Advanced Digital Signal Processing: Imaging and Image Processing

TECHNISCHE UNIVERSITÄT DARMSTADT

Exercise 6: Segmentation and Classification **Due date: 21.07.2015**

Problem 1 - k-means

In this task you will implement the k-means algorithm for image segmentation. In particular, we are considering a real ultrasound image taken in the SPG lab. Your task is to segment the targets and non-targets (background).

a) Load **data.mat** containing the image $\mathbf{Y}$.

b) As features, we will only consider the intensity at each position. Thus, create the feature matrix $\mathbf{F}$ of size $N \times D$ where N denotes the number of samples and D the number of features (here $D = 1$).

c) Initialize the cluster centers C_1 and C_2 by the 0.3 and 0.6-quantile of $\mathbf{F}$. Use **quantile**.

d) For I iterations (e.g. $I = 10$), repeat the following:
   1. For each sample, compute the distances (d_1 and d_2) to C_1 and C_2.
   2. Update $r_{n,k}$: Assign the sample to the cluster with the minimal distance.
   3. Estimate the mean: Compute the mean of each cluster with respect to the samples contained in this cluster.

e) For each sample, the assigned cluster is stored in $r_{n,k}$. Reorganize $r_{n,k}$ to obtain the estimated labeling $\hat{\mathbf{X}}$.

Problem 2 - Iterated Conditional Modes

Next, you will implement a segmentation algorithm based on the Iterated Conditional Modes (ICM) performing the same task as in Problem 1. Remember that ICM solves the problem

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} P(\mathbf{X}|\mathbf{Y}) = \arg\max_{\mathbf{X}} P(\mathbf{Y}|\mathbf{X})P(\mathbf{X})$$

where $\mathbf{X}$ is the noise free and $\mathbf{Y}$ the observed image. The noise in radar, as well as ultrasound images, is often assumed to be Rayleigh-distributed. Hence, the likelihood is given by

$$P(y|\mathbf{X}) = \frac{y}{\sigma_{\mathbf{X}}^2} \exp\left(\frac{-y^2}{2\sigma_{\mathbf{X}}^2}\right)$$

For the prior, we will use a Markov Random Field as defined in Exercise 5. Note, we assume that $\mathbf{X} = (x_1, \dots, x_N)$ is a binary image where $x = 1$ indicates a target and $x = 0$ background. You can use the file **problem02.m** as a template for your code.

a) For convenience, we use simple thresholding to initialize $\mathbf{X}$. Thus, if y is larger than $T = 0.5$, we consider it as a target $(x = 1)$ and otherwise as background $(x = 0)$.

b) For the likelihood, $\sigma_{\mathbf{x=1}}$ and $\sigma_{\mathbf{x=0}}$ need to be estimated. Thus, parameter estimation needs to be performed based on the observed image $\mathbf{Y}$ considering either all pixels being classified as targets $\{y_i : x_i = 1\}$ or background $\{y_i : x_i = 0\}$. The parameter of a Rayleigh distribution can be estimated by **raylfit**.

c) Estimate the log-likelihood $\log P(y|\mathbf{X})$ using **raylpdf**.

d) Estimate the prior probabilities $P(x = 1|\mathcal{N}_x)$ and $P(x = 0|\mathcal{N}_x)$ where $\mathcal{N}_x$ denotes the first order neighborhood of x. Note that you can neglect the normalizing factor here. Working in the log domain simplifies the computation.

e) Finally, compute the (unnormalized) log probabilities $P(x = 1|y)$ and $P(x = 0|y)$ and estimate $\hat{x}$. Repeat these steps for every pixel.

f) Having estimated $\hat{\mathbf{X}}$, assign $\mathbf{X} = \hat{\mathbf{X}}$ and repeat the estimation process. This iterative procedure can be repeated for a fixed number of iterations or until convergence.

Note: The following part of the exercise is voluntarily and can be completed if interested. These problems will not be assessed.

Problem 3 - K-Nearest Neighbor Classifier (voluntarily)

In this task you will implement the K-Nearest Neighbor (K-NN) classifier. You can use the file **knn.m** as a template for your code.

a) For each feature vector f, that is to be classified, compute the Euclidean distances to all training samples.

b) Find the K training samples that are closest to f.

c) Compute which class appears most frequently in this neighborhood and assign it to f.

d) Try different K and comment on your results.

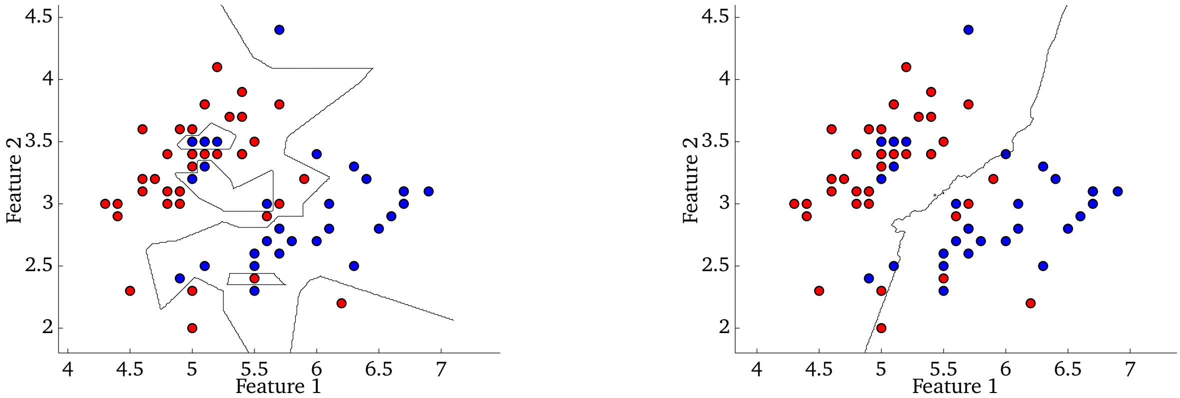

Figure 1: Decision boundary for $K = 1$ (left) and $K = 15$ (right).

Problem 4 - Classifying the *Yeast* data set (voluntarily)

Now we will use the K-NN classifier from the previous task to classify the *Yeast* data set. This data set consists of 10 classes. Each sample is described by eight features.

a) Plot the *Overall Accuracy* which is defined as the number of all correctly classified test samples divided by the number of all test samples for $1 \leq K \leq 60$.

b) What do you observe? Give a plausible explanation for this effect.

c) What is problematic about considering only the *Overall Accuracy*? Think about unbalanced data sets, i.e. different numbers of samples for each class. What happens in the extreme case? Think of another accuracy measure.

d) Implement your proposed method and comment on your results.