Machine learning lab course
Problem set 2: Clustering, EM

ROHRMANN Till, Matrnr: 343756
till.rohrmann@campus.tu-berlin.de

May 23, 2013

Part I
Introduction

The second problem set dealt with the clustering problem. Clustering analysis is the problem to find groups of data points which which are more similar to each other than to data points of other groups. This plays an important role in a wide variety of fields such as machine learning, bioinformatics and data mining for example.

In the context of the work, I implemented and evaluated three different clustering algorithms: `k-means` as a representative of a centroid model, `hierarchical clustering` as a representative of a connectivity model and `EM algorithm` as a representative of distribution models. The evaluation was done by clustering provided test data sets and investigating how well the different groups have been found.

The report is structured as follows: In part II I will explain shortly the subtlenesses of the implementation and the encountered pitfalls while coding the algorithms. In part **??** the implemenations are applied to the test data sets. The interpretation of the results and the performance evaluation is given in this part as well.

Part II
Implementation

One problem I encountered while implementing the hierarchical clustering was that the `mergeidx` return value of `kmeans_agglo` did not work properly with the function `scipy.cluster.hierarchy.dendrogram`. In order to fix it, I changed slightly the semantic of `mergeidx`. The array `mergeidx` is a $k-1 \times 2$ vector with k being the number of initial clusters which shall be merged. `mergeidx[i][0]` and `mergeidx[i][1]` contain the identifier of the clusters which are merged in the ith step. But instead of giving the newly merged cluster the index `mergeidx[i][1]`, a new index $k+i$ is introduced and assigned to the cluster. However, this change does not affect the overall semantics of the algorithm.

A problem of the `EM algorithm` is a possible overfitting of a mixture component to one data point. Since one maximizes the likelihood of the Gaussian mixture components, it might be possible that one component is centered on one data point. By letting the covariance matrix converge to 0, the overall likelihood increases to ∞. One can prevent that from happening by regularizing the estimated covariance matrices:

$$CovMatrix \quad = \quad \frac{(\hat{X} - \mu)(\hat{X} - \mu)^T)}{\mathbb{1}^T \gamma} + \delta \cdot Id$$

Where $(\gamma)_i$ is the probability of the data point x_i belonging to the cluster μ, the columns of $\hat{X}$ are $\sqrt{(\gamma)_i} \cdot x_i$ and δ is the regularization constant.

Furthermore, while running the expectation-maximization clustering on the test data set `USPS`, I encountered such small determinants of the covariance matrices while computing the probability density function which lead to cancellation and consequently causing the algorithm to fail. In order to cope with those problems, one could have increased the regularization constant of the covariance matrices and thus preventing a too small probability. However, I decided to go another way and implemented the algorithm executing its calculations in the log space. By taking the log of a probability and calculating with those values, we can avoid the numerical problems of really small determinants. But nothing comes for free. While products

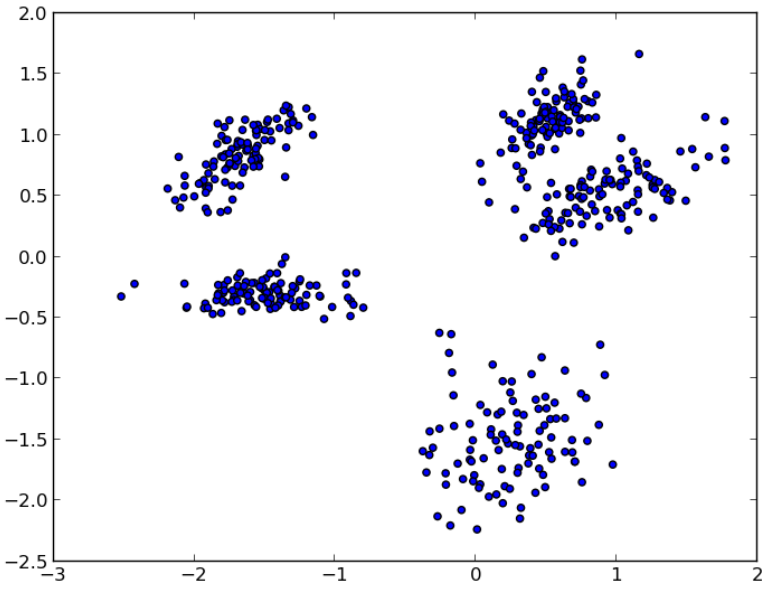


Figure 1: Scatter plot showing the data set `5gaussians`.

become sums in the log space, sums have to be treated differently. A numerically stable variant is the following. Given to variables a and b of which we want to calculate the sum in log space.

$$\begin{aligned} \log(a+b) &= \log a + \log\left(1+\frac{b}{a}\right) \\ &= \log a + \log\left(1+\exp\left(\log b - \log a\right)\right) \end{aligned}$$

Assuming that $a > b$, gives us now a numerically stable sum operation in the log space. For further details see the implementations in the file `ps2_implementation.py`.

Part III
Application

1 Assignment 6

In this assignment, we were supposed to apply the `k-means`, the `EM-algorithm` and the `hierarchical clustering` onto the data set `5gaussians`. This data set contains 500 2-dimensional data points sampled from a Gaussian mixture distribution. The distribution consists of 5 components with different means and covariance matrices. A scatter plot of the initial data can be seen in Figure 1. The distinct Gaussian distributions can be clearly seen.

Since it is apriori not known how many clusters a data set contains, the analysis is run for the number of clusters $k = 2, \ldots, 10$. The `k-means` algorithm was executed with a maximum iterations of 500. The `EM-algorithm` was run as well with a maximum iterations of 500. Additionally, the regularization constant was set to $\delta = 1e-5$ and the termination threshold of the log-likelihood to $\epsilon = 1e-3$. That is to say, as soon as the log-likelihood between the current iteration and the preceding iteration surpasses this threshold, the iteration procedure is stopped. Furthermore, I investigated the influence of a preceding initialization with `k-means` of

the EM-algorithm on the obtained results. Since I did not encounter any numerically problems with this data set, I used the EM-algorithm implementation which calculates in the normal space (see em_mog). Since the algorithms strongly depend on the initialization, the performance as well as the result, I executed all algorithms 30 times and took the result with minimal error and maximum log-likelihood, respectively. The 30 runs are also used to calculate the average number of iterations and the average time per iteration needed.

The error function for the k-means algorithm is simply the sum of the distances of each data point to its assigned cluster. For the EM-algorithm it is the sum of the distances of each data point to all clusters weighted by its membership probability of a cluster:

$$Error_{EM} = \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{i,j} \sqrt{\|X_i - \mu_j\|_2}$$

1.1 Do k-means and the EM-algorithm find the 5 clusters reliably?

In Figure 2 one can see a selection of clustering results of the algorithms k-means and EM-algorithm for different numbers of clusters. Especially the third row is interesting, because it contains the result with the correct number of clusters, that is to say 5. Here we can see that all variants find reliably the 5 clusters. For the k-means the different clusters are color-coded while the Gaussian mixture model is indicated by the covariance ellipses around the mean at the distance one time and two times the standard deviation. Furthermore, we can observe a deviating behavior of the two algorithms for a cluster number higher than 5. As we can see in the last row of Figure 2, the k-means algorithm successfully sub-divides the individual clusters into the half. In contrast, the EM-algorithm finds additional Gaussians mixture components which seem to be highly degenerated, that is to say they have a covariance matrix with a highly dominating eigenvalue.

In order to evaluate the performance of the different algorithms, I plotted in Figure 3 the error with respect to the number of clusters, in Figure 4 the number of iterations until convergence and in Figure 5 the total number of time needed until convergence.

We can see in Figure 3 that the k-means algorithm performs best. It is noteworthy that the error of the EM-algorithm with and without initialization stays almost constant for more than 5 clusters. This illustrates the incapability of the EM-algorithm to further discriminate the clusters. In Figure 4 we can observe that the number of iterations of k-means increases only slowly with increasing number of clusters compared to the EM-algorithms. For the EM-algorithms we can see that the average number of iterations until convergence increases quickly with the number of clusters. If we take a look at the total time needed for the computation in Figure 5, the same observation can be made. While the k-means runtime increases slowly the EM-algorithms' runtime increases rapidly. In fact, the average time for one iteration of k-means is one magnitudes faster than an iteration of the EM-algorithm. This shows quite well that the k-means algorithm is superior to the EM-algorithms for the 5gaussians data set. The result is not really surprising considering the more complex computations involved in the EM-algorithm.

1.2 What role does the initialisation of the EM-algorithm play?

In Figure 3 we can observe that the EM-algorithm with initialization does not necessarily benefit from the k-means initialization in terms of accuracy if one runs the version without initialization often enough (here 30 times).

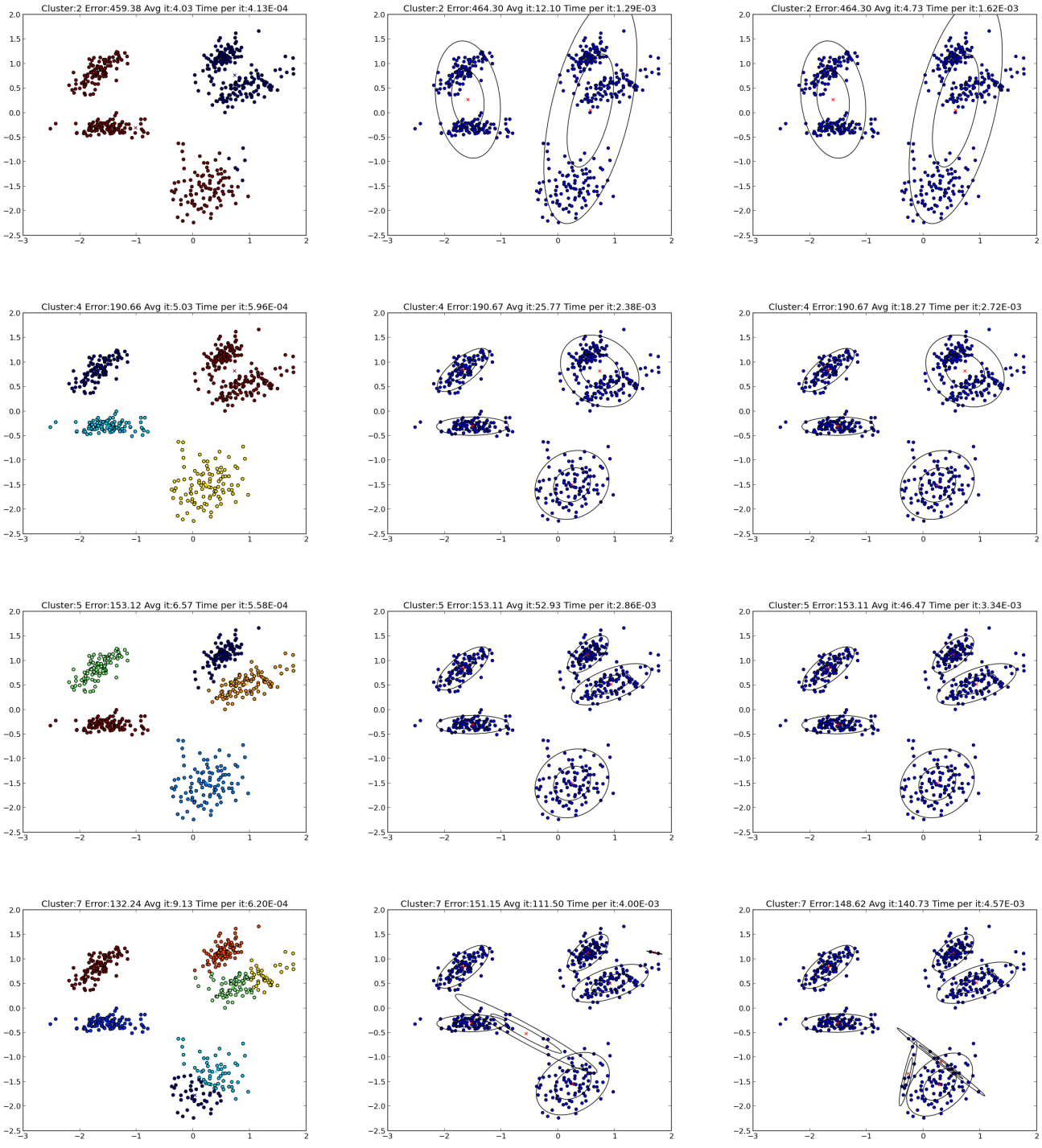


Figure 2: Results of clustering analysis for a subset of cluster numbers. Left column: Results of `k-means`. Middle column: Results of `EM-algorithm` without initialization. Right column: Results of `EM-algorithm` with `k-means` initialization. First row: 2 clusters, second row: 4 clusters, third row: 5 clusters and fourth row: 7 clusters.

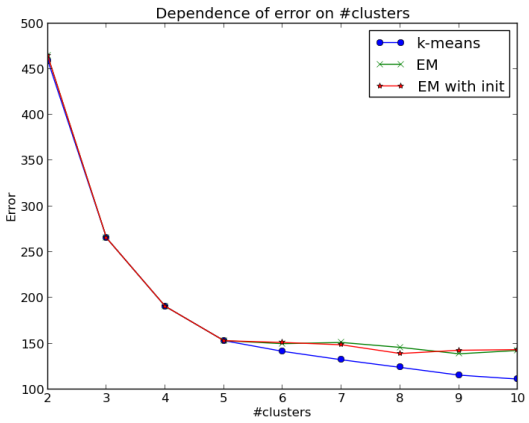


Figure 3: Error of clustering of `k-means`, `EM-algorithm` without and `EM-algorithm` with initialization with respect to the number of clusters.

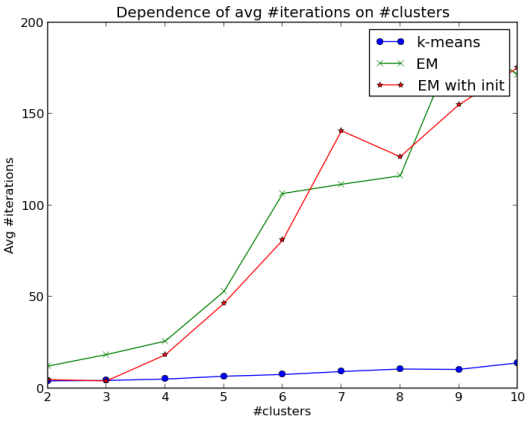


Figure 4: Number of iterations until convergence of `k-means`, `EM-algorithm` without and `EM-algorithm` with initialization with respect to the number of clusters.

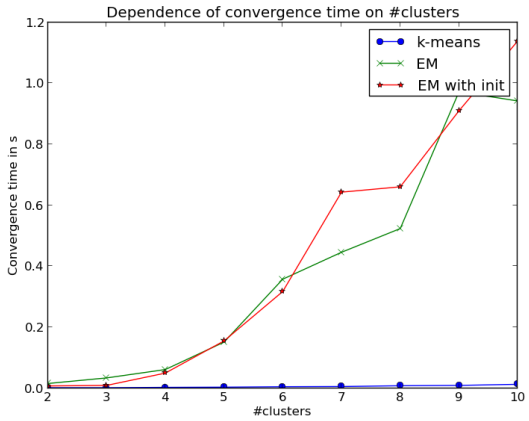


Figure 5: Total time until convergence of `k-means`, `EM-algorithm` without and `EM-algorithm` with initialization with respect to the number of clusters.

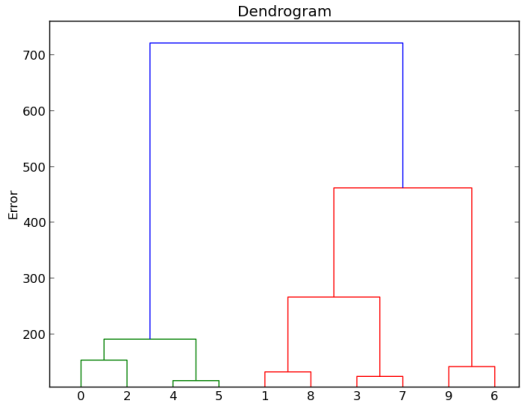


Figure 6: Dendrogram of the hierarchical clustering of the data set `5gaussians`.

However, the convergence rate benefits from the `k-means` initialization step. A preceding `k-means` initialization speeds the `EM-algorithm` noticeably up if the number of clusters is smaller or equal to the real number of clusters. However, once exceeded the real number of clusters, a prior initialization does not guarantee to improve the convergence rate. In that case, we see clearly in Figure 4 that the `EM-algorithm` with initialization performs sometimes better and sometimes worse than the same version without initialization. If we take a look at the convergence time, we can observe a similar behavior: The initialized `EM-algorithm` performs first better than the non initialized version if the number of clusters is smaller or equal to 5. If the number of assumed clusters is higher than 5, an initialization does not necessarily speed the convergence time up.

1.3 What does the dendrogramm of the hierarchical clustering look like?

In Figure 6 we can see the dendrogram of the hierarchical clustering algorithm which has initially started with 10 clusters. Estimating the number of cluster k from the dendrogram, I would have chosen $k = 4$. The reason for that is that there is jump of almost 50% of the preceding error.

2 Assignment 7

In this assignment, we were supposed to analyse the `2gaussians` data set with the `k-means` and `EM-algorithm`. The data set consists of 200 2-dimensional data points which are sampled from two Gaussian distributions which are stretched in one direction. Furthermore, they are aligned parallelly. A scatter plot of the data set is shown in Figure 7. The maximum iterations was again chosen to be 500 for both algorithms. The regularization constant was set to $\delta = 1e - 5$ and the termination threshold for the log-likelihood was set to $\epsilon = 1e - 3$. Moreover, the algorithms were executed with $k = 2$ (number of clusters). Each algorithm was run 40 times and the best result in terms of minimal error and maximum log-likelihood, respectively, was chosen.

Figure 8 shows the results of the clustering algorithms `k-means` and `EM-algorithm` with and without initialization. We can clearly see that `k-means`, Figure 8a, fails to cluster the data correctly, because it assumes the data to be circularly located around its cluster center. However, if the data comes from a Gaussian distribution which is highly stretched, this assumption does not hold anymore. In contrast, the `EM-algorithm`, Figure 8b can correctly model the data

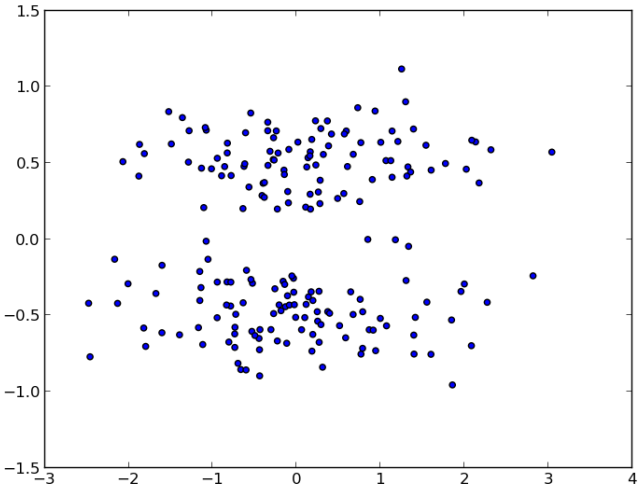


Figure 7: Scatter plot of the `2gaussians` data set.

set and seems to estimate the covariance matrices correctly as well. But this only works if the `EM-algorithm` is not initialized. If we apply a `k-means` initialization phase before executing the `EM-algorithm`, we cannot find the correct clusters, as it can be seen in Figure 8c. Apparently, the `EM-algorithm` cannot properly recover from the preadjustment of the initialization phase.

3 Assignment 8

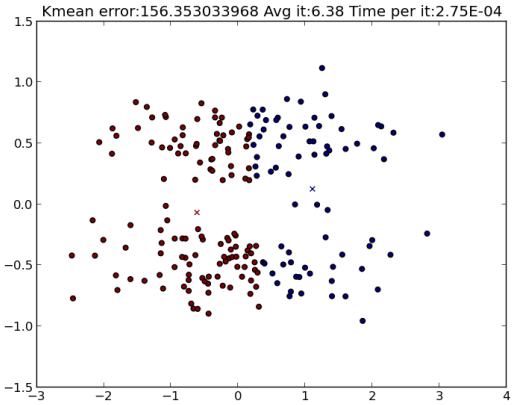
In the context of this assignment, I applied `k-means` and the `EM-algorithm` with and without initialization to the `USPS` data set. The `USPS` data set contains 16×16 images of hand-written numbers from 0 to 9. The data set contains 2007 data points. The maximum iterations was chosen to be 500 and 100 for `k-means` and `EM-algorithm` respectively. The regularization constant was set to $\delta = 1e-5$ and the termination threshold for the log-likelihood was set to $\epsilon = 1e-3$. Moreover, the algorithms were executed with $k = 10$ (number of clusters). All algorithms are executed 20 times and the best result with respect to the minimal error and the maximum log-likelihood, respectively, is chosen.

3.1 Which algorithm delivers better results?

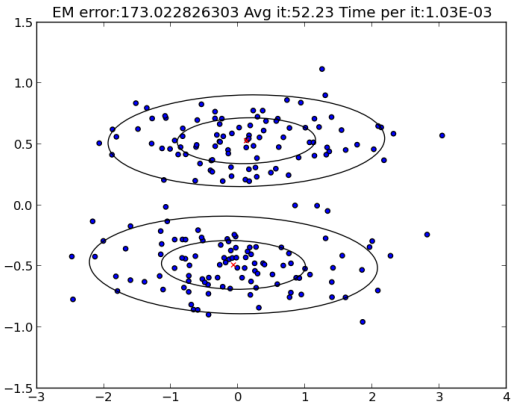
Figure 9 shows the centroids of clusters found by `k-means`, Figure 9a, by the `EM-algorithm` without initialization, Figure 9b, and by the `EM-algorithm` with initialization, Figure 9c. Regarding the centroids, one sees that `k-means` produces the most discriminative centroids. The only number which is missing is the 5, because it is usually very similar to the 3 and thus represented by the centroid of the 3s. The `EM-algorithm` with initialization delivers comparable results which have the same problem with the 5. But using this algorithm without an initialization phase produces highly ambiguous centroids, e.g. the 3 and 8, 0 and 5, 7 and 9 are mixed.

3.2 Dendrogram

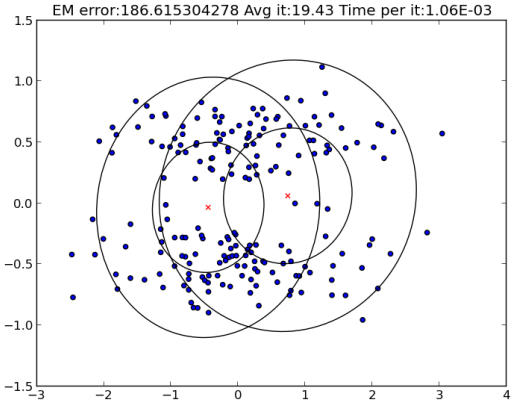
Figure 10 shows the dendrogram of the hierarchical clustering applied on the `USPS` data set. Figure 11 shows the initial centroids found by `k-means` and which are used for the hierarchical clustering algorithm.



(a)

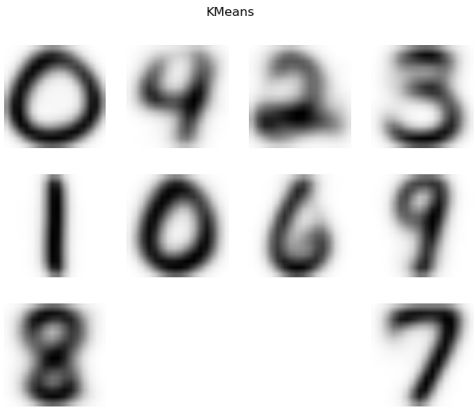


(b)

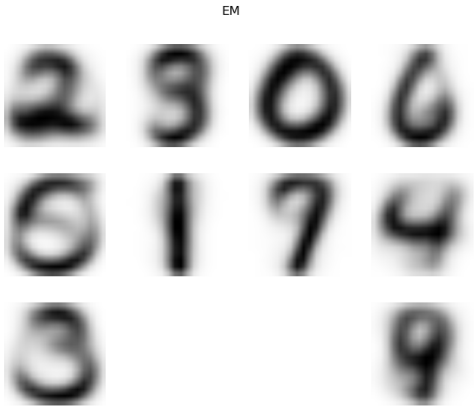


(c)

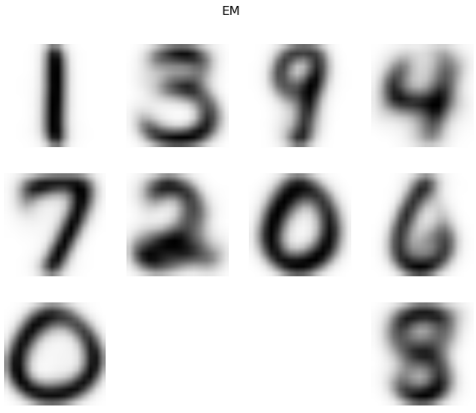
Figure 8: Clustering results of the data set `2gaussians`. (a) `k-means`, (b) `EM-algorithm` without initialization and (c) `EM-algorithm` with initialization.



(a)



(b)



(c)

Figure 9: Cluster centroids of the data set `USPS`. (a) `k-means`, (b) `EM-algorithm` without initialization and (c) `EM-algorithm` with initialization.

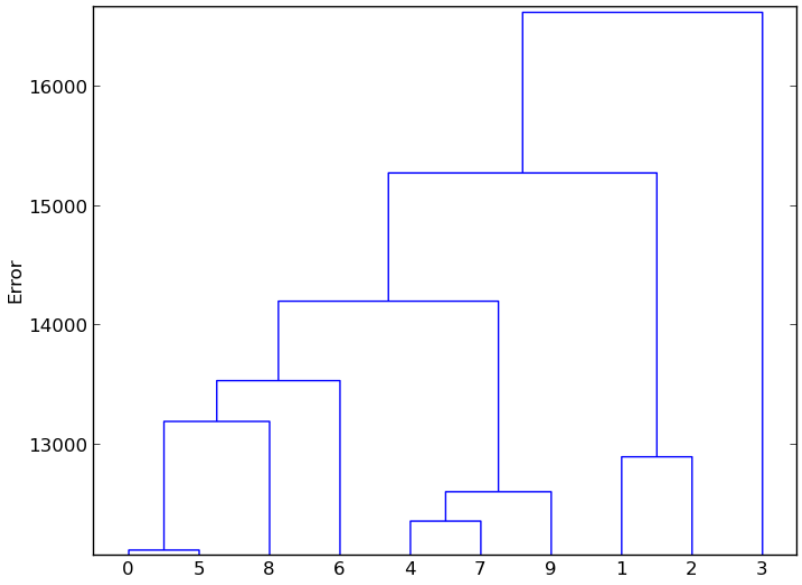


Figure 10: Dendrogram of the hierarchical clustering of data set `USPS`.

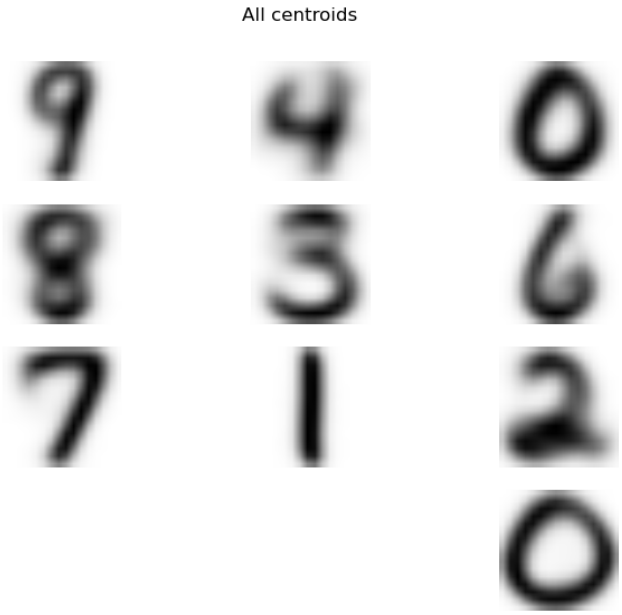


Figure 11: All initial centroids found by `k-means`.

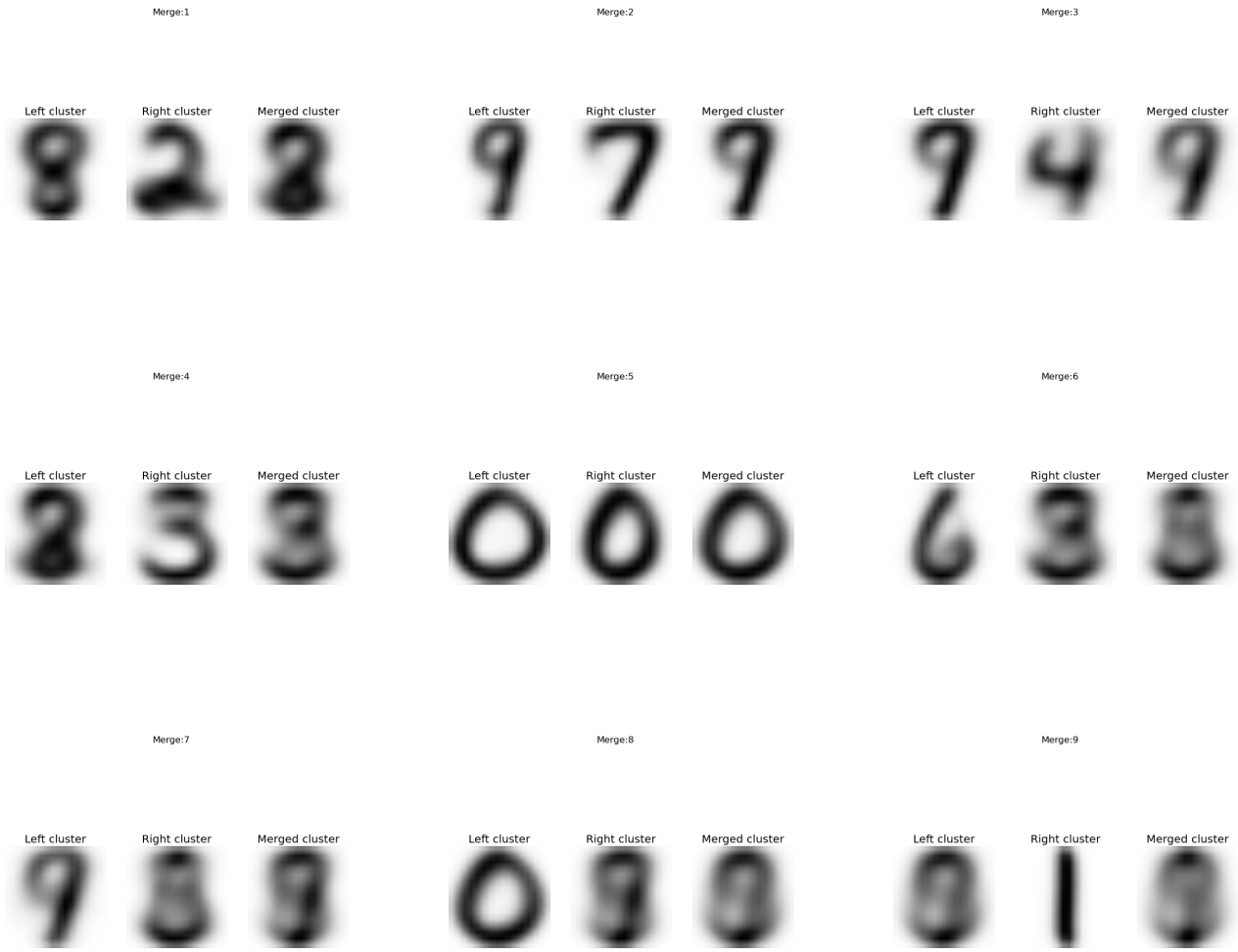


Figure 12: Centroids before and after merge operation. From left to right and top to bottom is the chronological order.

Figure 12 shows in chronological order from left to right and top to bottom the different cluster merges. The left and middle image show the centroids of the two clusters chosen to be merged and the right image shows the resulted centroid. We can see that at first clusters with similar centroids are merged and that at each step the cluster differ a little bit more from each other. Furthermore, we can observe that the resulting centroid loses more and more its distinctness. This is not surprising considering the fact that after each merge operation the resulting cluster has to represent more and more different numbers.