1 Written Problems

1. (1) $P(x_i | z_i = k, \mu_k) = \prod_{i=1}^{D} \mu_{kj}^{x_{ij}} (1-\mu_{kj})^{1-x_{ij}}$

$$L(\mu_k) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \ln P(x_i | z_i = k, \mu_k)$$

$$\frac{\partial L}{\partial \mu_{kj}} = \frac{\partial}{\partial \mu_{kj}} \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \left(x_{ij} \ln(\mu_{kj}) + (1-x_{ij}) \ln(1-\mu_{kj}) \right)$$

$$= \sum_{i=1}^{N} r_{ik} \cdot \left(\frac{x_{ij}}{\mu_{kj}} - \frac{1-x_{ij}}{1-\mu_{kj}} \right)$$

Let $\frac{\partial L}{\partial \mu_{kj}} = 0 \Rightarrow \sum_{i=1}^{N} r_{ik} \left(\frac{x_{ij} - x_{ij}\mu_{kj} - \mu_{kj} + x_{ij}\mu_{kj}}{\mu_{kj}(1-\mu_{kj})} \right) = 0$

Since $\mu_{kj} \in (0,1)$, so $\sum_{i=1}^{N} r_{ik} \cdot (x_{ij} - \mu_{kj}) = 0$

$$\sum_{i=1}^{N} r_{ik} x_{ij} - \sum_{i=1}^{N} r_{ik} \mu_{kj} = 0$$

$$\mu_{kj} = \frac{\sum_{i=1}^{N} r_{ik} x_{ij}}{\sum_{i=1}^{N} r_{ik}}$$

(2) The lower bound $l(\mu_k)$ of the original likelihood function:

$$l(\mu_k) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \ln P(x_i, z_i = k, \mu_k)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \ln P(z_i = k | \mu_k) \cdot P(x_i | z_i = k, \mu_k) \cdot \beta(\mu | \alpha, \beta)$$

$$\frac{\partial l(\mu_k)}{\partial \mu_{kj}} = \frac{\partial}{\partial \mu_{kj}} \left[\sum_{i=1}^{N} r_{ik} \cdot \left(\ln \pi_k + x_{ij} \ln(\mu_{kj}) + (1-x_{ij}) \ln(1-\mu_{kj}) \right) \right] +$$

$$\frac{\partial}{\partial \mu_{kj}} \ln \frac{\Gamma(\alpha+\beta) \mu_{kj}^{\alpha-1} (1-\mu_{kj})^{\beta-1}}{\Gamma(\alpha)\Gamma(\beta)} 1(\mu \in (a,b))$$

where $1(A)$ is an indicator function in the set(A), $0 \leq a < b < 1$.

Since $\frac{\partial \ln \beta}{\partial \mu_{kj}} = \frac{\partial}{\partial \mu_{kj}} \ln \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu_{kj}^{\alpha-1} (1-\mu_{kj})^{\beta-1} = \frac{\alpha-1}{\mu_{kj}} - \frac{\beta-1}{1-\mu_{kj}}$

$$\Rightarrow \frac{\partial l(\mu_k)}{\partial \mu_{kj}} = \sum_{i=1}^{N} r_{ik} \left(\frac{x_{ij}}{\mu_{kj}} - \frac{1-x_{ij}}{1-\mu_{kj}} \right) + \frac{\alpha-1}{\mu_{kj}} - \frac{\beta-1}{1-\mu_{kj}}$$

Let $\frac{\partial l(\mu_k)}{\partial \mu_{kj}} = 0$

$$\Rightarrow \mu_{kj} = \frac{\sum_i r_{ik} x_{ij} + \alpha - 1}{\sum_i r_{ik} + \alpha + \beta - 2}$$

2. From the question, $e_{ij} = g(x_i^+) - g(x_j^-)$.

$$u(e_{ij}) = \begin{cases} 1 & \text{if } e_{ij} > 0 \\ 0.5 & \text{if } e_{ij} = 0 \\ 0 & \text{if } e_{ij} < 0 \end{cases} \qquad AUC = \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(e_{ij})$$

If we'd like to prove $AUC = \dfrac{\sum_{i=1}^{m^+} rank_i - (m^+)(m^+ + 1)/2}{m^+ m^-}$,

it's the same to prove $\sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(e_{ij}) = \sum_{i=1}^{m^+} rank_i - (m^+)(m^+ + 1)/2$

We assume x_i^+ is the sample with i-th smallest $g(x_i^+)$, its rank is $rank_i$.

$g(x_i^+)$ is larger than $i-1$ predictions in the positive samples, also $g(x_i^+)$ is larger than $rank_i - 1$ predictions in the total samples.

Thus, $g(x_i^+)$ is larger than $rank_i - 1 - (i-1) = rank_i - i$ negative samples. $\Rightarrow \sum_{j=1}^{m^-} u(g(x_i^+) - g(x_j^-)) = \sum_{j=1}^{m^-} u(e_{ij}) = rank_i - i$

(Since $g(x_i^+) > g(x_j^-) \Rightarrow e_{ij} > 0 \Rightarrow u(e_{ij}) = 1$).

$$\sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(e_{ij}) = \sum_{i=1}^{m^+} (rank_i - i) = \sum_{i=1}^{m^+} rank_i - \frac{m^+(m^+ + 1)}{2}$$

$$AUC = \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(e_{ij}) = \frac{1}{m^+ m^-} \cdot \left(\sum_{i=1}^{m^+} rank_i - \frac{m^+(m^+ + 1)}{2} \right) \quad \text{Q.E.D.}$$

3. $\mu = \frac{1}{N} \sum_{i=1}^{N} x^{(n)}$, $\Sigma = \frac{1}{N} \sum_{i=1}^{N} (X^{(n)} - \mu)(X^{(n)} - \mu)^T$

The code is attached below:

```python
import numpy as np

X = np.array([[2,0,1,-3,-2],
              [0,2,-3,-3,-2],
              [1,2,1,3,-2],
              [-1,1,3,2,-1],
              [1,0,1,-1,1],
              [2,3,-1,1,-2],
              [-2,3,-3,3,2],
              [-2,-2,2,3,-2],
              [-2,-3,1,-2,-3],
              [-3,2,0,-1,-2]])

X = X.T
mu = np.mean(X, axis = 1)
cov_matrix = np.cov(X, ddof = 0)
eigen = np.linalg.eig(cov_matrix)
eigenvalue = eigen[0]
eigenvector = eigen[1]
U = eigenvector[:, 3:]
mumatrix = np.array([mu]*10).T
new = np.dot(U.T, (X-mumatrix))
```

```
In [29]: print(mu)
[-0.4  0.8  0.2  0.2 -1.3]
```

μ

Σ

```
In [30]: print(cov_matrix)
[[ 3.04  0.82 -0.02 -0.82 -0.12]
 [ 0.82  3.76 -2.16  1.04  1.04]
 [-0.02 -2.16  3.56  0.76 -0.84]
 [-0.82  1.04  0.76  5.56  1.16]
 [-0.12  1.04 -0.84  1.16  2.21]]
```

We choose the largest 2 eigenvalues and the corresponding eigenvectors

```
In [31]: print(eigenvalue)
[0.82311524 1.53027334 3.07614543 5.93067614 6.76978985]
```

Eigenvalue

```
In [32]: print(eigenvector)
[[-0.33192081  0.16894209  0.87848693  0.29809153 -0.02625442]
 [ 0.60584079 -0.33313495  0.17625638  0.39493632  0.57873744]
 [ 0.61980825  0.08283824  0.40532527 -0.58025264 -0.32862419]
 [-0.33144989 -0.15734291  0.14382432 -0.64618454  0.65356276]
 [ 0.16960016  0.91041788 -0.11054589  0.03031732  0.35949345]]
```

Eigenvector

Final projection: $U^T(X^{(n)} - \mu)$

```
In [33]: print(new)
[[ 1.98183693  4.49653708 -1.40348923 -2.87861204  0.48232827  1.74241301
   0.53945236 -4.45776178 -1.07184006  0.56913546]
 [-3.13194616 -0.60746566  1.97315969  0.4956134  -0.72008587  1.87576557
   5.38313097 -0.591651   -4.46907149 -0.20744945]]
```

Final new projection