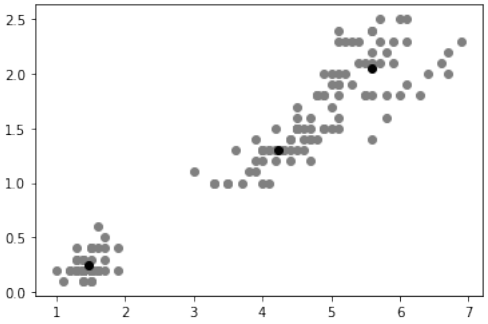

Fig. 29: The best results given the criteria is k = 3

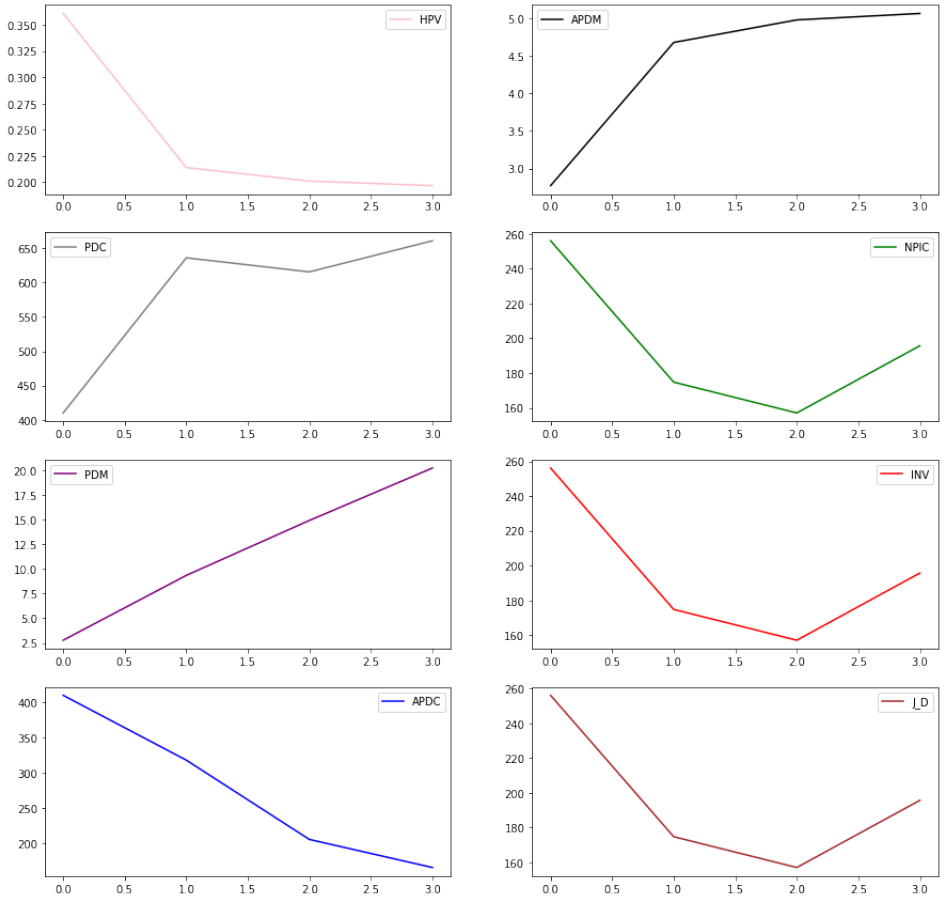

Fig. 30: All the criteria depending on k.

FAIL CASE- Circular data set:

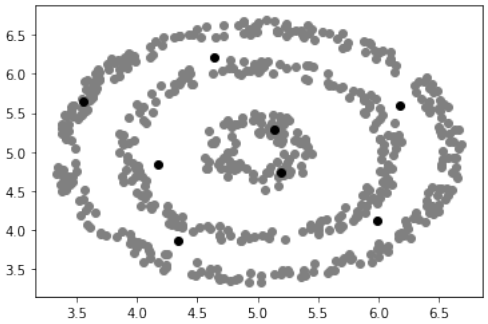

Fig. 31: The results given k = 8

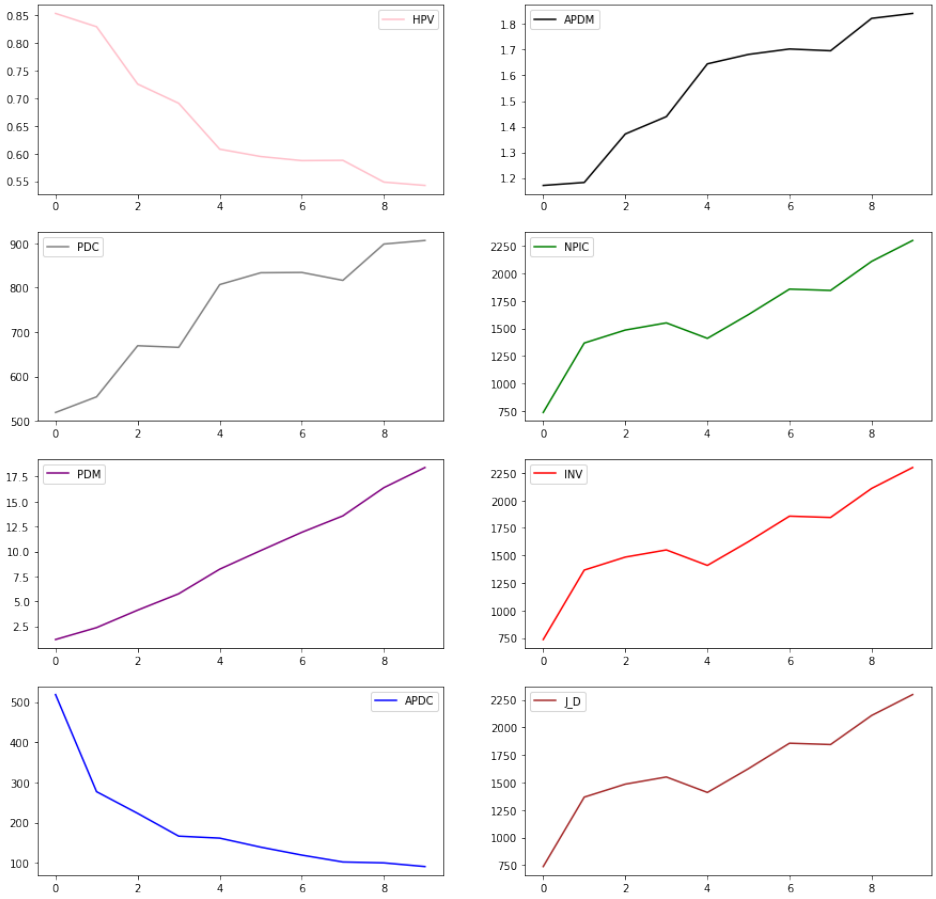

Fig. 32: All the criteria depending on k.

UOFC Pros and Cons:

Pros:

- Fuzzy clustering very sensitive to initialization UOFC not that much.
- Don't need to know k ahead.

Drawbacks:

- Needed to scale q - fuzziness parameters.
- Out layers data points also part of the clusters calculation.
- not good for not circular data.

Assignment 3 - Agglomerative Hierarchical Clustering

The agglomerative clustering also known as Agglomerative Nesting is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. Agglomerative clustering works in a "bottom-up" manner. That is, each object is initially considered as a single-element cluster (leaf). At each step of the algorithm, the two clusters that are the most similar are combined into a new bigger cluster (nodes). This procedure is iterated until all points are member of just one single big cluster (root).

Algorithm 3 Agglomerative Hierarchical Clustering

1: Initialize k number of clusters = n number of data points.
2: Initialize $\forall i = \{1, ..., n\} : D_i = \{X_i\}$ each cluster D contains only one data point.
3: **while** $c > 1$ **do**
4: Find the nearest clusters for example: D_i, D_j
5: Merge D_i, D_j to be one cluster.
6: c = c - 1

We tested 4 different types of distance measures for the nearest clusters. For all $x_i \in D_i, x_j \in D_j$ we calculate the distance between D_i, D_j as follow:

- $d_{min}(D_i, D_j) = min_{x \in D_i, x' \in D_j} ||x - x'||$
- $d_{max}(D_i, D_j) = max_{x \in D_i, x' \in D_j} ||x - x'||$
- $d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \Sigma_{x \in D_i} \Sigma_{x' \in D_j} ||x - x'||$
- $d_{mean}(D_i, D_j) = ||m_i - m_j||$

Results

Data points Drown from 6 Gaussian's:

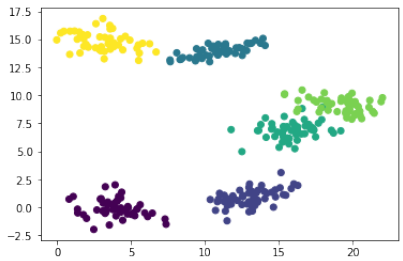

Fig. 33: The data points original labels.

Fig. 34: The results of all 4 methods when c = 5, there are 5 clusters.

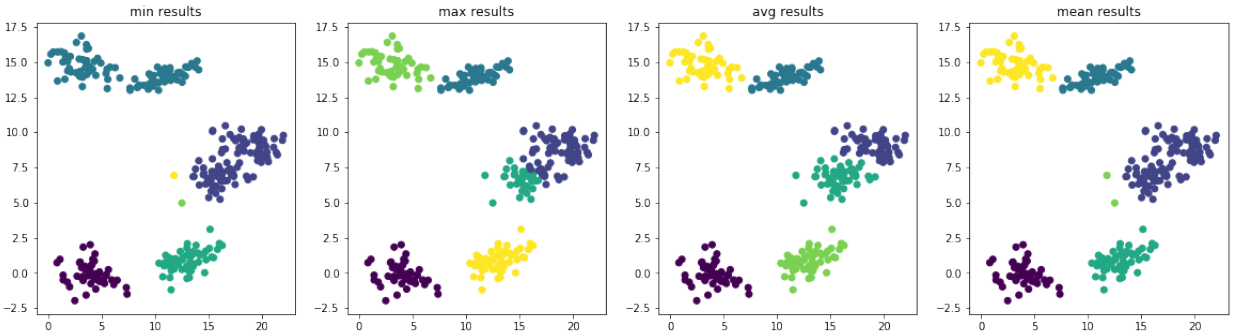

Fig. 35: The results of all 4 methods when c = 6, there are 6 clusters. As you can see the best results is given by the avg distance.

Iris data set:

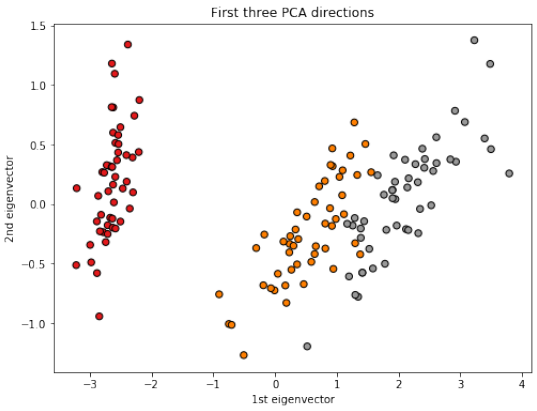

Fig. 36: The iris data set in 2D using PCA and taking the two biggest eigenvalues.

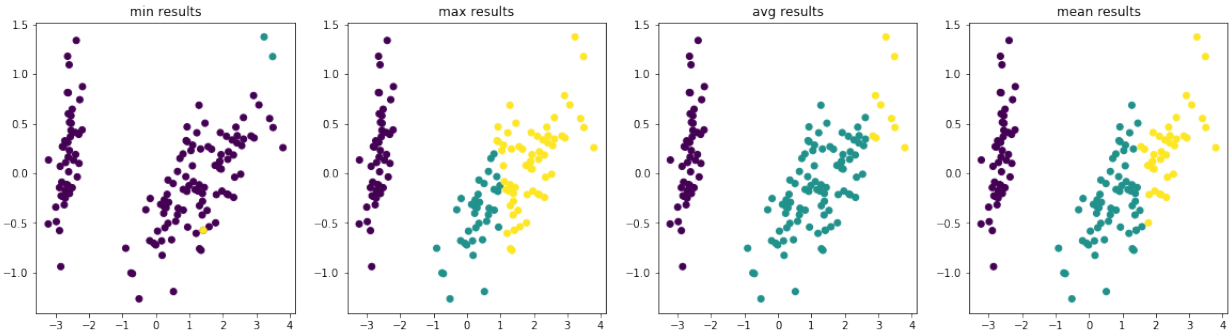

Fig. 37: The results of all the methods, the best results is between max and mean. The worst result is min criteria.

Circular data set:

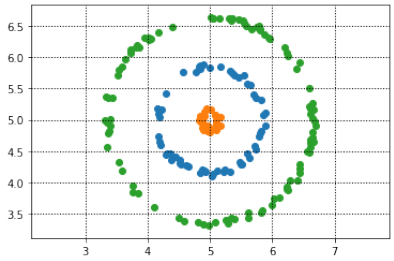

Fig. 38: The circular data set.

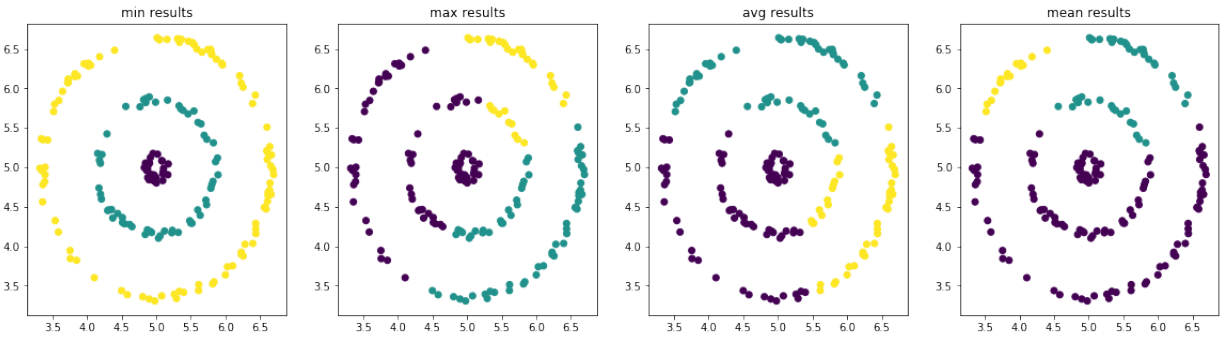

Fig. 39: The only distance criteria that manage to find the circles is the minimum criteria, and it only work if the distance between the circles is big enough.

AHC Pros and Cons:

Pros:

- Best for capturing clusters of different sizes and shapes
- Computationally possible for large datasets

Drowbacks:

- Sensitive to the method selection.
- sensitive to noise.
- complete link and group average are not affected by noise, but have a bias towards finding global patterns.
- Doesn't give good results for large data sets because it uses too little information.

Assignment 4 - "The Chinese restaurant process" Dirichlet Process Gaussian Mixture Model:

Bayesian nonparametric (BNP) models approach is to fit a single model that can adapt its complexity to the data and allow the complexity to grow as more data are observed.
The BNP approach finesses the problem of choosing the number of clusters by assuming that it is infinite, while specifying the prior over infinite groupings $P(c)$ in such a way that it favors assigning data to a small number of groups.
The prior over groupings is called the Chinese Restaurant Process (CRP).
Imagine a restaurant with an infinite number of tables, and imagine a sequence of customers entering the restaurant and sitting down.
The first customer enters and sits at the first table.
The second customer enters and sits at the first table with probability $\frac{1}{1+\alpha}$, and the second table with probability $\frac{\alpha}{1+\alpha}$, where α is a positive real and called the concentration parameter.
More formally, let c_n be the table assignment of the nth customer. The probability he/she will sit in the kth table is given by the following probability:

$$P(c_n = k|c_{1:n-1}) = \begin{cases} \textbf{if } k \leq K_+ : \ \frac{m_k}{n-1+\alpha} \\ \textbf{(i.e k is a previously occupied table)} \\ \textbf{otherwise: } \frac{\alpha}{n-1+\alpha} \\ \textbf{(i.e k is the next unoccupied table)} \end{cases}$$

Where m_k is the number of customers sitting at table k, and K_+ is the number of tables for which $m_k > 0$. Intuitively, a larger value of α will produce more occupied tables (and fewer customers per table).

Dirichlet Process Gaussian Mixture Model:
The Dirichlet process (DP) is a distribution over distributions. It is parameterized by a concentration parameter $\alpha > 0$ and a base distribution G_0, which is a distribution over a space Θ. A random distribution G drawn from a DP is denoted by $G \sim DP(\alpha, G_0)$. One property of the Dirichlet process [2] is that random distributions drawn from the Dirichlet process are discrete and called "atoms".

$G = \Sigma_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}$

Where π_k is the probability assigned to the kth atom and $\delta_{\theta_k^*}$ is the location or value of that atom, and these atoms are drawn independently from the base distribution G_0.

The second property connects the Dirichlet process to the Chinese restaurant process. Consider a random distribution drawn from a DP followed by repeated draws from that random distribution,

$G \sim DP(\alpha, G_0)$
$\theta_i \sim G \quad i \in \{1, ..., n\}$

The joint distribution of $\theta_{1:n}$, which is obtained by marginalizing out the random distribution G is ,

$p(\theta_1, ..., \theta_n | \alpha, G_0) = \int (\Pi_{i=1}^{n} p(\theta_i | G))\, dP(G | a, G_0)$

Ferguson (1973) [2] showed that, under this joint distribution, the θ_i will exhibit a clustering property — they will share repeated values with positive probability. (Note that, for example, repeated draws from a Gaussian do not exhibit this property.) The structure of shared values defines a partition of the integers from 1 to n, and the distribution of this partition is a Chinese restaurant process with parameter α. Finally, he showed that the unique values of θ_i shared among the variables are independent draws from G_0.
A DP mixture adds a third step to the model above [1]:

$G \sim DP(\alpha, G_0)$
$\theta_i \sim G \quad i \in \{1, ..., n\}$
$x_i \sim p(\dot{|}\theta_i)$

Marginalizing out G reveals that the DP mixture is equivalent to a CRP mixture. Good Gibbs sampling algorithms for DP mixtures are based on this representation.

The stick-breaking construction of DP [5]:
Consider a stick with unit length. We divide the stick into an infinite number of segments π_k by the following process. First, choose a beta random variable $\beta_1 \sim Beta(1, \alpha)$ and break of β_1 of the stick. For each remaining segment, choose another beta distributed random variable, and break off that proportion of the remainder of the stick. This gives us an infinite collection of weights π_k,

$\beta_k \sim Beta(1, \alpha)$
$\pi_k = \beta_k \Pi_{j=1}^{k-1}(1 - \beta_j) \quad k = 1, 2, 3, ...$

Finally, we construct a random distribution using

$G = \Sigma_{k=1}^{\infty} \pi_k \delta_{\theta_k^\star}$,

where we take an infinite number of draws from a base distribution G_0 and draw the weights as in the second equation. Sethuraman [5] showed that the distribution of this random distribution is a $DP(\alpha, G_0)$. This representation of the Dirichlet process, and its corresponding use in a Dirichlet process mixture, allows us to compute a variety of functions of posterior DPs [4] and is the basis for the variational approach to approximate inference.

DPGMM Pros and Cons:

Pros:

- Very stable to changes of the parameters, leading to more stability and less tuning.
- We do not need to know the number of clusters.

Drowbacks:

- the extra parametrization can and will make inference slower, although not by much.
- There are many implicit biases in the Dirichlet process and the inference algorithms, and whenever there is a mismatch between these biases and the data it might be possible to fit better models using a finite mixture.

Results

Iris 2D data set:

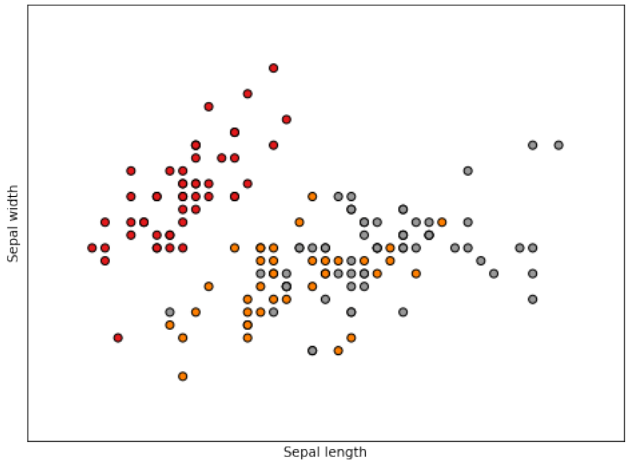

Fig. 40: The iris 2D data set.

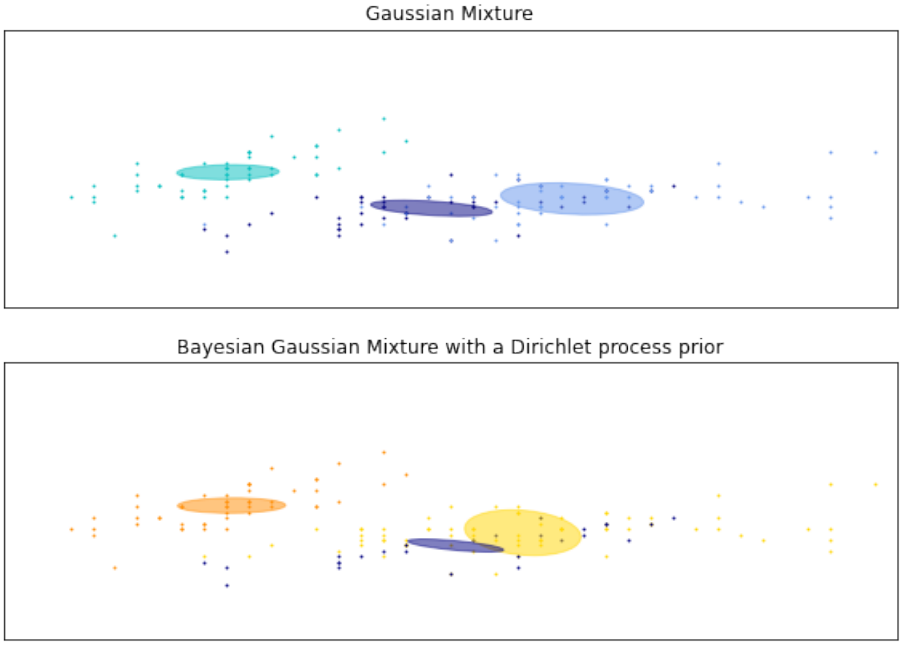

Fig. 41: The results of the DPGMM over iris data set vs of gaussian mixture model.

Iris 3D data set:

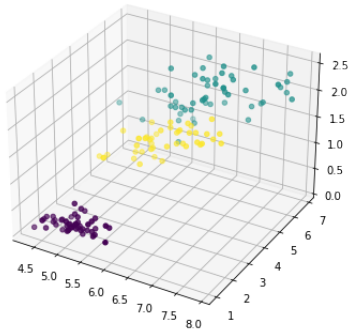

Fig. 42: The iris data set with the the first third and forth values.

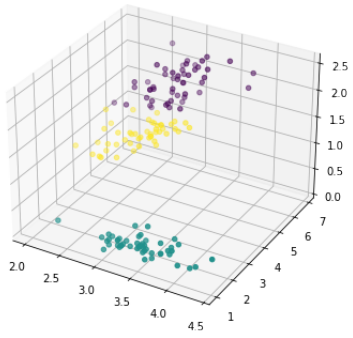

Fig. 43: The results of the **DPGMM** over iris data set, almost prefect!.

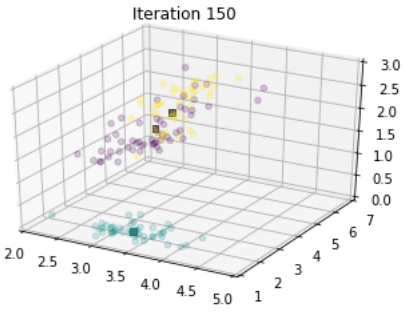

Fig. 44: The results of the GMM over iris data set. For comparison.

Circular data set:

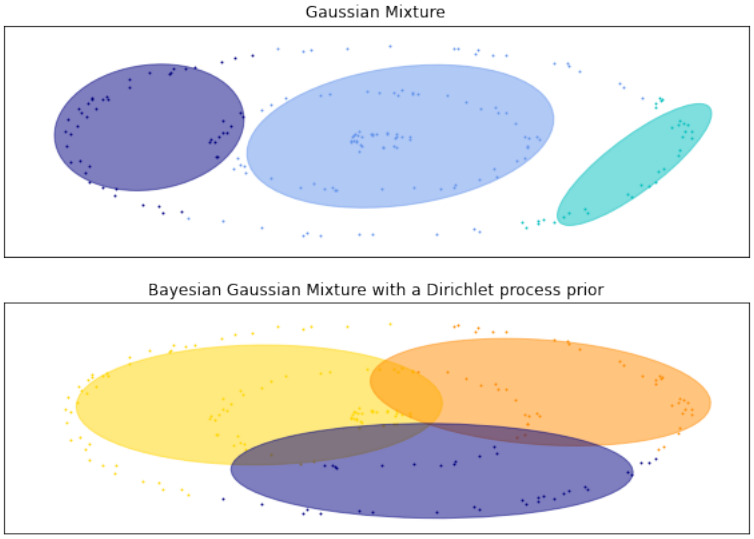

Fig. 45: The results the circular data set with both DPGMM and GMM.

Random Gaussian data set:

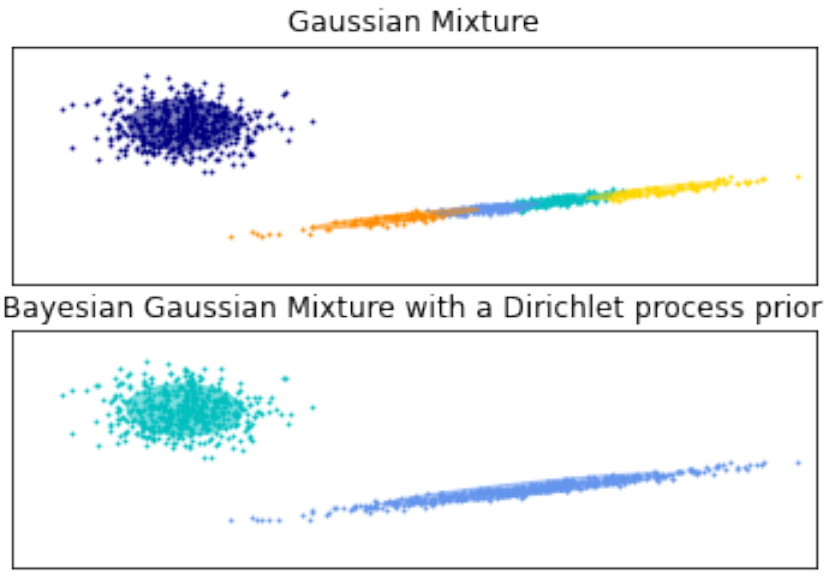

Fig. 46: The results of DPGMM and GMM over random Gaussian data set.

References

[1] C. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1974.

[2] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1(2):209–230, 03 1973.

[3] I. Gath and A. B. Geva. *Unsupervised optimal fuzzy clustering*, volume 11. 1989.

[4] Alan E Gelfand and Athanasios Kottas. A computational approach for full nonparametric bayesian inference under dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 11(2):289–305, 2002.

[5] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.