Problem 1.

(a) [5 points] Suppose we use the sigmoid function as the activation function for h_1, h_2, h_3 and o. What is the gradient descent update to $w_{1,2}^{[1]}$, assuming we use a learning rate of α? Your answer should be written in terms of $x^{(i)}$, $o^{(i)}$, $y^{(i)}$, and the weights.

$$l = \frac{1}{m} \sum_{i=1}^{m} (o^{(i)} - y^{(i)})^2$$

$$\frac{\partial}{\partial w_{1,2}^{[1]}} l = \frac{2}{m} \sum_{i=1}^{m} (o^{(i)} - y^{(i)}) \cdot \frac{\partial o^{(i)}}{\partial w_{1,2}^{[1]}}$$

$$\frac{\partial o}{\partial w_{1,2}^{[1]}} = \frac{\partial o}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial a^{[1]}} \cdot \frac{\partial a^{[1]}}{\partial z^{[1]}} \cdot \frac{\partial z^{[1]}}{w_{1,2}^{[1]}}$$

$$o(1-o) \cdot w^{[2]} \cdot a^{[1]}(1-a^{[1]}) \cdot x_1$$

$$a^{[1]} = \sigma(w^{[1]} \cdot x)$$

$$\text{or } = \sigma(w_{0,2}^{[1]} + x_1^{(i)} w_{1,2}^{[1]} + x_2^{(i)} w_{2,2}^{[1]})$$

$$\frac{\partial}{\partial w_{1,2}^{[1]}} l = \frac{2}{m} \cdot w_2^{[2]} \cdot \sum_{i=1}^{m} o^{(i)}(1-o^{(i)})(o^{(i)} - y^{(i)}) a^{[1]}(1-a^{[1]}) x_1^{(i)}$$

$$\therefore \; w_{1,2}^{[1]} := w_{1,2}^{[1]} - 2 \cdot \frac{2}{m} \cdot w_2^{[2]} \cdot \sum_{i=1}^{m} o^{(i)}(1-o^{(i)})(o^{(i)} - y^{(i)}) a^{[1]}(1-a^{[1]}) x_1^{(i)}$$

where $a^{[1]} = \sigma(w_{0,2}^{[1]} + x_1^{(i)} w_{1,2}^{[1]} + x_2^{(i)} w_{2,2}^{[1]})$

(b) [10 points] Now, suppose instead of using the sigmoid function for the activation function for h_1, h_2, h_3 and o, we instead used the step function $f(x)$, defined as

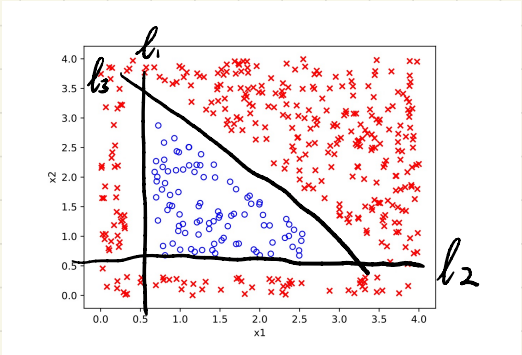
$$f(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

Is it possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy?

If it is possible, please provide a set of weights that enable 100% accuracy by completing `optimal_step_weights` within `src/p01_nn.py` and explain your reasoning for those weights in your PDF.

If it is not possible, please explain your reasoning in your PDF. (There is no need to modify `optimal_step_weights` if it is not possible.)

Hint: There are three sides to a triangle, and there are three neurons in the hidden layer.



yes, as the dataset could be classified by three linear classifers also

$w^{[2]} = \begin{bmatrix} w_1^{[2]} \\ w_2^{[2]} \\ w_3^{[2]} \end{bmatrix}$ the outcome "0" just simply one of the three lines

l_1: $x_1 = 0.5$

$-0.5 + 1 \cdot x_1 + 0 x_2 = 0$

l_2: $x_2 = 0.5$

$-0.5 + 0 \cdot x_1 + 1 \cdot x_1 = 0$

l_3: $x_1 + x_2 = 4$

$-4 + 1 \cdot x_1 + 1 \cdot x_2 = 0$

(c) [10 points] Let the activation functions for h_1, h_2, h_3 be the linear function $f(x) = x$ and the activation function for o be the same step function as before.

Is it possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy?

If it is possible, please provide a set of weights that enable 100% accuracy by completing `optimal_linear_weights` within `src/p01_nn.py` and explain your reasoning for those weights in your PDF.

If it is not possible, please explain your reasoning in your PDF. (There is no need to modify `optimal_linear_weights` if it is not possible.)

Impossible. The outcome would be a linear combination of three lines. However, the dataset can't be seperated by a single line

Problem 2.

(a) [5 points] **Nonnegativity.** Prove the following:

$$\forall P, Q \quad D_{\mathrm{KL}}(P\|Q) \geq 0$$

and

$$D_{\mathrm{KL}}(P\|Q) = 0 \quad \text{if and only if } P = Q.$$

Hint: You may use the following result, called **Jensen's inequality**. If f is a convex function, and X is a random variable, then $E[f(X)] \geq f(E[X])$. Moreover, if f is strictly convex (f is convex if its Hessian satisfies $H \geq 0$; it is *strictly* convex if $H > 0$; for instance $f(x) = -\log x$ is strictly convex), then $E[f(X)] = f(E[X])$ implies that $X = E[X]$ with probability 1; i.e., X is actually a constant.

$D_{KL}(P\|Q) = \sum_{x\in X} P(x) \log \frac{P(x)}{Q(x)}$ $\qquad E[f(x)] \leq f(E(x))$

$-D_{KL}(P\|Q) = \sum_{x\in X} P(x) \log \frac{Q(x)}{P(x)}$

$= E\left(\log \frac{Q(x)}{P(x)}\right) \leq \log E\left(\frac{Q(x)}{P(x)}\right) = \log \sum_{x\in X} P(x) \frac{Q(x)}{P(x)}$

$= \log \sum_{x\in X} Q(x) = \log 1 = 0$

$\therefore \; -D_{KL}(P\|Q) \leq 0 \quad$ or $\quad D_{KL}(P\|Q) \geq 0$

when $D_{KL}(P\|Q) = 0 \Rightarrow \frac{Q(x)}{P(x)}$ is a constant, say C.

$D_{KL}(P\|Q) = \log C \cdot \sum_{x\in X} P(x) = \log C \Rightarrow C = 1$ or $Q(x) = P(x)$ with prob of 1

also, if $Q(x) = P(x) \Rightarrow D_{KL}(P\|Q) = 0$

(b) [5 points] **Chain rule for KL divergence.** The KL divergence between 2 conditional distributions $P(X|Y), Q(X|Y)$ is defined as follows:

$$D_{\mathrm{KL}}(P(X|Y)\|Q(X|Y)) = \sum_y P(y)\left(\sum_x P(x|y)\log\frac{P(x|y)}{Q(x|y)}\right)$$

This can be thought of as the expected KL divergence between the corresponding conditional distributions on x (that is, between $P(X|Y=y)$ and $Q(X|Y=y)$), where the expectation is taken over the random y.

Prove the following chain rule for KL divergence:

$$D_{\mathrm{KL}}(P(X,Y)\|Q(X,Y)) = D_{\mathrm{KL}}(P(X)\|Q(X)) + D_{\mathrm{KL}}(P(Y|X)\|Q(Y|X)).$$

$$D_{KL}(P(X,Y)\|Q(X,Y)) = \sum_y \sum_x P(x,y)\log\frac{P(x,y)}{Q(x,y)} \quad (1)$$

$$D_{KL}(P(X)\|Q(X)) = \sum_x P(x)\log\frac{P(x)}{Q(x)} \quad (2)$$

$$D_{KL}(P(Y|X)\|Q(Y|X)) = \sum_x P(x)\left(\sum_y P(y|x)\log\frac{P(y|x)}{Q(y|x)}\right)$$

$$= \sum_x P(x)\sum_y \frac{P(x,y)}{P(x)}\left[\log\frac{P(x,y)}{Q(x,y)} - \log\frac{P(x)}{Q(x)}\right]$$

$$= \sum_x \sum_y P(x,y)\left(\log\frac{P(x,y)}{Q(x,y)}\right) - \sum_x \sum_y P(x,y)\left(\log\frac{P(x)}{Q(x)}\right)$$

$$= \sum_x \sum_y P(x,y)\left(\log\frac{P(x,y)}{Q(x,y)}\right) - \sum_x \log\frac{P(x)}{Q(x)} \sum_y P(x,y)$$

$$= \sum_x \sum_y P(x,y)\left(\log\frac{P(x,y)}{Q(x,y)}\right) - \sum_x \log\frac{P(x)}{Q(x)} P(x) \quad (3)$$

$(1) = (2) + (3)$

(c) [5 points] **KL and maximum likelihood.** Consider a density estimation problem, and suppose we are given a training set $\{x^{(i)}; i = 1, \ldots, m\}$. Let the empirical distribution be $\hat{P}(x) = \frac{1}{m} \sum_{i=1}^{m} 1\{x^{(i)} = x\}$. ($\hat{P}$ is just the uniform distribution over the training set; i.e., sampling from the empirical distribution is the same as picking a random example from the training set.)

Suppose we have some family of distributions P_θ parameterized by θ. (If you like, think of $P_\theta(x)$ as an alternative notation for $P(x; \theta)$.) Prove that finding the maximum likelihood estimate for the parameter θ is equivalent to finding P_θ with minimal KL divergence from $\hat{P}$. I.e. prove:

$$\arg\min_\theta D_{\mathrm{KL}}(\hat{P} \| P_\theta) = \arg\max_\theta \sum_{i=1}^{m} \log P_\theta(x^{(i)})$$

$$D_{KL}(\hat{P} \| P_\theta) = \sum_{i=1}^{m} \hat{P}(x^{(i)}) \log \frac{\hat{P}(x^{(i)})}{P(x^{(i)})}$$

$$= -\sum_{i=1}^{m} \hat{P}(x^{(i)}) \log P(x^{(i)}) + \sum_{i=1}^{m} \hat{P}(x^{(i)}) \log \hat{P}(x^{(i)})$$

constant

constant for a given dataset.

whereas MLE:

$$\sum_{i=1}^{m} \log P(x^{(i)})$$

$$\therefore \arg\min_\theta D_{KL}(\hat{P} \| P_\theta) = \arg\max_\theta \sum_{i=1}^{m} \log P_\theta(x^{(i)})$$

Problem 3.

(a) [3 points] **Score function**

The score function associated with $p(y;\theta)$ is defined as $\nabla_\theta \log p(y;\theta)$, which signifies the sensitivity of the likelihood function with respect to the parameters. Note that the score function is actually a vector since it's the gradient of a scalar quantity with respect to the vector θ.

Recall that $\mathbb{E}_{y\sim p(y)}[g(y)] = \int_{-\infty}^{\infty} p(y)g(y)dy$. Using this fact, show that the expected value of the score is 0, i.e.

$$\mathbb{E}_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta')|_{\theta'=\theta}] = 0$$

$$E_{y\sim p(y;\theta)}\left[\nabla_{\theta'} \log p(y;\theta')\big|_{\theta'=\theta}\right]$$

$$= \int_{-\infty}^{\infty} p(y;\theta) \cdot \nabla_{\theta'} \log p(y;\theta')\big|_{\theta'=\theta}\, dy$$

$$= \int_{-\infty}^{\infty} p(y;\theta) \cdot \frac{1}{p(y;\theta)} \cdot \nabla_\theta p(y;\theta)\, dy$$

$$= \int_{-\infty}^{\infty} \nabla_\theta p(y;\theta)\, dy = \nabla_\theta \int_{-\infty}^{\infty} p(y;\theta)\, dy = \nabla_\theta(1) = 0$$

(b) [2 points] **Fisher Information**

Let us now introduce a quantity known as the Fisher information. It is defined as the covariance matrix of the score function,

$$\mathcal{I}(\theta) = \mathrm{Cov}_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta')|_{\theta'=\theta}]$$

Intuitively, the Fisher information represents the amount of information that a random variable Y carries about a parameter θ of interest. When the parameter of interest is a vector (as in our case, since $\theta \in \mathbb{R}^n$), this information becomes a matrix. Show that the Fisher information can equivalently be given by

$$\mathcal{I}(\theta) = \mathbb{E}_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta')\nabla_{\theta'} \log p(y;\theta')^T|_{\theta'=\theta}]$$

Note that the Fisher Information is a function of the parameter. The parameter of the Fisher information is both a) the parameter value at which the score function is evaluated, and b) the parameter of the distribution with respect to which the expectation and variance is calculated.

$$I(\theta) = \mathrm{Cov}_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta') |_{\theta'=\theta}]$$

$$= E_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta') \nabla_{\theta'} \log p(y;\theta')^T |_{\theta'=\theta}]$$

$$- E_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta')] \cdot (E_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta')])^T$$

$$= E_{y\sim p(y;\theta)}[\nabla_{\theta'} \log p(y;\theta') \nabla_{\theta'} \log p(y;\theta')^T |_{\theta'=\theta}]$$

(c) [5 points] **Fisher Information (alternate form)**

It turns out that the Fisher Information can not only be defined as the covariance of the score function, but in most situations it can also be represented as the expected negative Hessian of the log-likelihood.

Show that $\mathbb{E}_{y\sim p(y;\theta)}[-\nabla^2_{\theta'} \log p(y;\theta')|_{\theta'=\theta}] = \mathcal{I}(\theta)$.

$$\nabla^2_{\theta'} \log p(y;\theta')|_{\theta'=\theta}$$

$$= \nabla_{\theta'}\left(\nabla_{\theta'} \log p(y;\theta')|_{\theta'=\theta}\right) = \nabla_{\theta'}\left(\frac{\nabla_{\theta} p(y;\theta')}{p(y;\theta')}\right)$$

$$= \frac{\nabla^2_{\theta'} p(y;\theta')}{p(y;\theta')} - \frac{\nabla_{\theta} p(y;\theta') \cdot \nabla_{\theta} p(y;\theta')^T}{p(y;\theta')^2}$$

$$E\left(\frac{\nabla^2_{\theta'} p(y;\theta')}{p(y;\theta')}\right) = \int_{-\infty}^{\infty} p(y;\theta') \cdot \frac{\nabla^2_{\theta'} p(y;\theta')}{p(y;\theta')}\, dy$$

$$= \nabla^2_{\theta'} \int_{-\infty}^{\infty} p(y;\theta')\, dy = \nabla^2_{\theta'}(1) = 0$$

$$\therefore E_{y\sim p(y;\theta)}\left[-\nabla^2_{\theta'} \log p(y;\theta')|_{\theta'=\theta}\right] = E_{y\sim p(y;\theta)}\left[\frac{\nabla_{\theta} p(y;\theta')\, \nabla_{\theta} p(y;\theta')^T}{p(y;\theta')^2}\right] \quad (1)$$

Also:

$$\mathcal{I}(\theta) = E_{y\sim p(y;\theta)}\left[\nabla_{\theta'} \log p(y;\theta')\, \nabla_{\theta'} \log p(y;\theta')^T |_{\theta'=\theta}\right]$$

$$= E_{y\sim p(y;\theta)}\left[\frac{\nabla_{\theta} p(y;\theta')}{p(y;\theta')} \cdot \frac{\nabla_{\theta} p(y;\theta')^T}{p(y;\theta')} \Big|_{\theta'=\theta}\right]$$

$$= E_{y\sim p(y;\theta)}\left[\frac{\nabla_{\theta} p(y;\theta') \cdot \nabla_{\theta} p(y;\theta')^T}{p(y;\theta')^2} \Big|_{\theta'=\theta}\right] = (1)$$

(d) [5 points] **Approximating D_{KL} with Fisher Information**

As we explained at the start of this problem, we are interested in the set of all distributions that are at a small fixed D_{KL} distance away from the current distribution. In order to calculate D_{KL} between $p(y;\theta)$ and $p(y;\theta+d)$, where $d \in \mathbb{R}^n$ is a small magnitude "delta" vector, we approximate it using the Fisher Information at θ. Eventually d will be the natural gradient update we will add to θ. To approximate the KL-divergence with Fisher Infomration, we will start with the Taylor Series expansion of D_{KL} and see that the Fisher Information pops up in the expansion.

Show that $D_{\mathrm{KL}}(p_\theta || p_{\theta+d}) \approx \frac{1}{2} d^T \mathcal{I}(\theta) d$.

Hint: Start with the Taylor Series expansion of $D_{\mathrm{KL}}(p_\theta || p_{\tilde{\theta}})$ where θ is a constant and $\tilde{\theta}$ is a variable. Later set $\tilde{\theta} = \theta + d$. Recall that the Taylor Series allows us to approximate a scalar function $f(\tilde{\theta})$ near θ by:

$$f(\tilde{\theta}) \approx f(\theta) + (\tilde{\theta}-\theta)^T \nabla_{\theta'} f(\theta')|_{\theta'=\theta} + \frac{1}{2}(\tilde{\theta}-\theta)^T \left(\nabla^2_{\theta'} f(\theta')|_{\theta'=\theta}\right)(\tilde{\theta}-\theta)$$

$D_{KL}(P_\theta \| P_{\theta+d}) = E_{x\sim P_\theta(x)}[\log P_\theta(x)] - E_{x\sim P(x)}[\log P_{\theta+d}(x)]$ (1)

by Taylor series:

$\log P_{\theta+d}(x) \simeq \log P_\theta(x) + d^T \nabla_{\theta'} \log P_\theta(x)|_{\theta'=\theta} + \frac{1}{2} d^T (\nabla^2_{\theta'} \log P_\theta(x)|_{\theta'=\theta}) d$

$\therefore E_{x\sim P_\theta(x)}[\log P_{\theta+d}(x)] \simeq E_{x\sim P_\theta(x)}[\log P_\theta(x)] + d^T E_{x\sim P_\theta(x)}[\nabla_{\theta'} \log P_\theta(x)]$

$+ \frac{1}{2} d^T \cdot E_{x\sim P_\theta(x)}[\nabla^2_{\theta'} \log P_\theta(x)|_{\theta'=\theta}] d$ (2)

$\nabla_{\theta'} \log P_\theta(x) = 0$

$-I(\theta)$

$\therefore (1) - (2) = \frac{1}{2} d^T I(\theta) d.$

(e)

$$d^* = \arg\max_d \ell(\theta+d) \quad \text{subject to } D_{KL}(P_\theta \| P_{\theta+d}) = C$$

by Taylor series:

$$d^* = \arg\max_d \ell(\theta) + d^T \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}$$

subject to: $D_{KL}(P_\theta \| P_{\theta+d}) \approx \frac{1}{2} d^T I(\theta) d$

$$\therefore \mathcal{L}(d, \lambda) = \ell(\theta) + d^T \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} - \lambda\left[\frac{1}{2} d^T I(\theta) d - C\right]$$

$$\nabla_d \mathcal{L}(d,\lambda) = \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} - \lambda I(\theta) d \overset{\text{set}}{=} 0$$

$$d = \frac{1}{\lambda} (I(\theta))^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} \quad (1)$$

$$\nabla_\lambda \mathcal{L}(d,\lambda) = -\frac{1}{2} d^T I(\theta) d + C$$

$$-\frac{1}{2} \cdot \frac{1}{\lambda} \cdot (\nabla_{\theta'} \ell(\theta')|_{\theta'=\theta})^T (I(\theta)^{-1})^T I(\theta) \cdot \frac{1}{\lambda} I(\theta)^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} + C$$

$$= -\frac{1}{2\lambda^2} (I(\theta)^{-1} \cdot \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta})^T \cdot \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} + C \overset{\text{set}}{=} 0$$

$$\frac{(I(\theta)^{-1} \cdot \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta})^T \cdot \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}}{2C} = \lambda^2$$

$$\lambda = \sqrt{\frac{(I(\theta)^{-1} \cdot \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta})^T \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}}{2C}}$$

plugin (2) to (1)

$$\nabla_{\theta'} l(\theta')|_{\theta'=\theta} = \frac{\nabla_{\theta'} P(y;\theta')|_{\theta'=\theta}}{P(y;\theta)}$$

$$d = \frac{1}{\lambda} (I(\theta))^{-1} \cdot \nabla_{\theta'} l(\theta')|_{\theta'=\theta}$$

$$= \sqrt{\frac{2C}{(I(\theta)^{-1} \cdot \nabla_{\theta'} l(\theta')|_{\theta'=\theta})^T \nabla_{\theta'} l(\theta')|_{\theta'=\theta}}} \; I(\theta)^{-1} \cdot \nabla_{\theta'} l(\theta')|_{\theta'=\theta}$$

$$= \sqrt{\frac{2 \cdot P(y;\theta)^2 \cdot C}{(\nabla_{\theta'} P(y;\theta')|_{\theta'=\theta})^T \cdot (I(\theta)^{-1})^T \cdot \nabla_{\theta'} P(y;\theta')|_{\theta'=\theta}}} \; I(\theta)^{-1} \frac{\nabla_{\theta'} P(y;\theta')|_{\theta'=\theta}}{P(y;\theta)}$$

$$= \sqrt{\frac{2C}{(\nabla_{\theta'} P(y;\theta')|_{\theta'=\theta})^T \cdot (I(\theta)^{-1})^T \cdot \nabla_{\theta'} P(y;\theta')|_{\theta'=\theta}}} \; I(\theta)^{-1} \cdot \nabla_{\theta'} P(y;\theta')|_{\theta'=\theta}$$

* Notice the fisher information is always symmetric, we can replace $(I(\theta)^{-1})^T$ with $I(\theta)^{-1}$

(f) [2 points] **Relation to Newton's Method**

After going through all these steps to calculate the natural gradient, you might wonder if this is something used in practice. We will now see that the familiar Newton's method that we studied earlier, when applied to Generalized Linear Models, is equivalent to natural gradient on Generalized Linear Models. While the two methods (Netwon's and natural gradient) agree on GLMs, in general they need not be equivalent.

Show that the direction of update of Newton's method, and the direction of natural gradient, are exactly the same for Generalized Linear Models. You may want to recall and cite the results you derived in problem set 1 question 4 (Convexity of GLMs). For the natural gradient, it is sufficient to use $\tilde{d}$, the unscaled natural gradient.

Recall Newton's method

$$\theta := \theta - H^{-1} \nabla_\theta \ell(\theta) \quad \text{where } H \text{ is Hessian Matrix}$$

Natural Gradient:

$$I(\theta) = E_{y\sim p(y;\theta)}[-\nabla_\theta^2 \log p(y;\theta)] \qquad \ell(\theta) = \log p(y;\theta)$$

$$= -E_{y\sim p(y;\theta)}[H] \quad \text{where } H = H_\theta(\ell(\theta))$$

$$\tilde{d} = \frac{1}{\lambda}(I(\theta))^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}$$

$$\theta := \theta + \tilde{d}$$

$$= \theta + \frac{1}{\lambda}(I(\theta))^{-1} \cdot \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}$$

$$= \theta - \frac{1}{\lambda}\left(E_{y\sim p(y;\theta)}[H]\right)^{-1} \cdot \nabla_\theta \ell(\theta)$$

So the Newton's Method has the same direction with Natural Gradient

Problem 4.

(a) [5 points] **Convergence.** First we will show that this algorithm eventually converges. In order to prove this, it is sufficient to show that our semi-supervised objective $\ell_{\text{semi-sup}}(\theta)$ monotonically increases with each iteration of E and M step. Specifically, let $\theta^{(t)}$ be the parameters obtained at the end of t EM-steps. Show that $\ell_{\text{semi-sup}}(\theta^{(t+1)}) \geq \ell_{\text{semi-sup}}(\theta^{(t)})$.

$$ELBO(x; Q, \theta) = \sum_z Q(z) \log \frac{P(x, z; \theta)}{Q(z)}$$

$$Q^{(t)}(z^{(i)}) = P(z^{(i)} | x^{(i)}; \theta^{(t)})$$

For $ELBO(x^{(i)}; Q^{(t)}, \theta^{(t)})$

$$= \sum_z Q^{(t)}(z^{(i)}) \log \frac{P(z^{(i)} | x^{(i)}; \theta^{(t)}) \, P(x^{(i)}; \theta^{(t)})}{P(z^{(i)} | x^{(i)}; \theta^{(t)})}$$

$$= \sum_z P(z^{(i)} | x^{(i)}; \theta^{(t)}) \log P(x^{(i)}; \theta^{(t)})$$

$$= \log P(x^{(i)}; \theta^{(t)}) \sum_z P(z^{(i)} | x^{(i)}; \theta^{(t)}) = 1$$

$$= \log P(x^{(i)}; \theta^{(t)}) = \ell_{unsup}(\theta^{(t)})$$

$$\ell_{semi-sup}(\theta^{(t+1)}) = \ell_{unsup}(\theta^{(t+1)}) + 2 \cdot \ell_{sup}(\theta^{(t+1)})$$

$$\geq \sum_{i=1}^{m} ELBO(x^{(i)}; Q^{(t)}, \theta^{(t+1)}) + 2 \ell_{sup}(\theta^{(t+1)})$$

$$\geq \sum_{i=1}^{m} ELBO(x^{(i)}; Q^{(t)}, \theta^{(t)}) + 2 \cdot \ell_{sup}(\theta^{(t)})$$

$$= \ell_{unsup}(\theta^{(t)}) + 2 \ell_{sup}(\theta^{(t)}) = \ell_{semi-sup}(\theta^{(t)})$$

(b) [5 points] **Semi-supervised E-Step.** Clearly state which are all the latent variables that need to be re-estimated in the E-step. Derive the E-step to re-estimate all the stated latent variables. Your final E-step expression must only involve x, z, μ, Σ, ϕ and universal constants.

Since $\tilde{z}^{(i)}$ are not latent variables that are already labelled, $z^{(i)}$ are the only latent variables to be re-estimate.

$$Q(z^{(i)}) = p(z^{(i)}=j \mid x^{(i)}; \phi, \mu, \Sigma)$$

$$= \frac{p(x^{(i)} \mid z^{(i)}=j; \mu, \Sigma)\, p(z^{(i)}=j; \phi)}{\sum_{l=1}^{k} p(x^{(i)} \mid z^{(i)}=l; \mu, \Sigma)\, p(z^{(i)}=l; \phi)}$$

For $x^{(i)} \mid z^{(i)} \sim N(\mu^{(i)}, \Sigma^{(i)})$

$z^{(i)} \sim \text{Multi}(\phi)$

$$= \frac{\frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)}-\mu_j)^T \Sigma_j^{-1} (x^{(i)}-\mu_j)\right) \phi_j}{\sum_{l=1}^{k} \frac{1}{(2\pi)^{n/2} |\Sigma_l|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)}-\mu_l)^T \Sigma_l^{-1} (x^{(i)}-\mu_l)\right) \phi_l}$$

(c) [5 points] **Semi-supervised M-Step.** Clearly state which are all the parameters that need to be re-estimated in the M-step. Derive the M-step to re-estimate all the stated parameters. Specifically, derive closed form expressions for the parameter update rules for $\mu^{(t+1)}$, $\Sigma^{(t+1)}$ and $\phi^{(t+1)}$ based on the semi-supervised objective.

Construct the Lagrangian:

$$L(\phi) = \sum_{i=1}^{m} \sum_{j=1}^{k} w_j^{(i)} \log(\phi_j) + \alpha \sum_{i=1}^{\tilde{m}} \sum_{j=1}^{k} I(\tilde{z}^{(i)} = j) \log(\phi_j) + \beta \left(\sum_{j=1}^{k} \phi_j - 1 \right)$$

$$\nabla_{\phi_j} L(\phi) = \sum_{i=1}^{m} \frac{w_j^{(i)}}{\phi_j} + \alpha \sum_{i=1}^{\tilde{m}} \frac{I(\tilde{z}^{(i)} = j)}{\phi_j} + \beta \overset{\text{set}}{=} 0$$

$$\therefore \phi_j = - \frac{\sum_{i=1}^{m} w_j^{(i)} + \alpha \sum_{i=1}^{\tilde{m}} I(\tilde{z}^{(i)} = j)}{\beta}$$

$$\because \sum_{j=1}^{k} \phi_j = - \frac{\sum_{j=1}^{k} \sum_{i=1}^{m} w_j^{(i)} + \alpha \sum_{j=1}^{k} \sum_{i=1}^{\tilde{m}} I(\tilde{z}^{(i)} = j)}{\beta} = 1$$

$$= - \frac{m + \alpha \tilde{m}}{\beta} = 1 \Rightarrow -\beta = m + \alpha \tilde{m}$$

$$\therefore \phi_j = \frac{\sum_{i=1}^{m} w_j^{(i)} + \alpha \cdot \sum_{i=1}^{\tilde{m}} I(\tilde{z}^{(i)} = j)}{m + \alpha \tilde{m}} \quad (1)$$

$$\nabla_{\mu_j}(\ell_{unsup}(\phi, \mu, \Sigma))$$

$$= \sum_{i=1}^{m} w_j^{(i)} (\Sigma_j^{-1} x^{(i)} - \Sigma_j^{-1} \mu_j)$$

$$= \Sigma_j^{-1} \left(\sum_{i=1}^{m} w_j^{(i)} x^{(i)} - \mu_j \sum_{i=1}^{m} w_j^{(i)} \right)$$

$$\nabla_{\mu_j}(\ell_{sup}(\phi, \mu, \Sigma))$$

$$= \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\} (\Sigma_j^{-1} \tilde{x}^{(i)} - \Sigma_j^{-1} \mu_j)$$

$$= \Sigma_j^{-1} \left(\sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\} \tilde{x}^{(i)} - \mu_j \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\} \right)$$

$$\nabla_{\mu_j}(\ell_{semi\text{-}sup}) \overset{set}{=} 0$$

$$\Rightarrow \mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)} + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\} \tilde{x}^{(i)}}{\sum_{i=1}^{m} w_j^{(i)} + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\}} \quad (2)$$

From $\frac{\partial}{\partial A}|\log A| = A^{-T}$ also $\Sigma_j^T = \Sigma_j$

$(A^{-1})' = -A^{-1}AA^{-1}$

$$\nabla_{\Sigma_j}\left(l_{unsup}(\phi,\mu,\Sigma)\right)$$

$$= \nabla_{\Sigma_j}\left(-\frac{1}{2}\sum_{i=1}^{m} w_j^{(i)}\cdot\log|\Sigma_j| + \sum_{i=1}^{m} w_j^{(i)}\left(-\frac{1}{2}(x^{(i)}-\mu_j)^T\Sigma_j^{-1}(x^{(i)}-\mu_j)\right)\right)$$

$$= -\frac{1}{2}\sum_{i=1}^{m} w_j^{(i)}\Sigma_j^{-1} + \frac{1}{2}\cdot\sum_{i=1}^{m} w_j^{(i)}\cdot\Sigma_j^{-1}(x^{(i)}-\mu_j)^T(x^{(i)}-\mu_j)\Sigma_j^{-1}$$

$$\nabla_{\Sigma_j}\left(l_{sup}(\phi,\mu,\Sigma)\right)$$

$$= -\frac{1}{2}\sum_{i=1}^{\tilde{m}} I\{\tilde{z}^{(i)}=j\}\Sigma_j^{-1} + \frac{1}{2}\cdot\sum_{i=1}^{\tilde{m}} I\{\tilde{z}^{(i)}=j\}\cdot\Sigma_j^{-1}(\tilde{x}^{(i)}-\mu_j)^T(\tilde{x}^{(i)}-\mu_j)\Sigma_j^{-1}$$

$$\nabla_{\Sigma_j}\left(l_{semi\text{-}sup}(\phi,\mu,\Sigma)\right) \overset{Set}{=} 0$$

$$\Sigma_j^{-1} = \frac{\sum_{i=1}^{m} w_j^{(i)}(x^{(i)}-\mu_j)^T(x^{(i)}-\mu_j) + \alpha\cdot\sum_{i=1}^{\tilde{m}} I\{\tilde{z}^{(i)}=j\}(\tilde{x}^{(i)}-\mu_j)^T(\tilde{x}^{(i)}-\mu_j)}{\sum_{i=1}^{m} w_j^{(i)} + \alpha\cdot\sum_{i=1}^{\tilde{m}} I\{\tilde{z}^{(i)}=j\}} \quad (3)$$

(f) [3 points] **Comparison of Unsupervised and Semi-supervised EM.** Briefly describe the differences you saw in unsupervised *vs.* semi-supervised EM for each of the following:

i. Number of iterations taken to converge.

ii. Stability (*i.e.,* how much did assignments change with different random initializations?)

iii. Overall quality of assignments.

Note: The dataset was sampled from a mixture of three low-variance Gaussian distributions, and a fourth, high-variance Gaussian distribution. This should be useful in determining the overall quality of the assignments that were found by the two algorithms.

i.

About 1000 iterations for unsupervised EM

50 iterations for semi-supervised EM

ii.

Unsupervised EM is not stable, it does not guarantee to classify the same results because of the random initiation

Semi-supervised EM is very stable.

iii.

The overall quality of semisupervised EM is better than unsupervised.

Problem 5:

(b) [5 points] **Compression Factor.** If we represent the image with these reduced (16) colors, by (approximately) what factor have we compressed the image?

16 colors need $\log_2 16 = 4$ bits

Compression Factor $= \frac{24}{4} = 6$