Final Project

Yifan Yang

General Analysis

First of all, we should do the general analysis.

We can write the pdf for X_i as:

$$p(X_i) = \sum_{m=1}^{M} \pi_m p(X_i|\mu_m) = \sum_{m=1}^{M} \pi_m (\prod_{j=1}^{D} \mu_{mj}^{x_{ij}} (1-\mu_{mj})^{1-x_{ij}})$$

Based on the above pdf formula and the contents in class, we can define the latent variables Z_is for every X_i which contains M variables representing the class labels for X_i. Then, we should take expectation over those Z_is based on the parameters of μ and π, and do the maximization.

The log-likelihood can be separated as two parts as:

$$\begin{aligned} ln\ p(X, \mu, \pi) &= ln\ p(X \mid \mu, \pi) + ln\ p(\mu, \pi) \\ &= ln\ p(X \mid \mu, \pi) + ln\ p(\mu) + ln\ p(\pi) \end{aligned}$$

In which the first part we take it as complete-data log-likelihood and can be written as:

$$ln\ p(X \mid \mu, \pi) = \sum_{i=1}^{N} \sum_{m=1}^{M} Z_{im} (ln\ \pi_m + \sum_{j=1}^{D} ln\ p(x_{ij}, \mu_{mj})))$$

Denote the parameter $\{\mu_m, \pi_m\}$ as θ_m. We can take initial values of θ, defined as θ^{old}, and take expetation in term of θ^{old} on this formula, so we can obtain the form of $Q(\theta, \theta^{old})$

$$Q(\theta, \theta^{old}) = \int p(Z|X, \theta^{old}) ln(p(Z, X|\theta))dz = \sum_{i=1}^{N} \sum_{m=1}^{M} \gamma_{\theta^{old}}(z_{im})(ln\ \pi_m + \sum_{j=1}^{D} ln\ p(x_{ij}, \mu_{mj}))$$

In which $\gamma_{\theta^{old}}(z_{im})$ refers to the expected value of z_{im} based on θ^{old}.

Therefore, our aim is to maximize the following formula in terms of θ:

$$Q(\theta, \theta^{old}) + ln\ p(\theta) = \sum_{m=1}^{M} (ln\ p(\mu_m) + ln\ p(\pi_m)) + \sum_{i=1}^{N} \sum_{m=1}^{M} \gamma_{\theta^{old}}(z_{im})(ln\ \pi_m + \sum_{j=1}^{D} ln\ p(x_{ij}, \mu_{mj}))$$

We do the EM algorithm to deal with the case, namely calculate θ iteratively until it converges to some standard.

(a)

Given the prior of μ_{mj} is $Beta(2, 2)$ and π_m is $Dirichlet(2, 2, ..., 2)$, We can calculate the pdf for μ_{mj}: $p(\mu_{mj}) = 6(1 - \mu_{mj})\mu_{mj}$. and π_m: $p_2(\pi_m) = (2M - 1)! \prod_{m=1}^{M} \pi_m$.

In order to solve the optimal parameter, we take derivatives here and make them 0.

$$\frac{\partial l}{\partial \mu_{mj}} = \frac{1 - 2\mu_{mj} + \sum_{i=1}^{N} \gamma_{\theta^{old}}(z_{im})(x_{ij} - \mu_{mj})}{\mu_{mj}(1 - \mu_{mj})}$$

Solve this equation, we obtain:

$$\hat{\mu}_{mj} = \frac{1 + \sum_{i=1}^{N} \gamma_{\theta^{old}}(z_{im}) x_{ij}}{2 + \sum_{i=1}^{N} \gamma_{\theta^{old}}(z_{im})}, \quad j = 1, 2...D \qquad m = 1, 2...M$$

In terms of π_m, we have the restriction of $\sum \pi_m = 1$, so we take Lagrangian here:

$$\frac{\partial l}{\partial \pi_m} = \frac{1 + \sum_{i=1}^{N} \gamma_{\theta^{old}}(z_{im})}{\pi_m} - \lambda$$

So we have:

$$\hat{\pi}_m = \frac{1 + \sum_{i=1}^{N} \gamma_{\theta^{old}}(z_{im})}{\lambda}, \qquad m = 1, 2...M$$

Then we use $\sum \hat{\pi}_m = 1$ to figure out that $\lambda = M + N$.

Thus,

$$\hat{\pi}_m = \frac{1 + \sum_{i=1}^{N} \gamma_{\theta^{old}}(z_{im})}{M + N}, \qquad m = 1, 2...M$$

(b)

Here we consider is the initialization step. Assign each example X_i at random to one of the M components of Z_i (for example: if assigned to class m, we have $z_{im} = 1$, and $z_{ik} = 0$ for all $k \neq m$).

Based on the results of (a) we only need to substitute the $\gamma_{\theta^{old}}(z_{im})$ part to simple z_{im} and do the calculation. The log-likelihood formula is:

$$Q(\theta, \theta^{old}) + ln\ p(\theta) = \sum_{m=1}^{M} (ln\ p(\mu_m) + ln\ p(\pi_m)) + \sum_{i=1}^{N} \sum_{m=1}^{M} z_{im}(ln\ \pi_m + \sum_{j=1}^{D} ln\ p(x_{ij}, \mu_{mj}))$$

So we can get:

$$\hat{\mu}_{mj} = \frac{1 + \sum_{i=1}^{N} z_{im} x_{ij}}{2 + \sum_{i=1}^{N} z_{im}}, \quad j = 1, 2...D \qquad m = 1, 2...M$$

$$\hat{\pi}_m = \frac{1 + \sum_{i=1}^{N} z_{im}}{M + N}, \qquad m = 1, 2...M$$

(c)

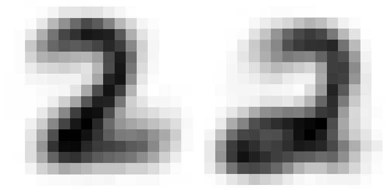
We can easily prove that both expressions are identical since when dividing $exp(l^*)$ on both sides of the fraction of the second form, we can have the first expression.

The reason why this transformation is important is that the value of every part of $\pi_m p(X_i|\mu_m)$ is very close to 0, so when we take log form and subtract the largest one (namely $exp(l - l^*)$), the value of such form would be much more computable than the original form which is pretty similar to $\frac{0}{0}$ to the computer.

(d)

Here are the results for `M=2,3,5,8`, the log-likelihood in every iteration and the image of every component in the mixture Bernoulli model:

M=2

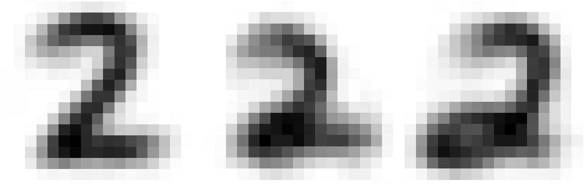


```
## Number of iterations: 15
```

Likelihood:

```
 [1] -51643.20 -49424.15 -48749.51 -48670.09 -48659.15 -48652.00 -48646.27
 [8] -48645.43 -48645.20 -48645.02 -48644.89 -48644.80 -48644.76 -48644.73
[15] -48644.72
```

M=3

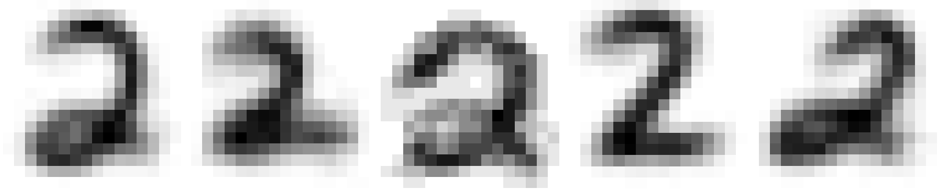


```
## Number of iterations: 80
```

Likelihood:

```
 [1] -52548.71 -51190.82 -49735.68 -48908.47 -48640.45 -48492.46 -48374.49
 [8] -48300.24 -48250.72 -48202.40 -48149.40 -48106.55 -48065.04 -48049.04
[15] -48034.35 -48023.89 -48016.26 -48004.77 -47993.21 -47983.02 -47971.42
[22] -47962.29 -47958.48 -47954.74 -47951.02 -47948.46 -47946.34 -47943.88
[29] -47940.07 -47932.89 -47924.35 -47918.67 -47912.30 -47902.77 -47896.62
[36] -47885.22 -47873.20 -47857.84 -47843.47 -47828.40 -47816.23 -47805.83
[43] -47797.32 -47788.63 -47777.31 -47767.88 -47761.52 -47757.79 -47753.13
[50] -47747.67 -47742.80 -47741.61 -47740.56 -47739.21 -47738.06 -47736.89
[57] -47735.35 -47733.92 -47732.84 -47730.93 -47729.39 -47728.04 -47727.31
[64] -47726.78 -47726.35 -47726.16 -47726.00 -47725.85 -47725.71 -47725.60
[71] -47725.49 -47725.35 -47725.18 -47724.97 -47724.77 -47724.61 -47724.51
[78] -47724.44 -47724.41 -47724.39
```

M=5

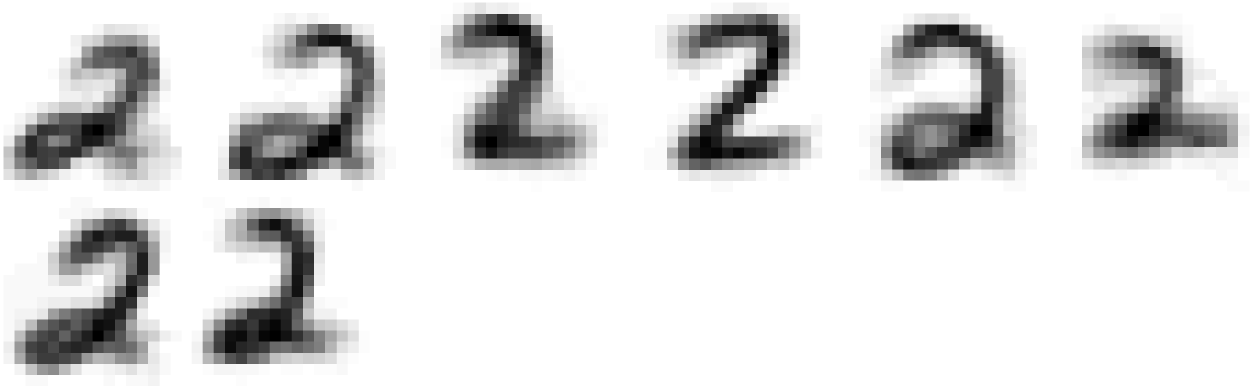


```
## Number of iterations: 34
```

Likelihood:

```
 [1] -53232.91 -49118.17 -47911.03 -47460.84 -47359.25 -47329.39 -47311.99
 [8] -47295.99 -47293.44 -47291.66 -47289.99 -47284.65 -47278.31 -47275.70
[15] -47274.08 -47269.84 -47265.58 -47263.24 -47262.08 -47261.23 -47260.13
[22] -47259.06 -47258.58 -47258.38 -47258.18 -47258.03 -47257.97 -47257.95
[29] -47257.95 -47257.96 -47257.96 -47257.97 -47257.97 -47257.97
```

M=8



```
## Number of iterations: 106
```

Likelihood:

```
  [1] -54795.23 -50125.26 -48574.14 -48069.87 -47880.38 -47757.59 -47646.27
  [8] -47586.77 -47536.66 -47518.68 -47500.76 -47483.50 -47469.02 -47457.30
 [15] -47443.90 -47439.72 -47438.07 -47435.43 -47433.54 -47430.94 -47420.83
 [22] -47412.22 -47408.92 -47408.97 -47409.08 -47409.18 -47409.27 -47409.35
 [29] -47409.45 -47409.54 -47409.36 -47408.26 -47407.76 -47406.99 -47404.76
 [36] -47403.71 -47403.16 -47403.12 -47403.11 -47402.99 -47402.86 -47402.80
 [43] -47402.82 -47402.87 -47402.94 -47402.99 -47402.97 -47402.80 -47402.60
 [50] -47402.71 -47402.59 -47398.49 -47393.14 -47388.51 -47379.71 -47372.96
 [57] -47367.53 -47362.64 -47359.43 -47357.32 -47354.29 -47351.71 -47350.03
 [64] -47350.07 -47349.63 -47346.97 -47345.56 -47344.24 -47340.60 -47336.75
 [71] -47336.78 -47336.99 -47337.19 -47337.40 -47337.65 -47337.90 -47337.18
 [78] -47333.20 -47328.51 -47324.11 -47320.92 -47314.77 -47307.26 -47306.15
 [85] -47303.61 -47302.05 -47301.92 -47301.82 -47301.63 -47301.05 -47299.94
 [92] -47298.92 -47298.89 -47298.95 -47298.99 -47299.01 -47299.02 -47299.01
 [99] -47298.96 -47298.88 -47298.77 -47298.66 -47298.60 -47298.56 -47298.55
[106] -47298.56
```

From all these components in the mixture model, we can see the difference among them is primarily the shape (or the type of writing) of the digit, and by using more components M, we can probably obtain larger likelihood and the number of iterations tends to increase, the uncertainty is due to the random allocation within the initial step.

(e)

Steps:

- Select two digits and a particular M
- Use EM algorithm in (d) to fit two models for both digits, and obtain their correponding π and μ
- Selecting the test data for both digits
- Based on π and μ, we can get the likelihood for every testing data in every part (results are two matrices with dimension 1000*M)

- Then we can use the formula for l in c to obtain the mixture likelihood for every testing data
- If the mixture likelihood is bigger in the "supposed" group than the other, we should label it as "correct", otherwise, it should be labeled "error"

Namely the mathmetical expresion is that (`n` stands for digits):

$$\hat{y}(X^*) = argmax_n \sum_{m=1}^{M} \hat{\pi}_{mn} p(X^*|\hat{\mu}_{mn})$$

Another thought is to compare the largest log-likelihood in M components (not the weighted mixture likelihood). The rationale is that a single data can only be generated with a particular component, so we can assume it derives from the component which has the largest likelihood.

The mathmetical expression for this is:

$$\hat{y}(X^*) = argmax_n \;\; max(p(X^*|\hat{\mu}_{mn}))$$

And the testing error can both be obtained by:

$$\hat{Err}(\hat{y}) = \frac{\#\{\hat{y}(X^*) \neq real\ digits\ of\ X^*\}}{\#TestingData}$$

I make the whole matrix (10*10) for every combination of two digits, and report the overall testing errors for both methods mentioned. Here, I use `M=5`.

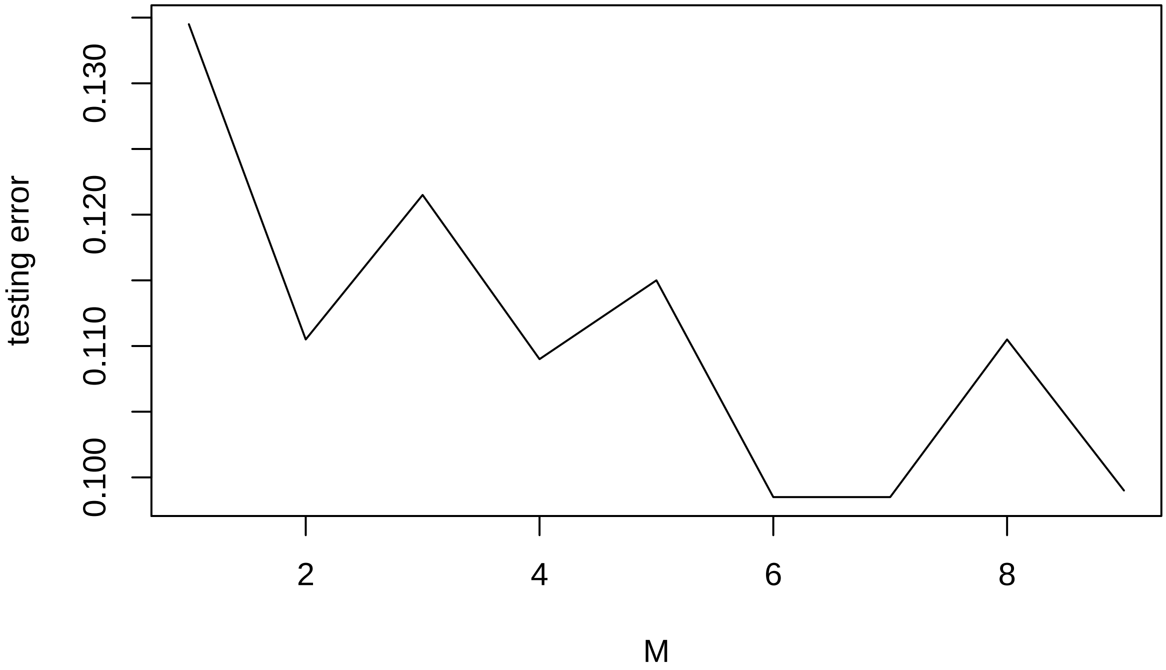
```
$MaxLikelihood_Standard
       0      1      2      3      4      5      6      7      8      9
0 0.0000 0.0025 0.0155 0.0115 0.0075 0.0205 0.0270 0.0055 0.0235 0.0105
1 0.0025 0.0000 0.0150 0.0130 0.0080 0.0085 0.0080 0.0145 0.0135 0.0105
2 0.0155 0.0150 0.0000 0.0330 0.0165 0.0185 0.0150 0.0195 0.0355 0.0155
3 0.0115 0.0130 0.0330 0.0000 0.0125 0.0485 0.0060 0.0120 0.0515 0.0235
4 0.0075 0.0080 0.0165 0.0125 0.0000 0.0140 0.0115 0.0225 0.0240 0.1200
5 0.0205 0.0085 0.0185 0.0485 0.0140 0.0000 0.0215 0.0080 0.0635 0.0215
6 0.0270 0.0080 0.0150 0.0060 0.0115 0.0215 0.0000 0.0020 0.0185 0.0020
7 0.0055 0.0145 0.0195 0.0120 0.0225 0.0080 0.0020 0.0000 0.0165 0.0605
8 0.0235 0.0135 0.0355 0.0515 0.0240 0.0635 0.0185 0.0165 0.0000 0.0335
9 0.0105 0.0105 0.0155 0.0235 0.1200 0.0215 0.0020 0.0605 0.0335 0.0000

$MixtureLikelihood_Standard
       0      1      2      3      4      5      6      7      8      9
0 0.0000 0.0025 0.0150 0.0115 0.0075 0.0200 0.0265 0.0055 0.0240 0.0100
1 0.0025 0.0000 0.0150 0.0130 0.0075 0.0085 0.0070 0.0155 0.0130 0.0105
2 0.0150 0.0150 0.0000 0.0330 0.0175 0.0180 0.0150 0.0200 0.0355 0.0155
3 0.0115 0.0130 0.0330 0.0000 0.0125 0.0490 0.0055 0.0120 0.0515 0.0235
4 0.0075 0.0075 0.0175 0.0125 0.0000 0.0145 0.0115 0.0230 0.0240 0.1180
5 0.0200 0.0085 0.0180 0.0490 0.0145 0.0000 0.0215 0.0080 0.0630 0.0215
6 0.0265 0.0070 0.0150 0.0055 0.0115 0.0215 0.0000 0.0020 0.0185 0.0030
7 0.0055 0.0155 0.0200 0.0120 0.0230 0.0080 0.0020 0.0000 0.0185 0.0585
8 0.0240 0.0130 0.0355 0.0515 0.0240 0.0630 0.0185 0.0185 0.0000 0.0330
9 0.0100 0.0105 0.0155 0.0235 0.1180 0.0215 0.0030 0.0585 0.0330 0.0000
```

It is easy to see that the test error are all quite small, the comparsion between 4 and 9 has the largest error rate since these two digits are quite alike.

Moreover, outputs from both standards are pretty similar, showing that these two methods can both be effective.

Besides, we can also test the effect of value of M on the testing error rate (here I use 4 and 9):



Which shows the decreasing trend in testing error when M gets larger. Again, it also depends on the randomness of the initial step.