Homework Set Five
ECE 175
Department of Electrical and Computer Engineering
University of California, San Diego
Nuno Vasconcelos

This HW set contains several problems. Only the problem labeled **Quiz** must be handed in and will be graded. The remaining problems are for practice. You should not submit them for grade. By submitting your Quiz solution, you agree to comply with the following.

1. The Quiz is treated as a **take-home test** and is an **INDIVIDUAL** effort. **NO collaboration is allowed**. The submitted work must be yours and must be original.
2. The work that you turn-in is your own, using the resources that are available to **<u>all</u>** students in the class.
3. You can use the help of **GENERAL** resources on programming, such as MATLAB tutorials, or related activities.
4. You are not allowed to consult or use resources provided by tutors, previous students in the class, or any websites that provide solutions or help in solving assignments and exams.
5. You will not upload your solutions or any other course materials to any web-sites or in some other way distribute them outside the class.
6. 0 points will be assigned if your work seems to violate these rules and, if recurrent, the incident(s) will be reported to the Academic Integrity Office.

1. onsider a random variable X whose distributed according to a Gaussian mixture

$$P_{\mathbf{X}}(\mathbf{x}) = \sum_{i=1}^{C} \pi_i \mathcal{G}(\mathbf{x}, \mu_i, \mathbf{\Sigma}_i).$$

Show that $\mathbf{X}$ has mean

$$\mu_x = E_{\mathbf{X}}[\mathbf{x}] = \sum_{i=1}^{C} \pi_i \mu_i$$

and covariance

$$\mathbf{\Sigma}_x = E_{\mathbf{X}}[(\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T] = \sum_{i=1}^{C} \pi_i [\mathbf{\Sigma}_i + (\mu_i - \mu_x)(\mu_i - \mu_x)^T]$$

2. The goal of this problem is to give you some "hands-on" experience on the very important case of EM as a tool for the estimation of the parameters of a mixture. Consider a mixture of two Gaussians

$$P_X(x) = \sum_{c=1}^{2} \pi_c \mathcal{G}(x, \mu_c, \mathbf{\Sigma}_c)$$

where the covariance matrices are diagonal, i.e. $\mathbf{\Sigma}_c = diag(\sigma^2_{c,1}, \sigma^2_{c,2})$, and a training sample of five points

$$\mathcal{D} = \{(-2.5, -1), (-2, 0.5), (-1, 0), (2.5, -1), (2, 1)\}.$$

a) Assume that the following hold

$$\begin{aligned}\mu_1 &= -\mu_2 \\ \Sigma_1 &= \Sigma_1 = \sigma^2 \mathbf{I}, \\ \pi_1 &= \pi_2 = \frac{1}{2}.\end{aligned}$$

Plot the log-likelihood surface $\log P_X(\mathcal{D})$ as a function of the mean parameters (entries of μ_1) for $\sigma^2 \in \{0.1, 1, 2\}$. Let the coordinate axis cover the range $([-5, 5])$. What can you say about the local maxima of the likelihood surface, and how it changes with σ^2? How does the convergence to the optimal depend on the location of the initial parameter guess?

b) Starting from the initial parameter estimate

$$\begin{aligned}\pi_1^{(0)} &= \pi_2^{(0)} = \frac{1}{2} \\ \mu_1^{(0)} &= -\mu_2^{(0)} = (-0.1, 0) \\ \Sigma_1^{(0)} &= \Sigma_1^{(0)} = \mathbf{I},\end{aligned}$$

compute all the quantities involved in the first 3 iterations of the EM algorithm. For each iteration produce

- plot 1: the posterior surface $P_{Z|\mathbf{X}}(1|\mathbf{x})$ for the first class as a function of $\mathbf{x}$,
- plot 2: the mean of each Gaussian, the contour where the Mahalanobis distance associated with it becomes 1, the points in $\mathcal{D}$, and the means of the solutions obtained the previous steps.

Let EM run until convergence, storing the mean estimates at each iteration. Produce the two plots above for the final solution. In plot 2, plot the values of the means as they progress from the initial to the final estimate.

Quiz (Computer) In this problem, we try to solve digit classification in an unsupervised manner, using K-means clustering. We assume that we only have the training images, **without labels**, and that there are 10 digit classes (as before). Hence, there are 10 clusters to learn. Each cluster has the same prior probability and gaussian distribution with identity covariances. Implement a K-means algorithm to learn the means of these clusters. A good stopping rule can be when the assignments of points to clusters do not change much in an iteration, say 0.2% (10 changes for a set of 5000 images).

1. First consider 10 random initializations suitably scaled to match the image intensity range. Run a K-means algorithm using these random initializations. Is there any problem that you encounter while running the algorithm? If yes, what is it and how can you tackle it (you need not implement the part of how to tackle it)? If not, submit the final class means as 28×28 images.
2. Now instead of choosing random initializations for the class means, choose 10 random images from the training data itself and assume it to be the initial class means. Run the K-means algorithm and display the final class means as grayscale images. Also submit the image number of the random image chosen for initialization.
3. Manually assign labels (0,1,2 ... 9) to the class means obtained in part 2. It is possible that some of the digit labels do not have a representation in the means obtained above, ignore those labels.

Also some labels will have more than one representation, choose the one you feel the best. Now using these means perform a classification using gaussian classifier of HW3 on the test data. As before compute and display the error rates per class and total error rates. (For the digits that do not have a representation, consider the error rate to be the 50%)

4. Repeat part 2 for another set of random images. Are these means different from the ones you obtained above? What can you say about the sensitivity to the initialization from the above experiment.