EECS 349 Fall 2015
Jason Brown - Homework 6

Gaussian Mixture Models

1) Making Gaussian Mixture Models

a) Gmmest function implemented as described. The keys steps were:

i) Initialization Step: Set w_k, μ_k, ${\sigma_k}^2$ for each k of K gaussians. The algorithm used the recommendations in *The Elements of Statistical Learning* chapter 8: $w_k = 1/\text{K}$, μ_k = randomly selected value from data set, ${\sigma_k}^2$ = variance of entire data set.

ii) Expectation Step: Compute the responsibilities, $\gamma_{i,k}$ for i in 0... N and k in 0...K. This was stored as a 2D numpy array and returned via a helper function.

iii) Maximization Step: Update w_k, μ_k, ${\sigma_k}^2$ for each k of K gaussians to increase the likelihood of achieving the data points given the K gaussians

iv) Do this until convergence or max number of iterations is achieved.

Ideally, this would be run multiple times and the result that achieves highest likelihood returned, since the EM algorithm is only guaranteed to find a *locally* optimal solution and we're starting with randomly selected μ.

b) Running the Gmmest function on training data:

i) Class 1: K = 2, Log likelihood: -3459
$w_0 = 0.60$, $\mu_0 = 9.77$, ${\sigma_0}^2 = 21.92$
$w_1 = 0.40$, $\mu_1 = 29.58$, ${\sigma_1}^2 = 9.78$

ii) Class 2: K = 3, Log likelihood: -8246
$w_0 = 0.20$, $\mu_0 = -24.82$, ${\sigma_0}^2 = 7.95$
$w_1 = 0.50$, $\mu_1 = -5.06$, ${\sigma_1}^2 = 23.32$
$w_2 = 0.30$, $\mu_2 = 49.62$, ${\sigma_2}^2 = 100.02$
(note, I also tried K = 2, also with randomly selected initial μ (see *Initialization Step* above) and with set μ after looking at histogram data, but K = 2 yielded a lower *best* log likelihood)

iii) Please see attached charts. The EM algorithm has likely converged since the log likelihood is increasing in very small steps as the number of iterations increases. Visually, the gaussian distributions seem to fit the data in the histogram. Since the algorithm starts with pseudorandom μ (see *Initialization Step* above), each run of the algorithm may be different. The algorithm ran on a loop of 20 runs, and reported the best results.

c) Gmmclassify function implemented as described. The keys steps were:

i) Determine the probability that x is from class 1 by comparing the weighted probabilities of the K gaussians comprising the class 1 GMM

ii) Do the same for class 2, and return whichever class has a higher probability for that x

d) Without using prior probability, the accuracy rate of the GMM classifier was 93.5%. Using prior probability increased the accuracy rate to 93.8%. Please the the attached ground-truth histogram & GMMs.

2) About the Math of GMMs

a) If the algorithm knew which one gaussian was responsible for each data point, then the following closed formula would reveal the optimal GMM. For each of k gaussians, $\mu_k = 1/n * \sum_{i=1}^{n} x_i$. Likewise, for each of the k gaussians, ${\sigma_k}^2 = 1/n * \sum_{i=1}^{n} (x_i - \mu_k)^2$ over the data points it was responsible for.

b) The EM algorithm will find a locally optimal solution, if it is unknown which gaussian is responsible for which data point. The EM algorithm will iterate, each time changing the $\gamma_{i,k}$, μ_k and ${\sigma_k}^2$ to increase the probability that the data points shown were from the k gaussians. It is best to run the EM algorithm many times, each with different starting μ and σ^2, and pick the best final values that get closest to a global optimum.