Exercises for
Pattern Analysis
Marco Bögel, Sebastian Käppler
Assignment 2, 23.04.2015

General Information:

Lecture (3 SWS):	Mo 08.30 – 10.00 (H16) and Tue 08.15 – 09.45 (H16)
Exercises (1 SWS):	Tue 12.15 – 13.15 (02.134-113) and Thu 8.30 – 9.30 (E1.12)
Certificate:	Oral exam at the end of the semester
Contact:	marco.boegel@fau.de
	sebastian.kaeppler@fau.de

Hard Clustering

Exercise 1 In this exercise, we study the K-means algorithm as one the simplest methods for *hard clustering*. Given a set of n unlabled samples $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ and K clusters, the objective of K-means is to cluster the samples in S such that the sum of intraclass distances over all clusters is minimized.

(a) Write down the underlying optimization problem for K-means clustering and outline the general structure of the K-means algorithm.

(b) Let S be the following example dataset that should be clustered into $K = 2$ clusters:

$$S = \left\{ \begin{pmatrix} 0.4 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.3 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.9 \\ 0.6 \end{pmatrix}, \begin{pmatrix} 0.8 \\ 0.7 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.9 \\ 0.8 \end{pmatrix} \right\}$$

Draw the given example data in the two-dimensional feature space.

(c) Initialize your cluster centers with $\mu_1 = (0, 0)^\top$ and $\mu_2 = (0.8, 0.8)^\top$ and use the squared Euclidean distance to measure the distance between samples and a cluster center. Perform 2 iterations of the K-means algorithm and compute for each iteration:

- The updated cluster centers μ_1 and μ_2
- The clustering matrix $\boldsymbol{C}$

(d) Consider the extended sample set $S' = S \cup \{(0.1, 5)^\top\}$. What happens if you apply K-means to S' compared to the clustering determined for S? Explain an intuitive modification of the K-means algorithm to avoid this issue.

(e) Now, set $\mu_1 = (0.1, 0.1)^\top$ and $\mu_2 = (3, 2)^\top$ and perform again the clustering for S using the squared Euclidean distance measure. Explain which problem occurs and describe how it might be solved.

Exercise 2 In conventional hard clustering based on the K-means algorithm, the squared Euclidean is used to calculate distances between samples and the different cluster centers for the assignment of a sample to a cluster. This is feasible for spherical distributed samples for each cluster. However, different distance measures or kernel-based methods are required to deal with non-spherical distributed samples. In this exercise, we replace the Euclidean distance with a simple measure to deal with non-spherical data.

(a) Explain why the Euclidean distance is not appropriate for clusters with samples that are non-spherical. Draw a two-dimensional example for K-means clustering to visualize this problem.

(b) Replace the Euclidean distance by the *Mahalanobis distance* and write down the objective function for K-means clustering. Explain the benefit of the Mahalanobis distance compared to the Euclidean distance.

(c) Derive the update formulas for the cluster centers in our modified K-means algorithm using the Mahalanobis distance.

Hint: You can assume that the cluster covariance matrix is known for the estimation of the cluster centers.

(d) Explain how the covariance matrix for each cluster can be estimated for K-means clustering.

Note: For this exercise, a simple strategy is sufficient. A more advanced method to estimate the covariance matrices has been presented in:

Jianchang M.; Jain, A.K., A self-organizing network for hyperellipsoidal clustering (HEC), IEEE Transactions on Neural Networks, vol. 7, no. 1, pp.16–29, Jan 1996

Exercise 3 Matlab exercise

K-means clustering can be employed for image data compression by means of color quantization. For this purpose, we consider 24-bit color images stored in the RGB color space such that each image point is described by a red (R), green (G) and blue (B) intensity value. In order to compress RGB images, we apply K-means to represent 24-bit color values by K different clusters.

(a) Implement the K-means algorithm for a general number of clusters and dimensions of the input samples in Matlab. You can assume that we use the Euclidean distance measure for this exercise.

(b) Load the `peppers.png` test image available for Matlab and reorganize the RGB color values for each pixel as three-dimensional feature vector.

(c) Cluster the RGB color values into $K = 24$ clusters using the K-means and visualize the final clustering. You can select the cluster centers randomly out of the RGB color space.

(d) Repeat the clustering for $K = 16$ and $K = 32$ and $K = 64$ and compare the results.