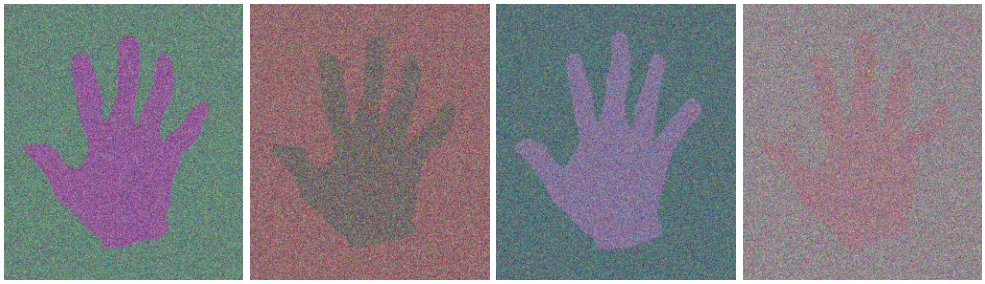
EM-ALGORITHM FOR A SIMPLE SHAPE MODEL

STATISTICAL MACHINE LEARNING (WS2020)
3. COMPUTER LAB (10+3P)

You are given a set of images, each of which was generated by a common, binary valued shape model. However, the appearance of the foreground and background segments differ in each of the images. The task is is to estimate the shape model and, simultaneously, to segment all images.



1. NOTATIONS & MODEL

Let us denote the common domain of the images by $D \subset \mathbb{Z}^2$. Each image is a mapping $\mathbf{x}\colon D \to \mathbb{R}^3$, where x_i, denotes the three dimensional colour vector in pixel $i \in D$. Binary valued segmentations are denoted by $\mathbf{s}\colon D \to \{0, 1\}$, where again, s_i denotes the segment label in pixel $i \in D$.

Assume the following simple, pixelwise independent shape model

$$p_{\mathbf{u}}(\mathbf{s}) = \prod_{i \in D} p_{u_i}(s_i) = \prod_{i \in D} \frac{e^{u_i s_i}}{1 + e^{u_i}} \tag{1}$$

parametrised (in the exponential domain) by the field $\mathbf{u}$ of real valued parameters $u_i \in \mathbb{R}$, $i \in D$. Furthermore, let us assume that the foreground and background appearance $p_\theta(\mathbf{x}_i \mid s_i = 0, 1)$ are multivariate normal distributions with $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denoting the mean and the covariance matrix. We assume that the appearance model is pixelwise independent (conditioned on $\mathbf{s}$) and that all foreground pixels (resp. background pixels) share the same θ_1 (resp. θ_0).

2. LEARNING TASK

You are given a set of images $\mathcal{T}^m = \{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$. All of them were generated from the common shape model (1) with unknown parameters $\mathbf{u}$. However, each of them was generated by its own appearance model with some unknown parameters θ_0^ℓ, θ_1^ℓ, $\ell = 1, 2, \ldots, m$. The task is to estimate all unknown parameters and to segment the images. We choose the *maximum likelihood estimator* and will apply the *EM-algorithm* for it.

Deduce that the log-likelihood $L_m(\mathbf{u}, \boldsymbol{\theta})$ of the training data $\mathcal{T}^m$ decomposes into a sum over images and pixels

$$L_m(\mathbf{u}, \boldsymbol{\theta}) = \frac{1}{m} \sum_{\ell=1}^{m} \sum_{i \in D} \log \sum_{s_i \in \{0,1\}} p_{u_i, \theta^\ell}(x_i^\ell, s_i) \tag{2}$$

By using the bound

$$\log \sum_{s_i \in \{0,1\}} p_{u_i, \theta^\ell}(x_i^\ell, s_i) \geqslant \sum_{s_i \in \{0,1\}} \alpha_\ell(s_i) \log p_{u_i, \theta^\ell}(x_i^\ell, s_i) - \sum_{s_i \in \{0,1\}} \alpha_\ell(s_i) \log \alpha_\ell(s_i), \tag{3}$$

we get the objective function for the EM algorithm as

$$\frac{1}{m} \sum_{\ell=1}^{m} \sum_{i \in D} \sum_{s_i \in \{0,1\}} \Big[\alpha_\ell(s_i) \log p_{u_i, \theta^\ell}(x_i^\ell, s_i) - \alpha_\ell(s_i) \log \alpha_\ell(s_i) \Big] \to \max_{\mathbf{u}, \boldsymbol{\theta}, \boldsymbol{\alpha}}. \tag{4}$$

The EM-algorithm solves this task by block-wise coordinate ascent, i.e. iterating maximisation over the auxiliary variables $\boldsymbol{\alpha}$ given the current estimate of the parameters $\mathbf{u}, \boldsymbol{\theta}$ (E-step) and vice versa (M-step).

E-step: Deduce that the E-step reads as simple as

$$\alpha_\ell(s_i) = p_{u_i, \theta^\ell}(s_i \mid x_i^\ell), \tag{5}$$

where $\mathbf{u}, \boldsymbol{\theta}$ denote the current estimate of the parameters.

M-step: Deduce that the maximisations w.r.t. the parameters $\mathbf{u}, \boldsymbol{\theta}$, given the current values of $\boldsymbol{\alpha}$ decomposes into two independent optimisation tasks.

(a) Show that the maximisation w.r.t. $\mathbf{u}$ further decomposes into independent tasks for each pixel $i \in D$

$$\frac{1}{m} \sum_{\ell=1}^{m} \alpha_\ell(s_i = 1)\, u_i - \log(1 + e^{u_i}) \to \max_{u_i} \tag{6}$$

Show that the function is concave and has a unique global maximum.

(b) Show that the maximisation w.r.t. the parameters $\boldsymbol{\theta}$ decomposes into independent tasks for each image and the foreground resp. background appearance parameters

$$\sum_{i \in D} \alpha_\ell(s_i = 1) \log p_{\theta_1^\ell}(\mathbf{x}_i^\ell \mid s_i = 1) \to \max_{\theta_1^\ell}$$
$$\sum_{i \in D} \alpha_\ell(s_i = 0) \log p_{\theta_0^\ell}(\mathbf{x}_i^\ell \mid s_i = 0) \to \max_{\theta_0^\ell}.$$

This are simple estimation tasks for the mean and covariance of a multivariate normal distribution (the α-s can be seen as multiplicities of data points).

3. ASSIGNMENTS

Data: The data can be downloaded from this link
`http://cmp.felk.cvut.cz/cmp/courses/SSU/shape_em/shape_em.tgz`
The (compressed) tar file contains

(1) `hand_nn.png` the images
(2) `hand_nn_seg.png` their segmentations (ground truth)

(3) `model_init.png` grey value encoded shape model initialisation (see below)

Assignment 1. (5p)
Fill in the details for the derivation of the EM-algorithm for the considered task.

Assignment 2. (3p)
Implement the following baseline approaches. Segment the images independently and without any shape model by

(1) clustering the colours of the image by k-means into two clusters,
(2) learning a mixture of two Gaussians for the colours in the image.

Initialise both approaches in the same way by assuming that the pixels in some appropriately chosen region at the image boundary are background pixels and the pixels in some circle in the image center are foreground pixels.

Compare the results against ground truth. Which of the two baseline approaches gives better results? Can you explain this?

Assignment 3. (5p)
Implement the EM-algorithm. Initialise it with the provided model (foreground probabilities $e^{u_i}/(1+e^{u_i})$ for all pixels $i \in D$ encoded as a grey valued image). Explain your choice of the stopping criterion. Report the final shape model (encoded as grey valued image) and a few segmentations. Report the average precision of segmentations (percentage of correctly segmented pixels) for the EM-algorithm by comparing with ground truth. (You obtain the segmentations from the EM-algorithm just by thresholding the final alpha's: $\alpha_\ell(s_i = 1) \gtrless 0.5$) Compare with the results obtained by the two baseline methods.

Hints:

(1) You may use the implementations of k-means and Gaussian mixture learning provided by `Scikit-learn` for the two baseline approaches in Assignment 2.
(2) When implementing the EM approach, it is recommended not to use loops over image pixels. Use instead `numpy` arrays (if you code in Python) and array operations. Similar recommendations apply for MatLab users.
(3) Recall that $0 \leqslant \alpha_\ell(s_i) \leqslant 1$ and $\alpha_\ell(s_i = 0) = 1 - \alpha_\ell(s_i = 1)$ hold. Consequently, you need only one array of alpha's (per image).
(4) Consider to use the functions `scipy.stats.multivariate_normal` as well as `numpy.cov` and `numpy.average` if you code in Python.