10907 Pattern Recognition

Lecturers
Prof. Dr. Thomas Vetter ⟨thomas.vetter@unibas.ch⟩
Prof. Dr. Ivan Dokmanić ⟨ivan.dokmanic@unibas.ch⟩
Dr. Dennis Madsen ⟨dennis.madsen@unibas.ch⟩
Dana Rahbani ⟨dana.rahbani@unibas.ch⟩

Tutors
Martin Ramm ⟨martin.ramm@unibas.ch⟩
Viktor Gsteiger ⟨v.gsteiger@unibas.ch⟩

Exercise 2 — Maximum likelihood and Skin Detection

Deadline **24.10** Upload .ZIP on Adam.

The exercises are done in groups of 2 students. Only upload 1 version of the exercise on Adam and specify your group partner.

Upload in .zip format containing 1 .pdf file with your answers to the questions and the code folder files (do NOT include the data folder). Create the .zip file with the `createSubmission.py` script.

In this exercise you will write a skin detector, which is able to classify pixels of an image as *skin* or *non-skin*. For the following experiments we provide two data files `skin.mat` and `nonskin.mat` containing RGB colour data (format: 3×#samples). This data represents pixel values from several photographs which were manually labelled as belonging to either *skin* or *non-skin* regions.

1 Maximum Likelihood - Skin detection

Estimate a Multivariate Gaussian distribution for each dataset (skin + non-skin). Use the maximum likelihood estimators known from the lecture. The *main* function for this exercise is found in the file `ex2-ML_1_Skin.py`. In this exercise you are allowed to use all the power of `numpy` to compute the MVND.

1.1 Detection

(a) **Multivariate Normal Distrbution** Implement the *mean*, *covariance* and *pdf* function of the MVND class. The general structure of the class is found in the file `myMVND.py`.

(b) **Prior Probabilities** Estimate the prior probabilities for each classification-class (skin & non-skin) in `imagePrior.py`. The prior probabilities are estimated from the image `image.png`. The skin label for the image is provided in `mask.png`.

(c) **Log Likelihood** Implement the `log_likelihood` function to compute the log likelihood of each datapoint in an array. Skelleton code given in `myMVND.py`. *Hint: The function takes as input a list of Gaussian distributions (MVND). In this exercise, only one list item will be given to the function. In Section 2, when you implement the GMM, you need to provide multiple gaussians items.*

(d) **Likelihood Classifier** Implement the remaining of the Bayes classifier function to distinguish between skin and non-skin. Provide the output as a binary image. In the file `classifyHelper.py` the *classify* function needs to be completed. You need to compute the false positive/negative and total error for the skin classification problem, both with and without the use of the skin-prior.

2 Gaussian Mixture Models

The task is now to extend the skin/non-skin classification with a Gaussian Mixture Model instead of modelling the distributions with single Gaussian distributions. The *main* functions for this exercise is found in the files `ex2-ML_2_GMM_toy.py` and `ex2-ML_2_GMM_skin.py`.

2.1 EM-Algorithm

The function `gmm_em()` in `myGMM.py` needs to be implemented. The EM-Algorithm for Gaussian Mixture Models can be found in the lecture slides *"Density_Estimation_2_GMM"*.

You can visualize the progress of the EM-Algorithm by using the function `gmm_draw` (provided in the `myGMM.py` file). After each iteration plot, the data points are coloured according to their current cluster assignment for a visual debugging.

Develop and test your EM algorithm using the two-dimensional toy data provided in the file `gmmdata.mat`, work with K=3 clusters (see function `gmmToyExample()`). To initialize, the algorithm uses a random cluster assignment for each data point. Run the algorithm long enough to converge (check graphically).

Make sure that your `gmm_em()` function works according to specifications:

- K - Number of GMM clusters, integer (>=1)
- iter - Number of iterations, integer (>=0)

2.2 Skin Detection

Train a GMM with two *K=3* components for each dataset (`skin` and `non-skin`) and use a Bayes classifier for classifying the pixels into skin and non-skin (see function `gmmSkinDetection()`).

3 Theory / coding questions[10p]

1. Provide your error percentages for the training and both test images (both with and without prior). Also insert the 3 final images titled: *Training-MVND*, *Test-portrait-MVND* and *Test-family-MVND* [1P].
2. List the prior values of the skin prior and the non-skin prior[0.5P].
3. Describe how the priors are computed[0.5P].
4. How does the introduced skin and non-skin priors influence the classification results on the test images and why does this happen [1p]?
5. Suggest an alternative approach that can be used to estimate the priors [0.5P].
6. Report the 3 obtained cluster means for the GMM toy example [0.5P].
7. Plot the first and last figures from the GMM steps of the toy example [0.5P].
8. The EM algorithm is used to solve the GMM problem. What are the hidden variables in this application [0.5P]?
9. What is the hyperparameter that you had to initialize the algorithm with, and how does its value influence the fitting results [0.5P]?
10. Does the EM algorithm always converge to a global optimum? Why/Why not [0.5P]?
11. Provide your error percentages for the training and both test images (both with and without prior). Also insert the 3 final images titled: *Training-GMM*, *Test-portrait-GMM* and *Test-family-GMM* [2P].
12. How did you use the Maximum Likelihood (ML) maximization principle in the skin classification exercise [1P]?
13. What is the difference between Maximum Likelihood (ML) and Maximum A Posteriori (MAP) parameter estimation? [1P]?