Machine Learning – Handin 4

Clara Albiñana Climent (au571323), Marcos Vinicius Cruz (au561532), Judit Kisistók (au567357)
Gruppe Hold 5 – 12

Summary

We implemented Lloyd's algorithm as well as the Expectation Maximization algorithm (EM), and we experimented with using Lloyd's algorithm to initialize EM. We computed the Silhouette coefficient and the F1 score to assess the quality. We also used Lloyd's algorithm for image compression purposes and finally, we experimented with MNIST.
Our implementations work, however, we are not impressed by our EM's performance and lack of robustness. We ran into the problem of obtaining NaNs in the covariance matrix, and we figured that this is likely due to an underflow problem. The numerical stability could be improved by implementing the algorithm in logspace and applying smoothing, however, we haven't had time to explore this possibility. We bypassed the problem by breaking the loop when a NaN is encountered.

Section 1

We compared the results of Lloyd's algorithm and the Expectation Maximization algorithm in a visual manner – we decided to plot the clusterings.

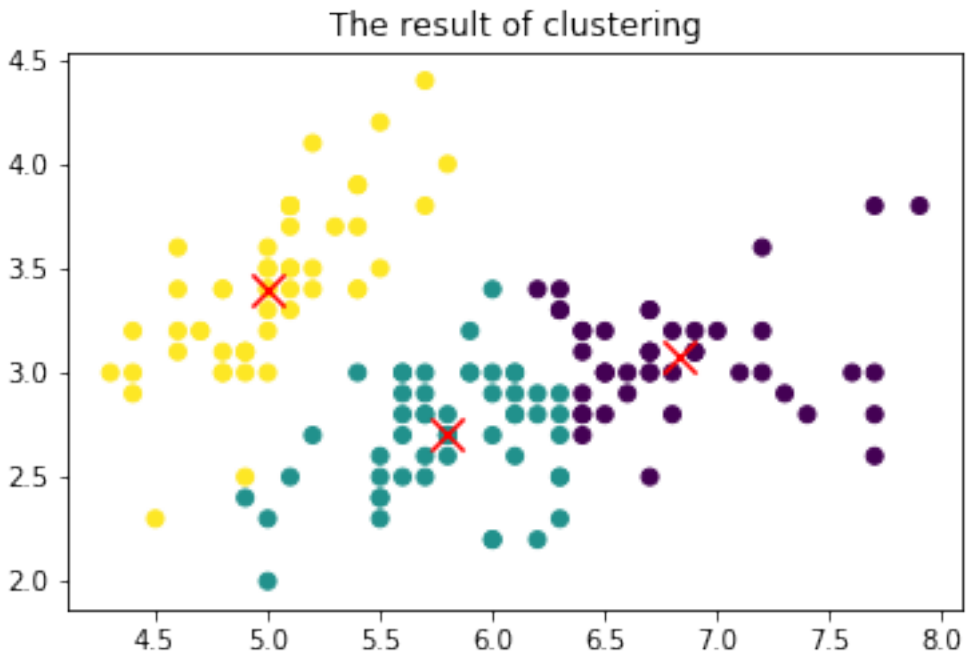
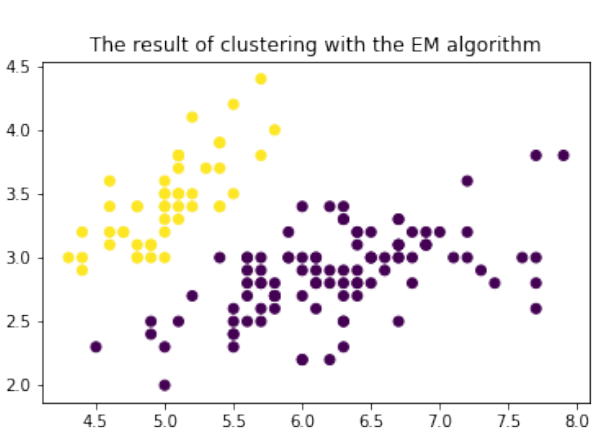


Figure 1. The result of Lloyd's algorithm.



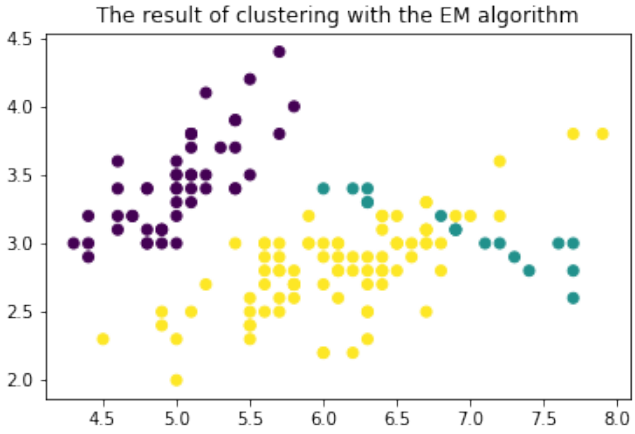
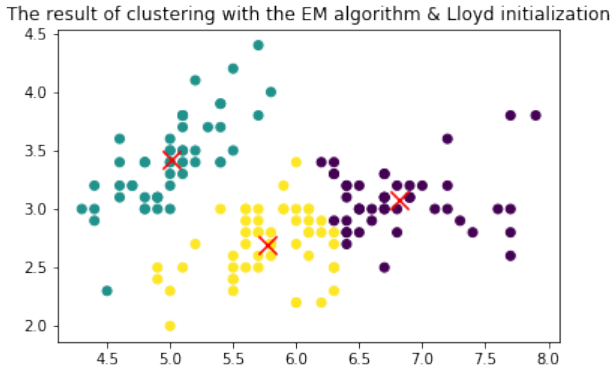


Figure 2a, b. Two results of the Expectation Maximization algorithm.



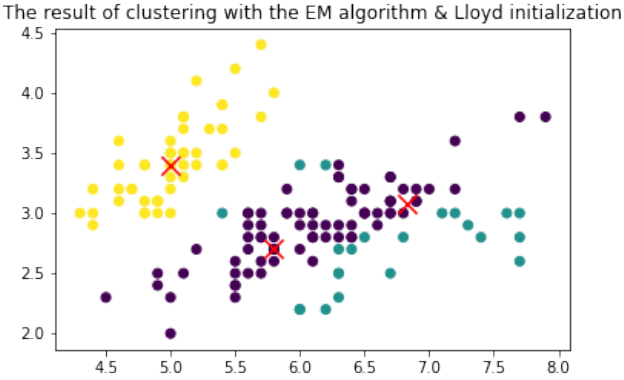


Figure 3 a, b. Two results of the Expectation Maximization algorithm when initializing the means with the centroids calculated by Lloyd's algorithm.

We can see that Figure 3a is rather comparable to Figure 1, so the EM algorithm didn't modify the clustering substantially in this case. On Figure 2a, we can see that we only got 2 clusters even though K was set to 3 – this can be due to the fact that the third cluster ended up being empty, because these two clusters can be very distinctly distinguished by eye, hence, we suspect that these two clusters yielded higher probabilities, overpowering the third cluster.

It is worth mentioning that the EM clustering changed quite a bit from run to run – this is illustrated by Figure 2b. We can see that in this case a third cluster does indeed show up, however, judging by eye it's not a particularly good result. The corresponding 3b plot shows that this clustering complemented with the Lloyd initiation still doesn't give very good results, especially in case of the green cluster.

Lloyd's algorithm gave us very stable results, therefore, in our implementation Lloyd's algorithm worked more reliably and we obtained reasonable clusterings as well.

Section 2.1

In order to evaluate the clusters resulting from our implementations of the algorithms we used both an internal (silhouette coefficient) and external (F1) validation measures, as indicated in the assignment.

Table 1 holds the the silhouette coefficient for the experiment.

k	Lloyd	EM_random	EM_lloyd_init
2	0.46407	0.44919	0.44919
3	0.44099	0.22758	0.25564
4	0.40119	0.44919	0.23529
5	0.35592	0.21877	0.23368
6	0.38756	0.08255	0.24800
7	0.37819	0.09698	0.09546
8	0.38522	0.05504	0.10041
9	0.36997	-0.03939	-0.04475

Table 1. Silhouette coefficient for k = 2...9.

The silhouette coefficient should range between -1 (badly assigned clusters) to 1 (appropriately clustered), and values close to 0 indicate the existence of border clusters, or the fact that both cases above may be happening in the data. For all the algorithms, the best Silhouette coefficient is given by the clustering on 2 groups, even though in the Lloyd algorithm the values at 3 are very close. If we compare with our visualization of the clusters these results actually make sense, as we could see two of the clusters combining in a bigger one.

Section 2.2

Table 2 holds the F1 score for the experiment.

k	Lloyd	EM_random	EM_lloyd_init
2	0.77624	0.62984	0.82608
3	0.81984	0.67268	0.68365
4	0.65093	0.53993	0.59938
5	0.56043	0.46218	0.50440
6	0.52590	0.47776	0.48600
7	0.45432	0.44249	0.44068
8	0.42344	0.41998	0.39993
9	0.39203	0.34999	0.36614

Table 2. F1 score for k = 2...9.

The maximum value of F1 that we could get is 1 when the number of clusters match the true number of partitions. All our values are below 1, having the maximum in both the Lloyd and the EM_random at r = 3, which is the true number of partitions. We can see that the value for k = 2 for the EM initialized with the Lloyd values is higher than at k = 3 – this is likely due to the relatively poor cluster assignment depicted on Figure 3b. Nevertheless, all of the cases with more partitions give smaller F1 values.

Section 2.3

We expect differences between these quality measures as they are conceptually different – the Silhouette coefficient examines how similar a point is to its own cluster and in the same time, how different it is compared to other clusters, thus, it is an unsupervised measure as it

doesn't take into account the true labels. Meanwhile the F1 score is a supervised measure, the harmonic mean of precision and recall, hence, it takes into account the true labels.

Section 3

We used Lloyd's algorithm to compress an image. We obtained the image via the download_image function, then applied Lloyd's algorithm with k = 4 and T = 100. We obtained a good compression ratio without modifying the code further. The original size of the image was 106292 bytes and we obtained a compressed size of 16247, which corresponds to a compression ratio of 6.54225. With k = 5 we obtained a more color-rich image with a compression ratio of 6.37701 (from 106292 bytes to 16668 bytes).

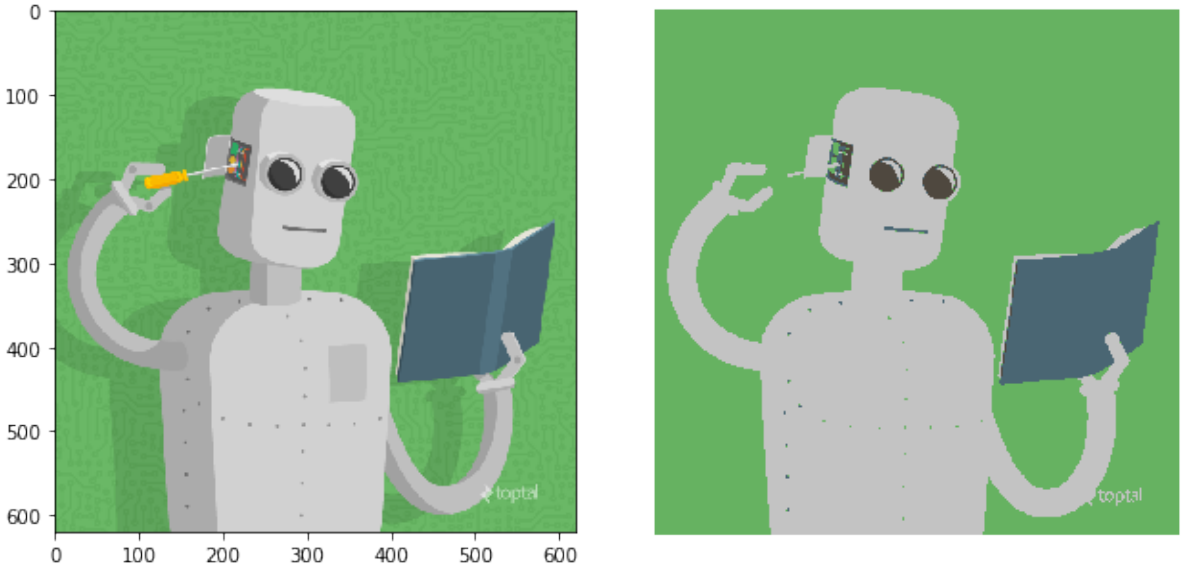


Figure 4. Image compression results for k = 4.

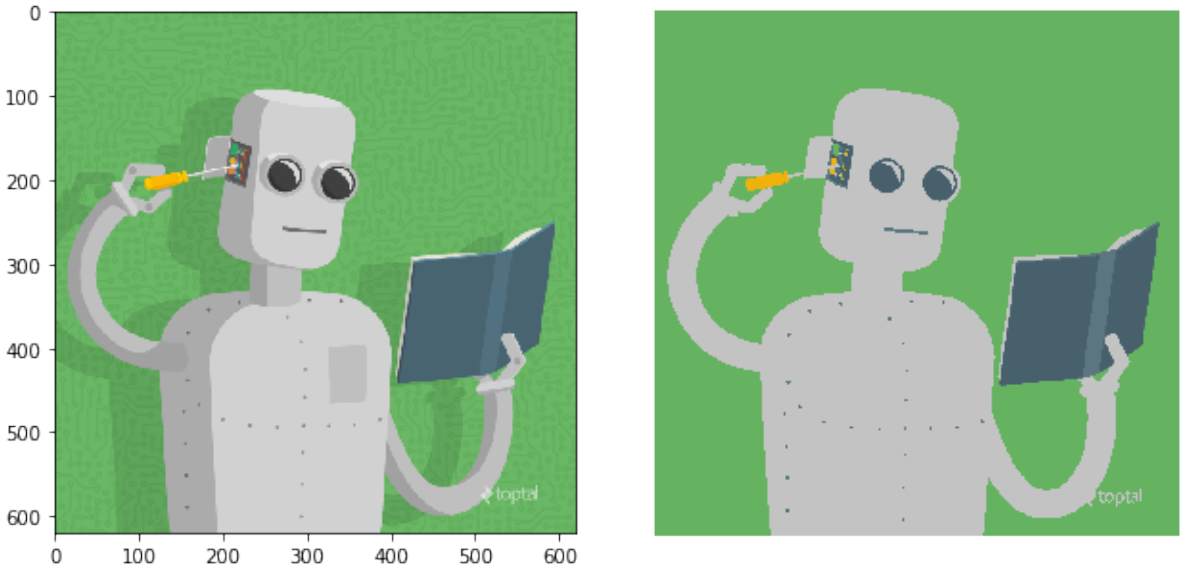


Figure 5. Image compression results for k = 5. We don't lose as many details, however, we don't get a substantially worse compression ratio either.

Section 4

We ran sklearn's EM on the MNIST dataset, which consists of handwritten digits. When we run EM, we fit to the handwritten digit training data and plot the means as a 28x28 image – these look like images of the digits, in this sense, the means correspond to pixel intensities. Apart from this, we are unsure about the details of this question.