Assignment #4 - DUE: Friday, November 6

1. Problem 8.1 from textbook.
2. Problem 8.3 from textbook.
3. Consider a signal which is a superimposition of p Gaussian shape pulses centered at different time delays. In other words,

$$x[n] = \sum_{i=1}^{p} s_i[n] + w[n], \; for \; n = 0, \ldots, N-1$$

where $w[n]$ is WGN with known variance σ^2 and

$$s_i[n] = e^{-\frac{(n-d_i)^2}{2\beta^2}}.$$

The time delays d_i are the unknown parameters in the range 0 to $N-1$ and β is a known constant representing the width of each Gaussian pulse. We will use the Expectation-Maximization algorithm to find the maximum likelihood estimators for the unknown parameter vector $\theta = [d_1 \ldots d_p]^T$ by considering the independent latent variables

$$y_i[n] = s_i[n] + w_i[n]$$

where w_i is WGN with $\sigma_i^2 = \sigma^2/p$. Let $\mathbf{s}_i = [s_i[0] \ldots s_i[N-1]]^T$ and $\mathbf{y}_i = [y_i[0] \ldots y_i[N-1]]^T$. Then, following the analysis in Appendix 7C, we can show that (you don't have to show this) the log-likelihood for the latent variables can be written in the form

$$\ln p_Y(y;\theta) = h(\mathbf{y}) + \sum_{i=1}^{p} \frac{1}{\sigma_i^2} \mathbf{s}_i^T \mathbf{y}_i$$

where $h(\mathbf{y})$ is a function of $\mathbf{y}$. In other words, show equation (7C.1) holds for this case as well. Note $\mathbf{y}$ is all the $\mathbf{y}_i$ concatenated into one vector.

Again following Appendix 7C, it can be shown that (again you don't have to show this)

$$U(\theta|\theta_k) = E[h(\mathbf{y})|\mathbf{x};\theta_k] + \sum_{i=1}^{p} \mathbf{s}_i^T \hat{\mathbf{y}}_i$$

where

$$\hat{\mathbf{y}}_i = e^{-\frac{(n-d_{i,k})^2}{2\beta^2}} - \frac{1}{p}\left(x[n] - \sum_{i=1}^{p} e^{-\frac{(n-d_{i,k})^2}{2\beta^2}}\right) \quad (1)$$

which forms the Expectation-Step. The values $d_{i,k}$ are our estimates at the k'th iteration. Then, the maximization step is

$$d_{i,k+1} = \arg\max_d \mathbf{s}_i^T \hat{\mathbf{y}}_i = \arg\max_d \sum_{n=0}^{N-1} e^{-\frac{(n-d)^2}{2\beta^2}} \hat{y}_i[n]. \quad (2)$$

(a) Change the MATLAB program I have posted online for the class example to work for the case here.

- You'll have to change the data model and the latent variable estimate in the code.
- Fix the pulse width $\beta = 5$.
- You can use a grid search for implementing equation (2), but make sure it matches the range of the parameters (0 to N-1). Use a grid interval size of 0.1

Try the following example. $N = 200$ and $\sigma^2 = 0.05$. Three pulses are expected with approximate delay parameters $d_1 = 40$, $d_2 = 100$ and $d_3 = 120$. Use these values as your initial guesses for the delay parameters. Now run your code with the true unknown parameters $d_1 = 45$, $d_2 = 105$ and $d_3 = 128$. **Attach a printout of the plot generate by the code for the first iteration and the last iteration.**

(b) Now run your code with the true unknown parameters $d_1 = 45$, $d_2 = 105$ and $d_3 = 118$. Note in this case the second and third pulses are closer together. Next, run your code with the true unknown parameters $d_1 = 45$, $d_2 = 108$ and $d_3 = 112$ where the second and third pulses are even closer in time. Briefly explain what you observe the algorithm do.

(c) Run your code with the true unknown parameters $d_1 = 20$, $d_2 = 60$ and $d_3 = 110$. In this case the initial estimates $d_1 = 40$, $d_2 = 100$ and $d_3 = 120$ are quite far off the truth. Briefly explain what the algorithm does.

(d) Again for the true unknown parameters $d_1 = 20$, $d_2 = 60$ and $d_3 = 110$: use a random initialization with the command $ceil(rand(1,3) * N)$. Run the code 100 times with random initializations, pick the best solution based on whichever run returns the highest log-likelihood value. What is the highest log-likelihood value you got?

Hint: Comment out the graph plots to make your code run faster. Call the function from a for loop from the MATLAB command window and store the returned J values into an array, then pick the best J.