E0 270 Machine Learning
Assignment - 3

Sruthi Gorantla
M. Tech. CSA
SR No. 15190

April 12, 2018

1. **Kernel K-Means**
   (a) (5 points) k-means
   Solution:

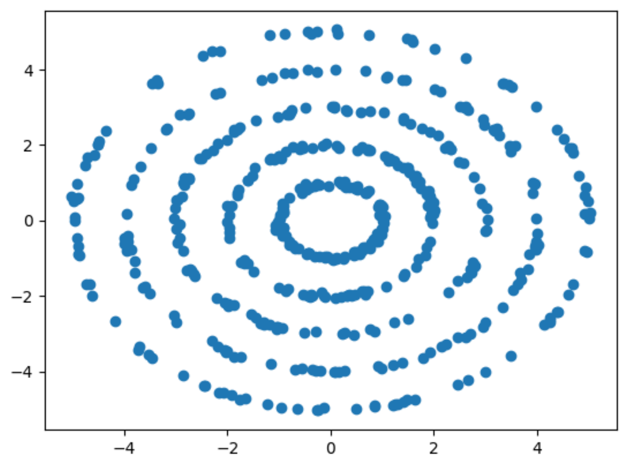

Figure 1: Data for k-means clustering plot 1

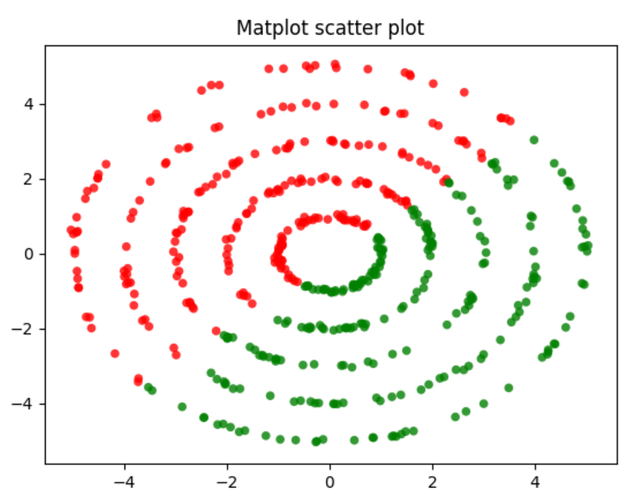

Figure 2: regular k-means clustering for k = 2 plot 1

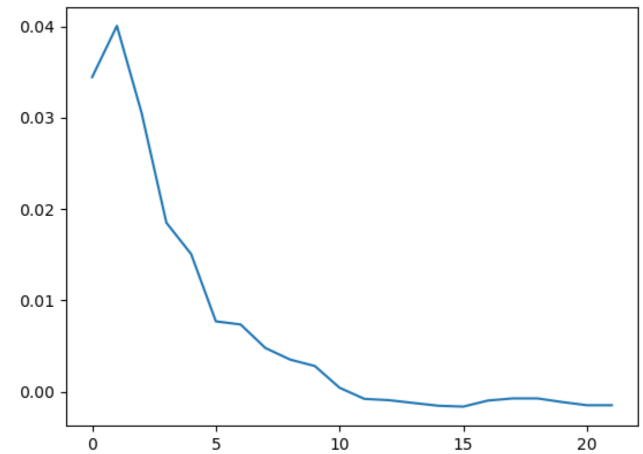

Figure 3: RAND for k-means clustering

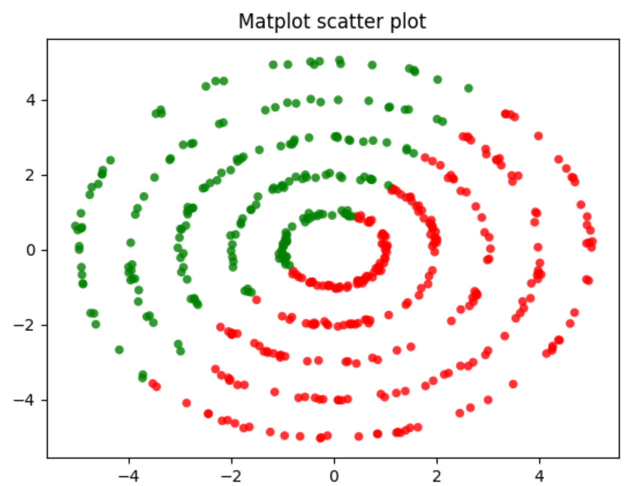

Figure 4: regular k-means clustering for k = 2 plot 2

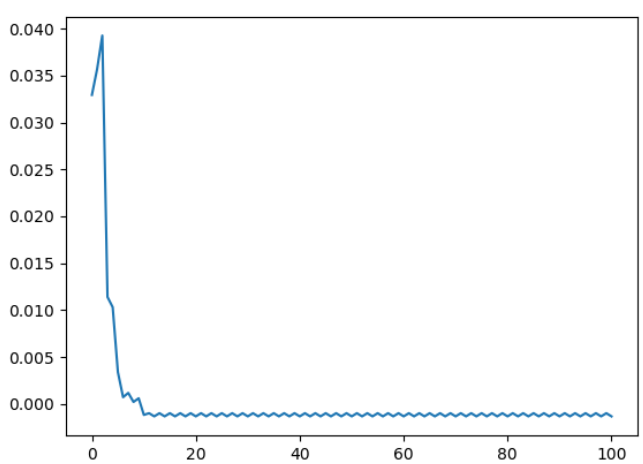

Figure 5: RAND for k-means clustering plot 2

We choose k = 2 for the given data and clearly k-means is not able to cluster the data properly. The reason for this is that normal k-means can only classify a linearly separable data. The given data is not linearly separable. If projected into higher dimensions where the data is linearly separable, then k-means clustering algorithm should be able to cluster the data.

(b) (5 points) kernel k-means

Solution:

In kernel k-means we replace the euclidean distance by kernelized versions. For e.g., $d(x_n, \mu_k) = \left\|\phi(x_n) - \phi(\mu_k)\right\|$ by

$$\begin{aligned} \left\|\phi(x_n) - \phi(\mu_k)\right\|^2 &= \left\|\phi(x_n)\right\|^2 + \left\|\phi(\mu_k)\right\|^2 - 2\phi(x_n)^\top \phi(\mu_k) \\ &= k(x_n, x_n) + k(\mu_k, \mu_k) - 2k(x_n, \mu_k) \end{aligned} \tag{1}$$

Here $k(.,.)$ denotes the kernel function and ϕ is its feature map. **NOTE:** When computing $k(\mu_k, \mu_k)$ and $k(x_n, \mu_k)$, remember that $\phi(\mu_k)$ is the average of ϕ's the data points assigned to the cluster k. The given matrix is the similarity measure, the distance between two points is inversely proportional to the similarity measure. Since we only consider the argmin of the distance values from a point to all the cluster centroids, we need not worry about the exact proportionality constants. Therefore:

$$d(x_a, x_b) = c\frac{1}{H_{a,b}} \tag{2}$$

$$cluster[x_a] = \underset{k=1\ldots K}{\text{argmin}}(d(x_a, \mu_k))$$

Now we run the regular k-means algorithm with this distance measure.

(c) (5 points) **Solution:**

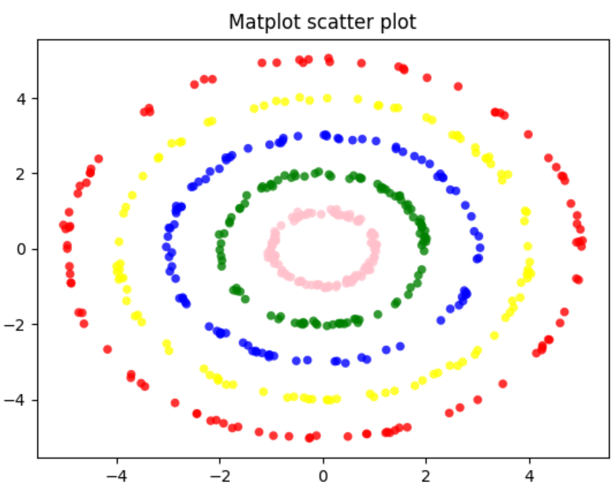

Figure 6: kernel k-means clustering for k = 5 plot 1

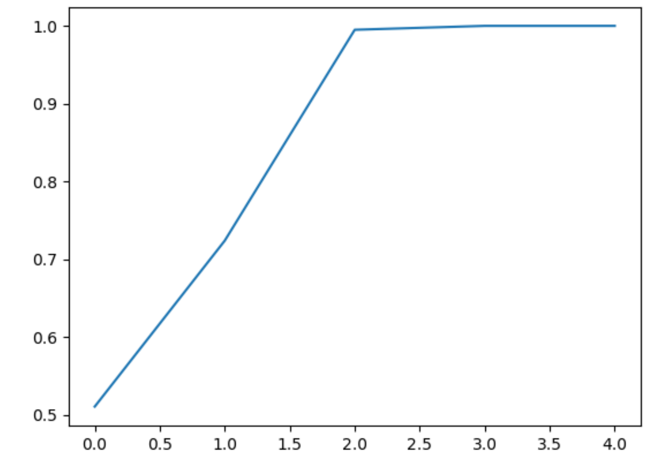

Figure 7: RAND for kernel k-means clustering plot 1

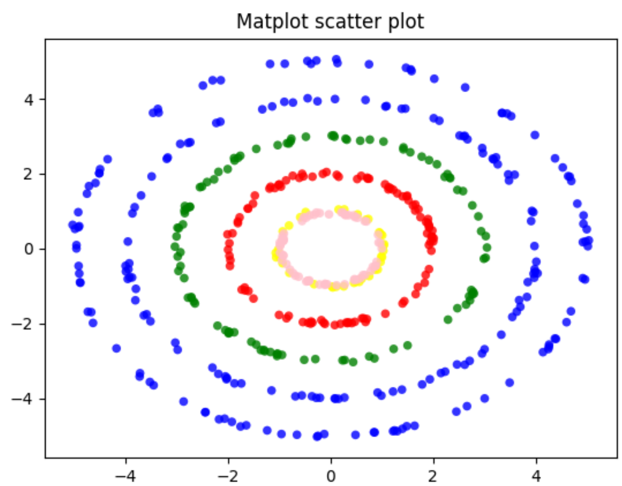

Figure 8: kernel k-means clustering for k = 5 plot 2

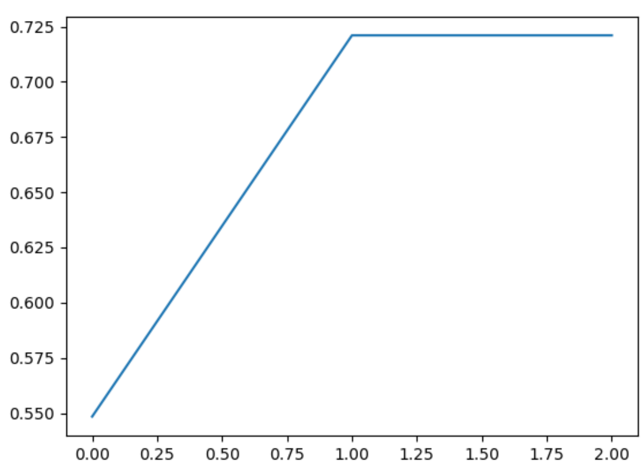

Figure 9: RAND for kernel k-means clustering plot 2

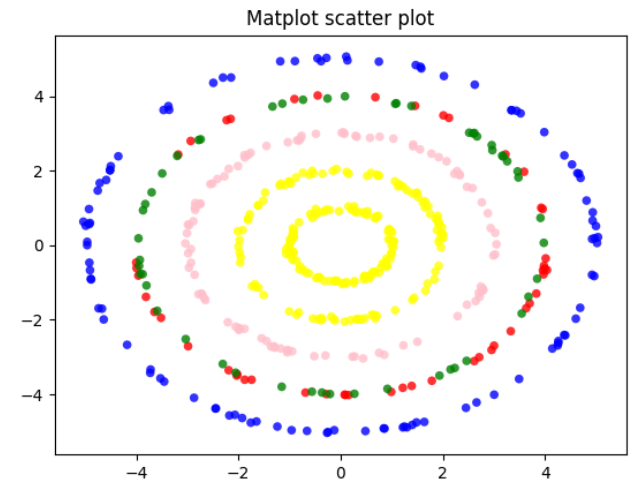

Figure 10: kernel k-means clustering for k = 5 plot 3

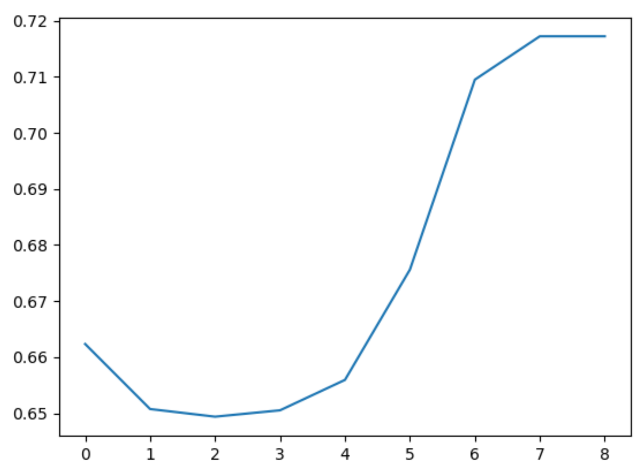

Figure 11: RAND for kernel k-means clustering plot 3

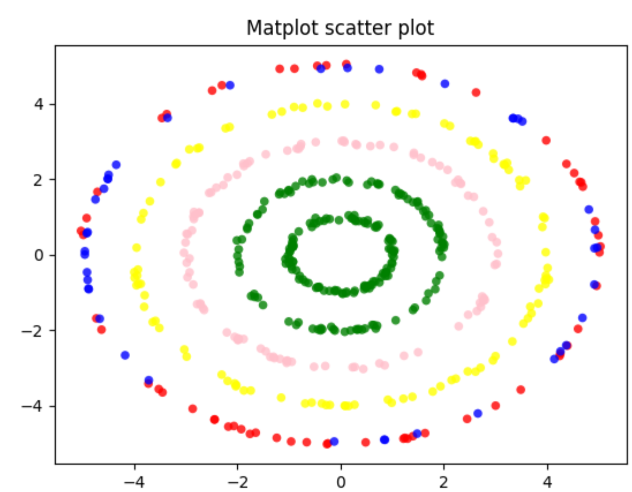

Figure 12: kernel k-means clustering for k = 5 plot 4

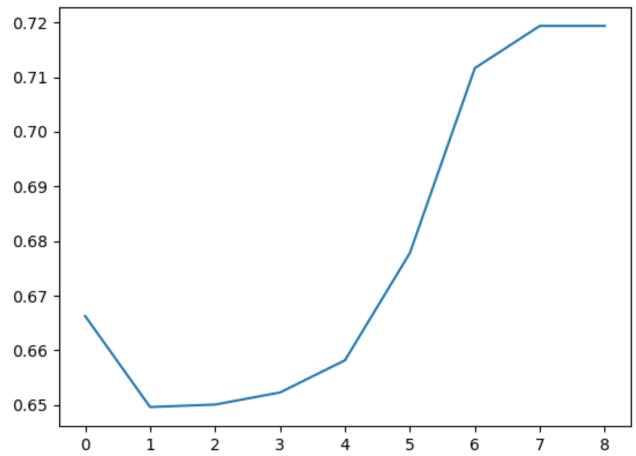

Figure 13: RAND for kernel k-means clustering plot 4

2. **Restricted Boltzmann Machines (RBM)**

   (a) (3 points) Derivation of likelihood and gradient updates

   Solution:

   An RBM is a Boltzmann Machine with a bi-partite graph of n visible and m hidden units, i.e., no connections between visible units and between hidden units. The energy has parameters, $\theta \in \Theta := \{w_{ij}, b_j, c_i : 1 \leq j \leq n, 1 \leq i \leq m\}$

$$E(v,h) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}h_i v_j - \sum_{j=1}^{n} b_j v_j - \sum_{i=1}^{m} c_i h_i \tag{3}$$

   As RBM can learn a distribution p to approximate q on $D \subset S = \{0,1\}^n$. An asymmetric measure of difference between q and p is given by KL divergence.

$$KL(q||p) = \sum_{x \in S} q(x) \log \frac{q(x)}{p(x)} = \sum_{x \in S} q(x) \log q(x) - \sum_{x \in S} q(x) \log p(x)$$

   $KL(q||p)$ is non negative and is zero iff $p = q$. Minimizing $KL(q||p)$ corresponds to maximixing the likelihood of p for training items. Thus learning aims to determining all parameters $\theta \in \Theta$ to maximize likelihood w.r.t. D defined by:

$$L(\theta|D) = \prod_{k=1}^{l} p(x_k|\theta)$$

$$\log L(\theta|D) = \log \prod_{k=1}^{l} p(x_k|\theta) = \sum_{k=1}^{l} \log p(x_k|\theta)$$

   Since this doesn't have a closed form solution, we use gradient ascent on the parameters to maximize this log likelihood.

$$\begin{aligned}\theta_i^{(t+1)} &= \theta_i^{(t)} + \alpha \frac{\partial f}{\partial \theta_i}(\theta_i^{(t)}) - \lambda\theta^{(t)} + \nu\Delta\theta^{(t-1)} \\ &= \theta_i^{(t)} + \alpha \frac{\partial}{\partial \theta_i}\Big(\sum_{k=1}^{l} \log p(x_k|\theta^{(t)})\Big) - \lambda\theta^{(t)} + \nu\Delta\theta^{(t-1)}\end{aligned}$$

   where $-\lambda\theta^{(t)}$ is the decay weight and $\nu\Delta\theta^{(t-1)}$ is the momentum. We know that

$$p(v,h) = \frac{e^{-E(v,h)}}{Z} withZ = \sum_{v \in \{0,1\}^n} \sum_{h \in \{0,1\}^m} e^{-E(v,h)}$$

Since the only connections are between a visible and a hidden unit, the conditional probability distributions are:

$$p(h|v) = \prod_{i=1}^{m} p(h_i|v)$$
$$p(v|h) = \prod_{i=1}^{n} p(v_i|h)$$

Hence now we can find $p(v)$ by finding the marginal distribution:

$$p(v) = \sum_h p(v,h) = \frac{1}{Z}\sum_h e^{-E(v,h)}$$

Therefore the log likelihood is computed as

$$\begin{aligned}\log p(x|\theta) &= \log \frac{1}{Z}\sum_h e^{-E(v,h)} \\ &= \log \sum_h e^{-E(v,h)} - \log \sum_{x,h} e^{-E(v,h)}\end{aligned}$$

To compute the derivative of the log likelihood we need the following:

$$p(h|v) = \frac{p(v|h)}{p(v)} = \frac{\frac{1}{Z}e^{-E(v,h)}}{\frac{1}{Z}\sum_h e^{-E(v,h)}}$$

The derivative is computed as follows:

$$\begin{aligned}\frac{\partial}{\partial\theta}(\log p(v|\theta)) &= \frac{\partial}{\partial\theta}(\log \sum_h e^{-E(v,h)}) - \frac{\partial}{\partial\theta}(\log \sum_{v,h} e^{-E(v,h)}) \\ &= -\sum_h p(h|v)\frac{\partial E(v,h)}{\partial\theta} + \sum_{v,h} p(v,h)\frac{\partial E(v,h)}{\partial\theta}\end{aligned}$$

The first term can be easily computed as we have $E(v,h) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}h_iv_j - \sum_{j=1}^{n} b_jv_j - \sum_{i=1}^{m} c_ih_i$
Taking average of the log likelihood gradient of all training vectors for θ we have:

$$\frac{1}{l}\sum_{x\in D}\frac{\partial \log p(v|w_{ij})}{w_{ij}} =< h_iv_j >_{data} - < h_iv_j >_{model}$$

Thus by independence of visible units we have:

$$p(v_k = 1|h) = \sigma\Big(\sum_{i=1}^{m} w_{ik}h_i + b_k\Big)$$

By symmetry we have

$$p(h_k = 1|v) = \sigma\Big(\sum_{j=1}^{n} w_{kj}v_j + c_k\Big)$$

We can do Gibbs sampling in two steps in each stage as follows:
(i) Sample h based on $p(h|v) = \prod_{i=1}^{m} p(h_i|v)$
(ii) Sample v based on $p(v|h) = \prod_{i=1}^{n} p(v_i|h)$

Contrastive Divergence (CD-k) is an algorithm to approximate MCMC for an RBM. We simply run Gibbs block sampling for k steps:

- Start with a training vector $v^{(0)}$ and at step $0 \leq x \leq k-1$.
- Sample $h^{(s)}$ from $p(h|v^{(s)}$.
- Sample $v^{(s)}$ from $p(v|h^{(s)}$.
- Replace each term with $-p(h_i = 1|v^{(k)})v_j^{(k)}$.

we usually take $k = 1$ which is CD-1 algorithm.

(b) (7 points) **Solution:**

Seed for W, b, c = 15190 Seed for Visible and hidden units = 1049*16 The final sampled visible units are: [1. 0. 0. 0. 1. 1. 0. 1. 0. 1.]

3. **EM algorithm for a mixture of Bernoullis**

(a) (10 points) Derive the steps of EM algorithm

Solution:

Consider a vector of binary random variables $x \in \{0,1\}^M$ such that each x_i is governed by a Bernoulli distribution with parameter μ_i. Hence

$$p(x|\mu) = \prod_{i=1}^{M} \mu_i^{x_i}(1-\mu_i)^{(1-x_i)} \tag{4}$$

where $x = \{x_i, \ldots, x_M\}^\top$ and $\mu = \{\mu_i, \ldots, \mu_M\}^\top$.

For a mixture of K such Bernoullis we have

$$p(x|\mu,\pi) = \sum_{k=1}^{K} \pi_k p(x|\mu_k) \tag{5}$$

where $\mu = \{\mu_i, \ldots, \mu_M\}$ with $\mu_i = \{\mu_{i1}, \ldots, \mu_{iM}\}^\top$ and $\pi = \{\pi_i, \ldots, \pi_K\}^\top$, with $\pi_i \geq 0$ and $\Sigma_i \pi_i = 1$.

Suppose we have the input examples as $X = \{x^{(i)}\}_{i=1\ldots N}$, the log likelihood of the data X, $\log p(x^{(i)}|\mu,\pi)$ is given by,

$$\log L(\mu,\pi) = \sum_{i=1}^{N} \log p(x^{(i)}|\mu,\pi) \tag{6}$$

Now let $z^{(i)} \in \{0,1\}^K$ be an indicator vector such that $z_k^{(i)} = 1$ if $x^{(i)}$ was drawn from Bernoulli $\mu^{(k)}$, otherwise 0. Let $Z = \{z^{(i)}_{i=1\ldots N}\}$. Then,

$$p(z^{(i)}|\pi) = \prod_{k=1}^{K} \pi_k^{z_k^{(i)}} \tag{7}$$

$$p(x^{(i)}|z^{(i)},\mu,\pi) = \prod_{k=1}^{K} p(x^{(i)}|\mu^k)^{z_k^{(i)}} \tag{8}$$

The log likelihood of the data and the latent variables is given by

$$\begin{aligned} p(Z,X|\mu,\pi) &= \prod_{i=1}^{N} p(x^{(i)}, z^{(i)}|\pi,\mu) = \prod_{i=1}^{N} p(x^{(i)}|z^{(i)},\pi,\mu)p(z^{(i)}|\pi) \\ &= \prod_{i=1}^{N} \left[\prod_{k=1}^{K} p(x^{(i)}|\mu^k)^{z_k^{(i)}}\right]\left[\prod_{k=1}^{K} \pi_k^{z_k^{(i)}}\right] \end{aligned} \tag{9}$$

Let $\eta(z_k^{(i)}) = E[z_k^{(i)}|x^{(i)},\pi,\mu]$. Then,

$$\begin{aligned} \eta(z_k^{(i)}) &= E[z_k^{(i)}|x^{(i)},\pi,\mu] \\ &= p(z_k^{(i)} = 1|x^{(i)},\pi,\mu) \\ &= \frac{p(x^{(i)}|z_k^{(i)} = 1,\pi,\mu)p(z_k^{(i)} = 1|\pi,\mu)}{\sum_{k'} p(x^{(i)}|z_{k'}^{(i)} = 1,\pi,\mu)p(z_{k'}^{(i)} = 1|\pi,\mu)} \\ &= \frac{\pi_k \prod_{m=1}^{M} (\mu_m^{(k)})^{x_m^{(i)}}(1-\mu_m^{(k)})^{1-x_m^{(i)}}}{\sum_{k'} \pi_{k'} \prod_{m=1}^{M} (\mu_m^{(k')})^{x_m^{(i)}}(1-\mu_m^{(k')})^{1-x_m^{(i)}}} \end{aligned} \tag{10}$$

Now we compute log likelihood

$$
\begin{aligned}
\log p(Z, X|\pi, \mu) &= \sum_{i=1}^{N} \left[\sum_{k=1}^{K} z_k^{(i)} \log \left[p(x^{(i)}|\mu^{(k)} \right] \right] + \left[\sum_{k=1}^{K} z_k^{(i)} \log \pi_k \right] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left[(x^{(i)}|\mu^{(k)} + \log \pi_k \right] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left[\log \pi_k + \log \prod_{m=1}^{M} (\mu_m^{(k)})^{x_m^{(i)}} (1 - \mu_m^{(k)})^{1 - x_m^{(i)}} \right] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left[\log \pi_k + \sum_{m=1}^{M} x_m^{(i)} \log \mu_m^{(k)} + (1 - x_m^{(i)}) \log(1 - \mu_m^{(k)}) \right] \qquad (11)
\end{aligned}
$$

Taking the expected value and replacing $E[z_k^{(i)}] = \eta(z_k^{(i)})$ we get,

$$
E[\log p(Z, X|\tilde{\mu}, \tilde{\pi})|X, \pi, \mu] = \sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) \Big[\log \tilde{\pi}_k + \sum_{m=1}^{M} x_m^{(i)} \log \tilde{\mu}_m^{(k)} + (1 - x_m^{(i)}) \log(1 - \tilde{\mu}_m^{(k)}) \Big] \qquad (12)
$$

Where $\tilde{\pi}$ and $\tilde{\mu}$ are the new parameters that we like to maximize. This completes the **Expectation step** of the EM algorithm.

Now we need to maximize the equation (12) with respect to $\tilde{\pi}$ and $\tilde{\mu}$. By finding the derivative and equating it to 0 we get

$$
\frac{d}{d\mu_m^{(k)}} E[\log p(Z, X|\pi, \mu)] = \sum_{i=1}^{N} \eta(z_k^{(i)}) \left[\frac{x_m^{(i)}}{\mu_m^{(k)}} + \frac{1 - x_m^{(i)}}{1 - \mu_m^{(k)}} \right] = 0 \qquad (13)
$$

$$
\sum_{i=1}^{N} \eta(z_k^{(i)}) \Big[x_m^{(i)} (1 - \mu_m^{(k)}) + (1 - x_m^{(i)}) \mu_m^{(k)} \Big] = \sum_{i=1}^{N} \eta(z_k^{(i)}) \Big[-\mu_m^{(k)} + x_m^{(i)} \Big] = 0 \qquad (14)
$$

Solving for $\mu_m^{(k)}$ results in

$$
\mu_m^{(k)} = \frac{\sum_{i=1}^{N} \eta(z_k^{(i)}) x_m^{(i)}}{\sum_{i=1}^{N} \eta(z_k^{(i)})} \qquad (15)
$$

In equation (12), we only need to maximize $\sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) \log \pi_k$ since the rest of the term doesn't depend on π. We have a constraint on π that $\sum_{k=1}^{K} \pi_k = 1$. Let λ be the dual variable for this constraint. Then,

$$
L(\pi, \lambda) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) \log \pi_k + \lambda (\sum_{k=1}^{K} \pi_k - 1) \qquad (16)
$$

Taking the derivative w.r.t. π_k we get

$$
\frac{d}{d\pi_k} L(\pi, \lambda) = -\sum_{i=1}^{N} \frac{\eta(z_k^{(i)})}{\pi_k} + \lambda = 0 \qquad (17)
$$

Solving for π_k we get

$$
\pi_k = \frac{\sum_{i=1}^{N} \eta(z_k^{(i)})}{\lambda} = \frac{N_k}{\lambda} (say) \qquad (18)
$$

Now we solve for λ

$$
L(\lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) (\log N_k - \log \lambda) + (\sum_{k=1}^{K} N_k - \lambda) \qquad (19)
$$

Taking derivative w.r.t. λ and solving for λ we get

$$\lambda = \sum_{i=1}^{N}\sum_{k=1}^{K}\eta(z_k^{(i)}) = \sum_{k=1}^{K} N_k \tag{20}$$

This completes the **Maximization step** of the EM algorithm

(b) (7.5 points) **Solution:**

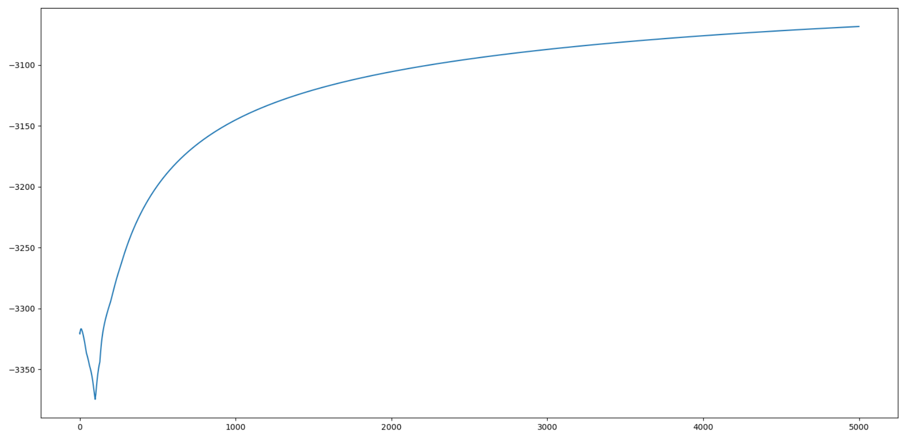

Figure 14: log-likelihood when k = 2, M = 5

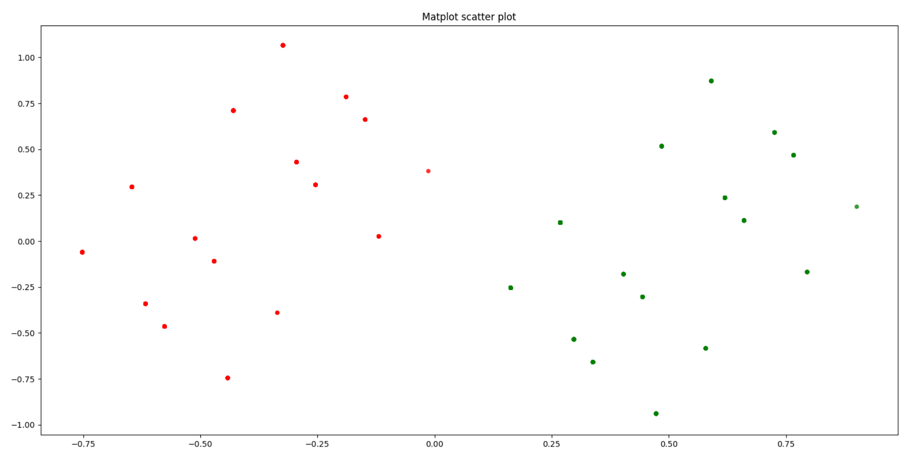

Figure 15: Clustered points obtained after performing a PCA on the points to reduce the dimension from 5 to 2

(c) (7.5 points) **Solution:**

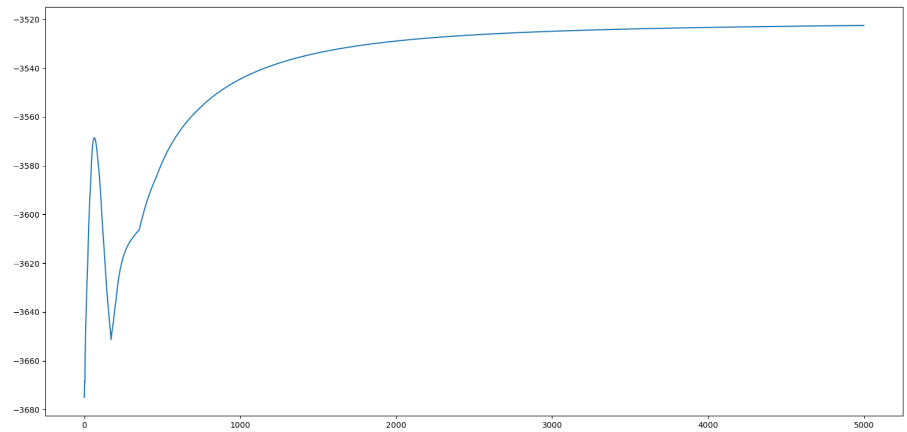

Figure 16: log-likelihood when k = 5, M = 5