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

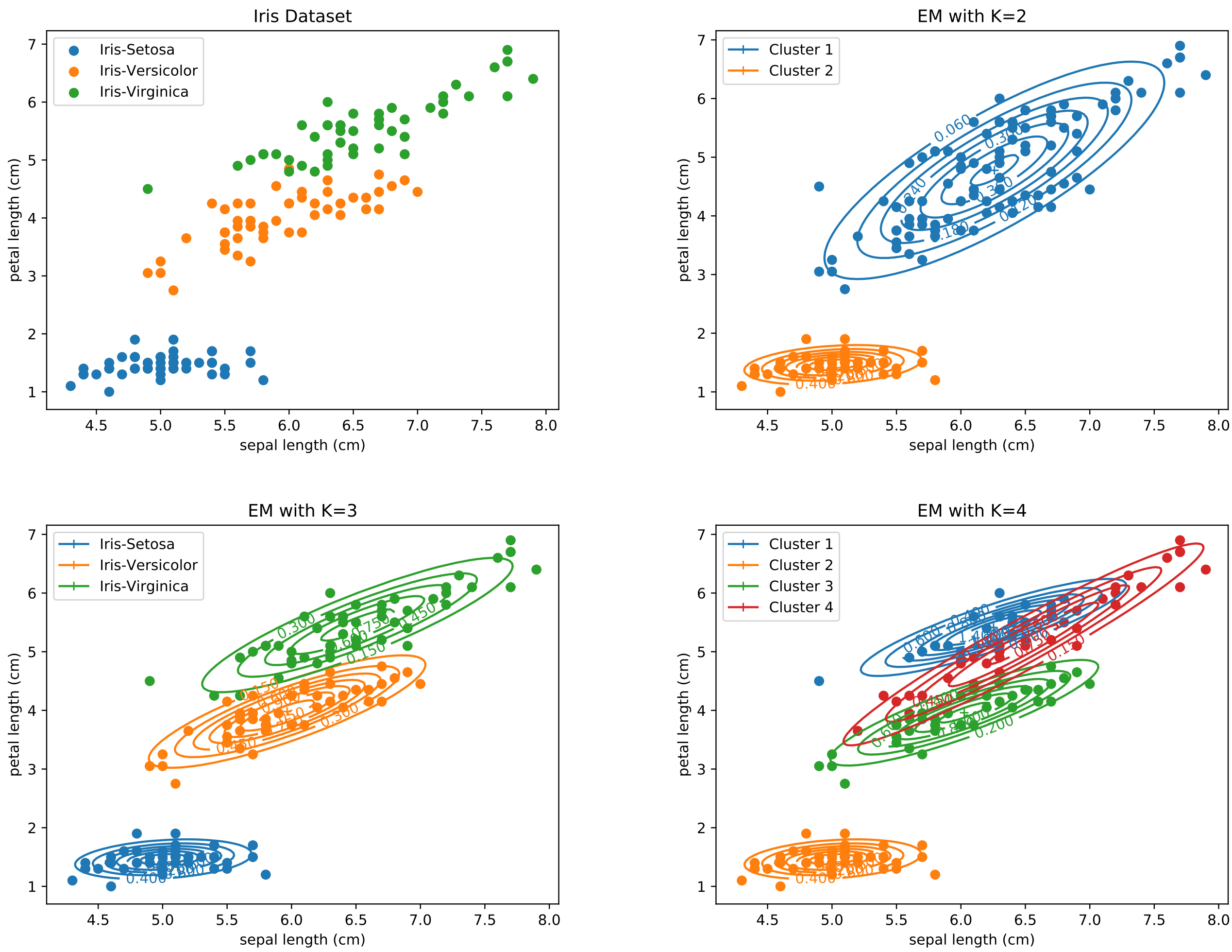


Figure 1: Results of EM classification with $K = 2 \ldots 4$ components.

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.

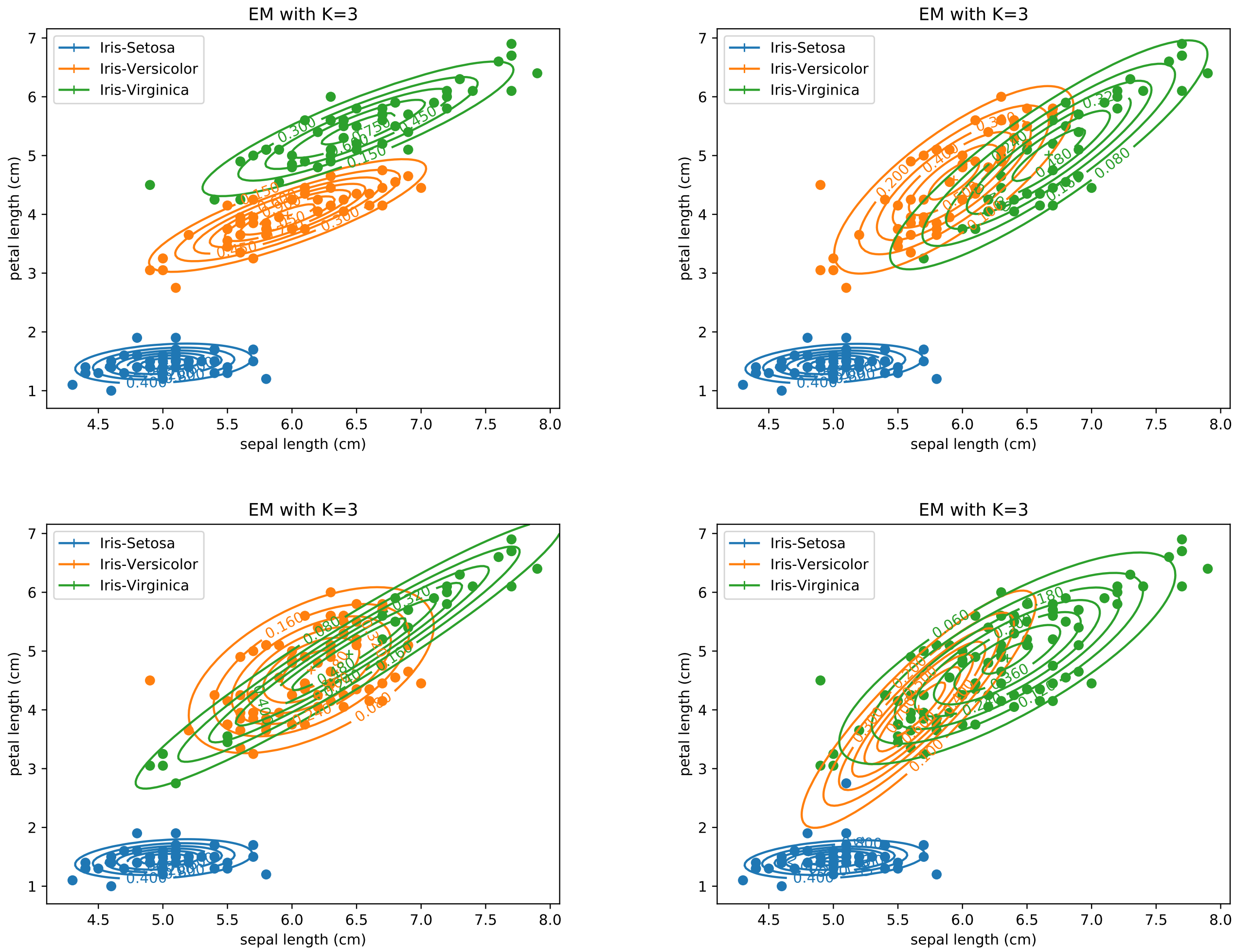


Figure 2: Results of EM classification for several random `mean0` starting samples.

The parameters $\mathbf{\Theta}_0$ 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 multivariate covariance of the dataset. Because of the non-convex form of the likelihood as a function of $\mathbf{\Theta}_0$, 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.

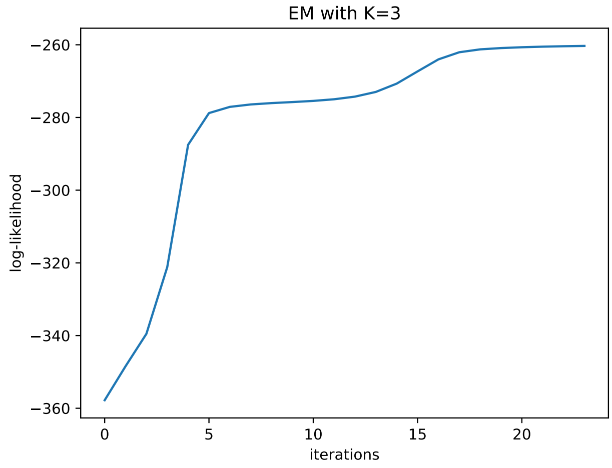


Figure 3: Log-likelihood function over iterations for $K = 3$ components.

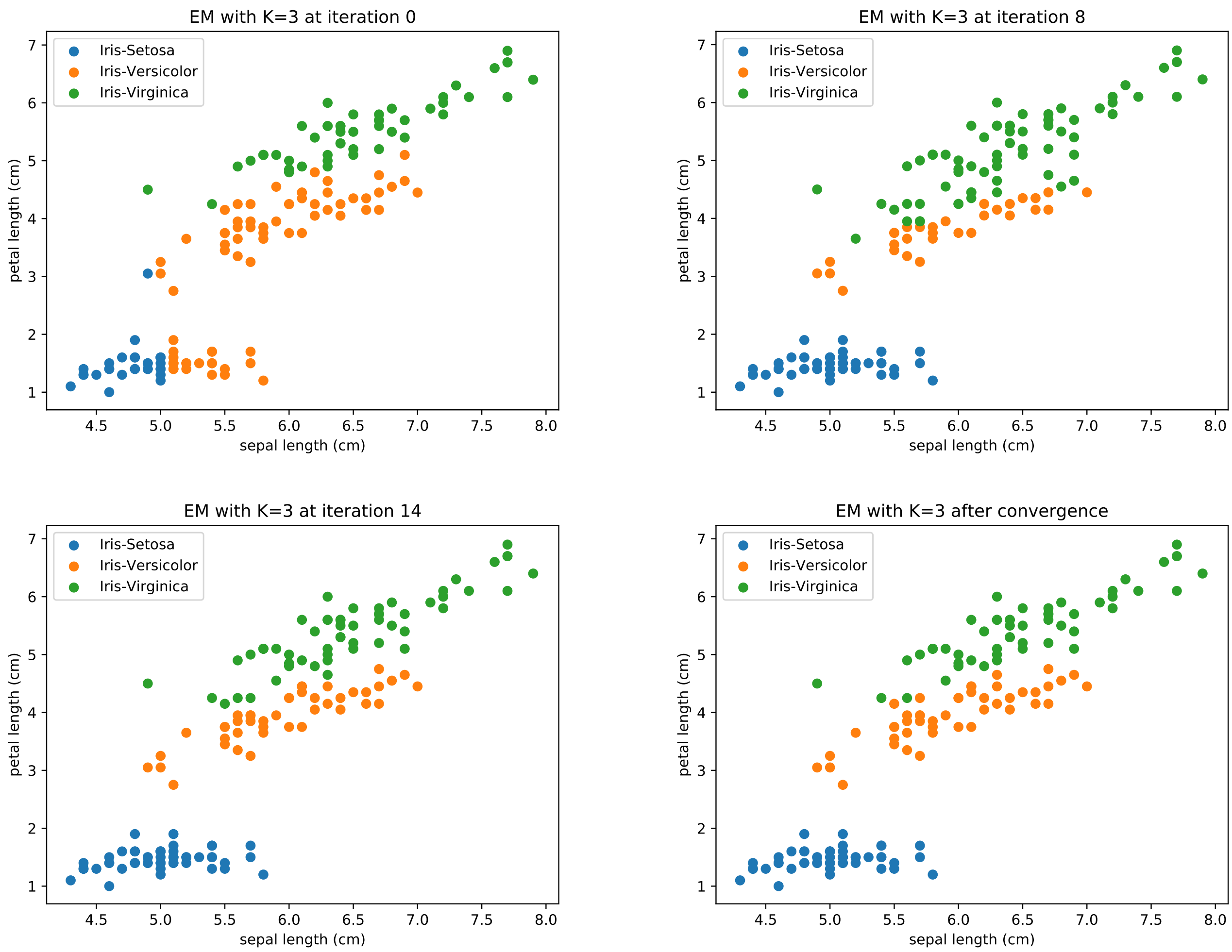


Figure 4: Results of soft-classification done in E-step.

1.2 K-means algorithm

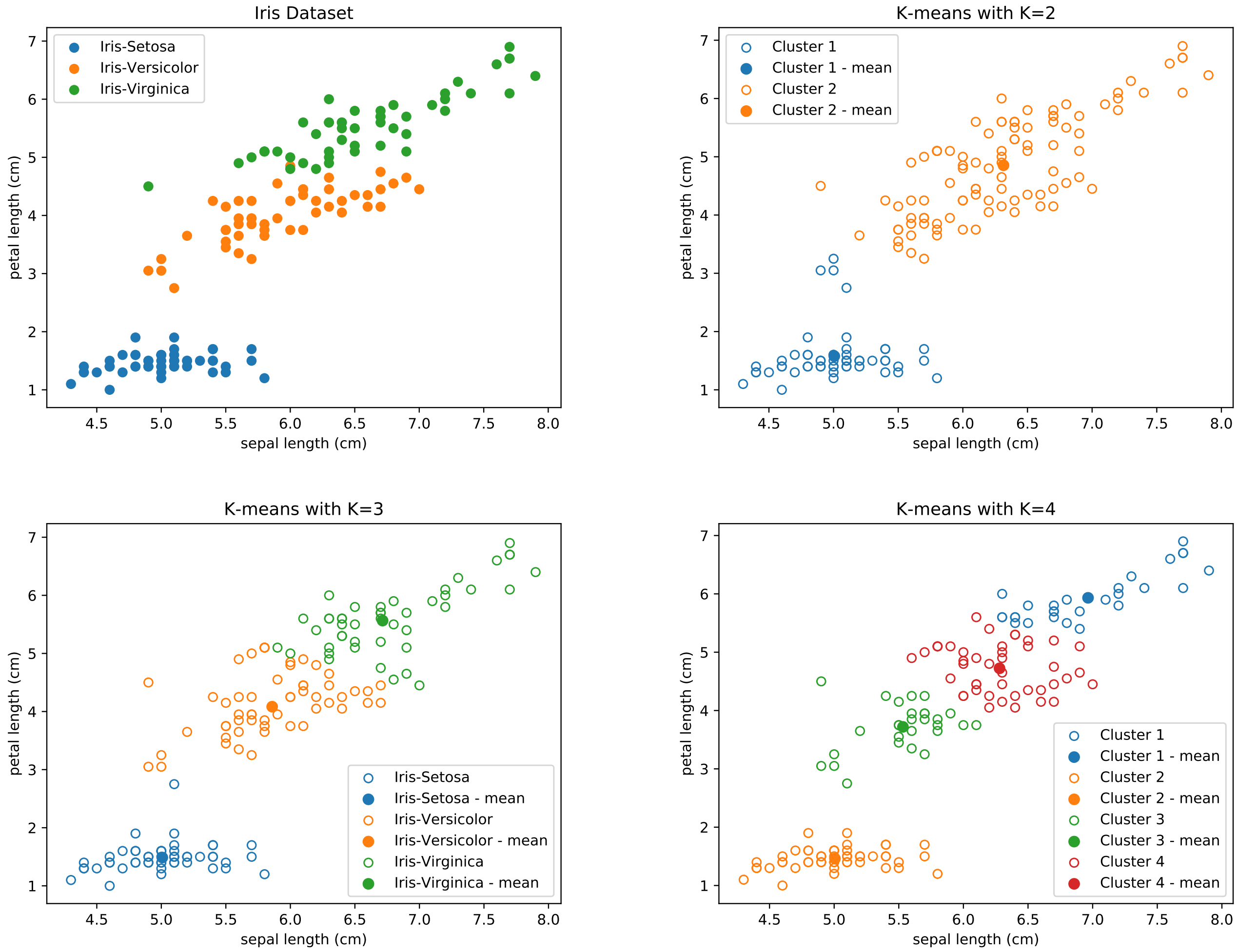


Figure 5: Results of K-means classification with $K = 2 \ldots 4$ components.

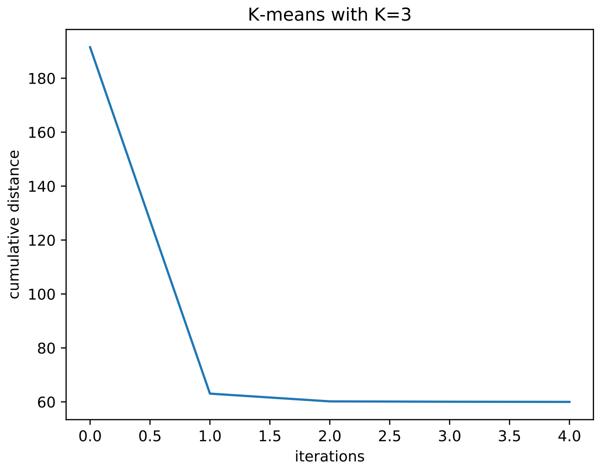


Figure 6: Cumulative distance function over iterations for $K = 3$ components.

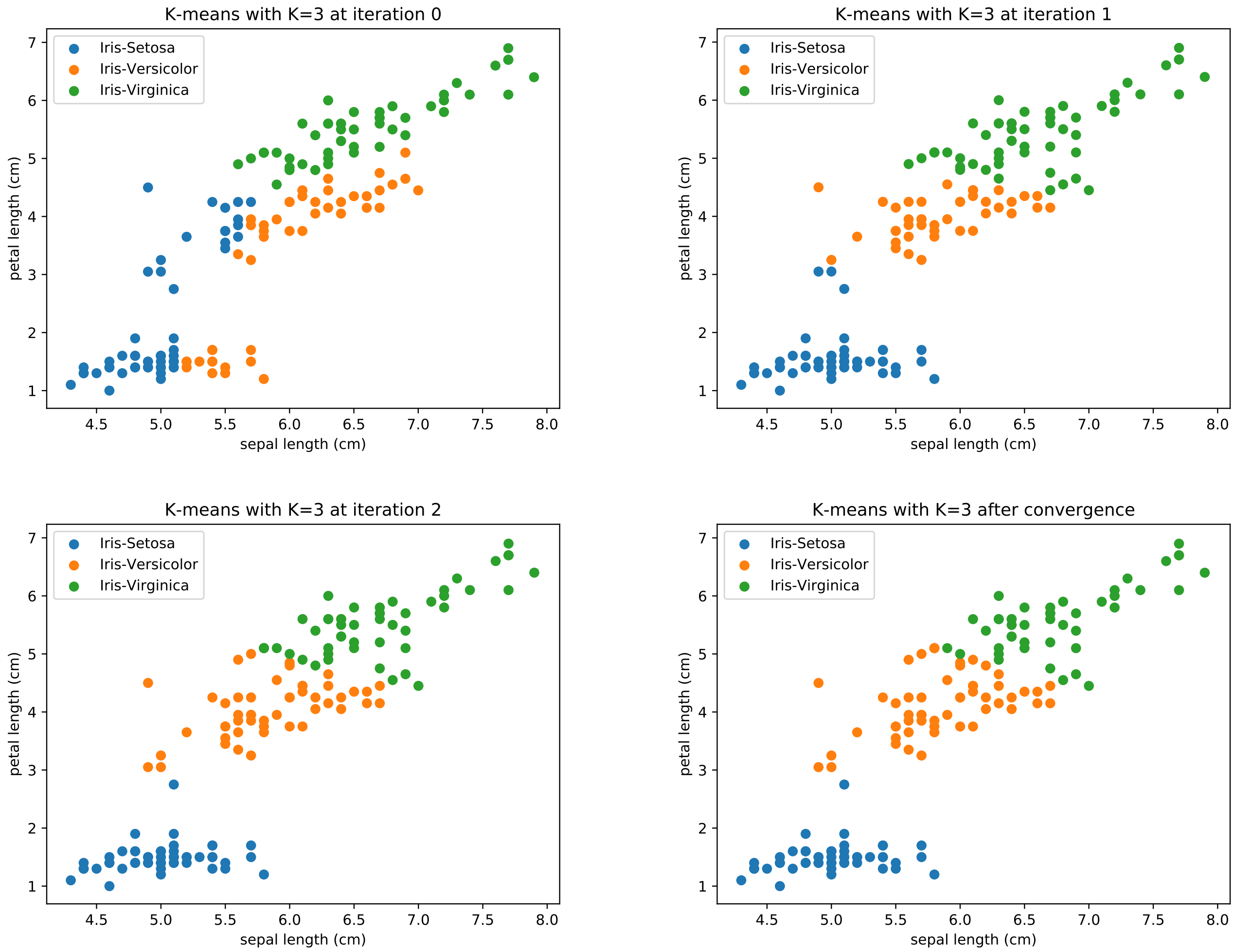


Figure 7: Results of hard-classification during optimisation.

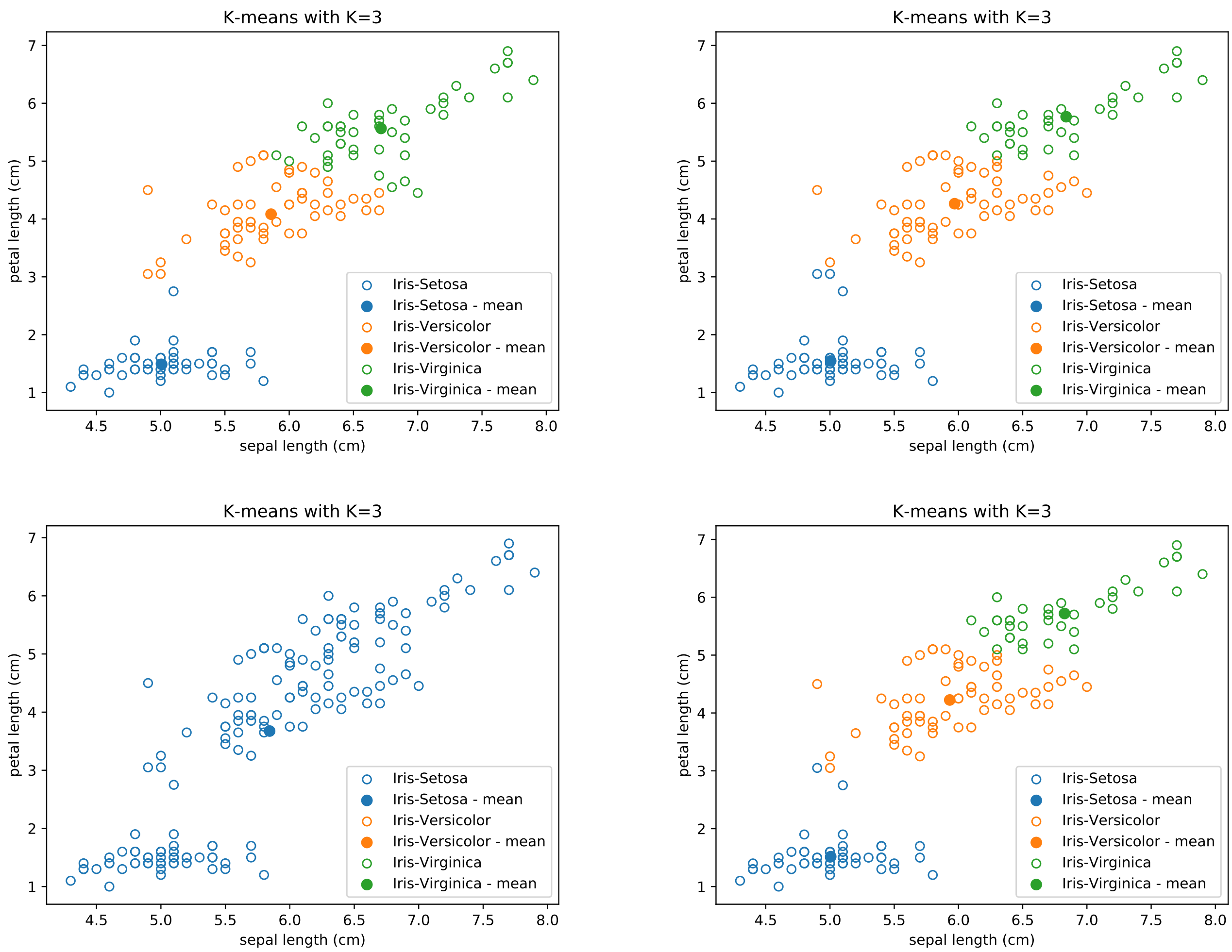


Figure 8: Results of K-means classification for several random `center0` starting samples.

2 Classification - 4 dimensional feature

2.1 EM algorithm

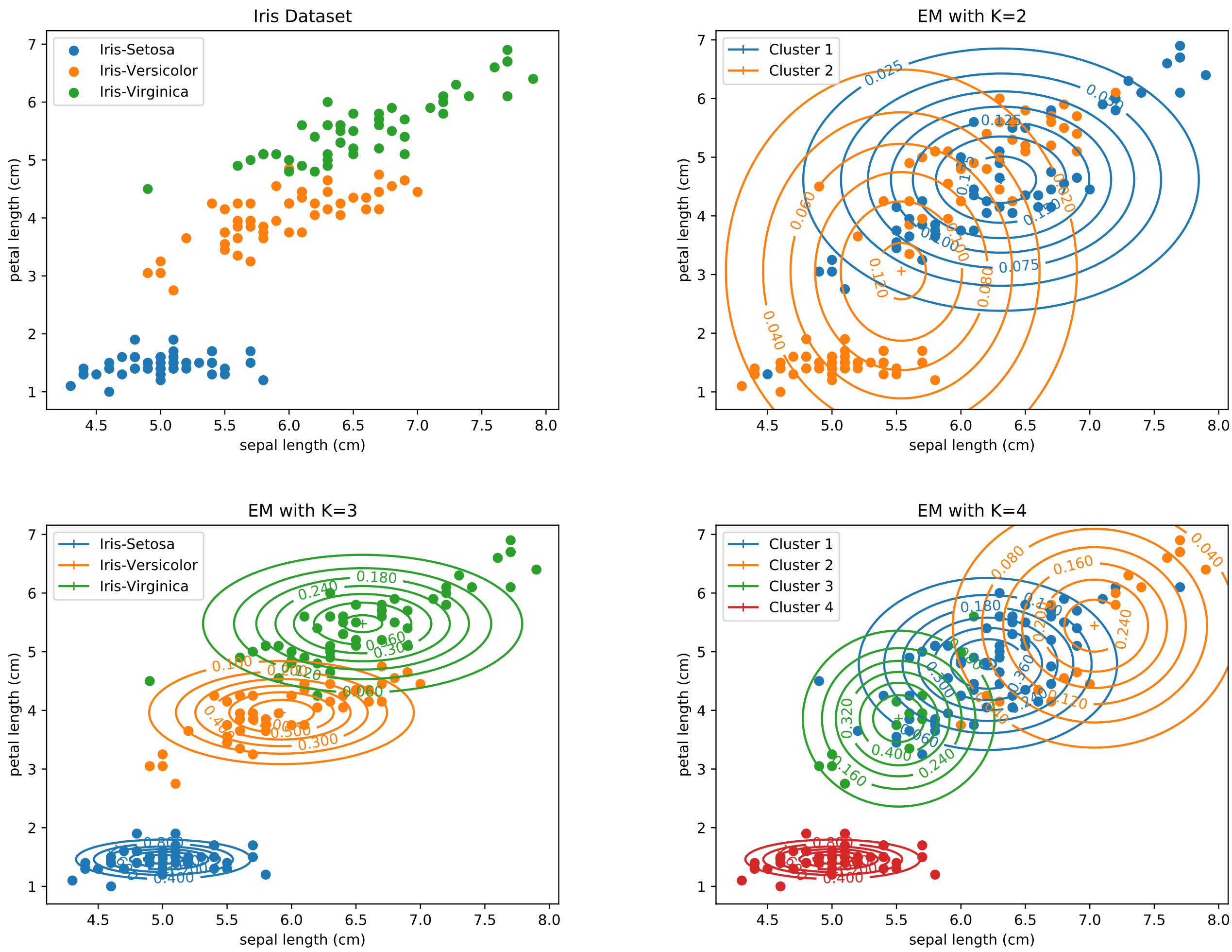


Figure 9: Results of EM classification with $K = 2 \dots 4$ components.

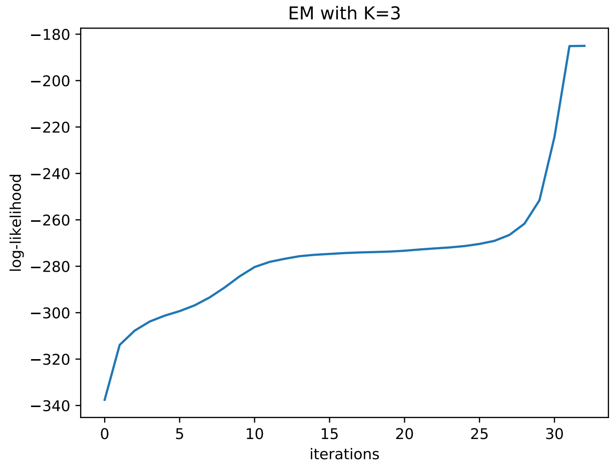


Figure 10: Log-likelihood function over iterations for $K = 3$ components.

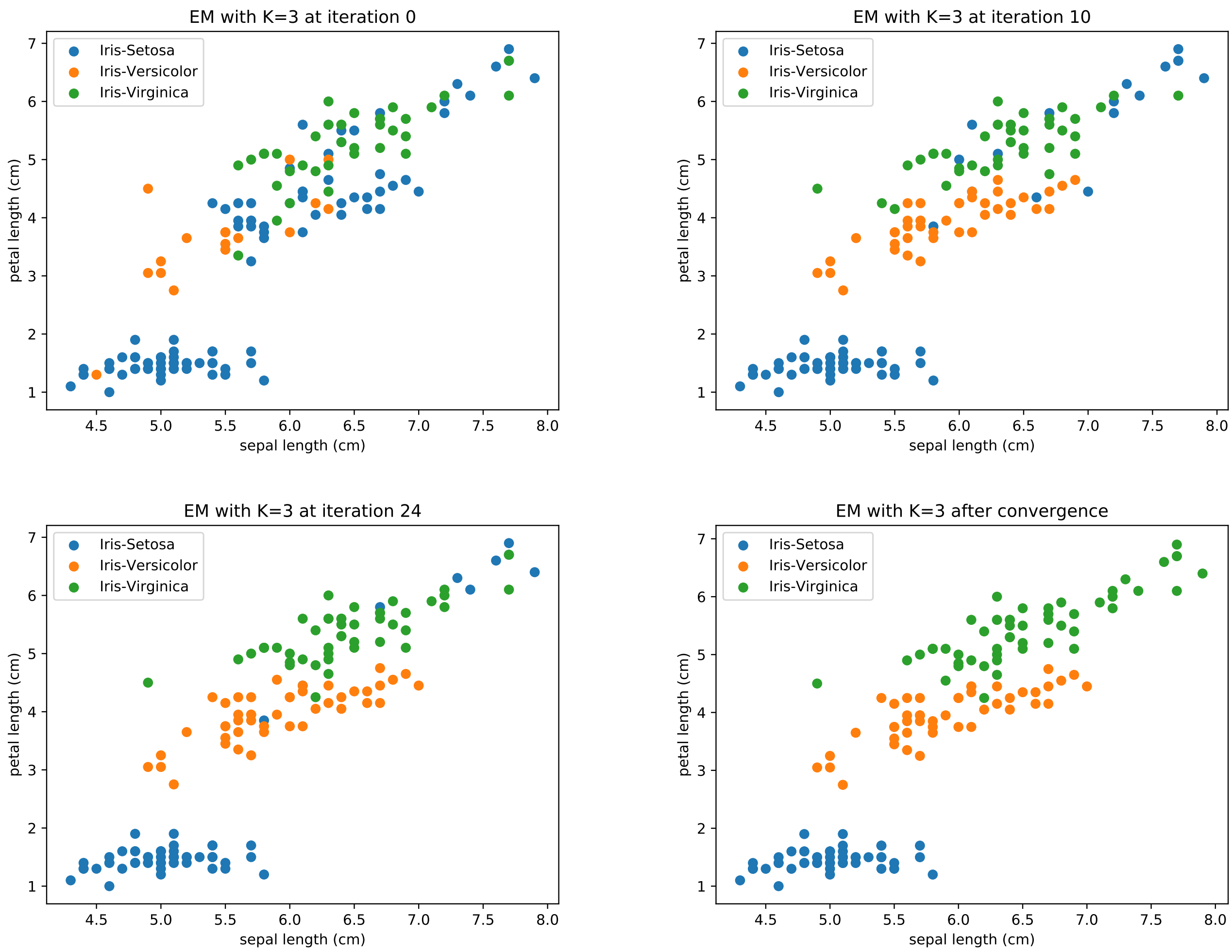


Figure 11: Results of soft-classification done in E-step.

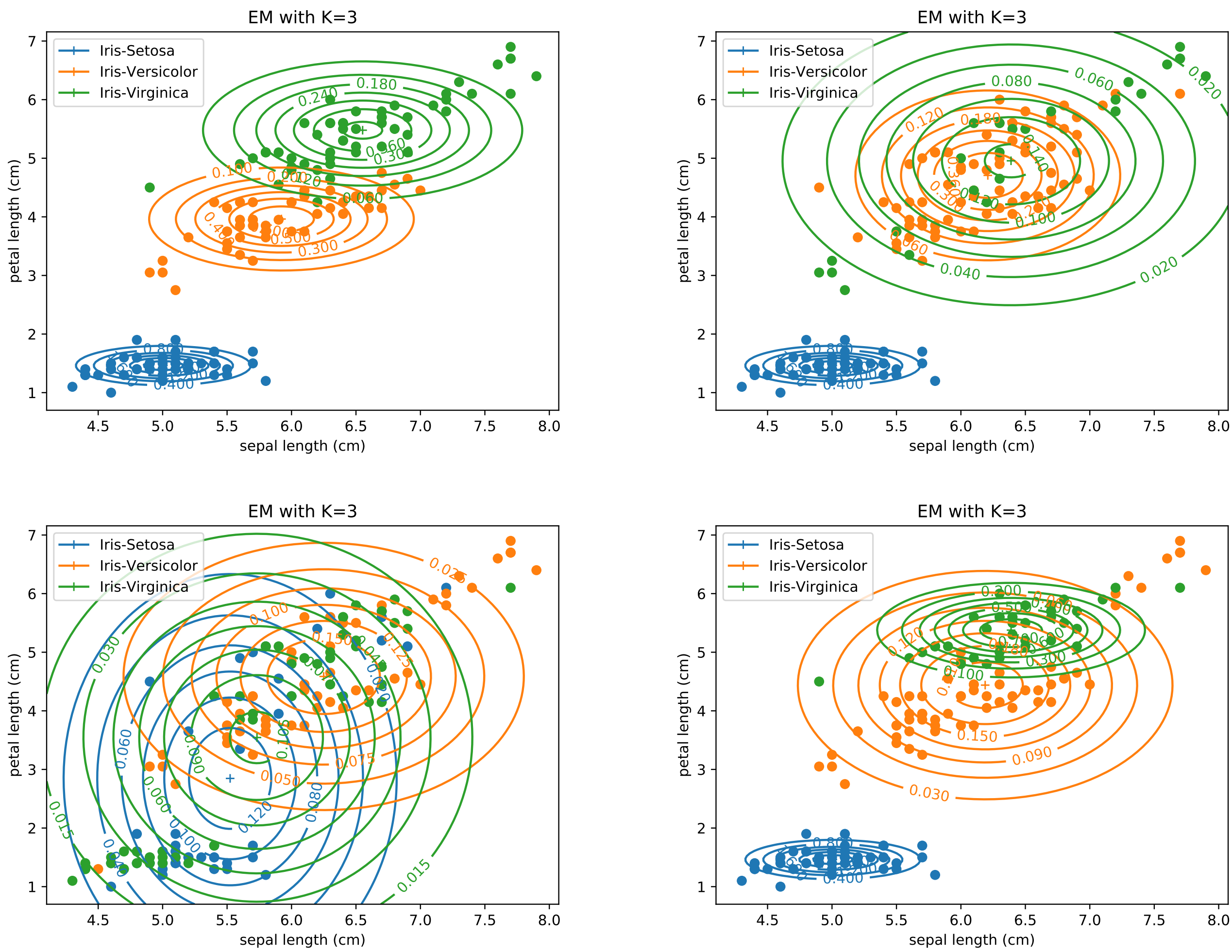
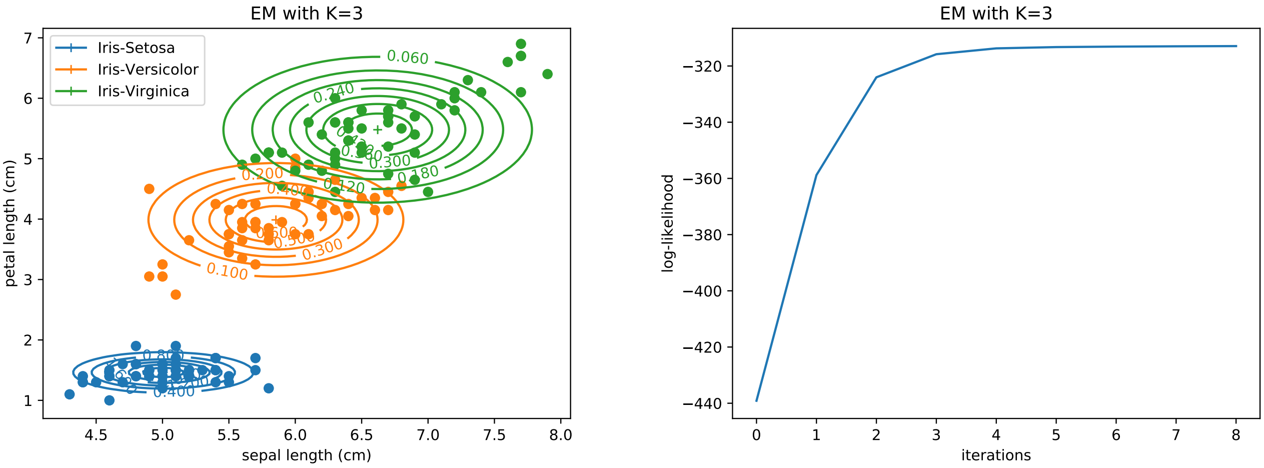


Figure 12: Results of EM classification for several random `mean0` starting samples.

2.2 EM algorithm with diagonal covariance matrices



(a) Results of classification

(b) Log-likelihood

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

2.3 K-means algorithm

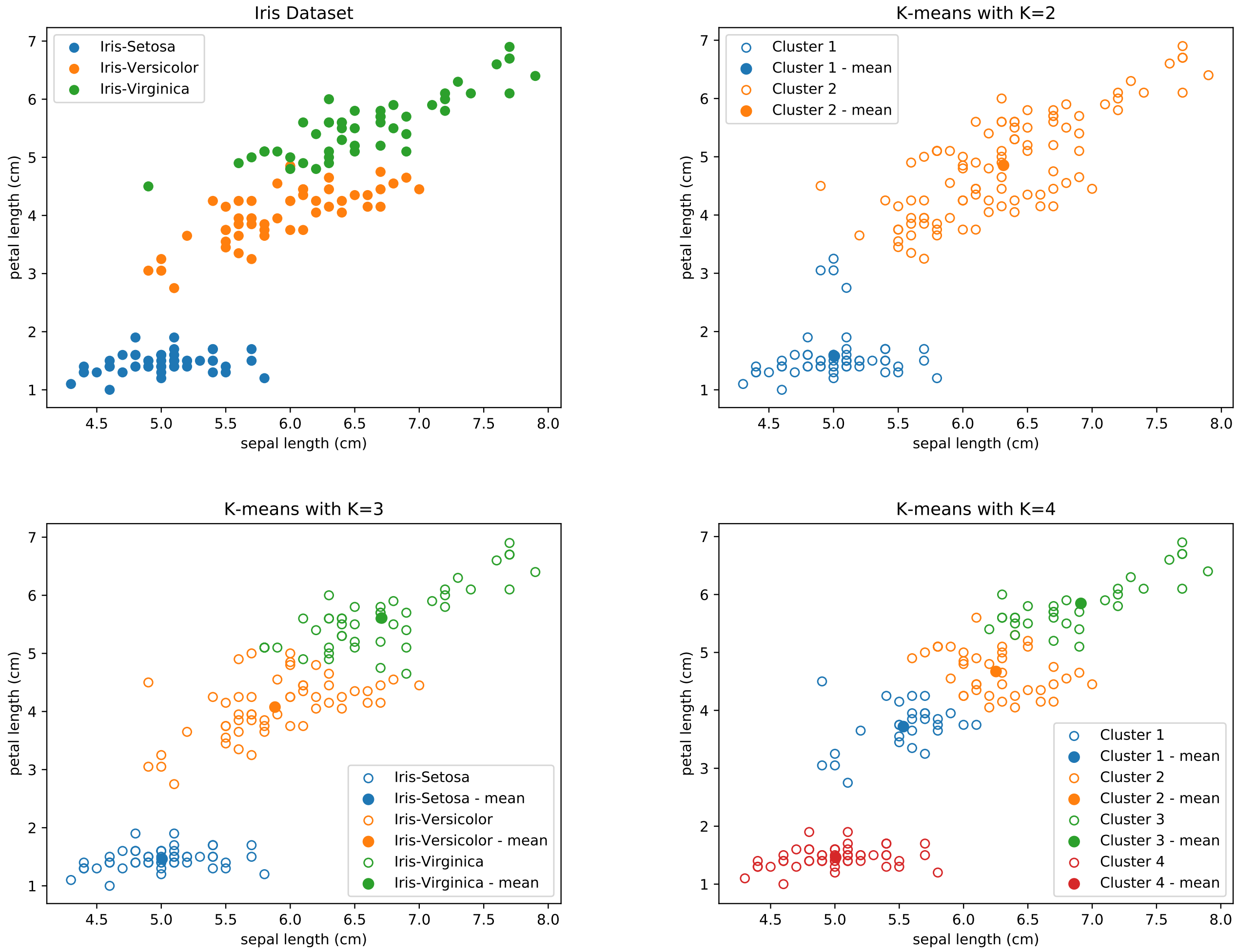


Figure 14: Results of K-means classification with $K = 2 \dots 4$ components.

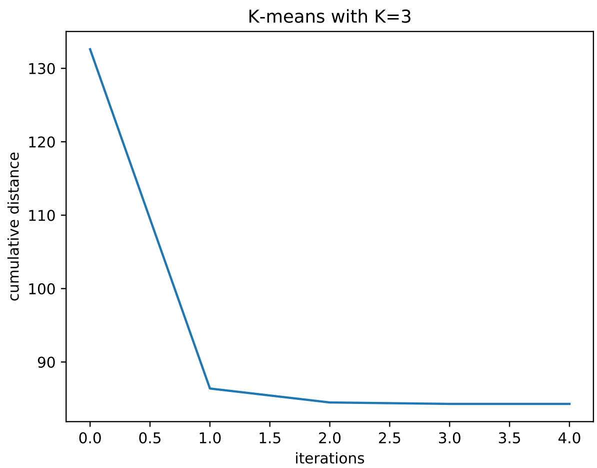


Figure 15: Cumulative distance function over iterations for $K = 3$ components.

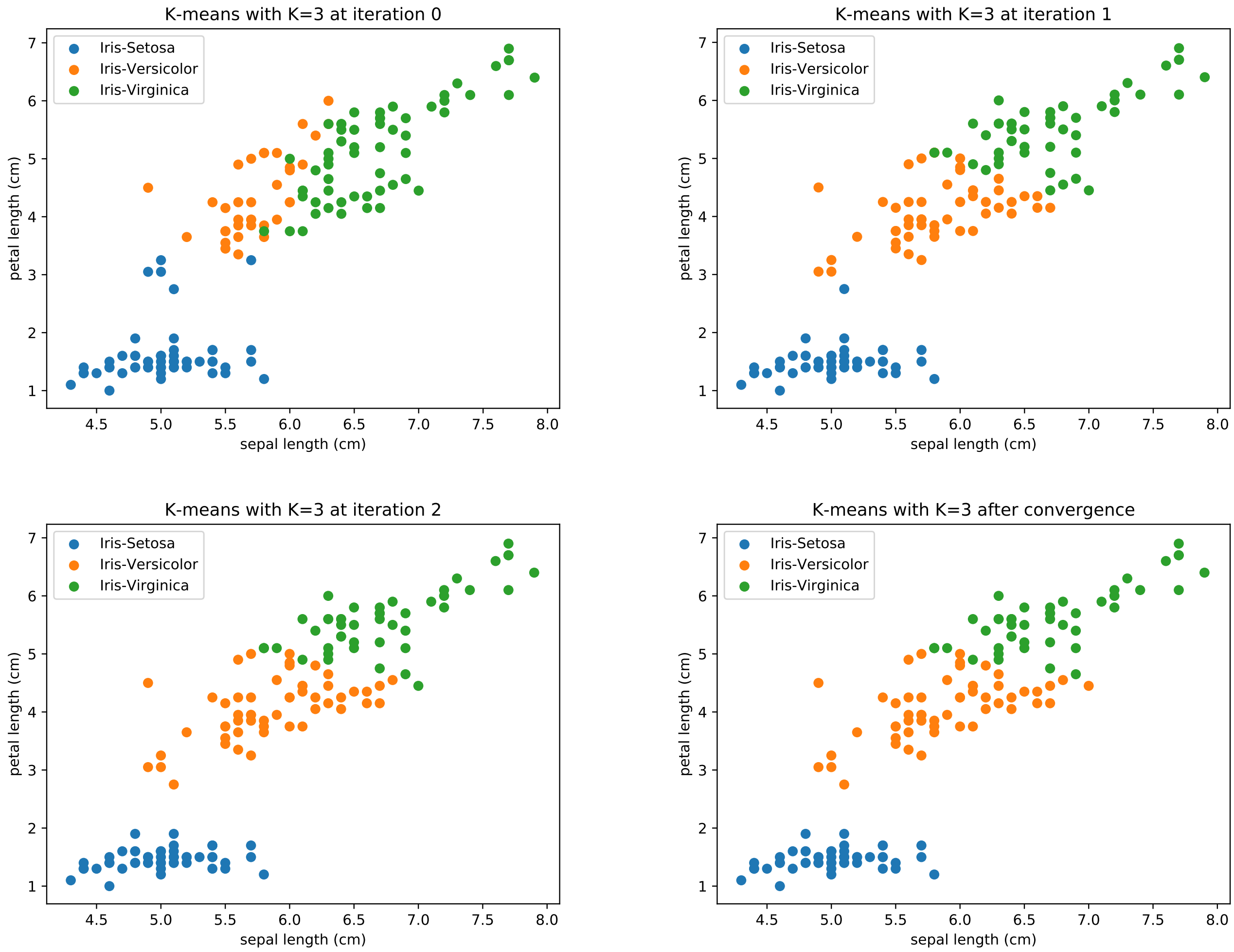


Figure 16: Results of hard-classification during optimisation.

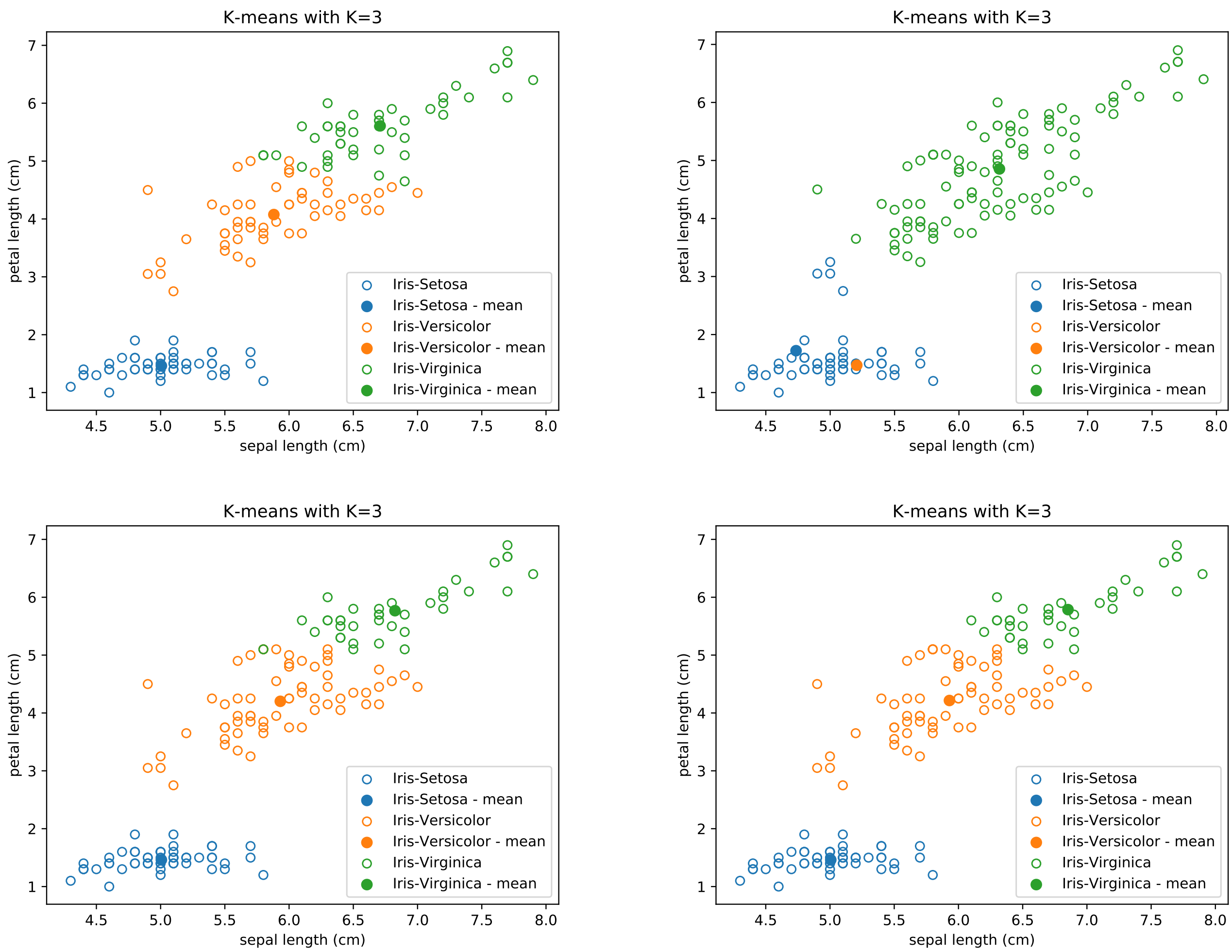


Figure 17: Results of K-means classification for several random `center0` starting samples.