CS 229 Autumn 2018 Problem Set #3

Ruining Li
University of Oxford

Problem 1

(a) We compute the partial derivative of l w.r.t. $w_{1,2}^{[1]}$ by the chain rule:

$$\begin{aligned}\frac{\partial l}{\partial w_{1,2}^{[1]}} &= \frac{1}{m}\sum_{i=1}^{m} 2(o^{(i)} - y^{(i)})\frac{\partial o^{(i)}}{\partial w_{1,2}^{[1]}} \\ &= \frac{1}{m}\sum_{i=1}^{m} 2(o^{(i)} - y^{(i)})o^{(i)}(1-o^{(i)})w_2^{[2]}\frac{\partial h_2^{(i)}}{\partial w_{1,2}^{[1]}} \\ &= \frac{1}{m}\sum_{i=1}^{m} 2(o^{(i)} - y^{(i)})o^{(i)}(1-o^{(i)})w_2^{[2]}h_2^{(i)}(1-h_2^{(i)})x_1^{(i)}.\end{aligned}$$

The gradient descent update to $w_{1,2}^{[1]}$ is given by:

$$w_{1,2}^{[1]} := w_{1,2}^{[1]} - \alpha\frac{\partial l}{\partial w_{1,2}^{[1]}}$$

where the partial derivative is given above.

(b) It is possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy.

Reasoning: If we plot the data, we observe that one of the decision boundaries that perfectly classifies this dataset is the triangle with vertices $(0.5, 0.5)$, $(0.5, 3.5)$, and $(3.5, 0.5)$. Therefore, we can tune the weights for the hidden layer so that $h_1 = 1$ if and only if the input is to the right of the line $x_1 = 0.5$, $h_2 = 1$ if and only if the input is above the line $x_2 = 0.5$, and $h_3 = 1$ if and only if the input is below the line $x_1 + x_2 - 4 = 0$. We can also tune the weights for the output layer so that $o = 0$ if and only if $h_1 = h_2 = h_3 = 1$. For an example of the weight values, please see `src/p01_nn.py`.

(c) It is not possible to have a set of weights that allow the neural network to classify this dataset with 100% accuracy.

Reasoning: Clearly the dataset is not linearly separable. If we take the activation functions for h_1, h_2, h_3 to be a linear function, the learned decision boundary will always be linear in the input space, and therefore cannot classify this dataset perfectly.

Problem 2

(a) $-\log$ is a strictly convex function. Applying Jensen's inequality, we obtain

$$D_{\mathrm{KL}}(P\|Q) = -\sum_x P(x)\log\frac{Q(x)}{P(x)} \geq \log\sum_x Q(x) = \log 1 = 0$$

where the equality holds if and only if $\frac{P(x)}{Q(x)}$ is constant. Because P and Q are both probability distributions and are therefore properly normalized, $\frac{P(x)}{Q(x)}$ is constant if and only if $P = Q$. This indicates $D_{\mathrm{KL}}(P\|Q) = 0$ if and only if $P = Q$.

(b) We state

$$\begin{aligned}
D_{\mathrm{KL}}(P(X,Y)\|Q(X,Y)) &= \sum_x\sum_y P(x,y)\log\frac{P(x,y)}{Q(x,y)} \\
&= \sum_x\sum_y P(x,y)\left(\log\frac{P(x,y)}{Q(x,y)} + \log\frac{Q(x)}{P(x)}\right) + \sum_x\sum_y P(x,y)\log\frac{P(x)}{Q(x)} \\
&= \sum_x\sum_y P(x,y)\left(\log\frac{Q(x)P(y|x)P(x)}{P(x)Q(y|x)Q(x)}\right) + \sum_x P(x)\log\frac{P(x)}{Q(x)} \\
&= \sum_x P(x)\left(\sum_y P(y|x)\log\frac{P(y|x)}{Q(y|x)}\right) + \sum_x P(x)\log\frac{P(x)}{Q(x)} \\
&= D_{\mathrm{KL}}(P(X)\|Q(X)) + D_{\mathrm{KL}}(P(Y|X)\|Q(Y|X)).
\end{aligned}$$

(c) By the definition of $\hat{P}$, we obtain

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

where the first term is constant.

This indicates that finding the maximum likelihood estimate for the parameter θ is equivalent to finding P_θ with minimal KL divergence from $\hat{P}$.

Problem 3

(a) We state

$$
\begin{aligned}
\text{left-hand side} &= \int_{-\infty}^{\infty} p(y;\theta)[\nabla_{\theta'} \log p(y;\theta')|_{\theta'=\theta}]\,dy \\
&= \int_{-\infty}^{\infty} p(y;\theta)\frac{[\nabla_{\theta'} p(y;\theta')|_{\theta'=\theta}]}{p(y;\theta)}\,dy \\
&= \left\{\nabla_{\theta'} \int_{-\infty}^{\infty} p(y;\theta')\,dy\right\}_{\theta'=\theta} \\
&= 0
\end{aligned}
$$

because probability distributions are normalized.

(b) By definition, $\text{Cov}[X] = E[(X - E[X])(X - E[x])^T]$.

By part (a), the expectation of the score function is 0. Therefore, its covariance matrix is given by the score function multiplied by its transpose.

(c) We state

$$
\begin{aligned}
\text{left-hand side} &= -\int_{-\infty}^{\infty} p(y;\theta)\mathbb{J}_{\theta'}\left\{\frac{\nabla_{\theta'} p(y;\theta')}{p(y;\theta)}\right\}_{\theta'=\theta} dy \\
&= \int_{-\infty}^{\infty} p(y;\theta)\left\{\nabla_{\theta'} p(y;\theta')\frac{\nabla_{\theta'} p(y;\theta')}{p(y;\theta)^2} - \frac{\nabla^2_{\theta'} p(y;\theta')}{p(y;\theta)}\right\}_{\theta'=\theta} dy \\
&= \int_{-\infty}^{\infty} p(y;\theta)\left[\nabla_{\theta'} p(y;\theta')\frac{\nabla_{\theta'} p(y;\theta')}{p(y;\theta)^2}\right]_{\theta'=\theta} dy - \int_{-\infty}^{\infty}[\nabla^2_{\theta'} p(y;\theta')|_{\theta'=\theta}]\,dy
\end{aligned}
$$

where $\mathbb{J}$ denotes Jacobian and the second equality comes from the chain rule of Jacobian.

Note that the second term above is 0 because

$$
\int_{-\infty}^{\infty}[\nabla^2_{\theta'} p(y;\theta')|_{\theta'=\theta}]\,dy = \left\{\nabla^2_{\theta'} \int_{-\infty}^{\infty} p(y;\theta')\,dy\right\}_{\theta'=\theta}
$$

where p as a probability distribution is normalized.

We conclude the proof by stating that

$$
\begin{aligned}
\mathcal{I}(\theta) &= \int_{-\infty}^{\infty} p(y;\theta)[\nabla_{\theta'} \log p(y;\theta)\nabla_{\theta'} \log p(y;\theta)^T|_{\theta'=\theta}]\,dy \\
&= \int_{-\infty}^{\infty} p(y;\theta)\left[\nabla_{\theta'} p(y;\theta')\frac{\nabla_{\theta'} p(y;\theta')}{p(y;\theta)^2}\right]_{\theta'=\theta} dy
\end{aligned}
$$

(d) Let $f(\tilde{\theta}) = D_{\mathrm{KL}}(p_\theta \| p_{\tilde{\theta}}) = \int_{-\infty}^{\infty} p_\theta(y) \log \frac{p_\theta(y)}{p_{\tilde{\theta}}(y)}\, dy$. Then,

$$\nabla_{\theta'} f(\theta')|_{\theta'=\theta} = -\int_{-\infty}^{\infty} p_\theta(y)[\nabla_{\theta'} \log p_{\theta'}(y)|_{\theta'=\theta}]\, dy = 0 \quad \text{by part (a);}$$
$$\nabla^2_{\theta'} f(\theta')|_{\theta'=\theta} = -\int_{-\infty}^{\infty} p_\theta(y)[\nabla^2_{\theta'} \log p_{\theta'}(y)|_{\theta'=\theta}]\, dy = \mathcal{I}(\theta) \quad \text{by part (c).}$$

Then, we approximate $f(\tilde{\theta})$ with its second-degree Taylor series expansion:

$$\begin{aligned} f(\tilde{\theta}) &\approx f(\theta) + (\tilde{\theta} - \theta)^T \nabla_{\theta'} f(\theta')|_{\theta'=\theta} + \frac{1}{2}(\tilde{\theta} - \theta)^T (\nabla^2_{\theta'} f(\theta')|_{\theta'=\theta})(\tilde{\theta} - \theta) \\ &= \frac{1}{2}(\tilde{\theta} - \theta)^T \mathcal{I}(\theta)(\tilde{\theta} - \theta). \end{aligned}$$

To obtain our desired result, set $\tilde{\theta} = \theta + d$.

(e) We construct the Lagrangian as follows:

$$\begin{aligned} \mathcal{L}(d, \lambda) &= \ell(\theta + d) - \lambda(D_{\mathrm{KL}}(p_\theta \| p_{\theta+d}) - c) \\ &\approx \ell(\theta) + d^T \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} - \lambda \left(\frac{1}{2} d^T \mathcal{I}(\theta) d - c \right) \end{aligned}$$

where we use Taylor approximation on various quantities.

Now we compute the gradient of the Lagrangian w.r.t d and λ respectively and set the gradients to 0:

$$\nabla_d \mathcal{L}(d, \lambda) = \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} - \lambda \mathcal{I}(\theta) d = 0 \tag{1}$$
$$\nabla_\lambda \mathcal{L}(d, \lambda) = -\left(\frac{1}{2} d^T \mathcal{I}(\theta) d - c \right) = 0 \tag{2}$$

From (1) we obtain

$$d = \frac{1}{\lambda} \mathcal{I}(\theta)^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} \tag{3}$$

Plug (3) into (2), we obtain

$$[\nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}]^T \mathcal{I}(\theta)^{-1} [\nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}] = 2\lambda^2 c \tag{4}$$

from which we can obtain an expression of λ that does not involve d.

Plug the expression for λ (without d) back into (3), we come up with an expression for d that does not include λ:

$$d^* = \sqrt{\frac{2c}{[\nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}]^T \mathcal{I}(\theta)^{-1} [\nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}]}} \mathcal{I}(\theta)^{-1} [\nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}].$$

(f) For Newton's Method, the update rule is

$$\theta := \theta - H^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}.$$

The direction of the natural gradient is given by

$$\mathcal{I}(\theta)^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta} = \left(E_{y \sim p(y;\theta)}[H]\right)^{-1} \nabla_{\theta'} \ell(\theta')|_{\theta'=\theta}$$

which is equivalent to Newton's Method.

Problem 4

(a) By Jensen's inequality, we obtain

$$\begin{aligned} l_{\text{unsup}}(\theta) &= \sum_{i=1}^{m} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \\ &= \sum_{i=1}^{m} \log \sum_{z^{(i)}} Q(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q(z^{(i)})} \\ &\geq \sum_{i=1}^{m} \sum_{z^{(i)}} Q(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q(z^{(i)})} \end{aligned}$$

for some distribution Q.

It follows that

$$\begin{aligned} l_{\text{semi-sup}}(\theta^{(t+1)}) &= l_{\text{unsup}}(\theta^{(t+1)}) + \alpha l_{\text{sup}}(\theta^{(t+1)}) \\ &\geq \sum_{i=1}^{m} \left(\sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} \right) + \alpha \left(\sum_{i=1}^{\tilde{m}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta^{(t+1)}) \right) \\ &\geq \sum_{i=1}^{m} \left(\sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} \right) + \alpha \left(\sum_{i=1}^{\tilde{m}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta^{(t)}) \right) \\ &= \sum_{i=1}^{m} \left(\sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log p(x^{(i)}; \theta^{(t)}) \right) + \alpha \left(\sum_{i=1}^{\tilde{m}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta^{(t)}) \right) \\ &= \sum_{i=1}^{m} \log p(x^{(i)}; \theta^{(t)}) + \alpha \left(\sum_{i=1}^{\tilde{m}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta^{(t)}) \right) \\ &= l_{\text{semi-sup}}(\theta^{(t)}). \end{aligned}$$

(b) In the E-step, we need to re-estimate $Q_i^{(t)}(z^{(i)})$, where $i \in \{1, \ldots, m\}$:

$$w_{ik} = Q_i^{(t)}(z^{(i)} = k) = p(z^{(i)} = k | x^{(i)}; \theta^{(t)}) = \frac{p(x^{(i)} | z^{(i)} = k; \theta^{(t)}) p(z^{(i)} = k; \theta^{(t)})}{\sum_j p(x^{(i)} | z^{(i)} = j; \theta^{(t)}) p(z^{(i)} = j; \theta^{(t)})}$$

which gives

$$w_{ik} = \frac{\mathcal{N}(x^{(i)}|\mu_k, \Sigma_k)\phi_k}{\sum_j \mathcal{N}(x^{(i)}|\mu_j, \Sigma_j)\phi_j}.$$

(c) In the M-step we need to re-estimate μ, Σ, ϕ. Unlike the unsupervised EM algorithm which we studies in class, we now need to consider the contribution from the complete dataset $\tilde{x}, \tilde{z}$. To be specific, we want to maximize

$$\sum_{i=1}^{m}\left(\sum_{j=1}^{k} w_{ij} \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{w_{ij}}\right) + \alpha\left(\sum_{i=1}^{\tilde{m}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta^{(t)})\right)$$

with respect to θ.

The following parameter update rules are derived by differentiating the above expression w.r.t each parameter, setting the result to 0, and solving the obtained equation:

$$\phi_j = \frac{\sum_{i=1}^{m} w_{ij} + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\}}{m + \alpha\tilde{m}}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_{ij} x^{(i)} + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\}\tilde{x}^{(i)}}{\sum_{i=1}^{m} w_{ij} + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\}}$$

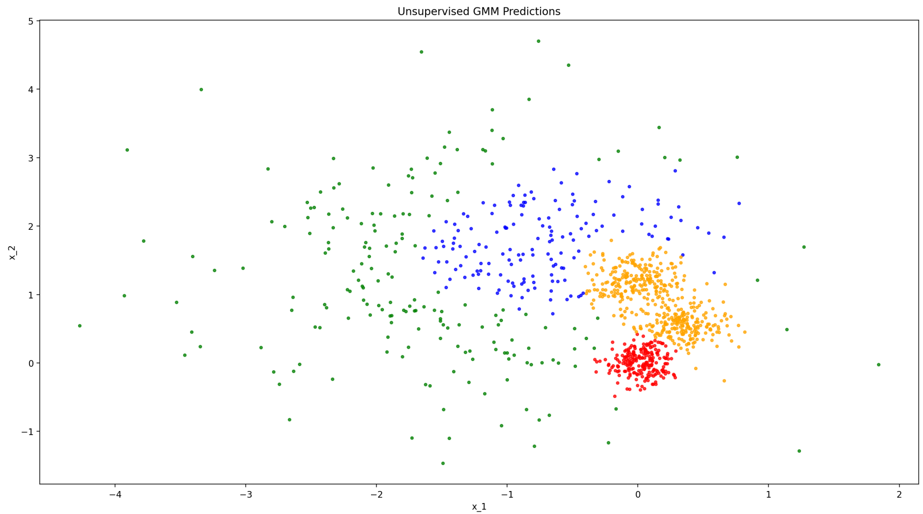
$$\Sigma_j = \frac{\sum_{i=1}^{m} w_{ij}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\}(\tilde{x}^{(i)} - \mu_j)(\tilde{x}^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_{ij} + \alpha \sum_{i=1}^{\tilde{m}} 1\{\tilde{z}^{(i)} = j\}}$$

It is worth emphasizing that the results do not constitute a closed-form solution for the parameters of the mixture model because w_{ij} depend on those parameters in a complex way.

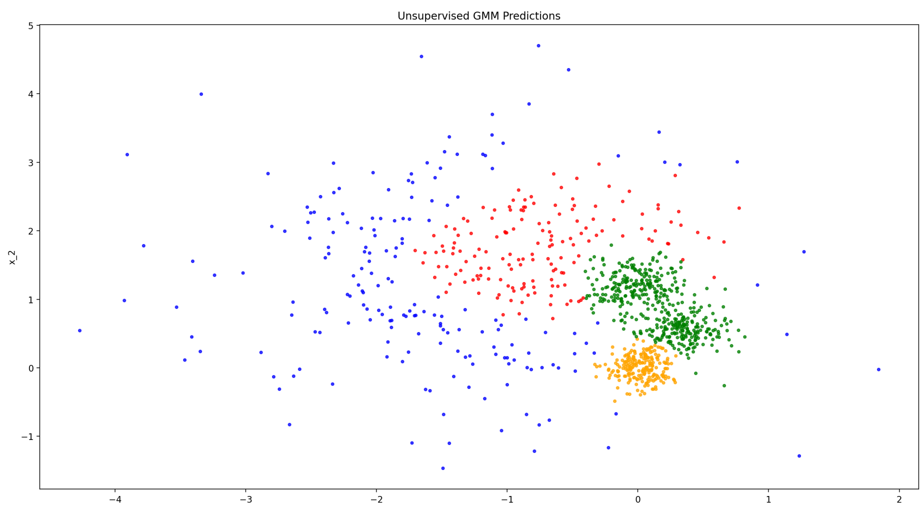
(d) Please see the code for a detailed implementation.

(e) Please see the code for a detailed implementation.

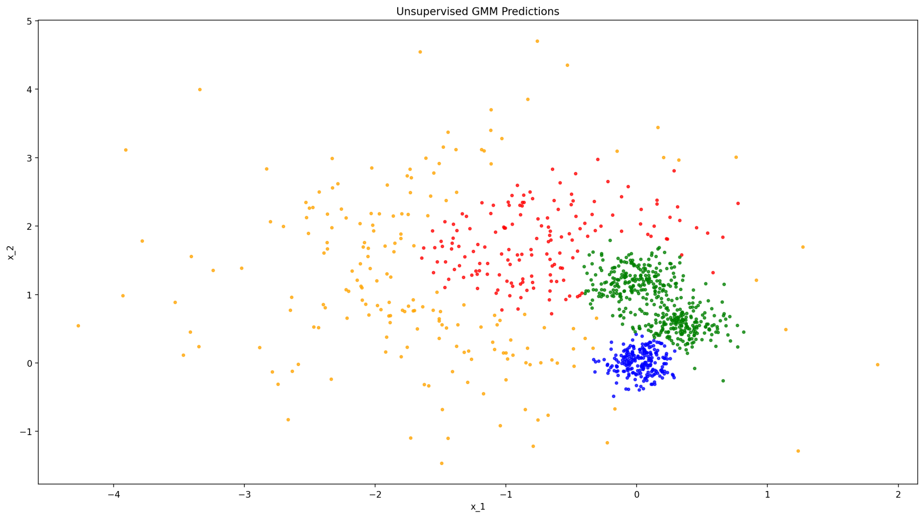
(f) The outputs of unsupervised EM with three random initialization of parameters are plotted below:



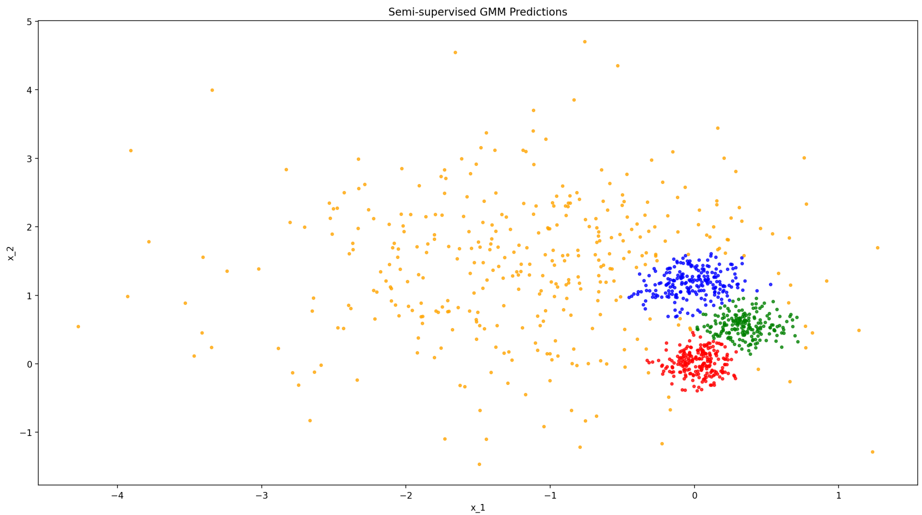
Unsupervised GMM Predictions
x_2
x_1

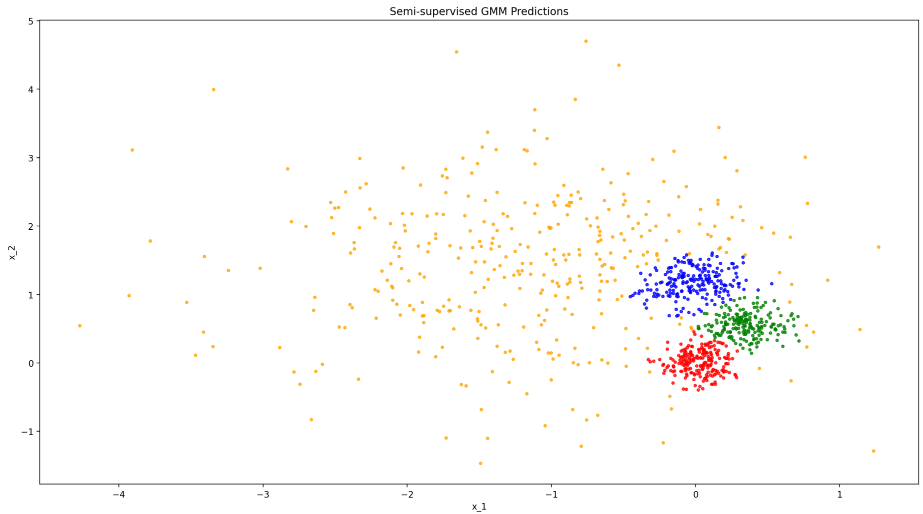


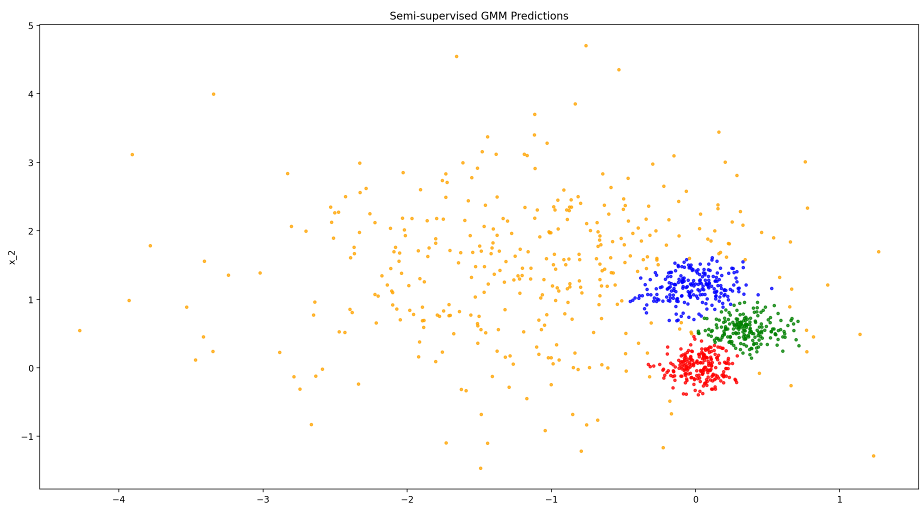
Unsupervised GMM Predictions
x_2
x_1



The outputs of semi-supervised EM with three random initialization of parameters are plotted below:







(i) Training takes less iterations to converge for semi-supervised EM than unsupervised EM.

(ii) The semi-supervised EM tends to be more stable than the unsupervised EM.

(iii) The semi-supervised EM tends to have better overall quality of assignments than the unsupervised EM. I.e., the output of the semi-supervised EM is closer to a mixture of three low-variance Gaussian distribution and a fourth, high-variance Gaussian distribution.

Problem 5

(a) The write-up is included in `p05_kmeans.ipynb`.

(b) For each pixel, we now need to store a 4-bit value (to indicate one of the 16 colors), instead of a 24-bit color. In addition, we also need to store the 24-bit color representation of the 16 colors (which is ignorable for an image with many pixels). Therefore, the compression factor is approximately $24/4 = 6$.