Exercises for
Pattern Analysis
Sebastian Käppler
Assignment 7, 16.06.2015

General Information:

Lecture (3 SWS):	Thu 14.15 – 15.45 (H16) and Tue 12.15 – 13.45 (H16)
Exercises (1 SWS):	Mo 12.15 – 13.45 (02.134-113) and Tue 12.15 – 13.45 (E1.12)
Certificate:	Oral exam at the end of the semester
Contact:	sebastian.kaeppler@fau.de

LLE / Mean Shift Smoothing

Exercise 1 Locally linear embedding (LLE) is a manifold learning technique to perform a dimensionality reduction in two steps. First, weighting coefficients w_{ij} are determined to reconstruct each sample $\boldsymbol{x}_i \in \mathbb{R}^D$ from the neighborhood $\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)$. In the second stage, the weights w_{ij} are used to find an embedding in a d-dimensional feature space ($d < D$) according to the minimization of:

$$E(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n) = \sum_{i=1}^{n} ||\boldsymbol{x}'_i - \sum_{\boldsymbol{x}'_j \in \mathcal{N}(\boldsymbol{x}'_i)} w_{ij}\boldsymbol{x}'_j||_2^2, \quad (1)$$

where $\boldsymbol{x}'_i \in \mathbb{R}^d$ are the embedded samples. Here, we examine the second step of the LLE algorithm.

(a) Let us assume that $d = 1$ and thus x_i is a scalar. Derive an optimization problem associated with Eq. (1) that enforces unit covariance for the embedded samples.

Hint: Use a Lagrangian multiplier in your derivation.

(b) Show that the derived optimization problem can be solved by an eigenvalue decomposition.

(c) Explain how $\boldsymbol{x}_i$ can be determined for $d > 1$. Therefore, make use of the fact that the smallest eigenvalue in the derived eigenvalue decomposition is always zero.

Exercise 2 **Python exercise:** In terms of image processing, the mean shift algorithm can be employed for edge-preserving smoothing. This filtering technique can be used to denoise images. The key idea of mean shift filtering is to represent each pixel of an image by a feature vector $\boldsymbol{x}$ and to define a joint probability density function $p(\boldsymbol{x})$ for the image. Mean shift iterations are performed to find a local maximum of $p(\boldsymbol{x})$ next to a given pixel. For the sake of simplicity, we consider 2-dimensional, intensity (gray value) images. For details of mean shift for edge-preserving smoothing please refer to

Comaniciu, D. and Meer, P. *Mean shift: a robust approach toward feature space analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence (2002), Volume 24, Issue: 5, pp. 603 - 619

Figure 1: Noisy (left) and denoised image (right) using mean shift filtering.

Images can be loaded using the *opencv* package[1].

(a) Define a feature vector $\boldsymbol{x}_i$ to model the i-th pixel for a given input image. Explain how the feature vector can be extended to handle color images represented in the RGB color space.

(b) Explain how the mean shift algorithm can be employed to denoise $\boldsymbol{x}_i$. In particular, describe which parameters are required and explain the influence of the parameters to the outcome of mean shift.

(c) Implement the edge-preserving smoothing using the mean shift algorithm. Without loss of generalization, we use the Epanechnikov kernel for the mean shift iterations.

(d) Test your algorithm using synthetic image data:

- Load the example *Cameraman* image.
- Apply your mean shift algorithm to smooth the noisy image.
- The width of the Epanechnikov kernel can be selected empirically by visual inspection of the denoised image.
- Compare the input and the denoised image qualitatively.

[1] http://www.lfd.uci.edu/ gohlke/pythonlibs/