

Assignment 5

Computational Intelligence, SS2020

Team Members		
Last name	First name	Matriculation Number
Blöcher	Christian	01573246
Bürgener	Max	01531577

1 Classification - 2 dimensional feature

1.1 EM algorithm

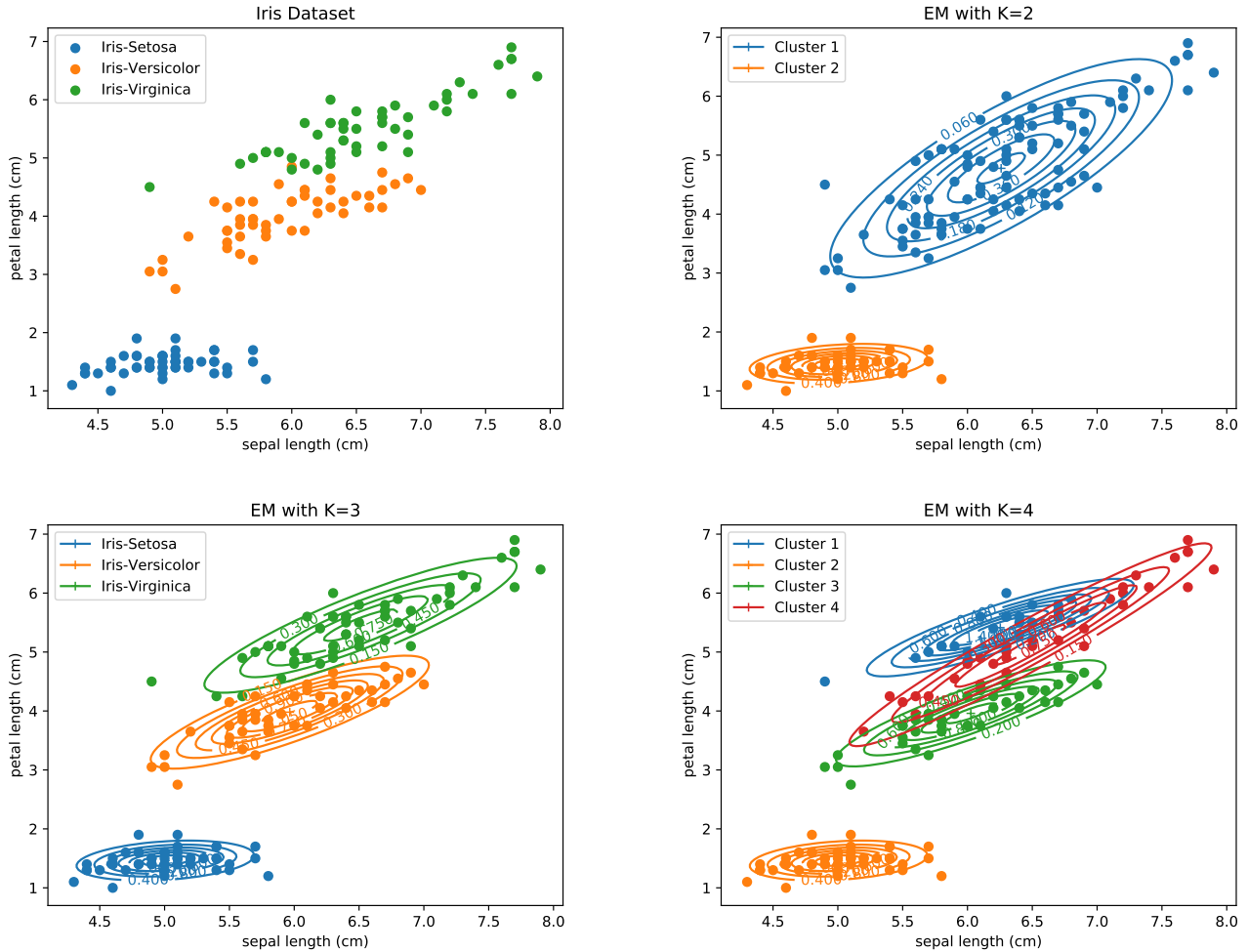


Figure 1: Results of EM classification with $K = 2 \dots 4$ components.

Figure 1 shows the dataset and classification results with contours of the used Gaussian kernels for $K = 2, 3$ and 4 components. The *Setosa*-Cluster is fully identified with all numbers of components. Using only two components *Versicolor* and *Virginica* are combined into a second cluster. With $K = 3$ both remaining classes are classified quite well with only *Versicolor*-samples being misclassified as *Virginica* within the class overlap-region. Using four components a third cluster is formed in this region, leading to incorrect classification. The best results are achieved when using the same number of clusters as the number of classes.

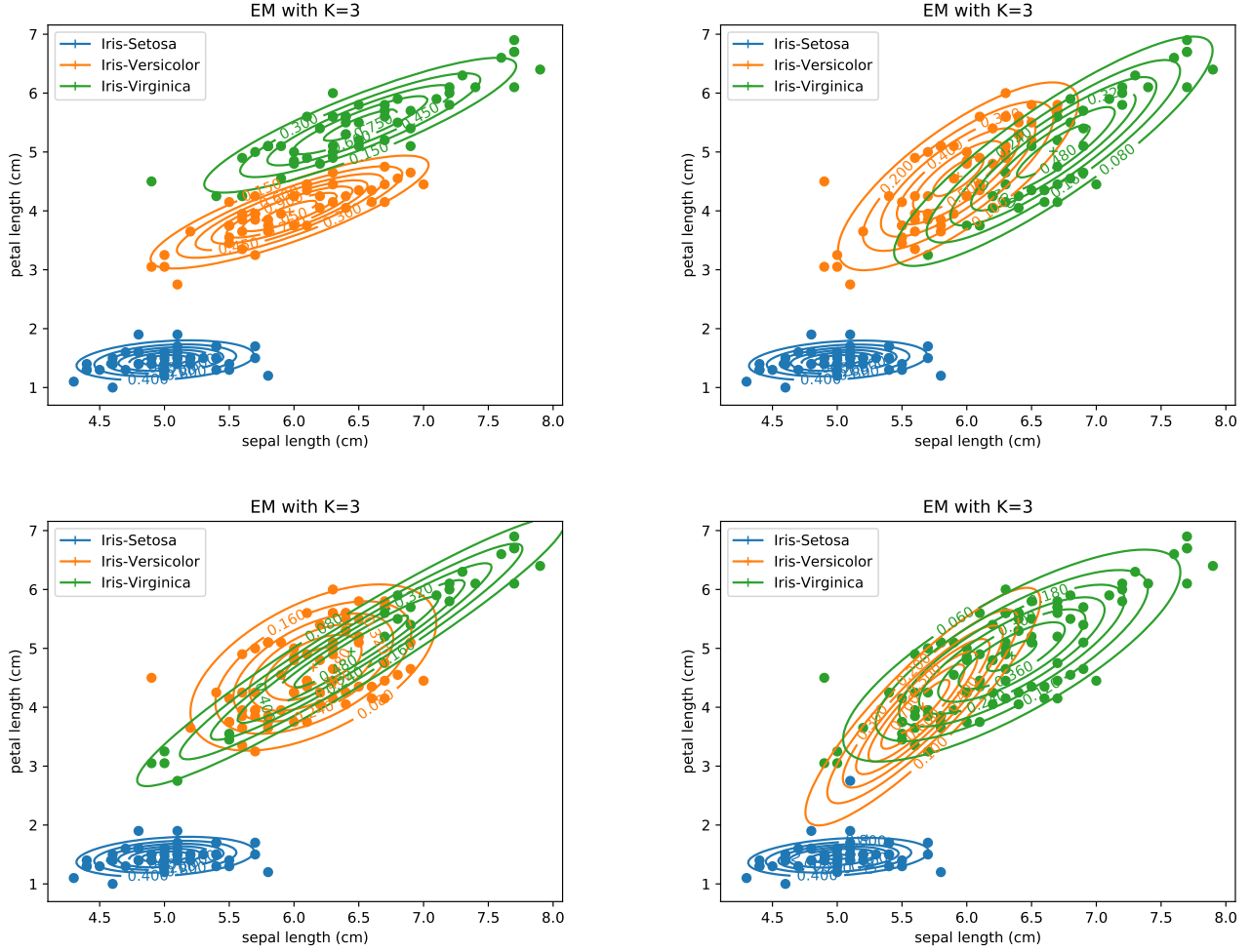


Figure 2: Results of EM classification for several random `mean0` starting samples.

The parameters Θ_0 are initialised as specified in the lecture notes for all K clusters: `alpha0` as $1/K$, `mean0` as random sample of the dataset (different for each cluster), and `cov0` as the multi-variate covariance of the dataset. Because of the non-convex form of the likelihood as a function of Θ_0 , finding the global maximum depends on the starting point of the optimisation process. With some initialisation values one might only find local maxima instead, even though the log-likelihood function increases monotonically over the iterations (s. figure 3). Thus the quality of the classification varies greatly for different `mean0` starting samples (s. figure 2 for some examples). In some instances one of the cluster weights `alpha` is even set to zero during the optimisation, resulting in only two classes being recognised.

Figure 4 shows the results of the soft-classification done in the E-step during the optimisation process. In the beginning the cluster borders (especially between *Setosa* and *Versicolor*) are random, mostly depending on the samples used to initialise `mean0`. At iteration 8 the *Setosa*-class is correctly classified. In between iteration 8 and the eventual convergence at iteration 23 only the border between *Versicolor*- and *Virginica*-clusters changes slightly. This is, because the log-likelihood (s. figure 3) increases only little after the big jump at iterations 3-5.

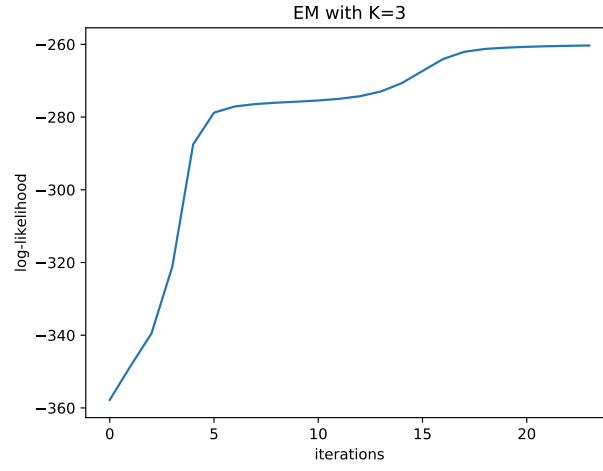


Figure 3: Log-likelihood function over iterations for $K = 3$ components.

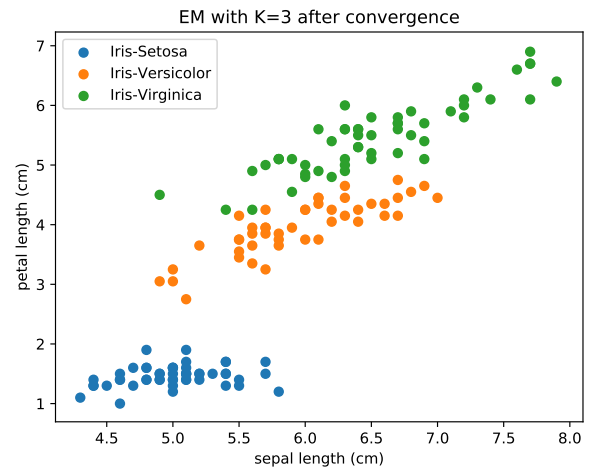
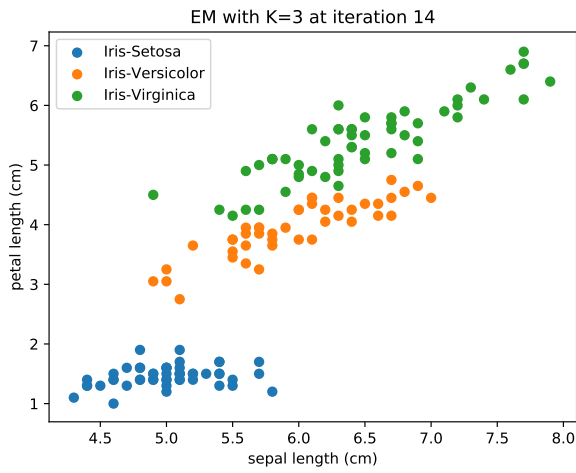
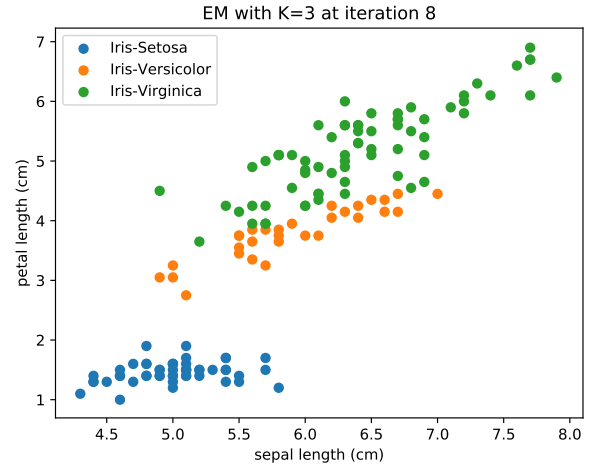
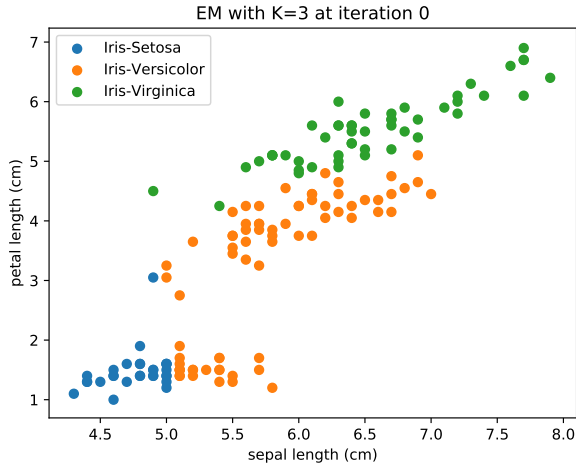


Figure 4: Results of soft-classification done in E-step.

1.2 K-means algorithm

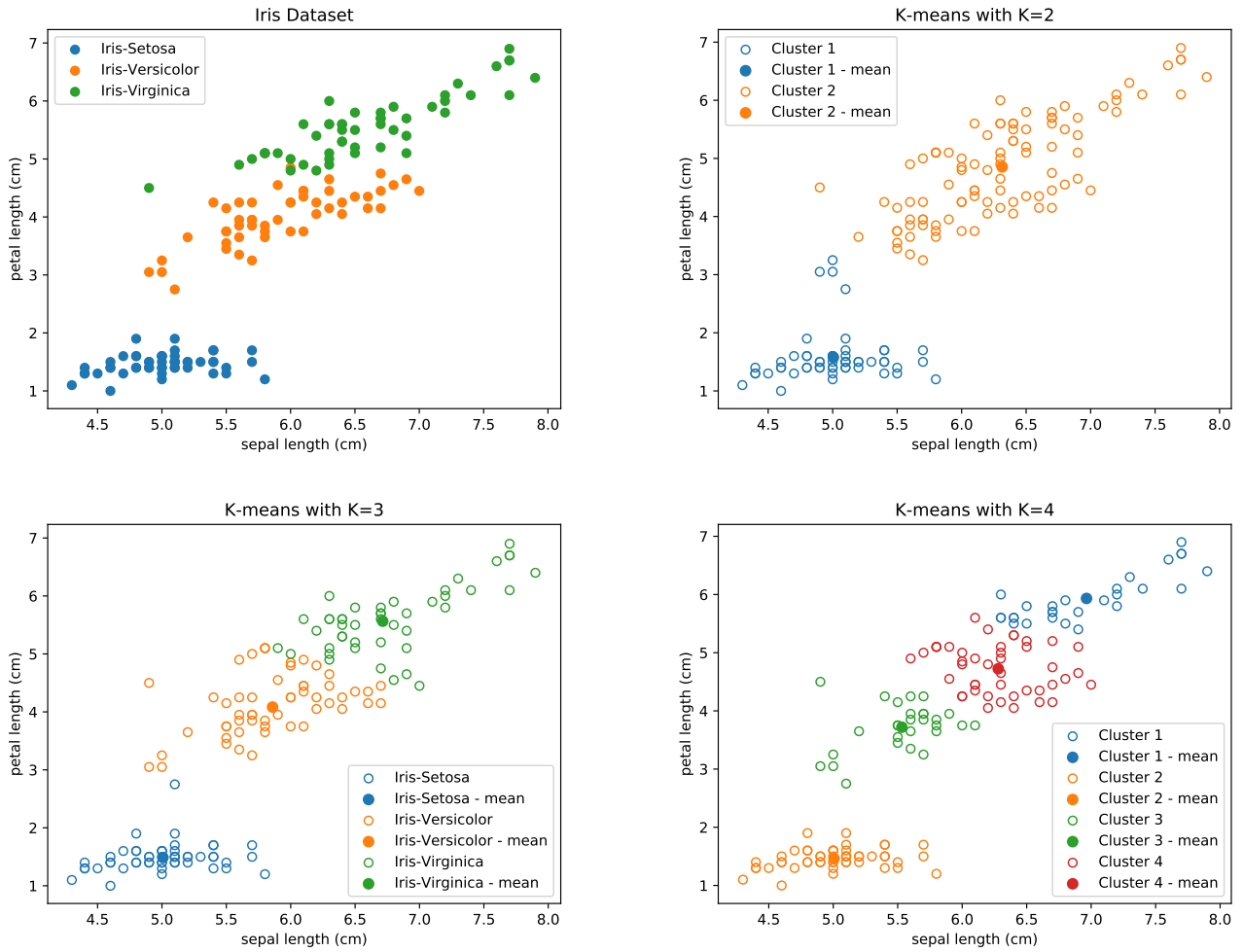


Figure 5: Results of K-means classification with $K = 2 \dots 4$ components.

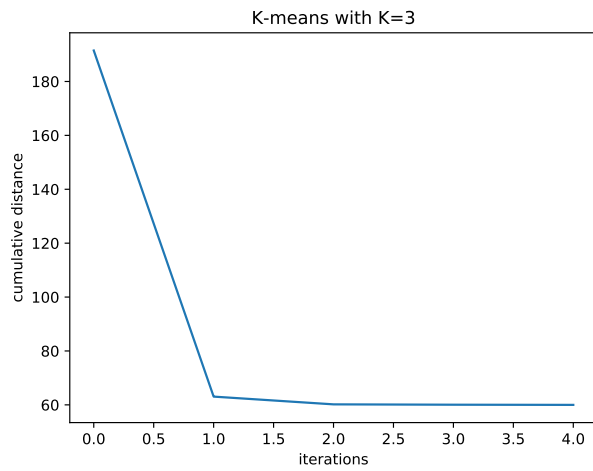


Figure 6: Cumulative distance function over iterations for $K = 3$ components.

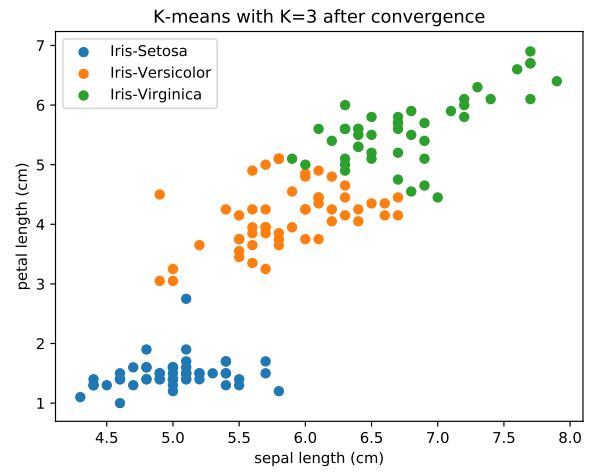
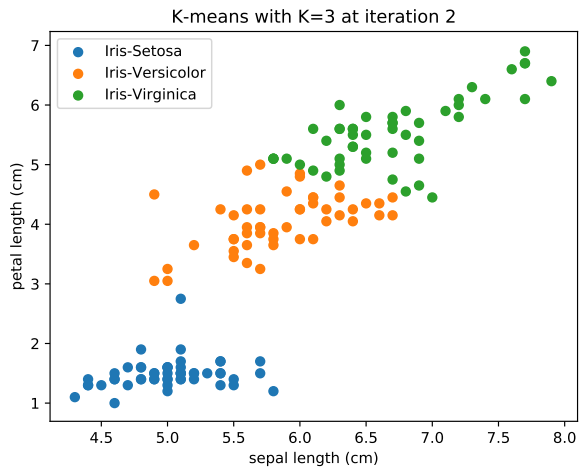
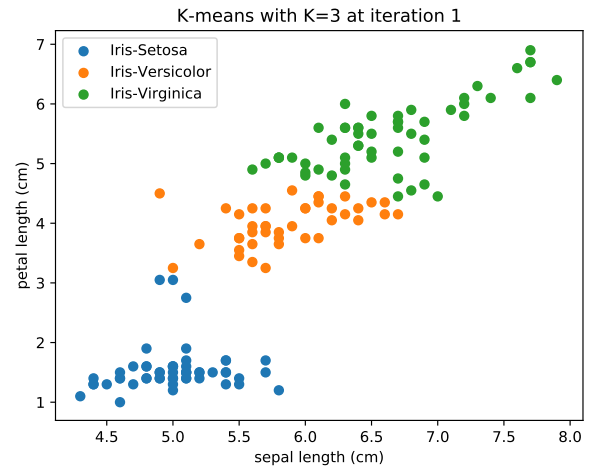
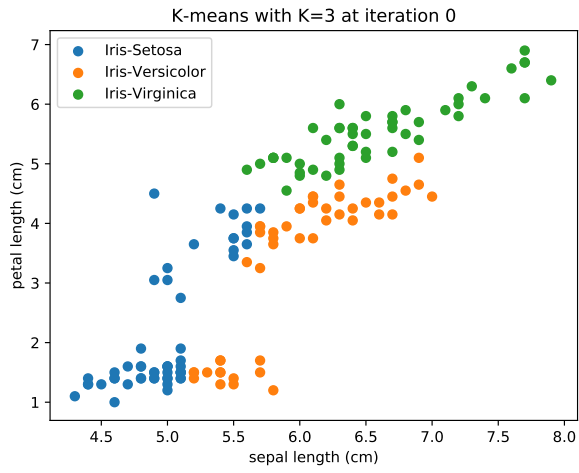


Figure 7: Results of hard-classification during optimisation.

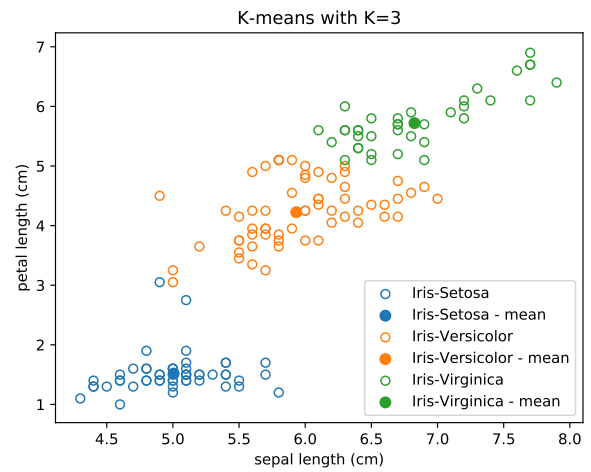
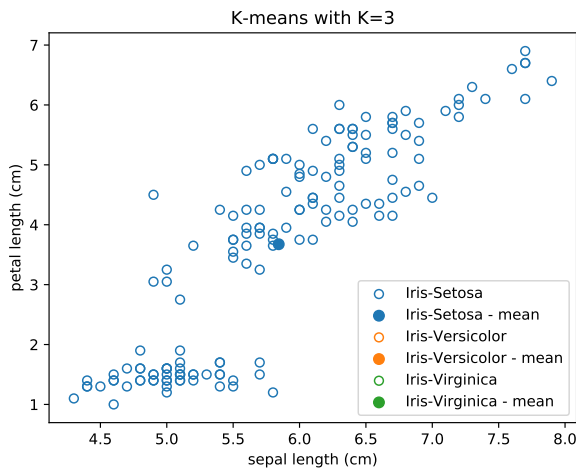
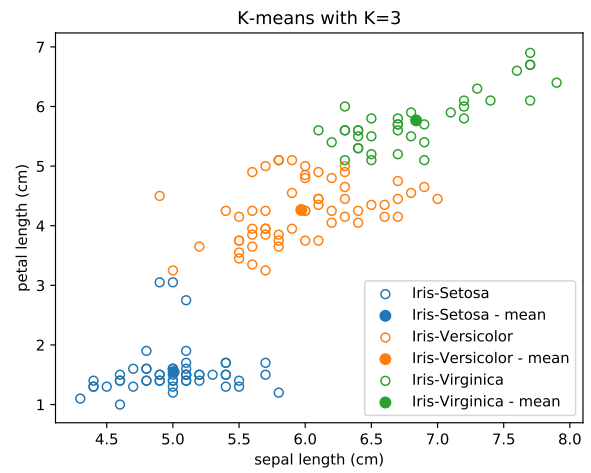
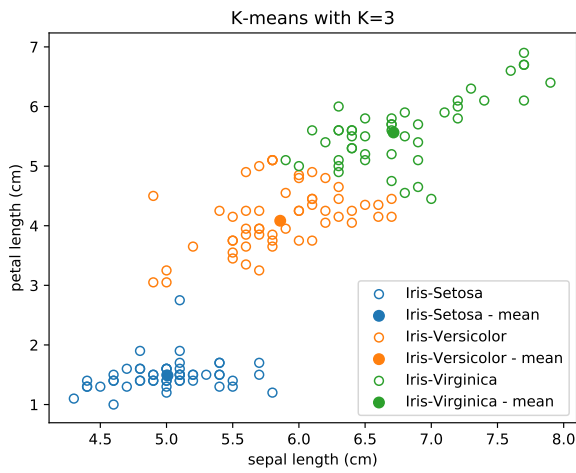


Figure 8: Results of K-means classification for several random `center0` starting samples.

2 Classification - 4 dimensional feature

2.1 EM algorithm

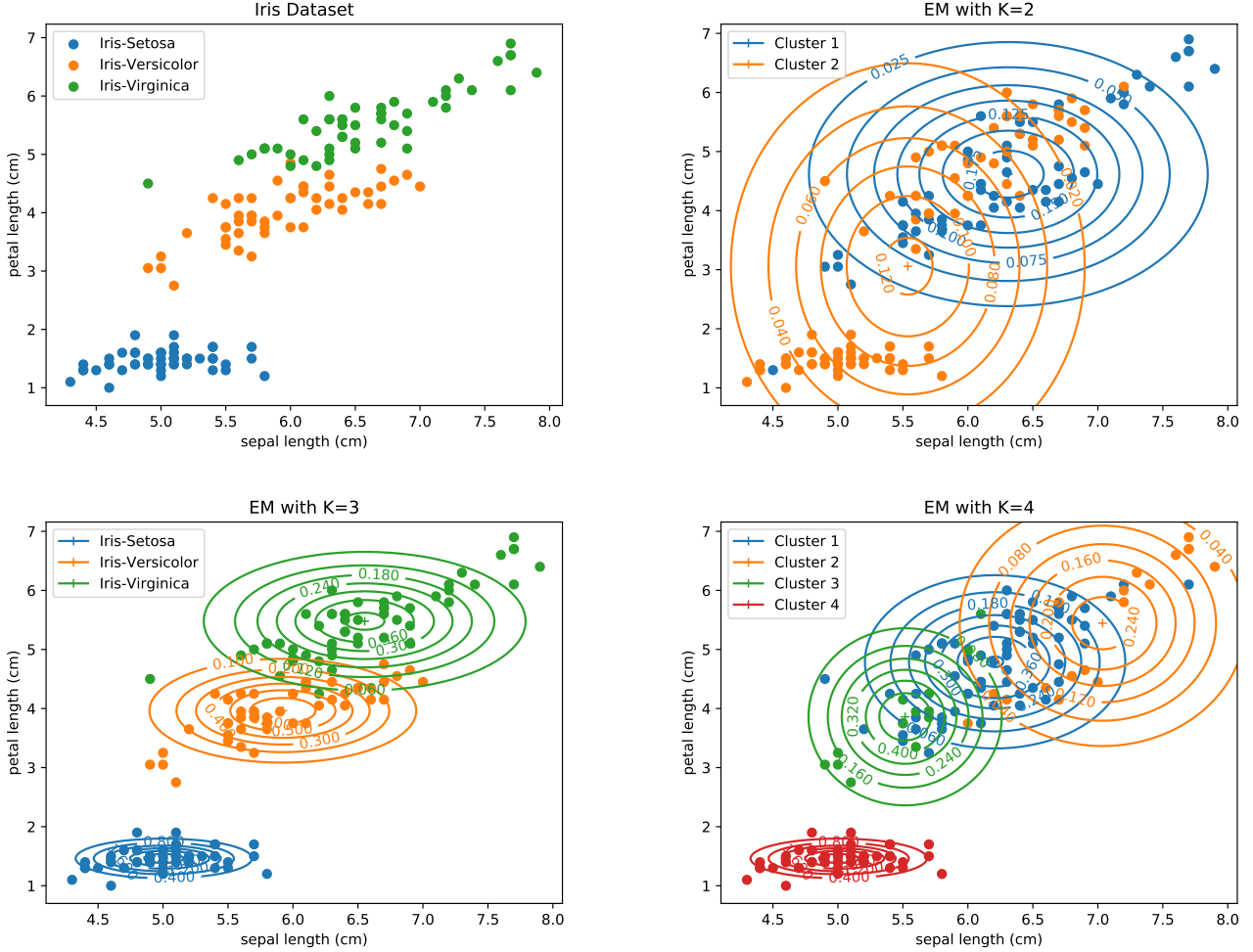


Figure 9: Results of EM classification with $K = 2 \dots 4$ components.

Applying the EM algorithm to the full dataset with $K = 3$ components the accuracy after classification stays roughly the same as in scenario 1 (s. figure 9). Using two components the clusters overlap, making it impossible to distinguish any classes. With four components the *Setosa*-class corresponds to Cluster 4, but the remaining clusters overlap as well. The overlap might also occur, because not the whole feature space is plotted. Classes may appear to be better separable when displaying other features.

Unfortunately the quality of the classification is still strongly dependent on the data samples chosen for `mean0` (s. figure 10).

The optimisation process takes longer to converge as with only two features (s. figure 11). Because only two features are plotted one cannot comprehend the cluster-formation due to the soft-classification updates in the E-step (s. figure 12) as easily as in scenario 1.

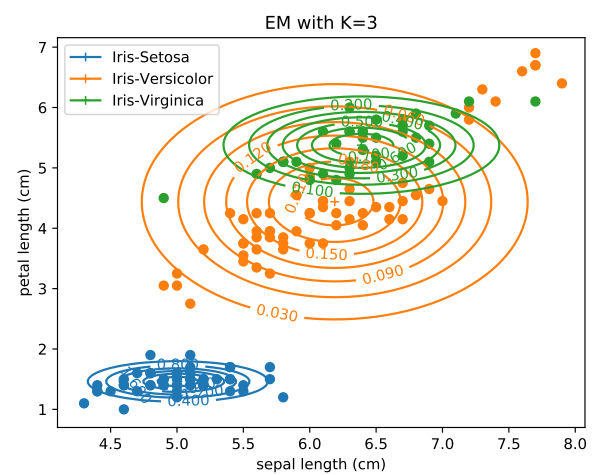
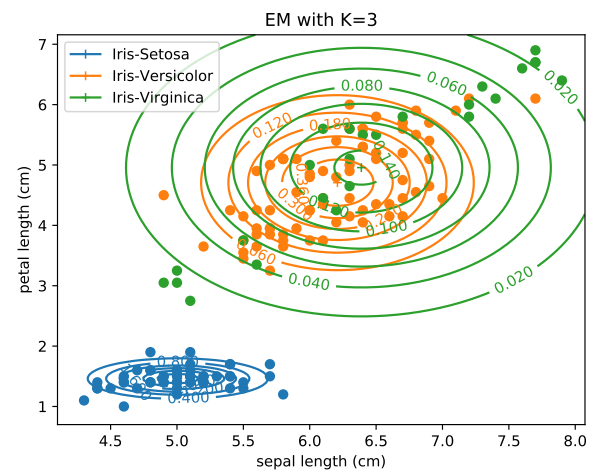


Figure 10: Results of EM classification for several random `mean0` starting samples.

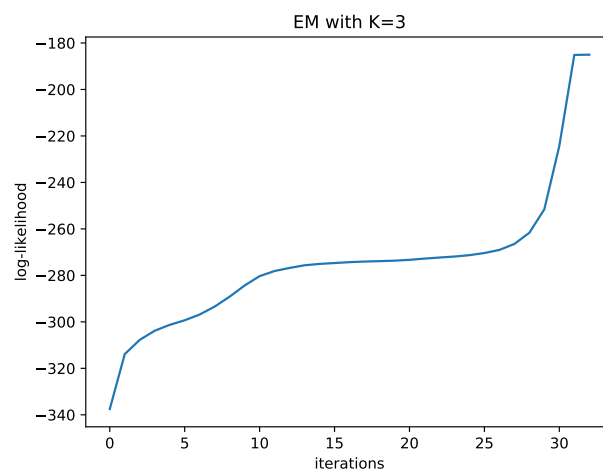


Figure 11: Log-likelihood function over iterations for $K = 3$ components.

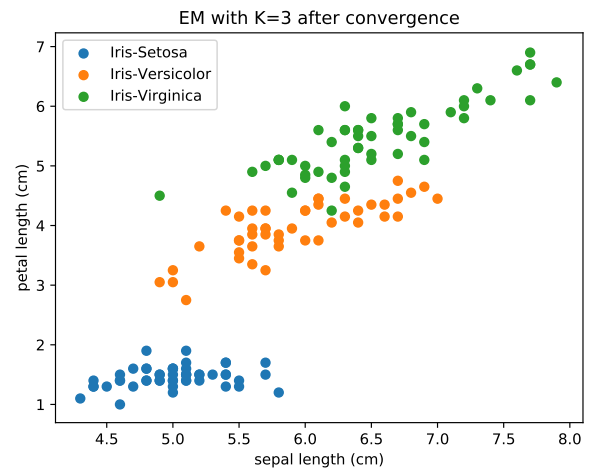
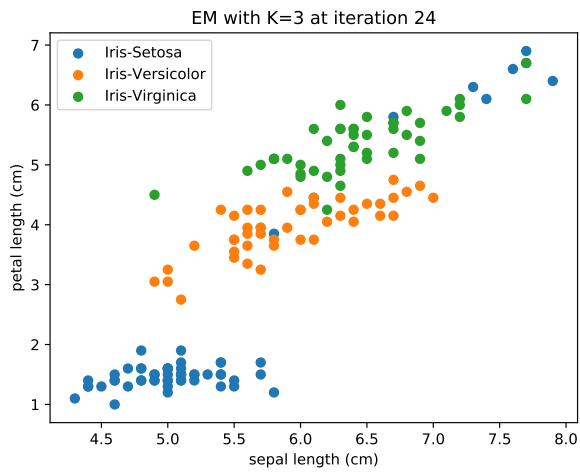
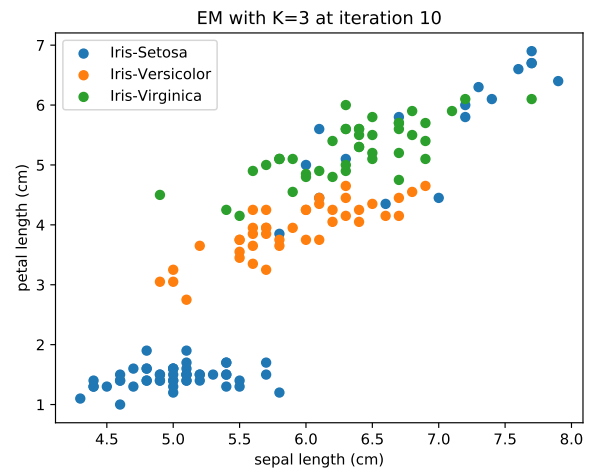
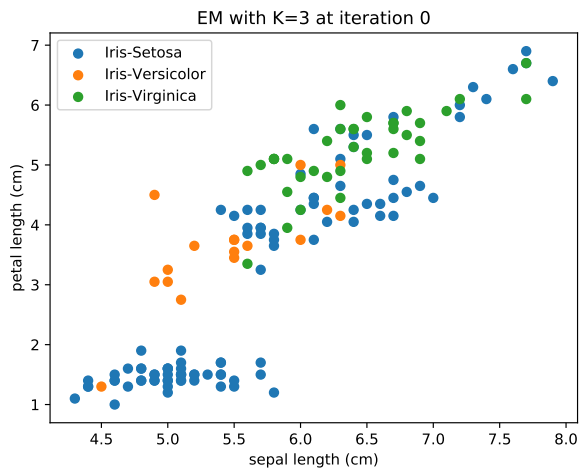


Figure 12: Results of soft-classification done in E-step.

2.2 EM algorithm with diagonal covariance matrices

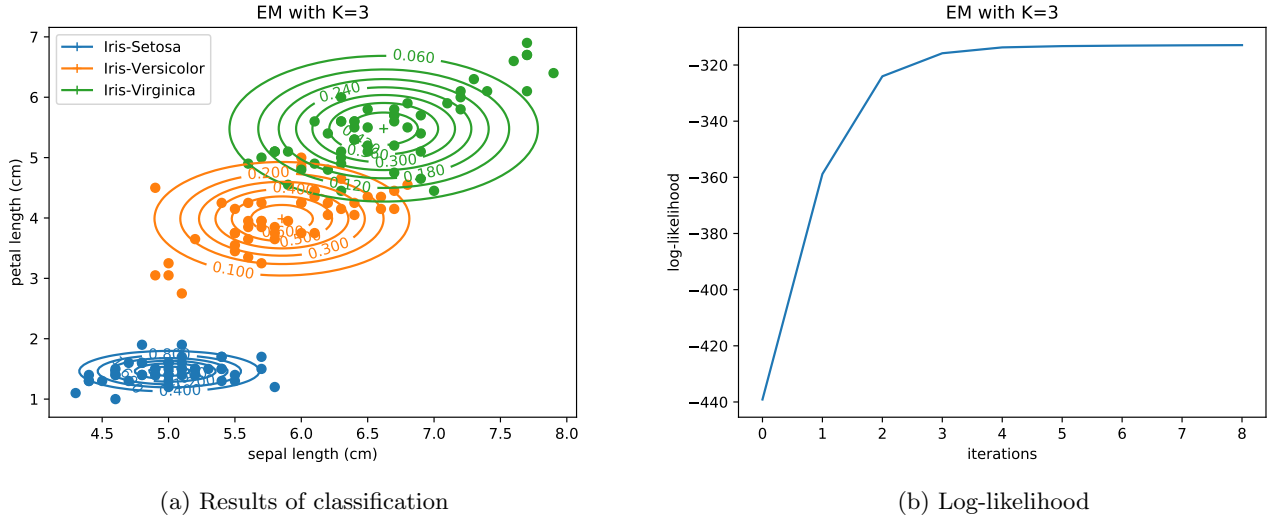


Figure 13: Results of EM classification with diagonal covariance matrices.

Confining the structure of the covariance matrices to diagonal matrices sets the covariance between the features to zero. This prevents the Gaussian kernels displayed in figure 13a from being tilted, which corresponds well with the dataset. While the resulting classification is not quite as accurate as with the very best results using non-diagonal covariance matrices (visible in the overlap region of *Versicolor* and *Virginica*), the quality of the results is not dependent on the initialisation of `mean0` anymore. Because there are fewer degrees of freedom with only diagonal covariance matrices the algorithm also converges much faster (s. figure 13b).

2.3 K-means algorithm

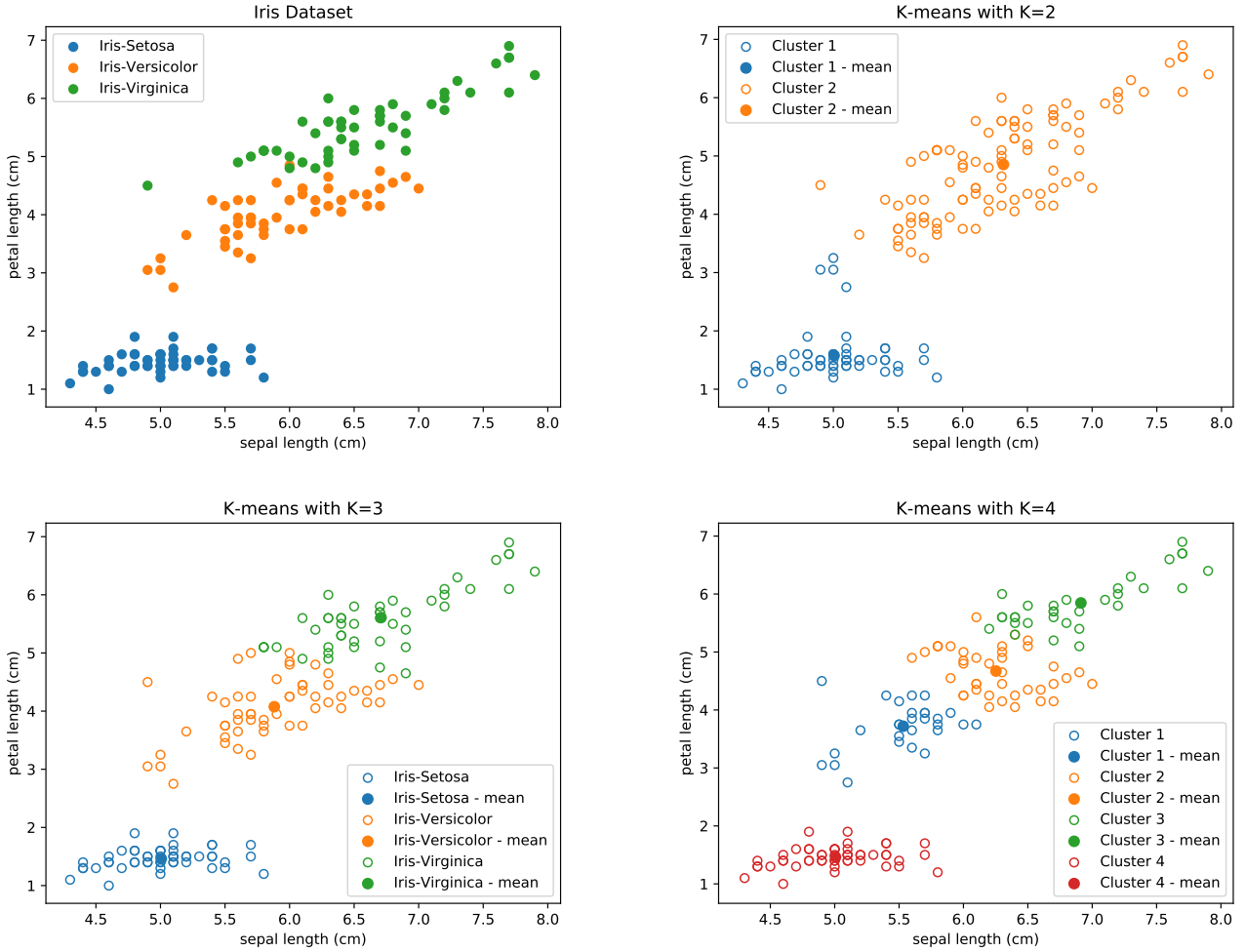


Figure 14: Results of K-means classification with $K = 2 \dots 4$ components.

While the classification result with two components using the whole feature set is the same as in scenario 1, the classification using three clusters improves (s. figure 14). Now *Setosa* is classified correctly and there are fewer misclassified samples on the border between *Versicolor* and *Virginica*. Clustering with four components leads to similar results as in scenario 1 with only a slight shift of the borders between Clusters 2 and 3 (1 and 4 in scenario 1, respectively).

Except for some instances where there were only two clusters found, the quality of the achieved classification results varies only slightly depending on the initialisation of `center0` - about the same as in scenario 1 (s. figure 17).

The optimisation process converges in the same number of iterations as in scenario 1 (s. 15 and 6), much faster than the EM algorithm. One can comprehend the cluster-updates during the optimisation process visualised in figure 16.

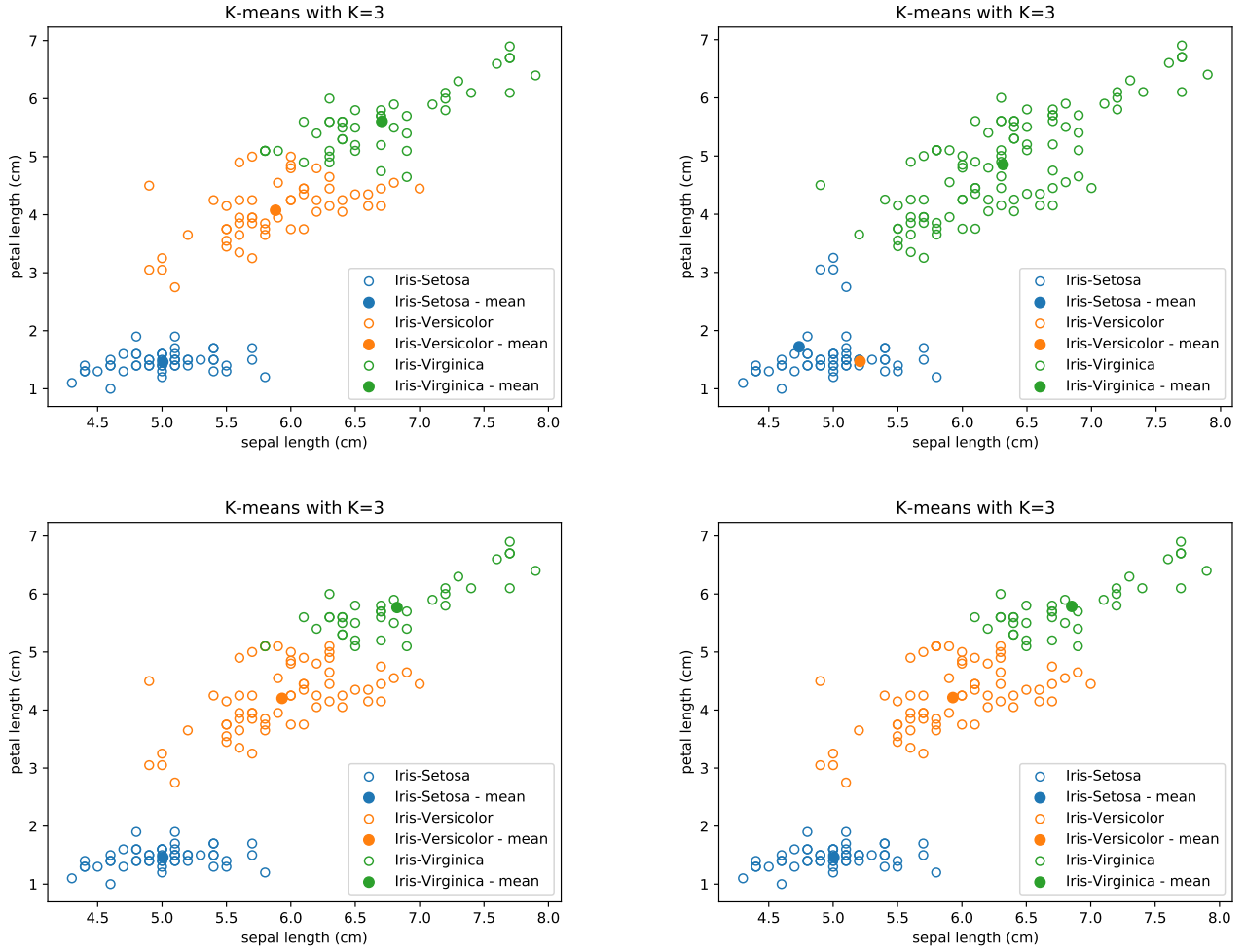


Figure 15: Results of K-means classification for several random `center0` starting samples.

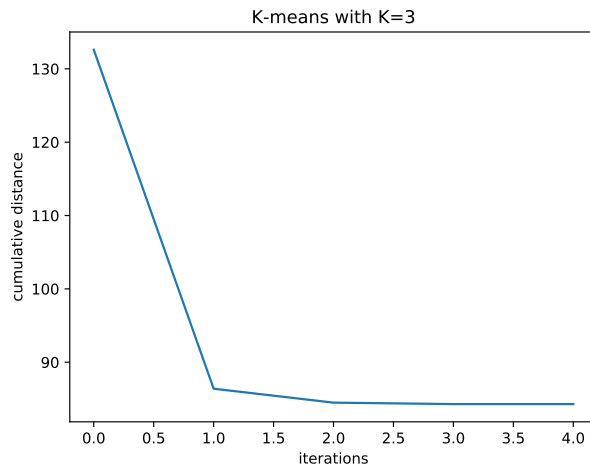


Figure 16: Cumulative distance function over iterations for $K = 3$ components.

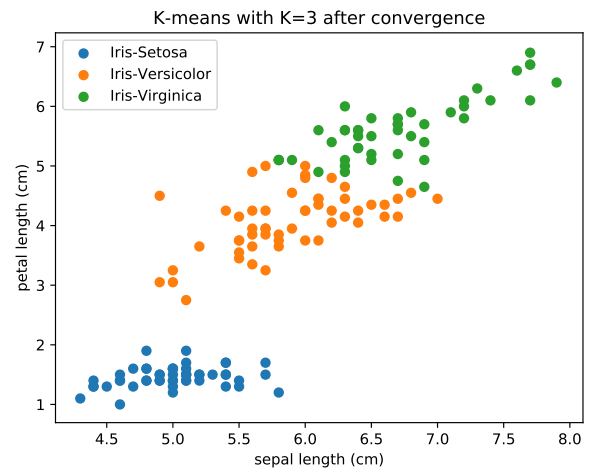
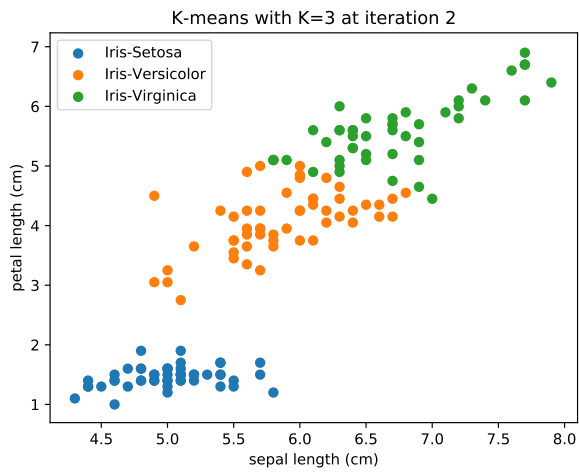
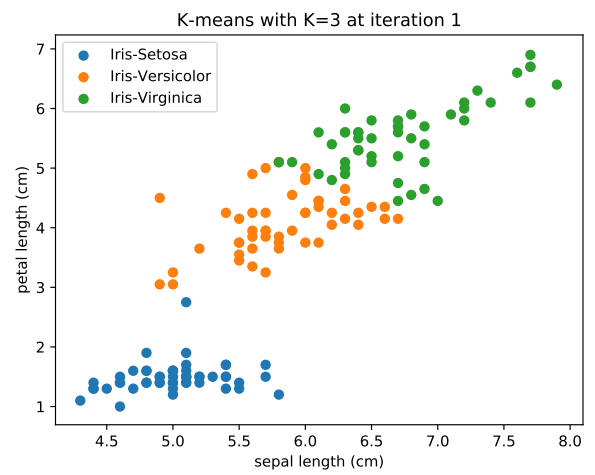
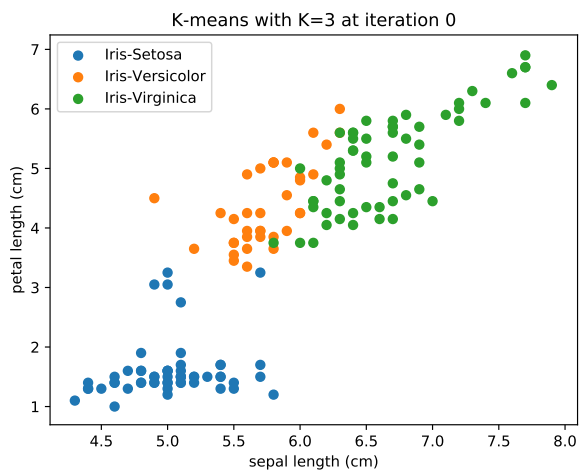


Figure 17: Results of hard-classification during optimisation.