Assignment 4

Computational Intelligence, SS2017

Team Members		
Last name	First name	Matriculation Number
Reeh	Lucas	00630128

Contents

List of Figures

1 Expectation Maximization Algorithm

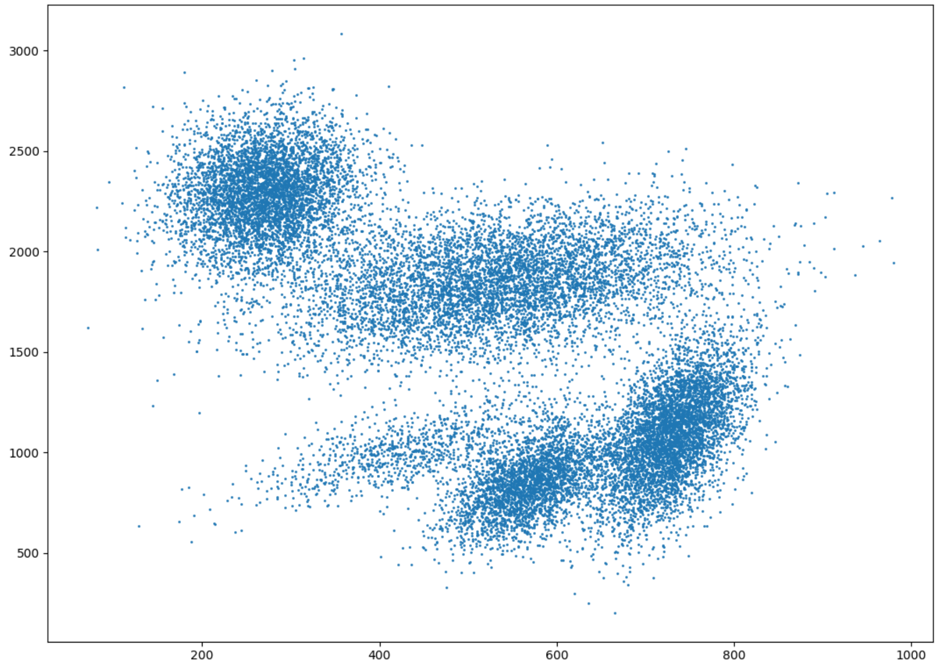

Figure 1: Training Data

2. Correct number of components $M = 5$ (fixed random seed)

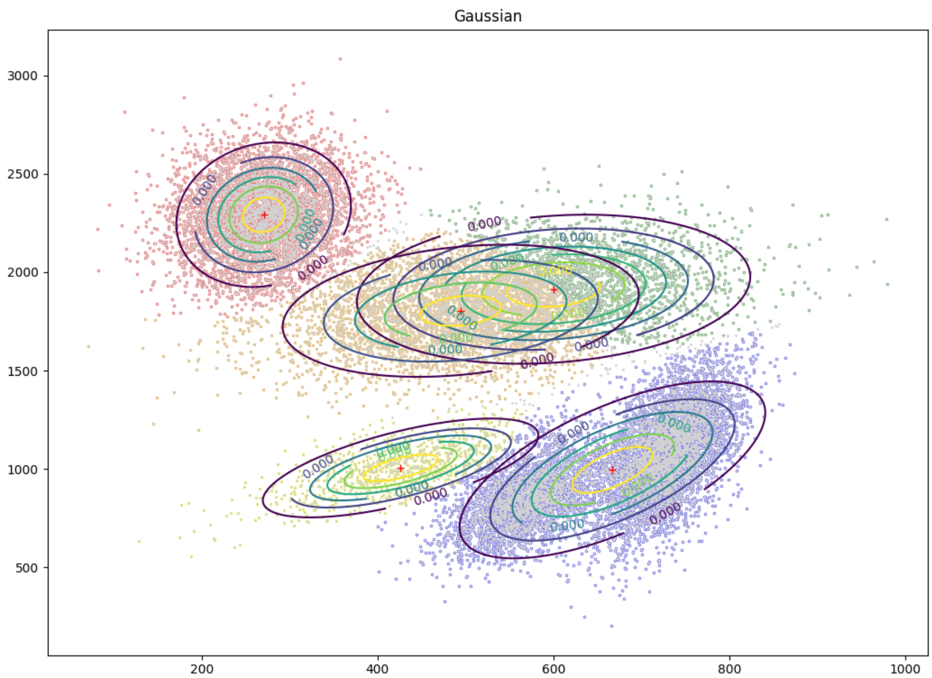

Figure 2: EM using $M = 5$

Figure 3 shows test-data overlayed by gaussian countours from EM-algorithm to compare to actual values.

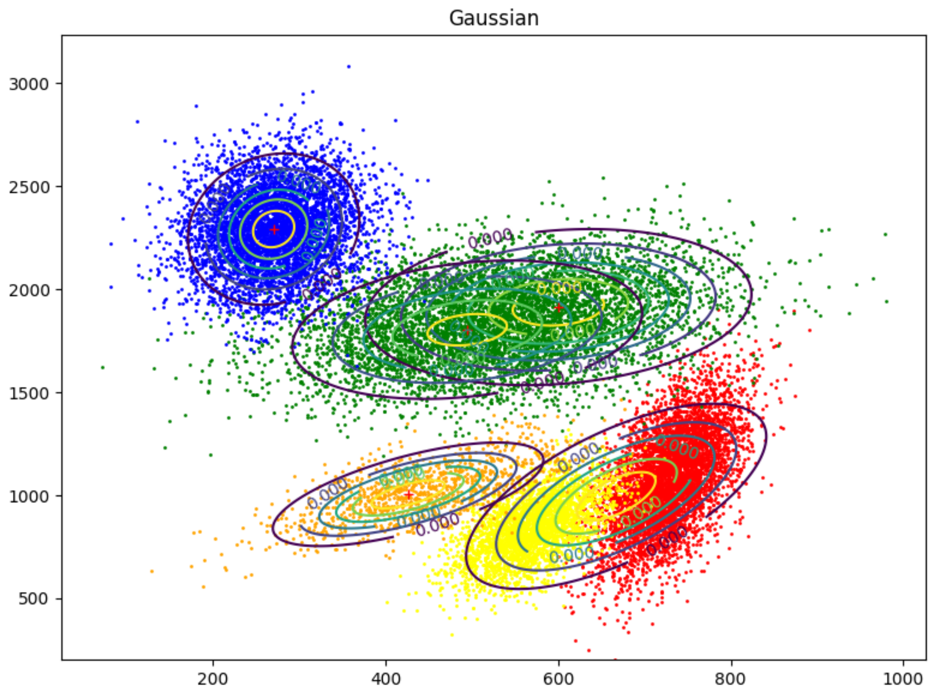

Figure 3: EM compared to test data

3. Initialization was choosen like mentioned in lecture noted[1]. Wrong number of components can lead to unexpected results ranging from complete jibberish to good matching distributions.

   Other findings with different Θ values:

   - Random seed can have great effect on the result, see Hint below.
   - divergence when choosing wrong parameters (eg. wrong Σ_0)
   - zero division when choosing $\alpha_0 = 0$.
   - weird linalg errors (`numpy.linalg.linalg.LinAlgError`) when using identity matrix for Σ_0.

4. Log-Likelihood function is converging very fast to the second iteration but slows down from there see Figure 4. Even logarithmic scaled y-axis on the plot had no improvement on the visualization.

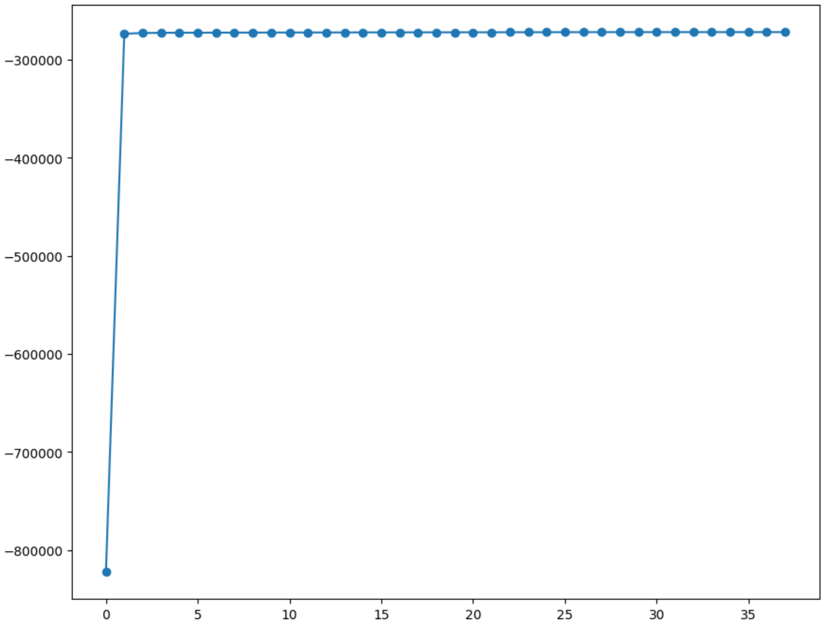

Figure 4: EM Log-Likelood progress over iterations

As you can see in Figure 5 Skipping data from the first iteration gives a much clearer picture. Now convergence (slightly logarithmic) is recognizable.

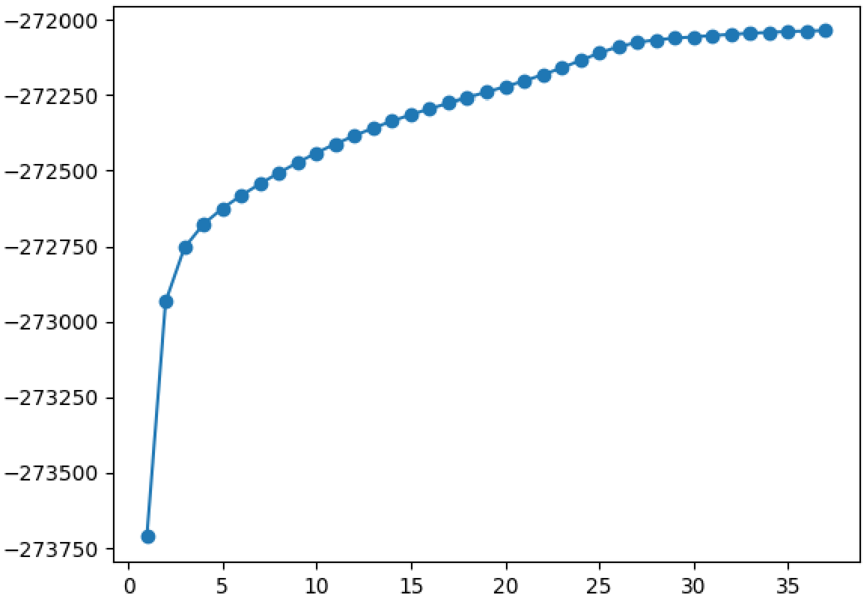

Figure 5: EM Log-Likelood progress over iterations (without the first)

5. If Σ is a diagonal matrix the distributions are not tilted and only expand along the axises (x, y). This sometimes leads to faster convergence but not so accurate matching.

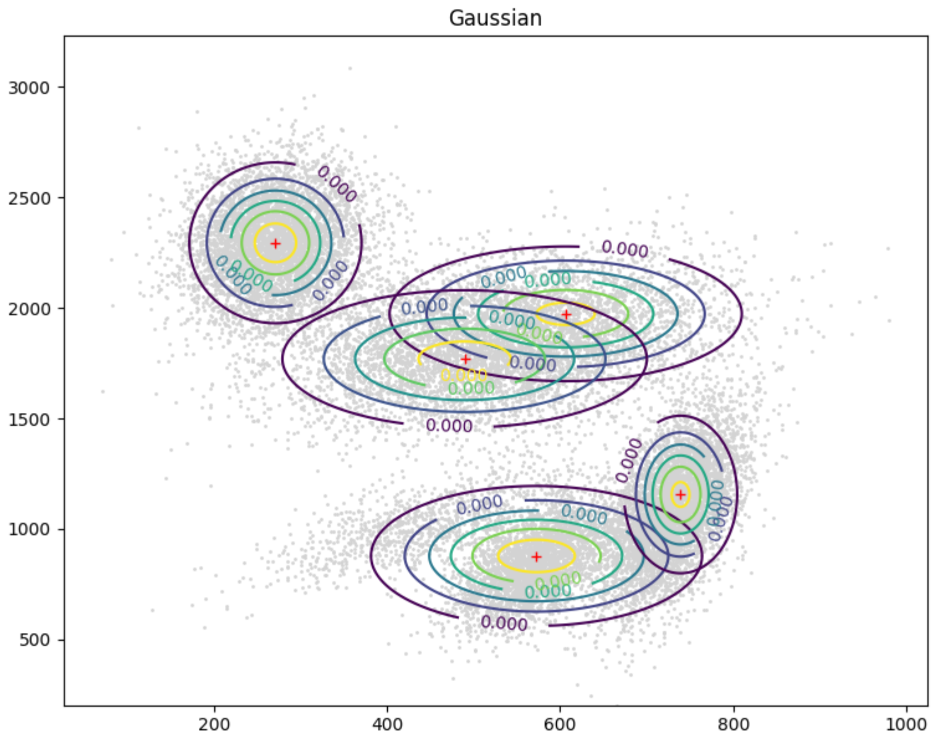

Figure 6: Using diagonal matrix for Σ

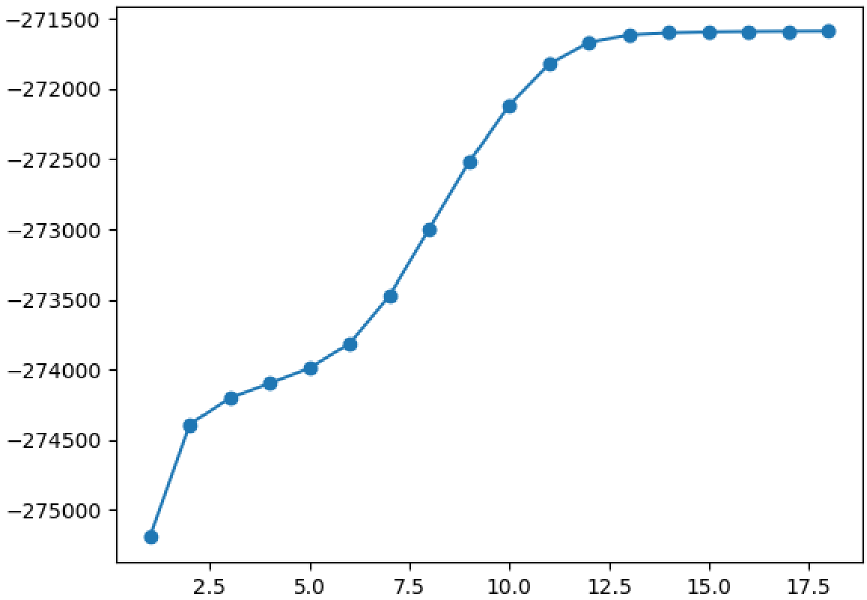

Figure 7: EM Log-Likelood using diagonal matrix for Σ

From time to time matrix computation errors occur within the given likelihood framework function: `ValueError: zero-size array to reduction operation minimum which has no identity` with no diagonal matrix, i guess rounding errors and zero values in the matrix.

6. Following Figures show snapshots of the soft-classification progress over iterations of the EM-Algorithm.

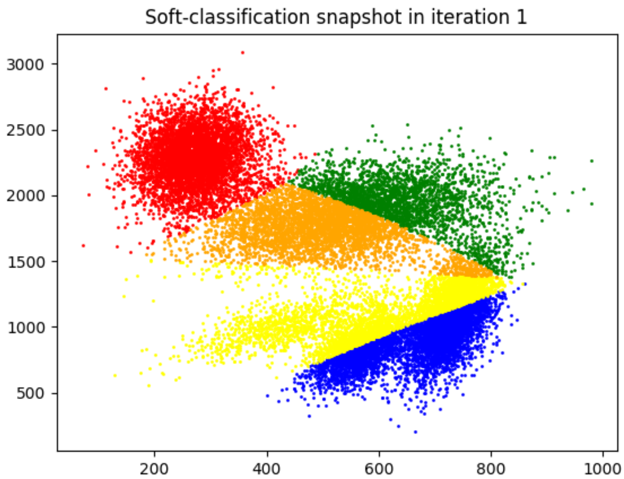

Figure 8: Soft-classification snapshot (iteration 1)

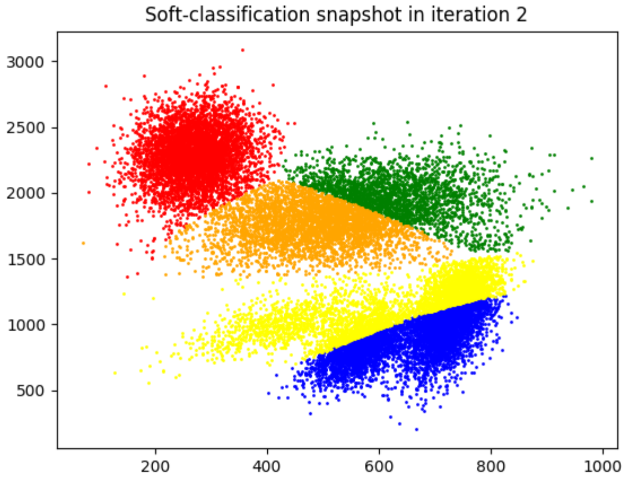

Figure 9: Soft-classification snapshot (iteration 2)

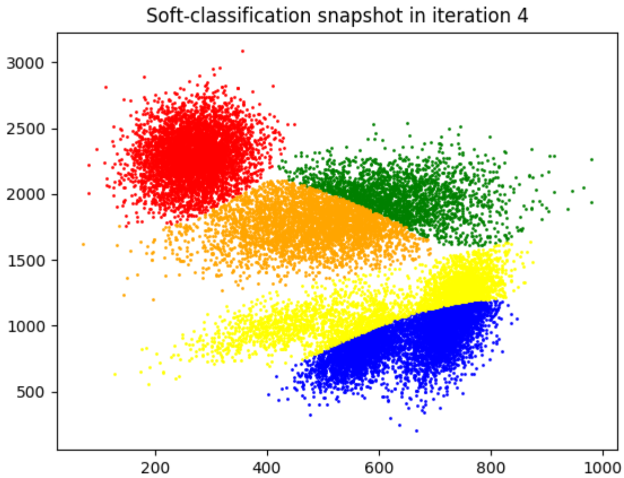

Figure 10: Soft-classification snapshot (iteration 4)

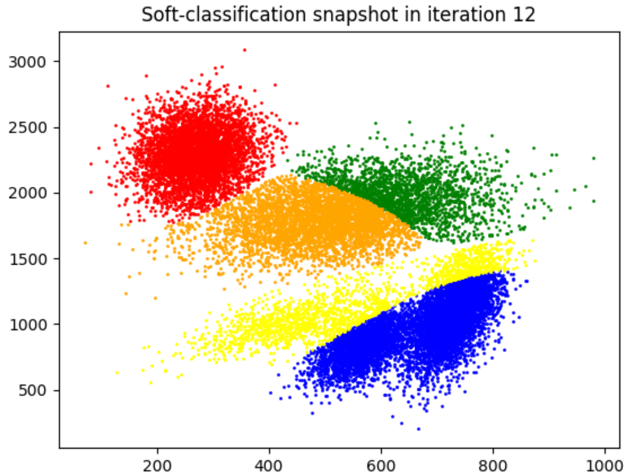

Figure 11: Soft-classification snapshot (iteration 12)

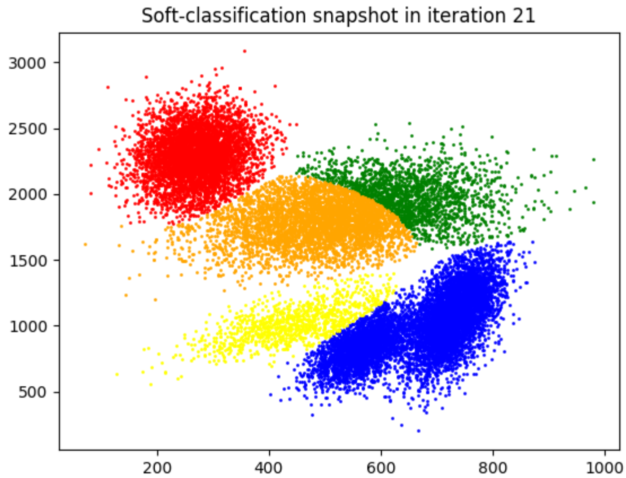

Figure 12: Soft-classification snapshot (iteration 21)

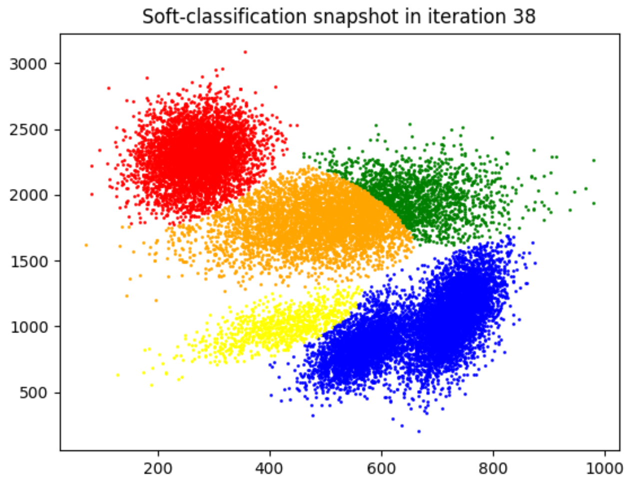

Figure 13: Soft-classification snapshot (iteration 38, final)

Hint

Changed random seed gives better classification (more like the one shown in practical classes). These distributions intiutively match the data better.

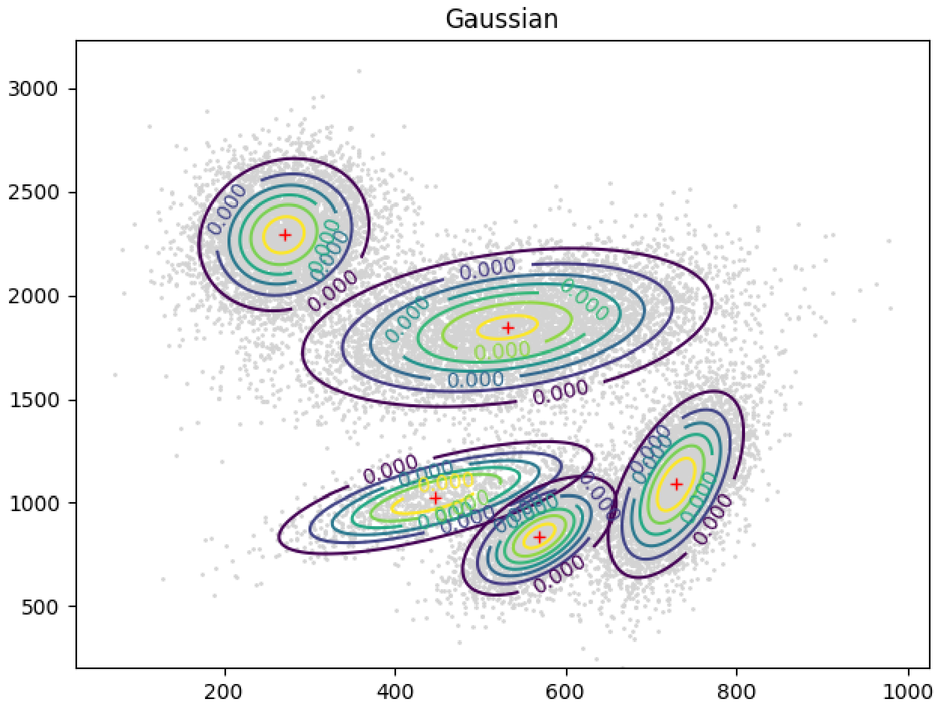

Figure 14: Better classification with no random seed set

Findings

Most important in doing EM would be choosing different starting values for μ_0 since it cleary effects classification tremendously. Even normalized values for the data can not prevent this problem. Compared to the test data fixed random seed from assignement template did not perform well. Different random starting points for μ gives high accuracy in some tests.

2 K-means algorithm

2. (a) Cumulative distance progress J with $\epsilon = 1$

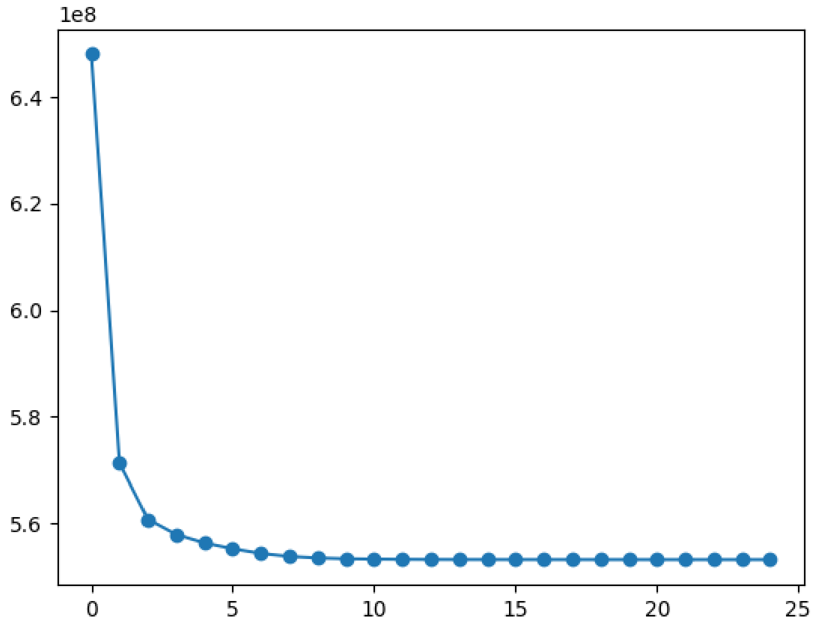

Figure 15: Distance (cumulative) progress over iterations

(b) Clustering progress

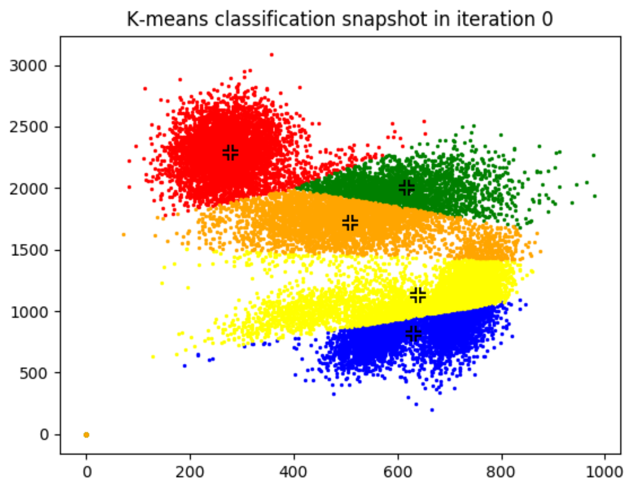

Figure 16: K-means classification snapshot in iteration 0 (inital)

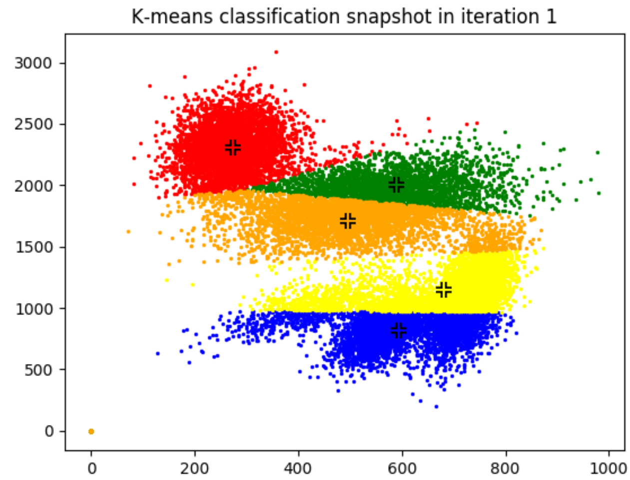

Figure 17: K-means classification snapshot in iteration 1

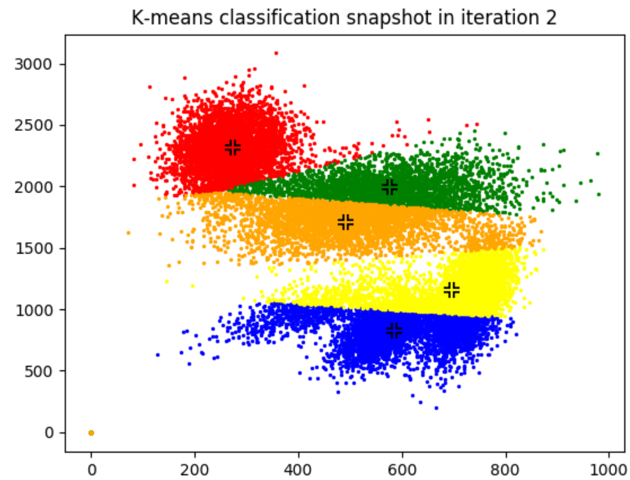

Figure 18: K-means classification snapshot in iteration 2

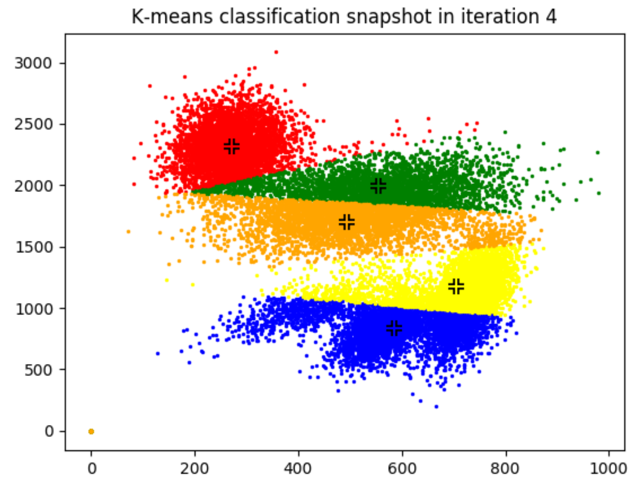

Figure 19: K-means classification snapshot in iteration 4

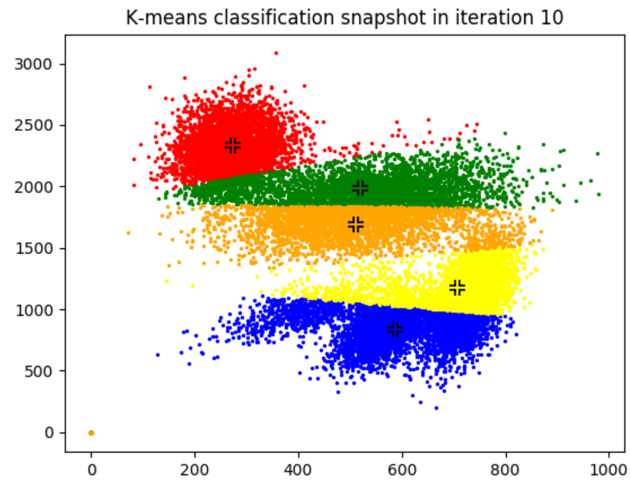

Figure 20: K-means classification snapshot in iteration 1

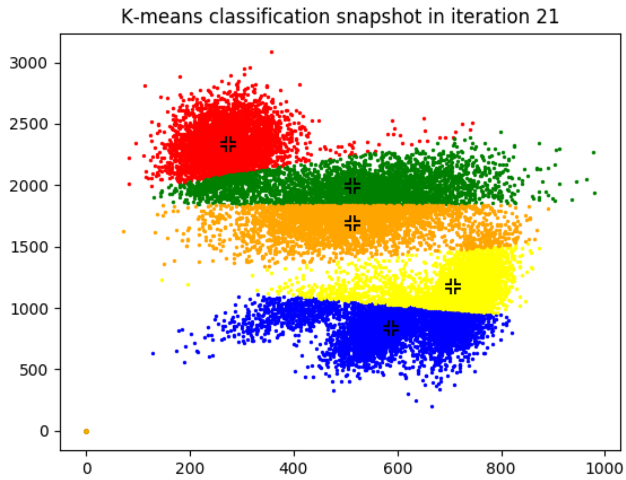

Figure 21: K-means classification snapshot in iteration 21

(c) Result

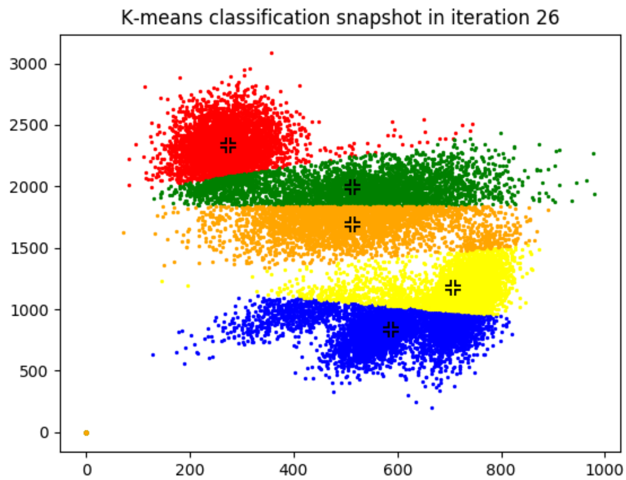

Figure 22: K-means classification snapshot in iteration 26 (final)

3. Wrong number of compenents

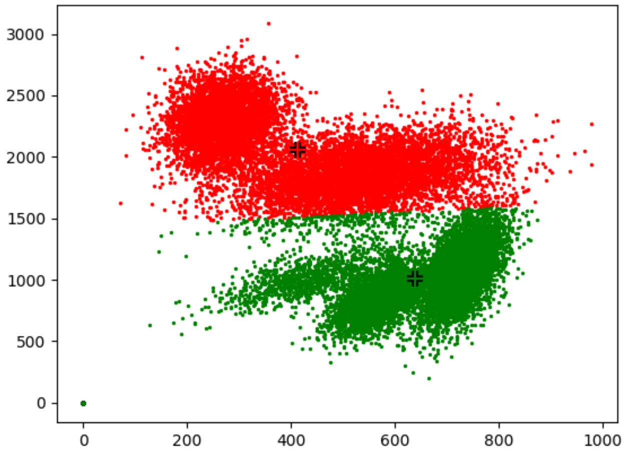

Figure 23: K-means with wrong number of compenents $m = 2$

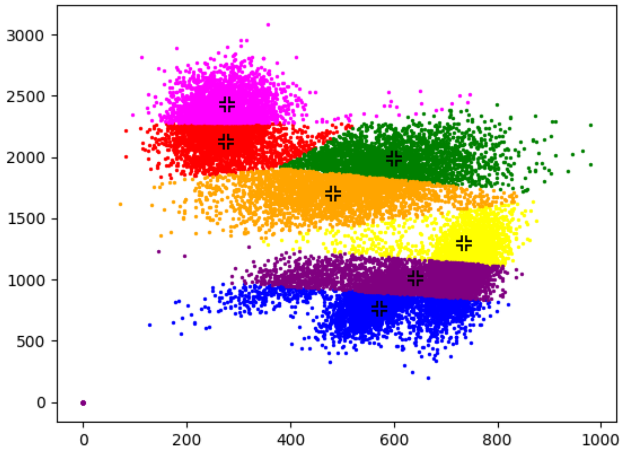

Figure 24: K-means with wrong number of compenents $m = 7$

Discussion

K-means does not perform as good as EM and also has problems with bad start values for μ_0. The same classes seem to be misclassfied. K-means (linearly separating class) seems not to be the best choice for this data (natural data, voice).

3 Samples from a Gaussian Mixture Model

1. Sampling GMM with $m = 5$, static intial parameters (linear) and 100 samples.

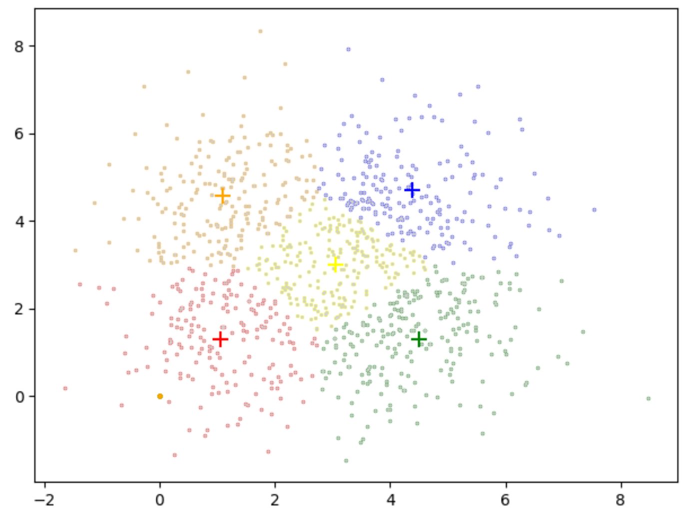

Figure 25: K-means for sampled GMM data $m = 5$

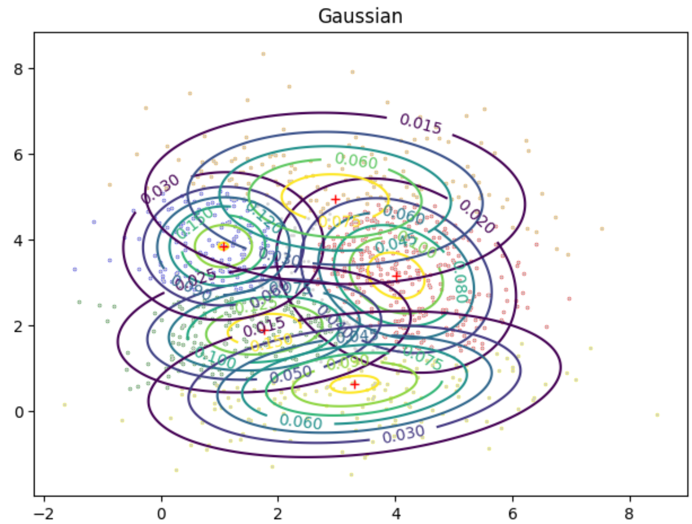

Figure 26: EM for sampled GMM data $m = 5$

Result

There is still a little TODO in the code for sampling GMM data.

References

[1] Franz Pernkopf. *Computational Intelligence: Teil 2 (Vorlesungsmitschrift)*. [Online; Stand 29. Juni 2016]. 2016.