EECS545(Section001): Homework #5

Due on Mar.16, 2022 at 11:59pm

Instructor:Honglak Lee

Tiejin Chen **tiejin@umich.edu**

Problem 1

(a)

We report the pixel error for each iteration, and here 0 iter means iteration before training and i-th iter pixel error means pixel error after this iteration. And our training index here will begin as 1 not 0 in standard python:

0 iter pixel error: 18.859335321207137
1 iter pixel error: 16.67936031119635
2 iter pixel error: 16.143609450732527
3 iter pixel error: 15.970290968605863
4 iter pixel error: 15.892206085164437
5 iter pixel error: 15.834913627786484
6 iter pixel error: 15.768480073452155
7 iter pixel error: 15.687771693156488
8 iter pixel error: 15.601217583736963
9 iter pixel error: 15.536757087435545
10 iter pixel error: 15.485654601938906
11 iter pixel error: 15.436685729603232
12 iter pixel error: 15.387321005422153
13 iter pixel error: 15.33410590568827
14 iter pixel error: 15.272616585063522
15 iter pixel error: 15.218145211186979
16 iter pixel error: 15.182122719392304
17 iter pixel error: 15.152439356260283
18 iter pixel error: 15.129369883097262
19 iter pixel error: 15.111557780299426
20 iter pixel error: 15.103583033161954
21 iter pixel error: 15.099361018287492
22 iter pixel error: 15.095234928045208
23 iter pixel error: 15.093837906041832
24 iter pixel error: 15.091478110011893
25 iter pixel error: 15.089711688746325
26 iter pixel error: 15.088328365576604
27 iter pixel error: 15.088921819015555
28 iter pixel error: 15.08808617415478
29 iter pixel error: 15.08661571895704
30 iter pixel error: 15.084845842682624
31 iter pixel error: 15.082765209498305
32 iter pixel error: 15.081155364501896
33 iter pixel error: 15.079653059006091
34 iter pixel error: 15.079320023450894
35 iter pixel error: 15.078008858643486
36 iter pixel error: 15.075496876716272
37 iter pixel error: 15.073818083987335
38 iter pixel error: 15.072800914692927
39 iter pixel error: 15.072250233212376
40 iter pixel error: 15.071963094357603
41 iter pixel error: 15.071818309061458
42 iter pixel error: 15.071222418342856

Problem 1 continued on next page. . .

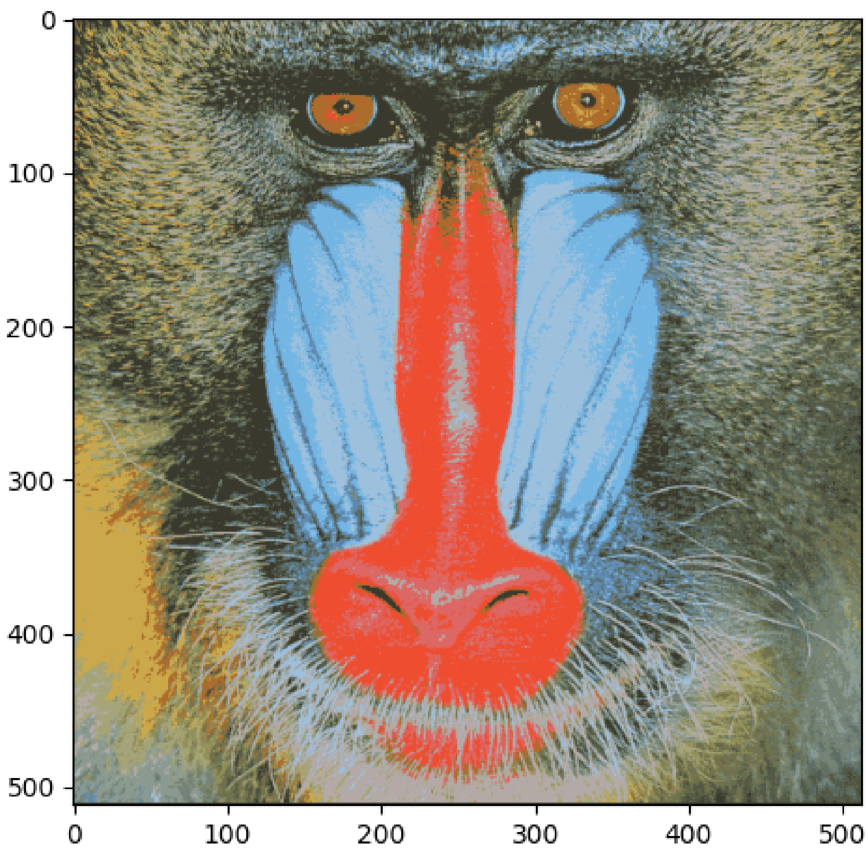
43 iter pixel error: 15.071184721460899
44 iter pixel error: 15.071477548605133
45 iter pixel error: 15.071271049803784
46 iter pixel error: 15.070787105245518
47 iter pixel error: 15.070253163924574
48 iter pixel error: 15.069261648168116
49 iter pixel error: 15.06893259416436
50 iter pixel error: 15.068497851713847

(b)

We use our own python function *own_pairwise_distances* to replace *pairwise_distances* in sklearn. And the only loop in our code is iteration from 0 to max iter which is 50 here.

(c)

We can get the new image



And pixel error result is 15.068497851713847.

(d)

We know for the original image, the total colors are $256^3 = 2^{24}$. Thus it needs 24 bits for represent one pixel in original data. And after compressing, we only have 16 colors, which is 2^4. And we only need 4 bits to represent one pixel in compressed data. We can get the compressing factor is $24/4 = 6$, which means we can compress the image to its $\frac{1}{6}$.

Problem 2

(a)

we report the loglikelihood for each iteration, and here 0 iter means iteration before training and i-th iter pixel error means pixel error after this iteration agian. And our training index here will begin as 1 not 0 in standard python:

0 iter loglikelihood: -255933.54971511522
1 iter loglikelihood: -234984.88906443462
2 iter loglikelihood: -233129.64799067134
3 iter loglikelihood: -231935.85967368353
4 iter loglikelihood: -231391.31718930256
5 iter loglikelihood: -231028.95582068275
6 iter loglikelihood: -230659.30989442748
7 iter loglikelihood: -230320.11067226104
8 iter loglikelihood: -230029.83169104904
9 iter loglikelihood: -229795.64212571736
10 iter loglikelihood: -229613.2274121245
11 iter loglikelihood: -229472.76352620716
12 iter loglikelihood: -229364.01200682926
13 iter loglikelihood: -229278.94124082162
14 iter loglikelihood: -229212.0307827594
15 iter loglikelihood: -229159.03222980574
16 iter loglikelihood: -229116.48405392538
17 iter loglikelihood: -229081.69448100665
18 iter loglikelihood: -229052.68830562648
19 iter loglikelihood: -229027.95033670843
20 iter loglikelihood: -229006.09288559164
21 iter loglikelihood: -228985.6749728887
22 iter loglikelihood: -228965.04711965926
23 iter loglikelihood: -228941.99004662823
24 iter loglikelihood: -228912.90774275622
25 iter loglikelihood: -228871.06338427393
26 iter loglikelihood: -228803.0644666749
27 iter loglikelihood: -228684.10170104238
28 iter loglikelihood: -228487.62811990466
29 iter loglikelihood: -228291.08036742482
30 iter loglikelihood: -228221.50789194135
31 iter loglikelihood: -228196.8736453044
32 iter loglikelihood: -228181.8939008806
33 iter loglikelihood: -228171.6457966613
34 iter loglikelihood: -228164.4831155889
35 iter loglikelihood: -228159.46611446305
36 iter loglikelihood: -228155.94997271523
37 iter loglikelihood: -228153.47761927452
38 iter loglikelihood: -228151.7295608838
39 iter loglikelihood: -228150.48565368415
40 iter loglikelihood: -228149.59486804536
41 iter loglikelihood: -228148.95322341964
42 iter loglikelihood: -228148.48859642656

Problem 2 continued on next page. . .

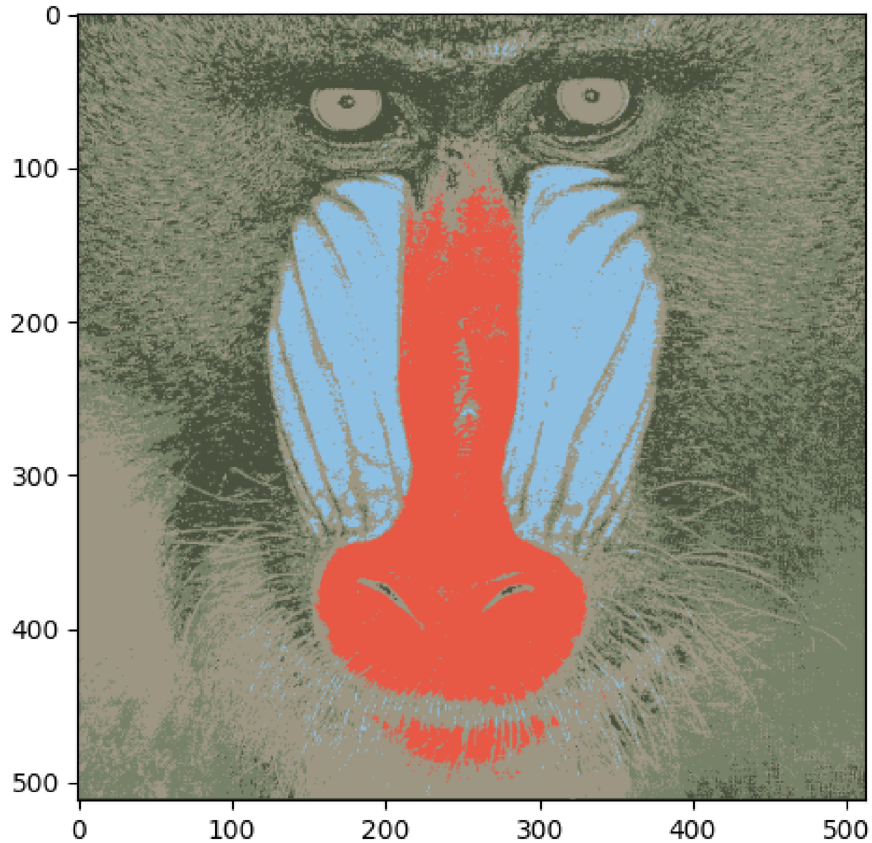
43 iter loglikelihood: -228148.15053252832
44 iter loglikelihood: -228147.90344925667
45 iter loglikelihood: -228147.72207494793
46 iter loglikelihood: -228147.58835451095
47 iter loglikelihood: -228147.4893251887
48 iter loglikelihood: -228147.41564186674
49 iter loglikelihood: -228147.36054344685
50 iter loglikelihood: -228147.3191226669

(b)

For both E and M step, we only use one loop.

(c)

We get the new image



And the pixel error is 33.22711066803432.

Now we report μ_k and σ^k:

1. μ_1:[141.77796344 191.0887889 226.67410958], $\Sigma_1 = \begin{bmatrix} 584.54417814 & 118.49096449 & -116.49649559 \\ 118.49096449 & 59.30202846 & 10.9545699 \\ -116.49649559 & 10.9545699 & 68.91164644 \end{bmatrix}$

2. μ_2:[156.26878194 150.27649259 130.14104438], $\Sigma_2 = \begin{bmatrix} 1230.3729375 & 403.0412842 & -605.66360609 \\ 403.0412842 & 792.77448118 & 706.18391346 \\ -605.66360609 & 706.18391346 & 2558.99102211 \end{bmatrix}$

3. μ_3:[231.56210529 88.14926648 69.73473432], $\Sigma_3 = \begin{bmatrix} 196.19321615 & -142.94254486 & -297.99006894 \\ -142.94254486 & 310.27086797 & 540.38232451 \\ -297.99006894 & 540.38232451 & 1163.38268786 \end{bmatrix}$

4. μ_4:[118.17293159 129.06302449 103.86034453], $\Sigma_4 = \begin{bmatrix} 673.70137018 & 573.16958607 & 172.65271639 \\ 573.16958607 & 635.69117341 & 400.17443826 \\ 172.65271639 & 400.17443826 & 775.83675746 \end{bmatrix}$

5. μ_5:[75.03056535 81.74551223 130.14104438], $\Sigma_5 = \begin{bmatrix} 347.72070436 & 415.66344678 & 254.68041925 \\ 415.66344678 & 598.86362876 & 410.83678682 \\ 254.68041925 & 410.83678682 & 371.21940727 \end{bmatrix}$

(d)
Agian, for original image we need to use 24 bits to represent one pixel. For compressed image, we have 5 colors, and $4 < 5 < 8$. Thus we need to use 3 bits to represent one pixel. Thus we can get the compressing factor is $24/3 = 8$.

Problem 3

(a)
First we prove $UU^Tx = \sum_{i=1}^K u_i u_i^T x$.We have:

$$U = [u_1, ..., u_k], UU^Tx = [u_1, ..., u_k][u_1^T, ..., u_k^T]^T x = \sum_{i=1}^k u_i u_i^T x$$

Then we will prove top K eigenvectors is the solution. We have:

$$\sum_{i=1}^N \left\|x_i - UU^Tx_i\right\|^2 = \sum_{i=1}^N (x_i - UU^Tx_i)^T(x_i - UU^Tx_i) = \sum_{i=1}^N -2\left\|U^Tx_i\right\|^2 + U^TU\left\|U^Tx_i\right\|^2 + \sum_{i=1}^N x_i^T x_i = -\sum_{i=1}^N \left\|U^Tx_i\right\|^2 + c$$

Where c is a fixed constant. Then we have $min - \frac{1}{N}\sum_{i=1}^N \left\|U^Tx_i\right\|^2 + c$ is equivalent to $max\frac{1}{N}\sum_{i=1}^N \left\|U^Tx_i\right\|^2$. Then it is equivalent to the PCA we talked in class. Thus we get u_i is the eigenvectors corresponding to the (ordered) eigenvalues λ_i.

Finally, we need to prove $\min \frac{1}{N}\sum_{i=1}^N \left\|x_i - UU^Tx_i\right\|^2 = \sum_{k=K+1}^d \lambda_k$. First we use $U_d = [u_1, ..., u_k, u_{k+1}, ..., u_d]$ to represent the eigenvector matrix of data covariance matrix. And we can get $U_d^TU_d = U_dU_d^T = I_d$. Then we have:

$$\frac{1}{N}\sum_{i=1}^N \left\|x_i - UU^Tx_i\right\|^2 = \frac{1}{N}\sum_{i=1}^N \left\|U_dU_d^Tx_i - UU^Tx_i\right\|^2 = \frac{1}{N}\sum_{i=1}^N \left\|\sum_{i=1}^d u_iu_i^Tx_i - \sum_{i=1}^k u_iu_i^Tx_i\right\|^2 = \frac{1}{N}\sum_{i=1}^N \left\|\sum_{i=k+1}^d u_iu_i^Tx_i\right\|^2$$

And we can get:

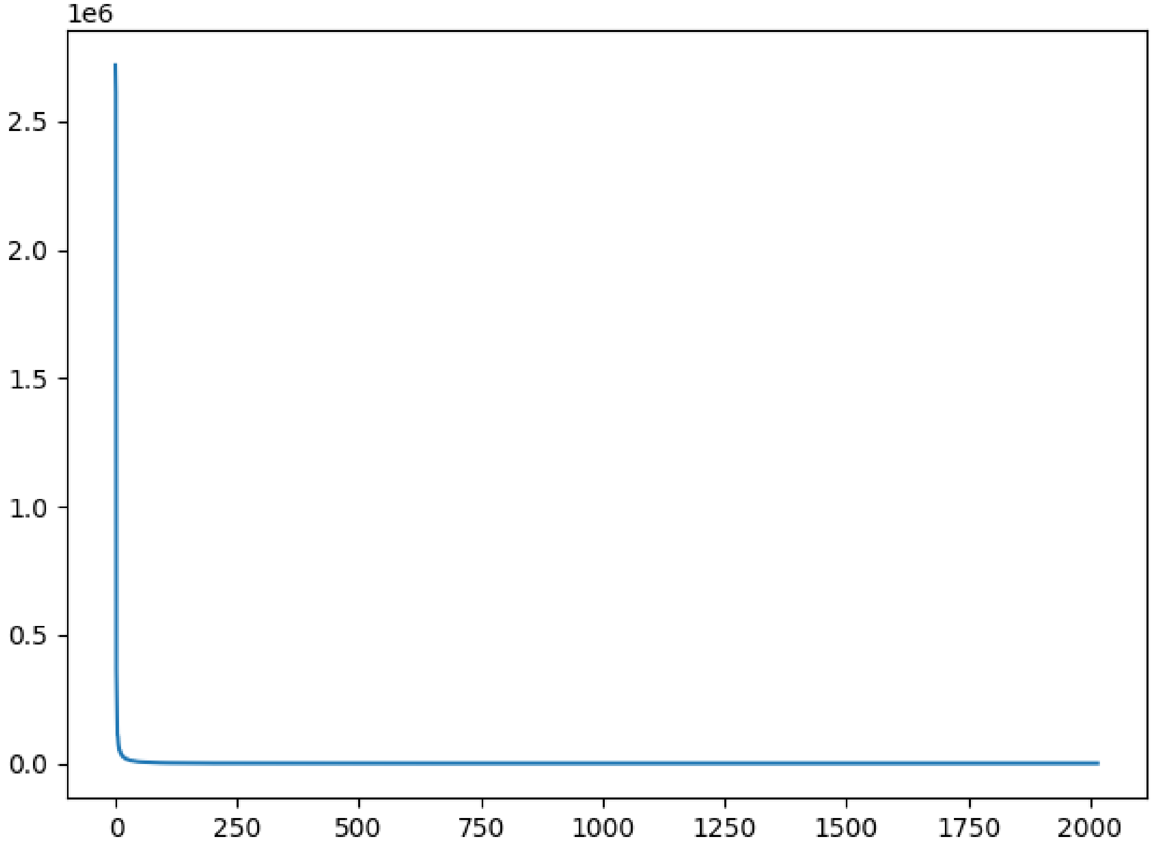
$$\begin{aligned} \frac{1}{N}\sum_{i=1}^N \left\|\sum_{i=k+1}^d u_iu_i^Tx_i\right\|^2 &= \frac{1}{N}\sum_{i=1}^N \sum_{i=k+1}^d x_i^Tu_iu_i^Tu_iu_i^Tx_i \\ &= \frac{1}{N}\sum_{i=1}^N \sum_{i=k+1}^d x_i^Tu_iu_i^Tx_i \\ &= \sum_{i=d+1}^k \lambda_i \end{aligned}$$

And that ends our proof.

Problem 3 continued on next page. . .

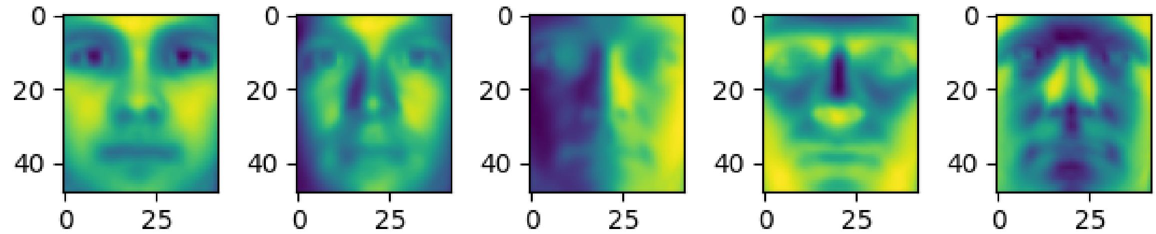
(b)

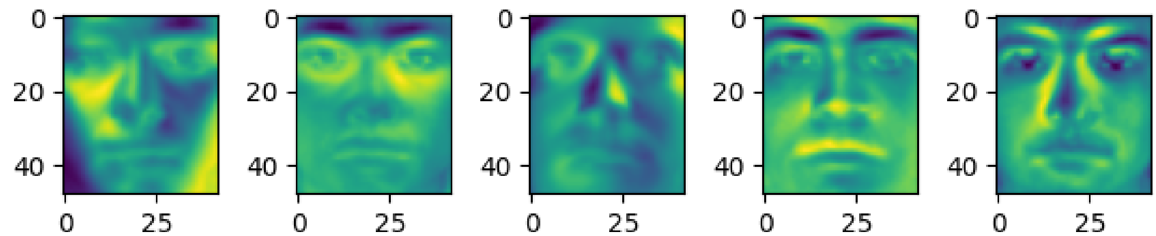
We can the top-10 eigenvalues are:[2719333.89451627, 2611865.95964518, 365515.29797577, 211149.83657024, 110603.22720593, 104715.79567702, 78512.9353901, 67032.06261281, 52592.80467658, 49197.08114141] And get the plot



(c)

We get the plot:





As we can see the real first component(second image) is very like mean of image. We can say the first component capture the overall information from all images. Then see the fourth component(fifth image), it captures information about nose from images. The sixth component(seventh image) captures the information around eyes. And eight component(ninth image) captures information about mouth. And the last image captures the sketch of near the nose.

(d)

for 95%, we need to use 43 components. For 99% represent, we need to use 167 componets.

Problem 4

(a)

x is a nomral distribution with probability of ϕ to decide whether we need to plus one to x.

(b)

for z, we have:

$$f(z_i) = \phi^z(1-\phi)^{1-z}$$

Thus ,we have log-likelihood for ϕ is:

$$logL(\phi|z_i) = \sum_{i=1}^{N} z_i log(\phi) + (1-z_i) log(1-\phi)$$

$$\frac{\partial logL(\phi|z_i)}{\partial \phi} = \frac{1}{\phi}\sum_{i=1}^{N} z_i - \frac{1}{1-\phi}\sum_{i=1}^{N}(1-z_i)$$

And we can get:

$$\hat{\phi}_{mle} = \frac{1}{N}\sum_{i=1}^{N} z_i$$

for ϵ, we have:

$$f(\epsilon_i) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2\sigma^2}(\epsilon-\mu)^2)$$

$$logL(\mu,\sigma^2|\epsilon) = -\frac{N}{2}log(2\pi) - \frac{N}{2}log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(\epsilon_i-\mu)^2$$

Then we have:

$$\frac{\partial[logL(\mu,\sigma^2|\epsilon)}{\partial\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{N}(\epsilon_i-\mu) = 0$$

$$\hat{\mu}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\epsilon_i$$

$$\frac{\partial[logL(\mu,\sigma^2|\epsilon)}{\partial\sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{N}(\epsilon_i-\mu)^2 = 0$$

$$\hat{\sigma^2}_{mle} = \frac{1}{N}\sum_{i=1}^{N}(\epsilon_i-\mu)^2$$

(c)

for x, we have:

$$f(x,z,\epsilon) = \phi^z(1-\phi)^{1-z}f_\epsilon(\epsilon) = \phi^z(1-\phi)^{1-z}f_\epsilon(x-z) = f(x,z)$$
$$f(x) = \phi f_\epsilon(x-1) + (1-\phi)f_\epsilon(x)$$

Then we have:

$$logL(\phi,\sigma^2,\mu) = \sum_{i=1}^{N} z_i log(\phi) + \sum_{i=1}^{N}(1-z_i)log(1-\phi) - \frac{N}{2}log(2\pi) - \frac{N}{2}log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-z_i-\mu)^2$$

which is the log-likelihood function of variables. Then In the E-step, we need to calculate $E(z_i|x), E((x_i - z_i-\mu)^2|x)$. In EM, we will use upper index to indicate the iteration in EM.

Then, we have in E-step:

$$z_i^j = E(z_i|x) = \frac{\phi^{j-1}f_{\epsilon|\mu^{j-1},\sigma^{j-1}}(x-1)}{\phi^{j-1}f_{\epsilon|\mu^{j-1},\sigma^{j-1}}(x-1) + (1-\phi^{j-1})f_{\epsilon|\mu^{j-1},\sigma^{j-1}}(x)}$$

$$E((x_i-z_i-\mu)^2|x) = \frac{(x-\mu^{j-1})^2(1-\phi^{j-1})f_{\epsilon|\mu^{j-1},\sigma^{j-1}}(x) + (x-\mu^{j-1}-1)^2\phi^{j-1}f_{\epsilon|\mu^{j-1},\sigma^{j-1}}(x-1)}{(1-\phi)f_{\epsilon|\mu^{j-1},\sigma^{j-1}}(x) + (1-\phi^{j-1})f_{\epsilon|\mu^{j-1},\sigma^{j-1}(x)}}$$
$$= (x_i-\mu^{j-1})^2(1-z_i^j) + (x_i-\mu^{j-1}-1)^2 z_i^j$$
$$E(\epsilon_i) = x_i - z_i^j = \epsilon_i^j$$

Then, for M-step, we have:

$$\frac{\partial logL(\phi,\sigma^2,\mu)}{\partial\phi} = \frac{\sum_{i=1}^{N} z_i^j}{\phi} - \frac{\sum_{i=1}^{N} 1-z_i^j}{1-\phi} = 0 \rightarrow \phi^j = \frac{1}{N}\sum_{i=1}^{N} z_i^j$$

$$\frac{\partial logL(\phi,\sigma^2,\mu)}{\partial\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{N} x_i - z_i^j - \mu = 0 \rightarrow \mu^j = \frac{1}{N}\sum_{i=1}^{N} x_i - z_i^j$$

$$\frac{\partial logL(\phi,\sigma^2,\mu)}{\partial\sigma^2} = -\frac{N}{2\sigma^2}+\frac{1}{2\sigma^4}\sum_{i=1}^{N}E[(x_i-z_i-\mu)^2] = 0 \to \sigma^{2^j} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu^{j-1})^2(1-z_i^j)+(x_i-\mu^{j-1}-1)^2 z_i^j$$

(d)

The original model only consider plus one or not. However,the modified model will consider plus all the constant,which is a more general model. For example. if we use modified model, Then we have:

$$E(x) = \mu + \lambda\phi, Var(x) = \sigma^2 + \lambda^2\phi(1-\phi)$$

And if we want the original model to have same expectation and variance, we will get μ_1, σ_1^2 for the original model:

$$\mu_1 = \mu + (\lambda-1)\phi, \sigma_1^2 = \sigma^2 + (\lambda^2-1)\phi(1-\phi)$$

Then, the parameter for original model contain two different unknown variables which means the original model will become much more complex than modified model if original model want to reach same result.

However, in other hand, if modified model want to reach same result with original model. Then we only need to use $\lambda = 1$. Hence, the modified model is a more general model and thus better.

Problem 5

We can get W:

$$W = \begin{bmatrix} 72.15081922 & 28.62441682 & 25.91040458 & -17.2322227 & -21.191357 \\ 13.45886116 & 31.94398247 & -4.03003982 & -24.0095722 & 11.89906179 \\ 18.89688784 & -7.80435173 & 28.71469558 & 18.14356811 & -21.17474522 \\ -6.0119837 & -4.15743607 & -1.01692289 & 13.87321073 & -5.26252289 \\ -8.74061186 & 22.55821897 & 9.61289023 & 14.73637074 & 45.28841827 \end{bmatrix}$$