Solutions to Homework Set Five
ECE 271A
Electrical and Computer Engineering
University of California San Diego

Nuno Vasconcelos Fall 2018

1.

a) In this case the BDR is to say

$$\begin{aligned} g(\mathcal{X}) &= \arg\max_i \log P_{\mathbf{X}|Y}(\mathcal{X}|i) + \log\frac{1}{c} \\ &= \arg\max_i \sum_k \log P_{\mathbf{X}|Y}(\mathbf{x}_k|i) \\ &= \arg\max_i \frac{1}{n}\sum_k \log P_{\mathbf{X}|Y}(\mathbf{x}_k|i) \\ &\to \arg\max_i E_{\mathbf{X}}[\log P_{\mathbf{X}|Y}(\mathbf{x}|i)] \quad (\text{as } n\to\infty) \end{aligned}$$

where the convergence is in probability and follows from the law of large numbers. This is equivalent to

$$\begin{aligned} g(\mathcal{X}) &= \arg\min_i -E_{\mathbf{X}}[\log P_{\mathbf{X}|Y}(\mathbf{x}|i)] \\ &= \arg\min_i E_{\mathbf{X}}[\log Q_{\mathbf{X}}(\mathbf{x})] - E_{\mathbf{X}}[\log P_{\mathbf{X}|Y}(\mathbf{x}|i)] \\ &= \arg\min_i E_{\mathbf{X}}\left[\log\frac{Q_{\mathbf{X}}(\mathbf{x})}{P_{\mathbf{X}|Y}(\mathbf{x}|i)}\right] \\ &= \arg\min_i \int Q_{\mathbf{X}}(\mathbf{x})\log\frac{Q_{\mathbf{X}}(\mathbf{x})}{P_{\mathbf{X}|Y}(\mathbf{x}|i)}d\mathbf{x} \\ &= \arg\min_i \mathcal{D}[Q_{\mathbf{X}}(\mathbf{x})||P_{\mathbf{X}|Y}(\mathbf{x}|i)] \end{aligned}$$

where $Q_{\mathbf{X}}(\mathbf{x})$ is the density from which $\mathcal{X}$ was sampled. Hence, the BDR is equivalent to search for the class-conditional pdf $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ that is closest, in the Kullback-Leibler sense, to $Q_{\mathbf{X}}(\mathbf{x})$.

b) Denoting

$$\begin{aligned} Q_{\mathbf{X}}(\mathbf{x}) &= \mathcal{G}(\mathbf{x},\mu_x,\mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{\Sigma}|}}e^{-\frac{1}{2}(\mathbf{x}-\mu_x)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_x)^T} \\ P_{\mathbf{X}|Y}(\mathbf{x}|i) &= \mathcal{G}(\mathbf{x},\mu_i,\mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{\Sigma}|}}e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_i)^T} \end{aligned}$$

we have

$$\begin{aligned} \mathcal{D}[Q_{\mathbf{X}}(\mathbf{x})||P_{\mathbf{X}|Y}(\mathbf{x}|i)] &= E_{\mathbf{X}}[\log\frac{Q_{\mathbf{X}}}{P_{\mathbf{X}|Y}(\mathbf{x}|i)}] \\ &= E_{\mathbf{X}}[-\frac{1}{2}(\mathbf{x}-\mu_x)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_x)^T + \frac{1}{2}(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_i)^T] \end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2} E_{\mathbf{X}}[\mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x} - 2\mathbf{x}^T \mathbf{\Sigma}^{-1} \mu_x + \mu_x^T \mathbf{\Sigma}^{-1} \mu_x - \mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x} + 2\mathbf{x}^T \mathbf{\Sigma}^{-1} \mu_i - \mu_i^T \mathbf{\Sigma}^{-1} \mu_i \\
&= \frac{1}{2} [\mu_x^T \mathbf{\Sigma}^{-1} \mu_x - 2\mu_x^T \mathbf{\Sigma}^{-1} \mu_i + \mu_i^T \mathbf{\Sigma}^{-1} \mu_i] \\
&= \frac{1}{2} (\mu_x - \mu_i)^T \mathbf{\Sigma}^{-1} (\mu_x - \mu_i).
\end{aligned}
$$

Hence the BDR is equivalent to the search for the class whose mean has the smalest Mahalanobis distance from the mean of the process from which $\mathcal{X}$ was drawn.

2. Problem 4.3.3 in DHS

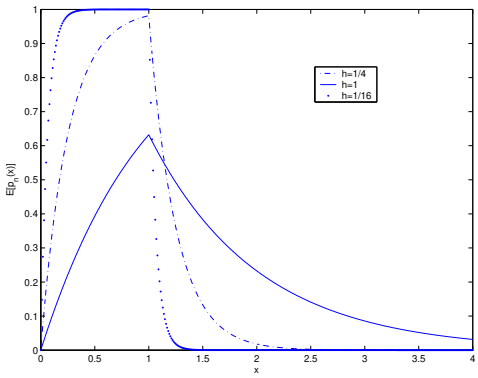
a) We know (equation 23 in DHS) that $\hat{p}_n(x)$ is the convolution of $p(x)$ with

$$\delta_n(x) = \frac{1}{h_n}\phi\left(\frac{x}{h_n}\right),$$

i.e.

$$\begin{aligned}
\hat{p}_n(x) &= p(x) * \delta_n(x) \\
&= \int p(t)\frac{1}{h_n}\phi\left(\frac{x-t}{h_n}\right)dt \\
&= \begin{cases} 0, & x < 0 \\ \int_0^x \frac{1}{ah_n} e^{\frac{t-x}{h_n}} dt, & 0 \le x < a \\ \int_0^a \frac{1}{ah_n} e^{\frac{t-x}{h_n}} dt, & x \ge a \end{cases} \\
&= \begin{cases} 0, & x < 0 \\ \int_{-\frac{x}{h_n}}^{0} \frac{1}{a} e^{u} du, & 0 \le x < a \\ \int_{-\frac{x}{h_n}}^{\frac{a-x}{h_n}} \frac{1}{a} e^{u} du, & x \ge a \end{cases} \\
&= \begin{cases} 0, & x < 0 \\ \frac{1}{a}(1 - e^{-\frac{x}{h_n}}), & 0 \le x < a \\ \frac{1}{a}(e^{\frac{a}{h_n}} - 1)e^{-\frac{x}{h_n}}, & x \ge a \end{cases}
\end{aligned}$$

b) The plot is shown in the figure below. It is interesting that $\hat{p}_n(x)$ converges to $p(x)$ reasonably quickly, given that the kernel is a very poor match to the shape of the pdf.



c) Let α be one minus the percent bias, i.e.

$$\begin{aligned}
\alpha &= 1 - \frac{p(x) - \hat{p}_n(x)}{p(x)} \\
&= \frac{\hat{p}_n(x)}{p(x)} \\
&= \begin{cases} 0, & x < 0 \\ 1 - e^{-\frac{x}{h_n}}, & 0 \le x < a \\ (e^{\frac{a}{h_n}} - 1)e^{-\frac{x}{h_n}}, & x \ge a \end{cases}
\end{aligned}$$

We want $\alpha > 0.99$ over 0.99 of $(0, a)$, which is equivalent to

$$\begin{aligned} e^{-\frac{x}{h_n}} &< 0.01, && \text{for .99 of } (0, a) \\ \frac{x}{h_n} &> \log(100), && \text{for .99 of } (0, a) \\ h_n &< \frac{x}{\log(100)}, && \text{for .99 of } (0, a) \end{aligned}$$

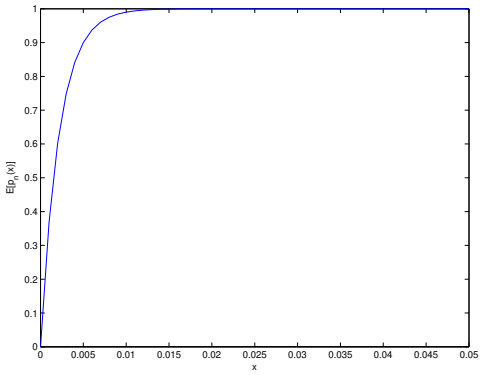
This is satisfied when

$$h_n < \frac{0.01a}{\log 100}$$

d) In this case

$$h_n < \frac{0.01}{\log 100} = 0.00217$$

and the plot is as shown below.



3.

a) This is a standard ML estimation problem. The solution is

$$\begin{aligned}
\Psi^* &= \arg\max_{\Psi} L(\Psi) \\
&= \arg\max_{\Psi} \log P_{\mathbf{X}}(\mathcal{D};\Psi) \\
&= \arg\max_{\Psi} \log \frac{n!}{x_1!x2!x_3!x_4!} + x_1 \log\left(\frac{1}{2}+\frac{1}{4}\Psi\right) + x_2\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_3\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_4\log\left(\frac{1}{4}\Psi\right) \\
&= \arg\max_{\Psi} x_1 \log\left(\frac{1}{2}+\frac{1}{4}\Psi\right) + x_2\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_3\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_4\log\left(\frac{1}{4}\Psi\right)
\end{aligned}$$

and, as usual, is obtained by setting derivatives to zero

$$\begin{aligned}
\frac{\partial L}{\partial \Psi} &= \frac{1}{4}\frac{x_1}{\left(\frac{1}{2}+\frac{1}{4}\Psi\right)} - \frac{1}{4}\frac{x_2}{\left(\frac{1}{4}-\frac{1}{4}\Psi\right)} - \frac{1}{4}\frac{x_3}{\left(\frac{1}{4}-\frac{1}{4}\Psi\right)} + \frac{1}{4}\frac{x_4}{\left(\frac{1}{4}\Psi\right)} \\
&= \frac{x_1}{(2+\Psi)} - \frac{x_2}{(1-\Psi)} - \frac{x_3}{(1-\Psi)} + \frac{x_4}{(\Psi)} \\
&= \frac{x_1\Psi(1-\Psi) - (x_2+x_3)\Psi(2+\Psi) + x_4(2+\Psi)(1-\Psi)}{\Psi(2+\Psi)(1-\Psi)} \\
&= \frac{-(x_2+x_3+x_1+x_4)\Psi^2 + (x_1-2(x_2+x_3)-x_4)\Psi + 2x_4}{\Psi(2+\Psi)(1-\Psi)} \\
&= -\frac{n\Psi^2 + [2(x_2+x_3)+x_4-x_1]\Psi - 2x_4}{\Psi(2+\Psi)(1-\Psi)} = 0
\end{aligned}$$

which leads to the solution

$$\Psi^* = \frac{(x_1 - 2(x_2+x3) - x_4) \pm \sqrt{(x_1-2(x_2+x3)-x_4)^2 + 8nx_4}}{2n}.$$

Without explicit values for $\mathcal{D}$ it is impossible to go any further. We expect that at least one of the solutions above will be admissible, i.e. within the range needed to keep all the probabilities in $P_{\mathbf{X}}(\mathbf{x};\Psi)$ between zero and one. In any case, we would need to examine the boundary points of Ψ and check if the maximum is not there.

b) We now have

$$\begin{aligned}
\Psi^* &= \arg\max_{\Psi} L_c(\Psi) \\
&= \arg\max_{\Psi} \log P_{\mathbf{X}}(\mathcal{D}_c;\Psi) \\
&= \arg\max_{\Psi} \log \frac{n!}{x_{11}!x_{12}!x2!x_3!x_4!} + x_{11}\log\left(\frac{1}{2}\right) + x_{12}\log\left(\frac{1}{4}\Psi\right) + x_2\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) \\
&\quad + x_3\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_4\log\left(\frac{1}{4}\Psi\right) \\
&= \arg\max_{\Psi} x_{12}\log\left(\frac{1}{4}\Psi\right) + x_2\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_3\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_4\log\left(\frac{1}{4}\Psi\right)
\end{aligned}$$

and setting derivatives to zero

$$\frac{\partial L_c}{\partial \Psi} = \frac{1}{4}\frac{x_{12}}{\left(\frac{1}{4}\Psi\right)} - \frac{1}{4}\frac{x_2}{\left(\frac{1}{4}-\frac{1}{4}\Psi\right)} - \frac{1}{4}\frac{x_3}{\left(\frac{1}{4}-\frac{1}{4}\Psi\right)} + \frac{1}{4}\frac{x_4}{\left(\frac{1}{4}\Psi\right)}$$

$$
\begin{aligned}
&= \frac{x_{12}}{\Psi} - \frac{x_2}{(1-\Psi)} - \frac{x_3}{(1-\Psi)} + \frac{x_4}{\Psi} \\
&= \frac{x_{12}+x_4}{\Psi} - \frac{x_2+x_3}{(1-\Psi)} \\
&= \frac{x_{12}+x_4-(x_{12}+x_4+x_2+x_3)\Psi}{\Psi(1-\Psi)}
\end{aligned}
$$

leads to the solution

$$\Psi^* = \frac{x_{12}+x_4}{x_{12}+x_4+x_2+x_3}.$$

This solution is significantly simpler, as is usually the case when the complete data is known. In fact, for most problems, it is not possible to obtain a closed-form solution unless we have the complete data. Because there is a closed form solution for ML estimation from incomplete data, EM is not strictly needed for this problem. However, as this solution indicates, it should be quite simple to implement.

c) We have seen in class that the E-step consists of computing the Q-function

$$Q(\Psi|\Psi^{(n)}) = E_{\mathbf{Z}|\mathbf{X}}[\log P_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z};\Psi)|\mathbf{x}=\mathcal{D}].$$

In this case, the hidden variables are $\mathbf{Z} = (X_{11}, X_{12})$ while the observed variables are $\mathbf{X} = (X_1, X_2, X_3, X_4)$ and the complete data is

$$P_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z};\Psi) = P_{X_{11},X_{12},X_2,X_3,X_4}(x_{11},x_{12},x_2,x_3,x_4;\Psi)$$

from which $\log P_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z};\Psi)$ is $L_c(\Psi)$ from **b)**. Hence,

$$Q(\Psi|\Psi^{(n)}) = E_{X_{11},X_{12}|X_1,X_2,X_3,X_4}[L_c(\Psi)|\mathcal{D}]$$

and the only quantity that needs to be computed, to know what this is, is

$$E_{X_{12}|X_1,X_2,X_3,X_4}[X_{12}|\mathcal{D}].$$

Note that x_{11} does not appear in $L_c(\Psi)$, because it does not affect the maximization, and is therefore irrelevant. In any case, since $x_{11}+x_{12}=x_1$ its expected value would be quite easy to compute (it is simply x_1 minus the expected value of x_{12}). To find the expected value of X_{12} we note that, given X_1, it does really not depend on X_2,X_3, or X_4. Furthermore, we note that X_{12} is the count of a subset (compact cars that are not bikes) of the total number of events counted by X_1, and therefore it has a binomial distribution. The binomial has two parameters: the total number of events, in this case x_1, and the probability of the event counted by X_{11}, which is

$$p = \frac{\frac{\Psi}{4}}{\frac{1}{2}+\frac{\Psi}{4}} = \frac{\Psi}{2+\Psi}.$$

Hence,

$$P_{X_{12}|X_1}(x_{12}|x_1) = \binom{x_1}{x_{12}} \left(\frac{\Psi}{2+\Psi}\right)^{x_{12}} \left(\frac{2}{2+\Psi}\right)^{x_1-x_{12}}$$

and the E-step consists of computing

$$\hat{x}_{12} = E_{X_{12}|X_1}(x_{12}|x_1) = \frac{\Psi^{(n)}}{\Psi^{(n)}+2}x_1.$$

The Q-function thus becomes

$$Q(\Psi|\Psi^{(n)}) = \hat{x}_{12}\log\left(\frac{1}{4}\Psi\right) + x_2\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_3\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_4\log\left(\frac{1}{4}\Psi\right)$$

and the M-step consists of finding

$$\Psi^{(n+1)} = \arg\max_{\Psi} Q(\Psi|\Psi^{(n)}).$$

This is exactly the problem that we solved in **b)** where we saw that the solution is

$$\Psi^{(n+1)} = \frac{\hat{x}_{12} + x_4}{\hat{x}_{12} + x_4 + x_2 + x_3}.$$

Thus, given an initial estimate $\Psi^{(0)}$, the algorithm iterates between
E-step:

$$\hat{x}_{12} = \frac{\Psi^{(n)}}{\Psi^{(n)} + 2} x_1.$$

M-step:

$$\Psi^{(n+1)} = \frac{\hat{x}_{12} + x_4}{\hat{x}_{12} + x_4 + x_2 + x_3}.$$

d) The fixed points of this algorithm are given by

$$\Psi^{(n+1)} = \Psi^{(n)} = \Psi$$

or

$$\begin{aligned} \Psi &= \frac{\frac{\Psi}{\Psi+2} x_1 + x_4}{\frac{\Psi}{\Psi+2} x_1 + x_4 + x_2 + x_3} \\ &= \frac{\Psi x_1 + x_4(\Psi + 2)}{\Psi x_1 + (\Psi + 2)(x_4 + x_2 + x_3)} \end{aligned}$$

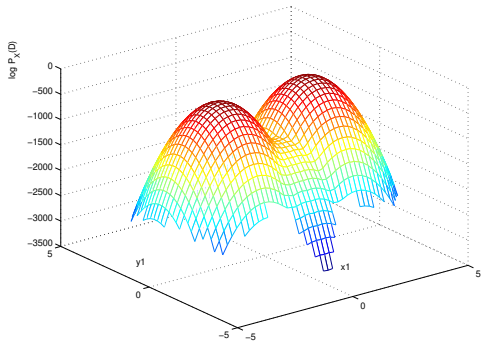
i.e.

$$\Psi^2(x_1 + x_2 + x_3 + x_4) + \Psi[2(x_2 + x_3) + x_4 - x_1] - 2x_4 = 0$$

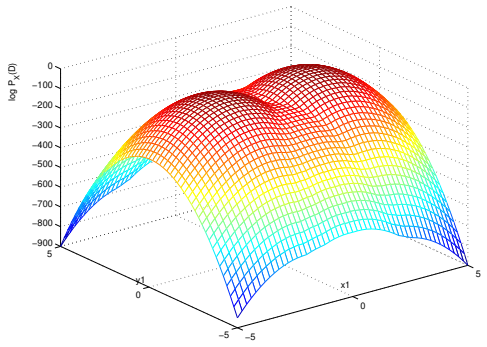
which is exactly the equation that we solved in **a)**. Hence, the solution is the same. This is not surprising, since we know that EM maximizes the likelihood of the incomplete data.

4.

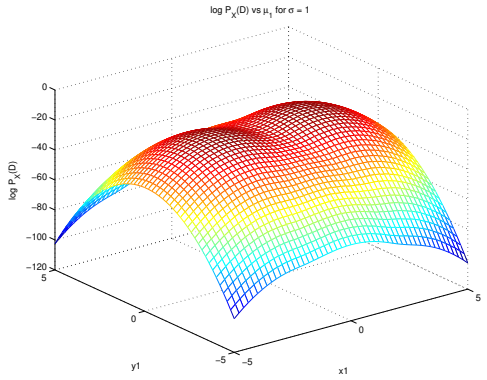
a) The surface plots are presented in the figure below. Note that there are two local maxima and the likelihood function is symmetric. This is a characteristic of all parameter estimation problems involving mixtures. In this case, $\{\mu_1, \Sigma_1, \pi_1\}$ can be the parameters of either of the Gaussians, calling one component 1 and the other component 2 is really just a question of convention. This is reflected in the likelihood surface that presents two identical local maxima for μ_1, co-located with the means of the two Gaussians. As far as ML is concerned, the two are equally good solutions. We thus see that a problem with C components will have at least $C!$ equivalent solutions and a likelihood surface with $C!$-fold symmetry. It turns out that this is not a big problem, since these solutions are indeed equivalent unless we care about a specific ordering of the mixture components. Typically we don't. There are usually also other local maxima or saddle points, and those can create problems. One such point is visible for the smaller values of σ, in between the two main bumps. Algorithms like EM can, and usually do, get trapped in such local optima.



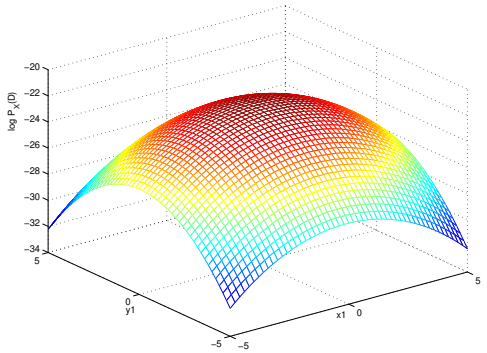
$\sigma^2 = 0.01$



$\sigma^2 = 0.1$



$\sigma^2 = 1$



$\sigma^2 = 10$

Regarding the role of σ, we see that it basically controls the smoothness of the likelihood function. For very small σ's the function is quite bumpy, becoming much smoother as σ increases. It is also visible from these plots that σ determines the bias and variance of the estimate. For small σ the function is very peaky, meaning that it is possible to determine the maxima with great accuracy. This means that the model has small bias, it is possible to approximate the underlying mixture very accurately. On the other hand, if we had 5 other points drawn from the same densities, these maxima would likely shift to other locations. Hence there is a significant amount of variance. As σ increases, bias increases as well. In fact, for $\sigma^2 = 10$ the bias is so large that it is practically impossible to say that there are two modes. In this case, the best solution is probably to assign the same mean to the two Gaussians. On

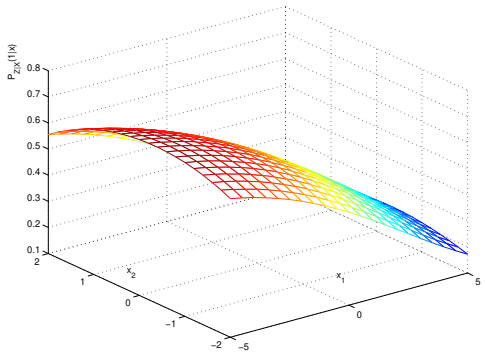
the other hand, the variance decreases. Because, with $\sigma^2 = 10$, there is so much smoothing going on, the likelihood surface is going to be like the one shown above for most sets of 5 points that one can draw from $P_{\mathbf{X}}(\mathbf{x})$. Hence, the estimates obtained with $\sigma^2 = 10$ are going to have small variance but very large bias.

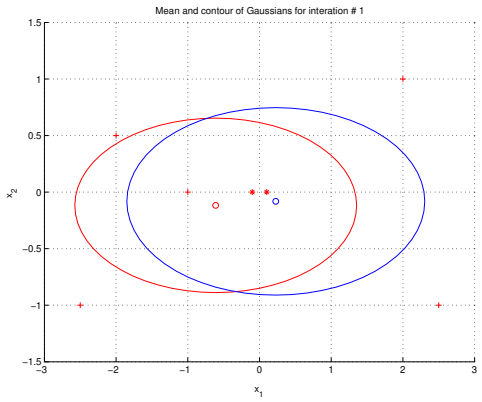
One of the nice properties of EM is that it automatically controls the bias and variance, by maximizing the likelihood with respect to all parameters. In areas where the true likelihood surface is very bumpy the mixture components will have small variance, while areas where the true likelihood surface is smoother will be covered by components of larger variance.

Finally, we see that convergence to a good solution is going to depend a lot on the initialization. If, for example, we had started from the location of the saddle point (visible when σ is small) an algorithm like EM might just get stuck in that initial point. In general there might be multiple local optima that are not close to the global optimum, and a good initialization can be critical.

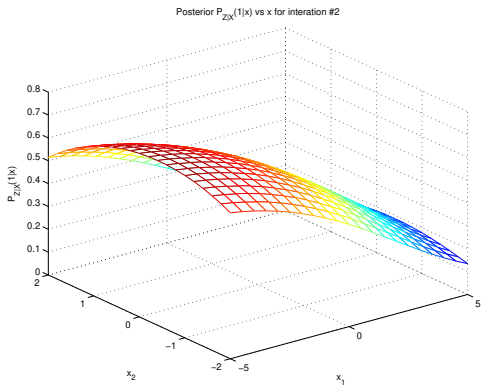
b) The plots of the posterior surface and the Gaussian estimates for the first three steps as well as after convergence (13 iterations) are shown in th next page. The surface plots are shown on the left column, while the Gaussian estimates are shown on the right. In the latter plots, the starting parameter estimates (for the iteration) are shown as '*', while the final parameter estimates are shown as 'o'. The last plot also shows the path traced by the mean estimates across iterations. A few observations can be made from these plots. First, the class assignments start very soft, becoming increasing harder, and are quite hard at convergence. The assignments are soft in the regions where the posterior is not 0 or 1, which means that there is a reasonable probability that points in those regions will be assigned to either of the classes. In the first three iterations this is true for the entire region covered by the plots. On the other hand, only points very close to the class boundary have soft assignments after convergence. Notice that this is very different from the sequence of assignments made by a greedy algorithm that assigns each point to only one of the classes at each iteration. In this case, because the classes are very separated, the result of such an algorithm would not be very different from the EM solution. However, when there is significant overlap between classes, greedy can be highly suboptimal.

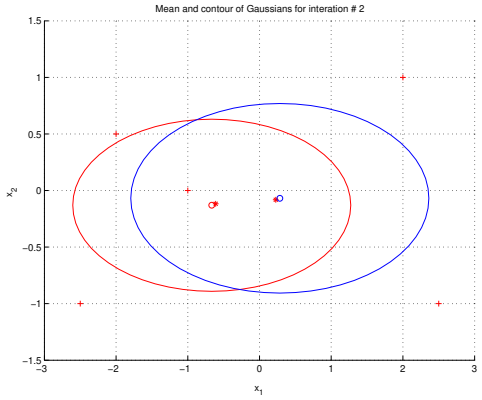
A second interesting point is the fact that convergence seems quite slow. In fact, after the first iteration, the progress within each iteration is quite small. This is a consequence of the soft assignments: because a point from the Gaussian on the right has a non-trivial probability under the Gaussian on the left, it also has a non-trivial contribution to its parameter updates (remember that e.g. the new mean is a weighted mean of all points, each point weighted by the its probability under the corresponding Gaussian). Hence, the points from the Gaussian on the right pull the parameters of the Gaussian on the left away from their true values, and vice-versa. The result is a slower convergence than that of a greedy algorithm based on hard assignments. This is the price to pay for the optimality of the EM solution.



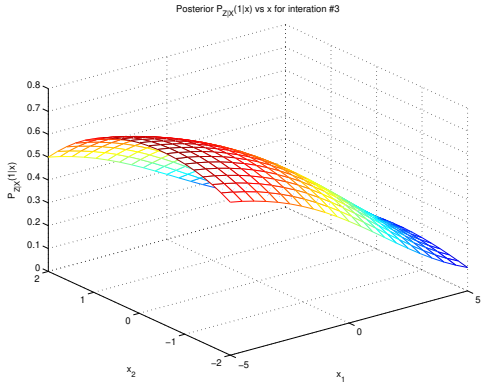


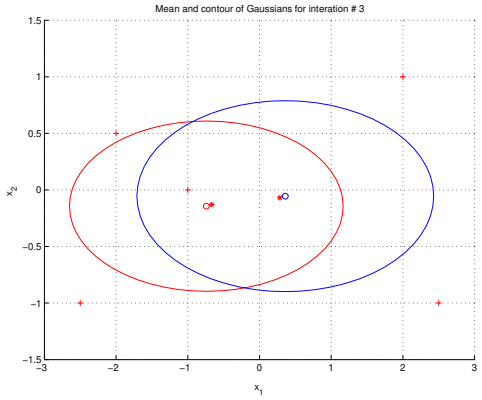
Iteration 1



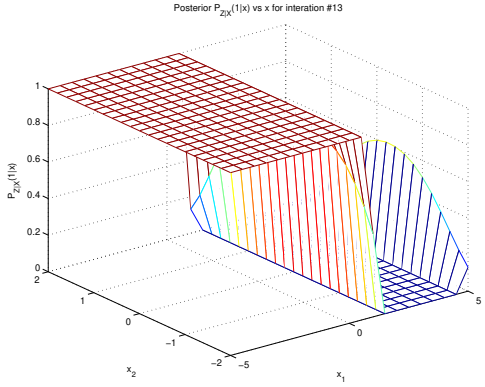


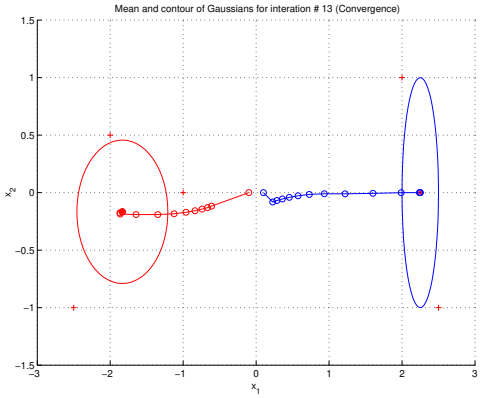
Iteration 2





Iteration 3





Iteration 13

5. We have seen in class that, for MAP estimation, the E-step is the same as that of ML estimation. So, the E-step is the one we computed on problem **3**, i.e. it consists of computing

$$\hat{x}_{12} = E_{X_{12}|X_1}(x_{12}|x_1) = \frac{\Psi^{(n)}}{\Psi^{(n)}+2}x_1.$$

The Q-function is

$$Q(\Psi|\Psi^{(n)}) = \hat{x}_{12}\log\left(\frac{1}{4}\Psi\right) + x_2\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_3\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + x_4\log\left(\frac{1}{4}\Psi\right)$$

and the M-step consists of finding

$$\begin{aligned}
\Psi^{(n+1)} &= \arg\max_{\Psi} Q(\Psi|\Psi^{(n)}) + \log P_\Psi(\Psi) \\
&= \arg\max_{\Psi} (\hat{x}_{12}+x_4)\log\left(\frac{1}{4}\Psi\right) + (x_2+x_3)\log\left(\frac{1}{4}-\frac{1}{4}\Psi\right) + (\nu_1-1)\log\Psi + (\nu_2-1)\log(1-\Psi) \\
&= \arg\max_{\Psi} (\hat{x}_{12}+x_4+\nu_1-1)\log\Psi + (x_2+x_3+\nu_2-1)\log(1-\Psi).
\end{aligned}$$

Setting the derivatives to zero, we get

$$(\hat{x}_{12}+x_4+\nu_1-1)(1-\Psi^{(n+1)}) = (x_2+x_3+\nu_2-1)\Psi^{(n+1)}$$

or

$$\Psi^{(n+1)} = \frac{\hat{x}_{12}+x_4+\nu_1-1}{\hat{x}_{12}+x_2+x_3+x_4+\nu_1+\nu_2-2}.$$

In summary, we have
E-step:

$$\hat{x}_{12} = \frac{\Psi^{(n)}}{\Psi^{(n)}+2}x_1,$$

M-step:

$$\Psi^{(n+1)} = \frac{\hat{x}_{12}+x_4+\nu_1-1}{\hat{x}_{12}+x_2+x_3+x_4+\nu_1+\nu_2-2}.$$

Note that this is exactly the same algorithm as that derived in problem **1** when $\nu_1 = \nu_2 = 1$. This makes sense since, in this case, the prior is uniform and the MAP and ML estimates should be the same.

b) Once again, the E-step does not change with respect to that of problem **1**, namely we need to compute

$$h_{ij} = \frac{g(\theta_j^{(t)})e^{\phi(\theta_j^{(t)})^T u(\mathbf{x}_i)}\pi_j^{(t)}}{\sum_k g(\theta_k^{(t)})e^{\phi(\theta_k^{(t)})^T u(\mathbf{x}_i)}\pi_k^{(t)}}, \forall i \in \{1,\ldots,n\}, j \in \{1,\ldots,C\}$$

The M-step consists of finding

$$\Psi^{(n+1)} = \arg\max_{\Psi} Q(\Psi|\Psi^{(n)}) + \log P_\Psi(\Psi).$$

In problem **3** we have seen that the Q-function is

$$\begin{aligned}
Q(\Psi|\Psi^{(n)}) &= \sum_{i=1}^{n}\sum_{j=1}^{C} h_{ij}\left[\log P_{\mathbf{X}|Z}(\mathbf{x}_i|j) + \log\pi_j\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{C} h_{ij}\left[\log g(\theta_j) + \phi(\theta_j)^T u(\mathbf{x}_i) + \log\pi_j\right] \\
&= \sum_{j=1}^{C}\log g(\theta_j)\sum_{i=1}^{n} h_{ij} + \sum_{j=1}^{C}\log\pi_j\sum_{i=1}^{n} h_{ij} + \sum_{j=1}^{C}\phi(\theta_j)^T\sum_{i=1}^{n} h_{ij}u(\mathbf{x}_i)
\end{aligned}$$

leading to

$$\begin{aligned}
\Psi^{(n+1)} &= \arg \max_{\{(\theta_1,\pi_1),\ldots,(\theta_C,\pi_c)\}} \sum_{j=1}^{C} \log g(\theta_j) \sum_{i=1}^{n} h_{ij} + \sum_{j=1}^{C} \log \pi_j \sum_{i=1}^{n} h_{ij} \\
&\quad + \sum_{j=1}^{C} \phi(\theta_j)^T \sum_{i=1}^{n} h_{ij} u(\mathbf{x}_i) + \sum_{j=1}^{C} \eta_j \log g(\theta_j) + \sum_{j=1}^{C} \phi(\theta_j)^T \nu_j \\
&= \arg \max_{\{(\theta_1,\pi_1),\ldots,(\theta_C,\pi_c)\}} \sum_{j=1}^{C} \log g(\theta_j) \left(\sum_{i=1}^{n} h_{ij} + \eta_j \right) + \sum_{j=1}^{C} \log \pi_j \sum_{i=1}^{n} h_{ij} \\
&\quad + \sum_{j=1}^{C} \phi(\theta_j)^T \left(\sum_{i=1}^{n} h_{ij} u(\mathbf{x}_i) + \nu_j \right).
\end{aligned}$$

This has basically the same form as the problem we solved in **3**, in fact the maximization with respect to the π's is exactly the same. Hence

$$\pi_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} h_{ij}.$$

For the maximization with respect to the θ's we now have

$$\theta_j^{(t+1)} = \arg \max_{\theta} \left\{ \log g(\theta) \left(\sum_{i=1}^{n} h_{ij} + \eta_j \right) + \phi(\theta)^T \left(\sum_{i=1}^{n} h_{ij} u(\mathbf{x}_i) + \nu_j \right) \right\}.$$

This can be solved in the same way as the corresponding maximization in problem **3**, and it follows that $\theta_j^{(t+1)}$ is the solution to

$$\nabla_\theta \phi(\theta_j)^T \left\{ E_{\theta_j}[u(\mathbf{x})] - \sum_{i=1}^{n} \frac{h_{ij} u(\mathbf{x}_i) + \nu_j}{\eta_j + \sum_{i=1}^{n} h_{ij}} \right\} = 0.$$

In summary,
E-step:

$$h_{ij} = \frac{g(\theta_j^{(t)}) e^{\phi(\theta_j^{(t)})^T u(\mathbf{x}_i)} \pi_j^{(t)}}{\sum_k g(\theta_k^{(t)}) e^{\phi(\theta_k^{(t)})^T u(\mathbf{x}_i)} \pi_k^{(t)}}, \forall i \in \{1,\ldots,n\}, j \in \{1,\ldots,C\}$$

M-step:

$$\pi_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} h_{ij} \qquad \nabla_\theta \phi(\theta_j)^T \left\{ E_{\theta_j}[u(\mathbf{x})] - \sum_{i=1}^{n} \frac{h_{ij} u(\mathbf{x}_i) + \nu_j}{\eta_j + \sum_{i=1}^{n} h_{ij}} \right\} = 0.$$

Comparing with problem **3** we see that the only difference between the ML and MAP estimates is the equation for the update of the θ_j's in the M-step. It is interesting to analyze this equation in light of what we have learned in problem set 3, where we saw that adding a conjugate prior was equivalent to augmenting the training sample with a virtual set of examples. Like then, the hyperparameters can be interpreted as

- η_j is a *virtual* number of samples from class j that are added to the training set. We thus have an extended set of $n + \sum_j \eta_j$ samples, where η_j samples are assigned to class j with probability 1.
- ν_j is the value that is added, by this additional set of virtual samples, to the sufficient statistic of the j^{th} class.