UNIVERSITÄT BASEL

Lecturer
Prof. Dr. Thomas Vetter
Departement Informatik
Bernoullistrasse 16
CH – 4056 Basel

Assistants
Sandro Schönborn <sandro.schoenborn@unibas.ch>
Christoph Jud <christoph.jud@unibas.ch>

Homepage
http://www.cs.unibas.ch

Pattern Recognition (CS254) - Sheet 3 [10 Points]

Preliminary Discussion	25.10.2012
Deadline	01.11.2012

Density Estimation using Gaussian Mixture Models

In this series you will improve the skin detector with a better model. Instead of a Gaussian for *skin* and one for *non-skin* you will use Gaussian Mixture Models.

Exercise 1 - Training a Gaussian Mixture Model [7 Points]

Write a Matlab function that implements the EM-Algorithm for Gaussian Mixture Models using the update formulas given in the lecture (slides "EM for Gaussian Mixtures", yellow box). Your function should have the following interface:

```
function gmm = gmmEM(data, K, iter)
%GMMEM - EM-algorithm for Gaussian Mixture Models
%  Usage: gmm = gmmEM(data, K, iter)
%
%  Parameters: data - Training inputs,           #(dims) x #(samples)
%                 K - Number of GMM components,  integer (>=1)
%              iter - Number of iterations,      integer (>=0)
%
%  Outputs:     gmm - Array of structures holding the GMM parameters
%                     Use gmm(i).mean, gmm(i).covm, gmm(i).p
%
```

Your implementation must be able to deal with arbitrary dimensional data as well as any (positive) number of mixture components.

In order to obtain a first guess of the GMM parameters you should initially assign each data sample to one of the K mixture components using the kmeans algorithm. Matlab already provides the function `kmeans`[1]. Using the initial assignments you can then compute the estimates of $\{\pi_i, \mu_i, \Sigma_i\}$. Iterate then between estimating the responsibilities of each component for the data points and the parameters of the Gaussian mixture as shown in the EM-algorithm.

Visualize the progress of the EM-Algorithm using the function `gmmDraw` (provided in `data.zip`). After each iteration plot the data points colored according to their current cluster assignment.

Test your EM algorithm on the data provided in the file `gmmdata.mat`, use 3 components.

[1] octave users use the file `myKmeans.m`, be careful regarding the dimensions

Exercise 2 - Skin Detection using GMMs [3 Points]

Train a GMM for each dataset (`skin` and `non-skin`, see exercise series 2), using two Gaussians, respectively. Based on the two Mixture Models, classify the pixels in the image `image.png` according to the rule:

$$\frac{P(\text{color}|\text{skin})}{P(\text{color}|\neg\text{skin})} > 1.$$

Compare your result with the ground truth stored in the image `mask.png` and report the obtained **true positive** and **false positive** rates.