CS 229, Problem Set #3

Summer 2023

Reyansh Gupta - reyansh - 06776137

Q1 - A

Uncompressed image

Compressed image

Q1 - B

Each pixel in the original picture is encoded using 24 bits. If we apply compression with 16 colors, each pixel now only uses 4 bits (because $2^4 = 16$).
As a result, the compression factor becomes $\frac{4}{24}$, which approximates to $\frac{1}{6}$.

Q2 - A

Given that:

$l_{semi-sup}(\theta) = l_{unsup}(\theta) + \alpha \cdot l_{sup}(\theta)$

So, for $(\theta+1)$:

$$l_{semi-sup}(\theta^{(t+1)}) = l_{unsup}(\theta^{(t+1)}) + \alpha\, l_{sup}(\theta^{(t+1)})$$

Jensen's inequality says that:

$$E[f(x)] \geq f(E[x])$$

$$\therefore l_{unsup}(\theta^{(t+1)}) + \alpha\, l_{sup}(\theta^{(t+1)}) \geq \sum_{i=1}^{n} ELBO(x^{(i)}, Q_i^{(t)}, \theta^{(t+1)}) + \alpha\, l_{sup}(\theta^{(t+1)})$$

$$\geq \sum_{i=1}^{n} ELBO(x^{(i)}, Q_i^{(t)}, \theta^{(t)}) + \alpha\, l_{sup}(\theta^{(t+1)})$$

From class notes → $\left[\theta^{(t+1)} \text{ chosen explicitly to be } \arg\max_{\theta} \sum_{i=1}^{n} ELBO(x^{(i)}; Q_i^{(t)}, \theta)\right]$

$$\geq \sum_{i=1}^{n} ELBO(x^{(i)}, Q_i^{(t)}, \theta^{(t)}) + \alpha\, l_{sup}(\theta^{(t)}) \quad [\text{from the E Step}]$$

$$= l_{unsup}(\theta^{(t)}) + \alpha\, l_{sup}(\theta^{(t)}) \quad [\text{from the M Step}]$$

$$= l_{semi-sup}(\theta^{t})$$

$\therefore$ Proved that $l_{semi-sup}(\theta^{(t+1)}) \geq l_{semi-sup}(\theta^{(t)})$ and thus, with every iteration, the algorithm will converge monotonically.

Q2 - B

In the E Step → we need to re-estimate all the latent variables $z^{(i)}$'s, for all $i = 1 \dots n$

We set:

$$w_j^{(i)} = Q_i(z^{(i)} = j)$$

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

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

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

Q2 - C

In the M step, we re-estimate the model parameters (μ, Σ, ϕ) to maximise the log likelihood function $\rightarrow$

$$\sum_{i=1}^{\hat{n}} \sum_{j=1}^{k} w_j^{(i)} \log \frac{p(x^{(i)}, z^{(i)}=j;\theta)}{w_j^{(i)}} + \alpha \sum_{i=1}^{\tilde{n}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)};\theta)$$

After removing the constant terms, we get $\rightarrow$

$$\sum_{i=1}^{n} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)}=j;\theta) + \alpha \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{k} 1\{z^{(i)}=j\} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)};\theta)$$

If we add the labeled dataset to the unlabeled dataset, we get the whole training set of $(n+\tilde{n})$ examples where $n \rightarrow$ unlabeled, $\tilde{n} \rightarrow$ labaled.

For labeled examples $w_j^{(i)} = \alpha\, 1\{z^{(i)}=j\}$, $i \in \{n, \dots, n+\tilde{n}\}$.
Now the objective can be written as:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)}=j, \theta) + \sum_{i=n+1}^{n+\tilde{n}} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)}=j;\theta)$$

$$= \sum_{i=1}^{n+\tilde{n}} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)}=j;\theta)$$

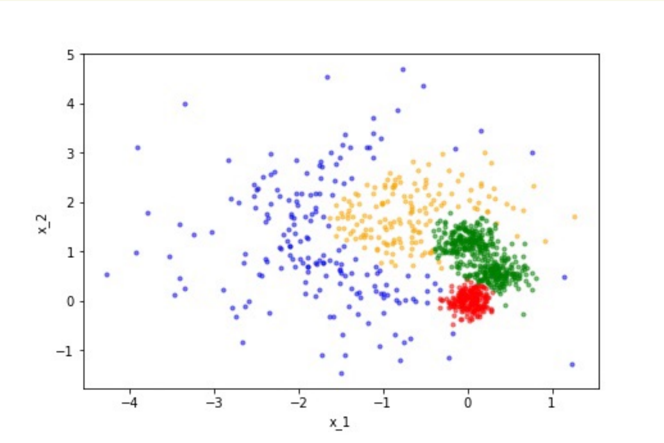
This is what we update in the classical Gaum model, hence, we can derive the update rule as follows:

$$\phi_j = \frac{1}{n+\alpha\tilde{n}} \sum_{i=1}^{n+\tilde{n}} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)}} \qquad \Sigma_j = \frac{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)} (x^{(i)}-\mu_j)(x^{(i)}-\mu_j)^T}{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)}}$$

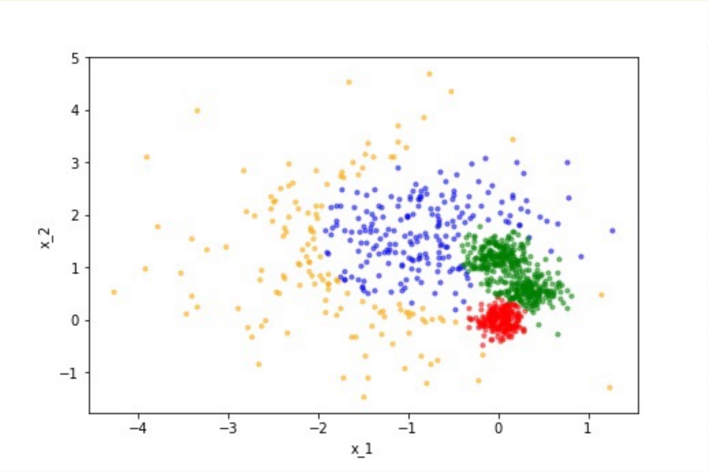
Q2 - D

Unsupervised EM



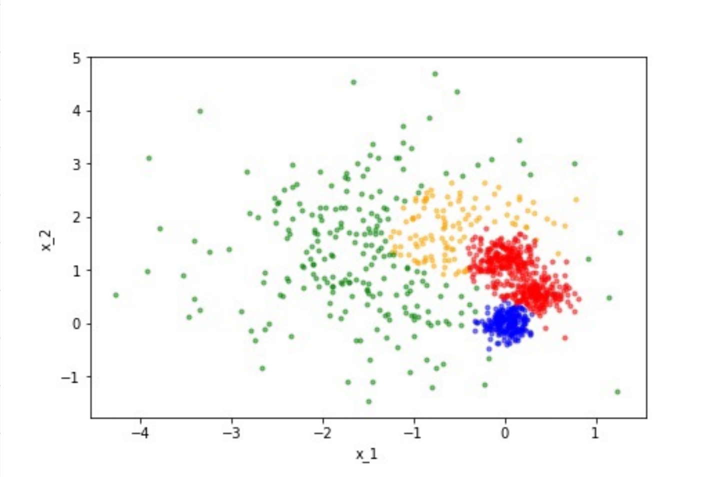
a) Converged after 145 iterations

loss = −1801.75



b) Converged after 128 iterations

loss = -1801.82

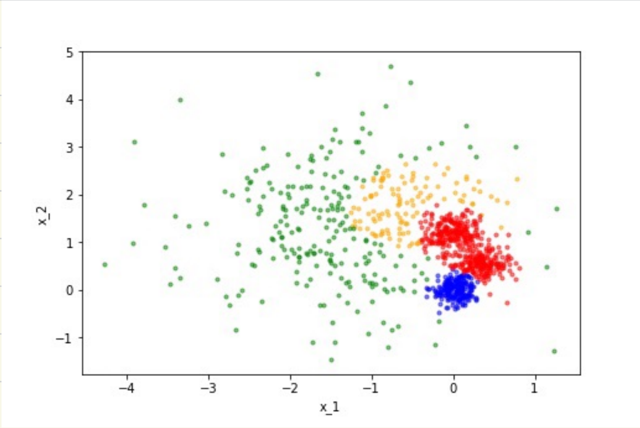


c) Converged after 123 iterations

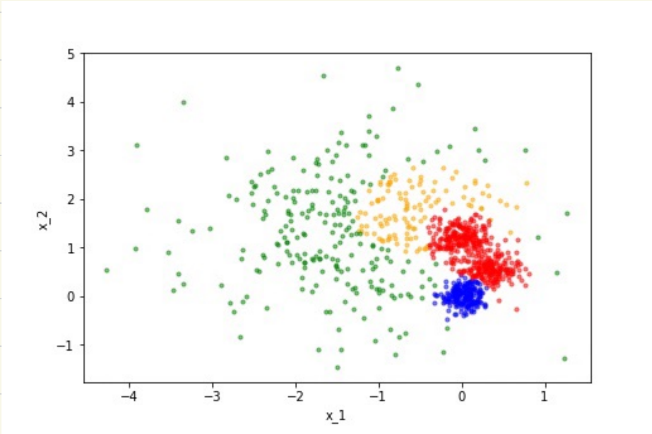
loss = -1801.74

Q2 - E

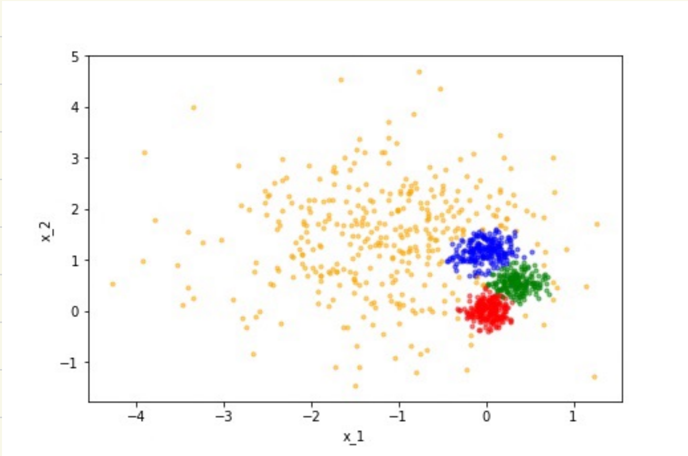
Semi Supervised EM



a) Converged after 25 iterations
loss = -1646.22



b) Converged after 29 iterations
loss = -1646.22



c) Converged after 22 iterations
loss = -1646.22

Q2 - F

i) The semi supervised EM took significantly less iterations to converge.

$\frac{145}{25} \approx 6$ times faster

ii) The clusters in the semi supervised EM are much more stable for random initializations

iii) This dataset has 3 Gaussian distributions with a low variance and a fourth gaussian distribution that overlaps the first three.

Regardless of this, semi-supervised EM could take advantage of this extra information about the cluster identities of known examples and could cluster them more accurately.

Q3

We need to prove that

$$\arg\min_{u \cdot u^T u = 1} \sum_{i=1}^{n} \|x^{(i)} - f_u(x^{(i)})\|_2^2 = \text{the first principle component}$$

(direction of most variance)

$$= \arg\max_{u:\|u\|=1} \left(\sum_{i=1}^{n} \|x^{(i)}\|^2 - \sum_{i=1}^{n} \|x^{(i)} - f_u(x^{(i)})\|_2^2 \right)$$

(maximising squared difference subtracted from constant)

$$= \arg\max_{u:\|u\|=1} \sum_{i=1}^{n} \left(\|x^{(i)}\|^2 - \|x^{(i)} - f_u(x^{(i)})\|_2^2 \right)$$

$$= \arg\max_{u:\|u\|=1} \frac{1}{n} \sum_{i=1}^{n} \|f_u(x^{(i)})\|_2^2$$

$\therefore$ minimiying our objective here is equivalent to maximising the variance of the projections along u: $\frac{1}{n} \sum_{i=1}^{n} \|f_u(x^{(i)})\|_2^2$ and that is the first principal component of PCA which will satisfy this

Q4 - A

In ICA, we maximise likelihood as a function of W →

$$\ell(W) = \sum_{i=1}^{n} \log p_x(x^{(i)})$$

$$= \sum_{i=1}^{n} \log (p_s(Wx^{(i)}) |W|)$$

$$= \sum_{i=1}^{n} \log \left(\frac{1}{(2\pi)^{d/2}} \exp\left\{ -\frac{1}{2} (Wx^{(i)})^T (Wx^{(i)}) \right\} |W| \right)$$

Applying log function inside →

$$= \sum_{i=1}^{n} \left(-\frac{d}{2} \log(2\pi) - \frac{1}{2} x^{(i)T} W^T W x^{(i)} + \log |W| \right)$$

∴ We got our objective function. To maximise this, we compute its gradient and set it equal to 0.

∴ The eqⁿ becomes:

$$\nabla_W \ell(W) = \sum_{i=1}^{n} \left(-\frac{1}{2} \nabla_W x^{(i)T} W^T W x^{(i)} + \nabla_W \log |W| \right)$$

$$= \sum_{i=1}^{n} (-Wx^{(i)} x^{(i)T} + (W^{-1})^T)$$

$$= -W \left(\sum_{i=1}^{n} x^{(i)} x^{(i)T} \right) + n(W^{-1})^T$$

$$= -W \left(\sum_{i=1}^{n} x^{(i)} x^{(i)T} \right) + n(W^{-1})^T$$

$$= -WX^TX + n(W^{-1})^T$$

$$\Rightarrow -WX^TX + n(W^{-1})^T = 0$$

Which means

$$W^TW = \left(\frac{1}{n} X^TX \right)^{-1}, \text{ assuming RHS is invertible.}$$

Let $Y = \left(\frac{1}{n} X^T X^{-1} \right)^{-1}$, then Y is positive semi definite

We can decompose W as $W = U \Sigma V^T$ where U, V are orthogonal and Σ is a diagonal.

∴ The final result becomes.

$$W^TW = (V\Sigma U^T)(U\Sigma V^T) = V\Sigma (U^TU) \Sigma V^T = V\Sigma^2 V^T = Y.$$

Thus, we can compute the eigen decomposition of Y to get Σ^*, V^* and can use an arbitary U to reconstruct $W = U\Sigma^* V^*$. This U can't be determined from data X which leads to ambiguity.

$\therefore$ The ICA fails at recovering the original sources.

Q4 - B

for any example $x^{(i)}$, we have:

$$l_i(W) = \sum_{j=1}^{d} \log p_s(w_j^T x^{(i)}) + \log |W|$$

$$= \sum_{j=1}^{d} \log\left(\frac{1}{2} \exp\left(-|w_j^T x^{(i)}|\right)\right) + \log |W|$$

$$= -d \log(2) - \sum_{j=1}^{d} |w_j^T x^{(i)}| + \log |W|$$

Taking it's gradient →

$$\nabla_W l_i(W) = -\sum_{j=1}^{d} \nabla_W |w_j^T x^{(i)}| + \nabla_W \log |W|$$

$$= \sum_{j=1}^{d} \text{sign}(w_j^T x^{(i)}) \left[x^{(i)T} \begin{pmatrix} 0 \\ j^{th}\ \text{row} \\ 0 \end{pmatrix} \right] + (W^{-1})^T$$

$$= -\begin{bmatrix} \text{sign}(w_1^T x^{(i)}) \\ \vdots \\ \text{sign}(w_d^T x^{(i)}) \end{bmatrix} x^{(i)T} + (W^{-1})^T$$

$\therefore$ The update rule becomes →

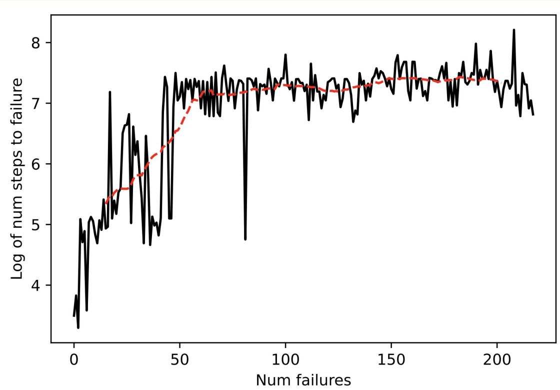
$$W := W + \alpha \left(-\begin{bmatrix} \text{sign}(w_1^T x^{(i)}) \\ \vdots \\ \text{sign}(w_d^T x^{(i)}) \end{bmatrix} x^{(i)T} + (W^{-1})^T \right)$$

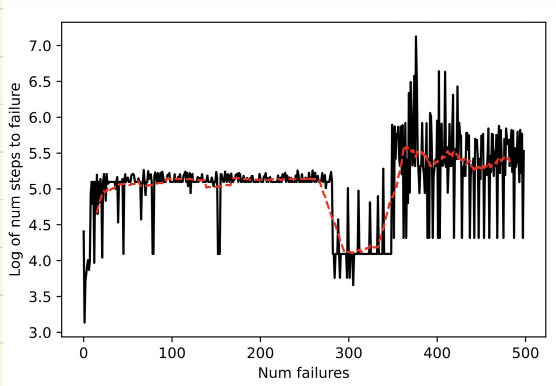
Q4 - C

W Matrix:

```
[[ 52.8352532   16.79619701  19.94171825 -10.19846303 -20.89757762]
 [ -9.9292747   -0.97875614  -4.67786427   8.04377382   1.7865852 ]
 [  8.31096507  -7.47675728  19.31500349  15.17429591 -14.32612384]
 [-14.66742843 -26.64517989   2.44081559  21.38210464  -8.4207738 ]
 [ -0.26929644  18.37414675   9.31198649   9.10287095  30.59463426]]
```

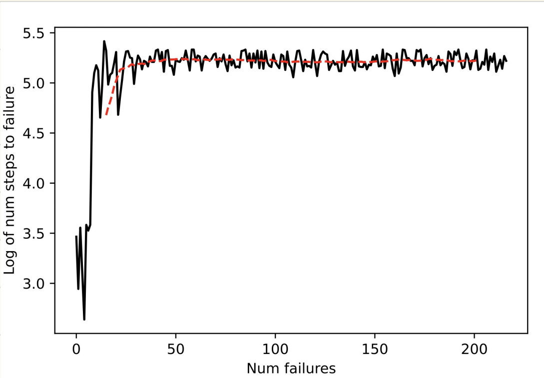
Q5

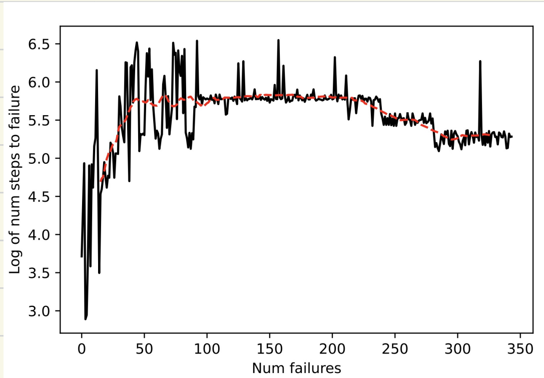




Seed 0 → 219 iterations to converge

Seed 1 → 500 + iterations (not converged)





Seed 2 → 218 iterations to converge

Seed 3 → 345 iterations to convergence

From these learning curves, we can see that the curve is affected with each seed.

With different seeds, different policies are made and our algoritum tries its best in a greedy approach to converge. Hence, we get a different learning curve for each seed