NPGR035 Homework 1

Mgr. Matúš Goliaš, Doc. RNDr. Elena Šikudová PhD.

October 31, 2023

This document gives a brief description of the first homework and our requirements for an acceptable solution. Additional details are provided on the practicals or directly in the source code.

Description

Your task is to finish an implementation of a simple Gaussian mixture model optimisation algorithm also known as expectation-maximisation. You are given a source file `hw1.py` containing a Python class representing the model with three methods without implementation (they are marked with TODO comments). You are required to finish these three methods such that the algorithm converges on the example data and produces roughly the same results as written in `results/example_results.txt` and shown in the associated images.

Gaussian mixture model was discussed on the third proper lecture. The formulas which you are supposed to implement are on slides 42 and 44 of `MLCV_3.pdf`. We are working with a multivariate normal distribution, specifically in 2 dimensions so, instead of simple variances, we have covariance matrices. Therefore, you have to be extra careful when implementing the σ formula as it results in a matrix.

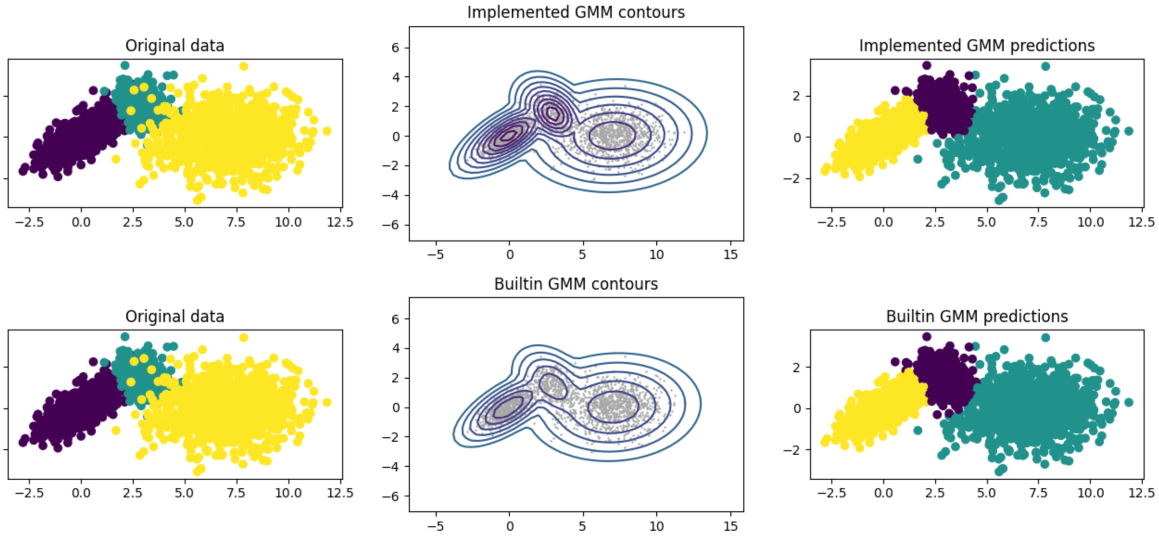

Figure 1: Example of the results given by running the evaluator.py script on test data. The top row shows custom implementation and the bottom row shows scikit-learn implementation.

Evaluation

Together with `hw1.py`, you get and evaluation script `evaluator.py` which can be used to test your solution. It runs your algorithm and displays the results by printing the optimised parameters and showing your mixture model alongside GMM implementation in scikit-learn. The two should be very similar but can't be exactly the same because of many factors. See Figure 1 for an example of the result. There are two generated datasets which you can use: `debug` and `test`. The exact parameters for running evaluation and comparing your results can be found in `results/example_results.txt`.

The correct implementation of the required formulas will be awarded full 100 points and mistakes will result in subtraction of points. Please, make sure that your solution runs and gives reasonable results before you submit because solutions that do not comply with `evaluator.py` will not be accepted.