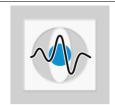
Exercises for Pattern Analysis Marco Bögel, Sebastian Käppler Assignment 7, 16.06.2015



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

Probability Density Estimation - Part II

- **Exercise 1** Adaptive binning techniques enable an automatic and adaptive selection of the bin size in a discrete histogram. This exercise considers adaptive binning that is based on a minimization of the approximation error of the discrete histogram with respect to the underlying probability density p(x).
 - (a) Write down the approximation error for an histogram with adaptive bin size as introduced in the lecture.
 - (b) How can we estimate the mean values and the bounds of the different bins based on the approximation error as underlying objective function?
 - (c) Derive a binning scheme if p(x) is a uniform distribution.
- Exercise 2 Matlab exercise The underlying probability density of the intensities in an image can be approximated by the image histogram. However, the image histogram is a discrete representation of this density. In order to obtain a continuous estimate, we employ the Parzen window approach.
 - (a) Download the the image fundus.png which shows the optic nerve head (bright, circular spot) on a human retina as well as retinal blood vessels converging at the optic nerve head. Calculate the discrete histogram of the image (Matlab hist).
 - (b) Estimate the probability density of the image intensities using the Parzen window approach. For this purpose, use N randomly selected intensity samples from the image to apply Parzen window estimation. Throughout your experiments, use a Gaussian kernel of width (standard deviation) λ for your experiments.

Implement the Parzen window estimation in Matlab and visualize the discrete histogram along with the estimated density.

- (c) Take a look at the results for
 - $\lambda = 0.25$ and $N = \{10, 100, 1000\},\$
 - $\lambda = 5$ and $N = \{10, 100, 1000\}$, and
 - $\lambda = 10 \text{ and } N = \{10, 100, 1000\}.$

- What happens if λ is chosen too small (too high) for a given N? Explain your observations.
- (d) Now, we perform an automatic and data-driven selection of an optimal kernel width λ . Write down the log-likelihood function to estimate λ in a leave-one-out cross validation scheme. Visualize your log-likelihood function for N=1000 samples and different parameters λ (e. g. $0.5 \le \lambda \le 10$).
- (e) Optimize the log-likelihood function for N=1000 samples in Matlab to find an optimal λ .

Hint: Use a gradient-based (first-order) optimization technique (Matlab: fminunc)). Therefore, derive the derivative of the log-likelihood function with respect to λ to provide a gradient for the optimization algorithm.