CO495 Coursework EM model

Name: Fanbo Meng (fm2616) **Student Number:** 01366977

Exercise 1

(a) Co-variance matrix in this case has a form of .

$$\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}$$

Here we can see that if co-variance matrix is isotropic, which means that $\Sigma_{xy} = \Sigma_{yx} = 0$, indicating that for each dimension of sample points, the Gaussian Distribution is independent. This may show in the graph that the ellipse showing Gaussian Distribution should be vertical or horizontal to x-axis or y-axis. Clearly in this case, the do not, so the covariance matrix is anisotropic.

(b) Coding part is included in files submitted .
Explanation of using K-means to initialize:
EM algorithm is a slow-converged model, so if we randomly set the initial value, it will not converge when 10 iterations are over, so we need to set an initial value near the final answer, and the the EM algorithm will do a good job. Here I use K-means to set means, covariances and co-efficients of four Gaussian. In fact, in the original code, if we randomly generate co-variance matrix, when computing the probability density function, it may have trouble due to the random matrix may not be positive-definite.
Of course the idea of using K-means is not originally to ensure the covariance matrix positive definite. For algorithm in K-means, we have:

$$\gamma(Z_{nk}) = 1 \qquad if k = argminDistance(Xn, \mu)$$
$$\gamma(Z_{nk}) = 0 \qquad else$$

So K-means is a so-called hard-classification method, for a specific sample, we only consider it is definitely belongs to one specific class. But, EM algorithm gives us a distribution when giving expectation of z. In this scenario, we can see that K-means gives us an estimate of what the parameters like and EM modify the parameters in details.
P.S:The original code is running error in my computer shows that write() argument 1 must be Unicode, not str", so I make a modification in the output code when exporting a csv file. (See the corresponding code part.)
To show the final outcome clearly anyway, I post an output in this report.

```
{'covars_1': array([[ 1.8906865 ,  0.49429292],
                    [ 0.49429292,  0.99539668]]),
 'covars_3': array([[ 1.06063285, -0.49514593],
                    [-0.49514593,  1.85981389]]),
 'covars_2': array([[ 1.03323426, -0.00788144],
                    [-0.00788144,  0.97366275]]),
 'covars_4': array([[ 0.49183114,  0.00589224],
                    [ 0.00589224,  0.45876576]]),
 'mixCoeff_2': 0.2498260331296627,
 'mixCoeff_3': 0.24480085345568717,
 'mixCoeff_1': 0.25476283360525526,
 'mixCoeff_4': 0.25061027980939476,
 'means_4': array([ 1.98717629,  2.00624766]),
 'means_1': array([ 5.01663696,  5.02286884]),
 'means_2': array([ 7.97655199,  9.07474059]),
 'means_3': array([ 7.04324422,  1.89327452])}
```

Exercise 2 First in this case, we calculate the form of expectation:

$$\gamma(Z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma)}{\sum_{l=1}^{4} \pi_l N(x_n|\mu_l, \Sigma)} \tag{1}$$

Then we put expectation (1) back to the likelihood function, we get

$$lnP(x, z_n|\Theta) = \sum_{n=1}^{N} \sum_{k=1}^{4} \gamma(z_{nk}) ln[N(x_n|\mu_k, \Sigma)] + \sum_{n=1}^{N} \sum_{k=1}^{4} \gamma(z_{nk}) ln(\pi_k) = G(\Theta) \tag{2}$$

where

$$\Theta = \{\Sigma, \mu_1, \mu_2, \mu_3, \mu_4\}$$

$$N(x_n|\mu_k, |\Sigma) = [(2\pi)^{d/2}|\Sigma|^{-1/2}]^{-1} exp[-\frac{1}{2}(x_n - \mu_k)^T \Sigma^{-1}(x_n - \mu_k)]$$

s.t $\sum_{l=1}^{4} \pi_l = 1$

Then we compute the gradient for maximization:

$$\frac{\partial G(\Theta)}{\partial \Sigma} = 0$$

Then we solve this equation, and get:

$$\frac{\partial G}{\partial \Sigma} = \sum_{n=1}^{N} \sum_{k=1}^{4} \{\gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T)\} - 4N\Sigma = 0$$

Finally, we get:

$$\Sigma = \frac{\sum_{n=1}^{N} \sum_{k=1}^{4} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{4N}$$

Intuitively, when adding up the constrain on all Σ are equal, the optimized Σ will be the mean of four Gaussian Distribution's covariance matrix with a weight of z_{nk}