"Machine Learning and Computational Statistics"

6th Homework

Exercise 1:

Consider the case where the data at hand are modeled by a pdf of the form

$$p(\boldsymbol{x}) = \sum_{j=1}^{m} P_j p(\boldsymbol{x} \mid j), \quad \sum_{j=1}^{m} P_j = 1, \quad \int_{-\infty}^{+\infty} p(\boldsymbol{x} \mid j) = 1$$

where $P_j, j = 1, \ldots, m$, are the a priori probabilities of the pdfs $p(\boldsymbol{x}|j)$, which involved in the definition of $p(\boldsymbol{x})$. In the "parameter updating" part of the EM-algorithm, which allows the estimation of the parameters of $p(\boldsymbol{x}|j)$'s as well as P_j's, we need to solve the problem

$$[P_1, P_2, \ldots, P_m] = argmax_{[P_1,P_2,\ldots,P_m]} \sum_{i=1}^{N} \sum_{j=1}^{m} P(j|\boldsymbol{x_i}) \ln P_j \,, \ subject \ to \ \sum_{j=1}^{m} P_j = 1,$$

for fixed $P(j|\boldsymbol{x_i})$'s (see also slide 6 of the 6th Lecture). Prove that, independently of the form adopted for each $p(\boldsymbol{x}|j)$, the solution of the above problem is

$$P_j = \frac{1}{N} \sum_{i=1}^{N} P(j|\boldsymbol{x_i}) \,,\ j = 1, \ldots, m.$$

Hint: In this case we have an equality constraint. Work as follows:

1. Define the Lagrangian function $L(P_1, P_2, \ldots, P_m) = \sum_{i=1}^{N} \sum_{j=1}^{m} P(j|\boldsymbol{x_i}) \ln P_j + \lambda(\sum_{j=1}^{m} P_j - 1),$
2. Solve the equations $\frac{\partial L(P_1,P_2,\ldots,P_m)}{\partial P_j} = 0, j = 1, \ldots, m$, expressing each P_j in terms of λ.
3. Substitute P_j's in the constraint equation $\sum_{j=1}^{m} P_j = 1$ and solve with respect to λ.
4. Compute P_j's from the equations derived in step 2 above.

Note: In the case of equality constraints, the final solution **does not** involve the Lagrangian multipliers.

Exercise 2 (python code):

Consider the two data sets $X_1$and X_2 contained in the attached file "Dataset.mat", each one of them containing 4-dimensional data vectors, in its rows. The vectors of X_1 stem from the pdf $p_1(\boldsymbol{x})$, while those of X_2 stem from the pdf $p_2(\boldsymbol{x})$.

(a) Based onX_1, estimate the values of $p_1(\boldsymbol{x})$ at the following points: $\boldsymbol{x}_1 = (2.01, 2.99,\ 3.98,\ 5.02)$, $\boldsymbol{x}_2 = (20.78,\ -15.26,\ 19.38,\ -25.02)$, $\boldsymbol{x}_3 = (3.08, 3.88,\ 4.15,\ 6.02)$.

(b) Based onX_2, estimate the values of $p_2(\boldsymbol{x})$ at the following points: $\boldsymbol{x}_1 = (0.05, 0.15, -0.12, -0.08)$, $\boldsymbol{x}_2 = (7.18,\ 7.98,\ 9.12, 9.94)$, $\boldsymbol{x}_3 = (3.48, 4.01,\ 4.55,\ 4.96)$, $\boldsymbol{x}_4 = (20.78,\ -15.26,\ 19.38,\ -25.02)$.

Hints:

- To load the data sets use the script “HW6.ipynb”.
- Use the Sklearn.mixture.GaussianMixture class (https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html), if you are willing to use Gaussian mixtures modelling.
- It could be proved useful for the modelling of each pdf to compute the mean of each data set and then to consider the distances of the data vectors from it. However, other methods can also be applied.

Exercise 3 (python code + text):

Consider the attached image (it is an image taken by the Huble telescope).

(a) Read and depict the image. Let *A* be the *MxN* array corresponding to the image

(b) Produce 15 **noisy** versions of the image adding Gaussian noise with zero mean and variance 256. ***In reality, these versions may be different images of exactly the same part of the sky, taken at different times*** (to produce a noisy version of *A*: **(i)** create an *MxN* array, *B*, each one of its entries steming from a zero mean, unit variance normal distribution, **(ii)** multiply the array with $\sqrt{256}$., **(iii)** the noisy version *C* of *A* is produced as C=A+B).

(c) Average the noisy versions of the images and compare with the original one *A*. Report and justify your findings.

Hint: To read and show an image, use the python commands

```
A = mpimg.imread('image_name')
plt.imshow(A)
plt.show()
```

at the beginning insert the instruction:

```
import matplotlib.image as mpimg
```