1.

(a)

iteration: 0 pixel error: 16.494066809275818 iteration: 1 pixel error: 13.89275138870801

iteration: 2 pixel error: 13.328692808059689 iteration: 3 pixel error: 13.1233066246321

iteration: 4 pixel error: 13.034582087480153 iteration: 5 pixel error: 12.983227794490372

iteration: 6 pixel error: 12.93367629070885 iteration: 7 pixel error: 12.875933115088028

iteration: 8 pixel error: 12.816562931736327 iteration: 9 pixel error: 12.767151109946882

iteration: 10 pixel error: 12.720603562593913 iteration: 11 pixel error: 12.68088966417506

iteration: 12 pixel error: 12.642245976071205 iteration: 13 pixel error: 12.599481943303926

iteration: 14 pixel error: 12.551401215661455 iteration: 15 pixel error: 12.511858789346679

iteration: 16 pixel error: 12.486356209991003 iteration: 17 pixel error: 12.466181265682264

iteration: 18 pixel error: 12.451005458969648 iteration: 19 pixel error: 12.441774514286925

iteration: 20 pixel error: 12.43661570515305 iteration: 21 pixel error: 12.431667221371725

iteration: 22 pixel error: 12.426945942833568 iteration: 23 pixel error: 12.424326446608193

iteration: 24 pixel error: 12.42184501662411 iteration: 25 pixel error: 12.41923602458894

iteration: 26 pixel error: 12.41673132522578 iteration: 27 pixel error: 12.414600880230076

iteration: 28 pixel error: 12.413147832323068 iteration: 29 pixel error: 12.412140964335943

iteration: 30 pixel error: 12.410813271425466 iteration: 31 pixel error: 12.409983569247101

iteration: 32 pixel error: 12.409189852486058 iteration: 33 pixel error: 12.407949204556655

iteration: 34 pixel error: 12.407021284097153 iteration: 35 pixel error: 12.406274112040713

iteration: 36 pixel error: 12.405849380261508 iteration: 37 pixel error: 12.40555504822065

iteration: 38 pixel error: 12.40535001206644 iteration: 39 pixel error: 12.40515954189744

iteration: 40 pixel error: 12.405009426243742 iteration: 41 pixel error: 12.404895799874296

iteration: 42 pixel error: 12.404819715080365 iteration: 43 pixel error: 12.404693590795524 iteration: 44	pixel error: 12.404600412247218 iteration: 45	pixel error: 12.404491855025428

iteration: 46	pixel error: 12.404314039457452 iteration: 47	pixel error: 12.404192509750667

iteration: 48	pixel error: 12.404106306581475 iteration: 49	pixel error: 12.40406554603991

(b)

only one loop

```
def train_kmeans(train_data, initial_centroids):
  ##### TODO: Implement here!! #####
  # Hint: pairwise_distances() might be useful
  N = 50
  centroids = initial_centroids
  #print(centroids.shape)
  data = np.zeros(train_data.shape)
  #print(data.shape)
  for i in range(N):
    #print(i)
    #map = np.argmin(pairwise_distances(train_data, centroids), axis = 1)
    map = np.argmin(vec_pairwise_distances(train_data, centroids), axis = 1)

    centroids_repeat = centroids[np.newaxis, :, :].repeat(train_data.shape[0], axis = 0)
    data = centroids_repeat[np.arange(0, train_data.shape[0]), map]
    temp = np.zeros(train_data.shape[0])
    map_new = np.zeros([train_data.shape[0], centroids.shape[0]])
    map_new[np.arange(0, train_data.shape[0]), map] = 1
    color = map_new[:, :, np.newaxis].repeat(3, axis = 2)
    data_temp = train_data[:, np.newaxis, :].repeat(centroids.shape[0], axis = 1) * color
    centroids = data_temp.sum(axis = 0) / color.sum(axis = 0)
    error = np.sqrt(np.mean(np.power(train_data - data, 2)))
    print('iteration:', i, '\t', 'pixel error:', error)

  states = {
      'centroids': centroids
  }
  ##### TODO: Implement here!! #####
  return states
```

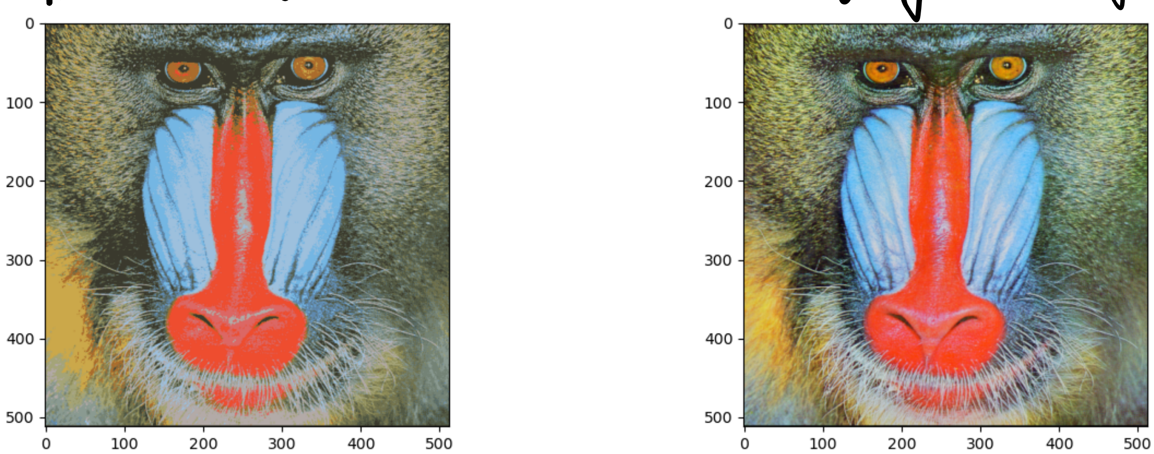
my own vectorized pairwise-distance function

```
def vec_pairwise_distances(x, y):
    xx = np.power(x, 2).sum(axis = 1)
    xx = xx[:, np.newaxis].repeat(y.shape[0], axis = 1)
    yy = np.power(y.T, 2).sum(axis = 0)
    yy = yy[np.newaxis, :].repeat(x.shape[0], axis = 0)
    distances = np.sqrt(xx + yy - 2*np.dot(x, y.T))
    return distances
```

(c)

Kmeans result= {'pixel-error': 15.068497851713847}

new image.　　　　　　　　　　origin image.



(d) $16 = 2^4 \Rightarrow 4$ bit (compressed)

$8 \times 3 = 24$ bit (origin)

$\frac{24}{4} = 6 \Rightarrow$ compressed 6 times.

2.

(a)

iteration: 0 log likelihood: [-255933.54971512] iteration: 1 log likelihood: [-234984.88906443]

iteration: 2 log likelihood: [-233129.64799067] iteration: 3 log likelihood: [-231935.85967368]

iteration: 4 log likelihood: [-231391.3171893] iteration: 5 log likelihood: [-231028.95582068]

iteration: 6 log likelihood: [-230659.30989443] iteration: 7 log likelihood: [-230320.11067226]

iteration: 8 log likelihood: [-230029.83169105] iteration: 9 log likelihood: [-229795.64212572]

iteration: 10 log likelihood: [-229613.22741212] iteration: 11 log likelihood: [-229472.76352621]

iteration: 12 log likelihood: [-229364.01200683] iteration: 13 log likelihood: [-229278.94124082]

iteration: 14 log likelihood: [-229212.03078276] iteration: 15 log likelihood: [-229159.03222981]

iteration: 16 log likelihood: [-229116.48405393] iteration: 17 log likelihood: [-229081.69448101]

iteration: 18 log likelihood: [-229052.68830563] iteration: 19 log likelihood: [-229027.95033671]

iteration: 20 log likelihood: [-229006.09288559] iteration: 21 log likelihood: [-228985.67497289]

iteration: 22 log likelihood: [-228965.04711966] iteration: 23 log likelihood: [-228941.99004663]

iteration: 24 log likelihood: [-228912.90774276] iteration: 25 log likelihood: [-228871.06338427]

iteration: 26 log likelihood: [-228803.06446668] iteration: 27 log likelihood: [-228684.10170104]

iteration: 28 log likelihood: [-228487.62811991] iteration: 29 log likelihood: [-228291.08036743]

iteration: 30 log likelihood: [-228221.50789194] iteration: 31 log likelihood: [-228196.8736453]

iteration: 32 log likelihood: [-228181.89390088] iteration: 33 log likelihood: [-228171.64579666]

iteration: 34 log likelihood: [-228164.48311559] iteration: 35 log likelihood: [-228159.46611446]

iteration: 36 log likelihood: [-228155.94997272] iteration: 37 log likelihood: [-228153.47761927]

iteration: 38 log likelihood: [-228151.72956088] iteration: 39 log likelihood: [-228150.48565368]

iteration: 40 log likelihood: [-228149.59486805] iteration: 41 log likelihood: [-228148.95322342]

iteration: 42 log likelihood: [-228148.48859643] iteration: 43 log likelihood: [-228148.15053253]

iteration: 44 log likelihood: [-228147.90344926] iteration: 45 log likelihood: [-228147.72207495] iteration: 46 log likelihood: [-228147.58835451] iteration: 47 log likelihood: [-228147.48932519]

iteration: 48 log likelihood: [-228147.41564187] iteration: 49 log likelihood: [-228147.36054345]

(b)

```python
def train_gmm(train_data, init_pi, init_mu, init_sigma):
  ##### TODO: Implement here!! #####
  # Hint: multivariate_normal() might be useful
  N = 50
  pi = init_pi
  mu = init_mu
  sigma = init_sigma
  nlength = train_data.shape[0]
  klength = init_sigma.shape[0]
  pn = np.zeros([klength, nlength])
  gamma = np.zeros([klength, nlength])
  #print(gamma.shape)
  #scale = multivariate_normal(mu)
  #gamma =
  for i in range(N):
    #print(i)
    #M Step:
    for j in range(klength):
        pn[j, :] = pi[j] * multivariate_normal.pdf(train_data, mu[j], sigma[j, :, :])
    gamma = pn / pn.sum(axis = 0)
    #print(gamma)
    #print(mu.shape)
    log_likelihood = np.log(pn.sum(axis = 0)).reshape((nlength,1)).sum(axis = 0)
    print('iteration:', i, '\t', 'log likelihood:', log_likelihood)
    #N Step:
    N_bigK = gamma.sum(axis = 1).reshape((klength, 1))
    #print(N_bigK.shape)
    gamma_data = np.dot(gamma, train_data)
    #data_new = np.zeros(train_data.shape)
    pi = N_bigK/ nlength
    mu = gamma_data / N_bigK
    for m in range(klength):
        sigma[m, :, :] = (train_data.T - mu[m, :].reshape((train_data.shape[1], 1))) @ np.diag(gamma[m]) @ (train_data.T - mu[m,
            :].reshape((train_data.shape[1], 1))).T / N_bigK[m, :]
        #x_sqrt = np.dot(data_new[], data_new.T)
        #sigma[m, :, :] =
  states = {
      'pi': pi,
      'mu': mu,
      'sigma': sigma,
  }
  ##### TODO: Implement here!! #####
  return states
```

one loop for M step

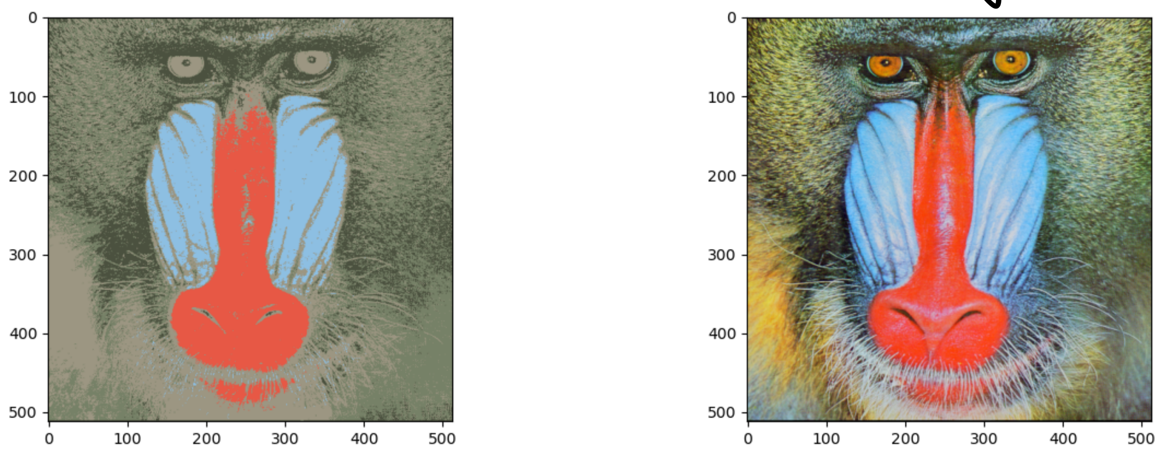
one loop for N step

(c)

pixel error:

GMM result= {'pixel-error': 33.227110668034236}

compressed image. Origin image



mu1: [141.77796344 191.0887889 226.67410958] mu2: [156.26878194 150.27649259 130.14104438]

mu3: [231.56210529 88.14926648 69.73473432] mu4: [118.17293159 129.06302449 103.86034453]

mu5: [75.03056535 81.74551223 63.65782859]

sigma1: [[584.54417814 118.49096449 -116.49649559] [118.49096449 59.30202846 10.9545699]

[-116.49649559 10.9545699 68.91164644]]

sigma2: [[1230.3729375 403.0412842 -605.66360609] [403.0412842 792.77448118 706.18391346]

[-605.66360609 706.18391346 2558.99102211]]

sigma3: [[196.19321615 -142.94254486 -297.99006894] [-142.94254486 310.27086797

540.38232451] [-297.99006894 540.38232451 1163.38268786]]

sigma4: [[673.70137018 573.16958607 172.65271639] [573.16958607 635.69117341 400.17443826]

[172.65271639 400.17443826 775.83675746]]

sigma5: [[347.72070436 415.66344678 254.68041925] [415.66344678 598.86362876 410.83678682]

[254.68041925 410.83678682 371.21940727]]

(d) $$\text{factor} = \frac{3\times 8}{\log_2 5} = \frac{24}{\log_2 5} = 10.336$$

3. PCA and eigenfaces.

(a) ① $K=0$

$$\min_{U\in\mathcal{U}} \frac{1}{N}\sum_{n=1}^{N}\|x^{(n)}-UU^T x^{(n)}\|_2^2 = \min_{U\in\mathcal{U}} \frac{1}{N}\sum_{n=1}^{N}\|x^{(n)}\|_2^2$$

$$\text{(unitary invariant)} = \min_{U\in\mathcal{U}} \frac{1}{N}\sum_{n=1}^{N}\|V^T\cdot x^{(n)}\|_2^2 \quad (V \text{ is unitary matrix})$$

$$= \frac{1}{N}\sum_{n=1}^{N} (V^T x^{(n)})\cdot(V^T x^{(n)})^T$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{d} V^T\cdot x^{(n)}\cdot x^{(n)T}\cdot V$$

$$= \frac{1}{N}\sum_{i=1}^{d}\cdot\sum_{n=1}^{N} v_i^T\cdot(x^{(n)}-\bar{x})\cdot(x^{(n)}-\bar{x})^T v_i \quad \text{(zero mean)}$$

define $S = \frac{1}{N}\sum_{n=1}^{N}(x^{(n)}-\bar{x})\cdot(x^{(n)}-\bar{x})^T$

$$= \frac{1}{N}\sum_{n=1}^{N} x^{(n)}\cdot x^{(n)T} = V\Sigma V^T \quad \text{(eigen decomposition)}$$

$$= \sum_{i=1}^{d} v_i^T\cdot S\cdot v^T \quad (v_i \text{ are the eigenvectors of } S)$$

$$= \sum_{i=1}^{d}\lambda_i \quad (\lambda_i \text{ are the corresponding eigenvalues of } S).$$

② $K\neq 0$

Since $U=[u_1, u_2, \cdots, u_K]$

$$U^T = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_K^T \end{bmatrix} \qquad UU^T=\sum_{i=1}^{K} u_i u_i^T$$

$$\Rightarrow \min_{U\in\mathcal{U}}\frac{1}{N}\sum_{n=1}^{N}\|x^{(n)}-UU^Tx^{(n)}\|^2 = \min_{U\in\mathcal{U}}\frac{1}{N}\sum_{n=1}^{N}\left\|x^{(n)}-\sum_{i=1}^{K}u_iu_i^Tx^{(n)}\right\|^2$$

$$=\min_{U\in\mathcal{U}}\frac{1}{N}\sum_{n=1}^{N}(x^{(n)}-UU^Tx^{(n)})^T(x^{(n)}-UU^Tx^{(n)})$$

$$=\min_{U\in\mathcal{U}}\frac{1}{N}\sum_{n=1}^{N}\left(x^{(n)T}\cdot x^{(n)} - x^{(n)T}\cdot UU^T\cdot x^{(n)} - \cancel{x^{(n)T}UU^Tx^{(n)} + x^{(n)T}UU^TUU^Tx^{(n)}}\right)$$

$(U^TU=I)$

$$=\min_{U\in\mathcal{U}}\frac{1}{N}\sum_{n=1}^{N}\left(x^{(n)T}\cdot x^{(n)} - x^{(n)T}\cdot UU^T\cdot x^{(n)}\right.$$

$$= \min_{U \in \mathcal{U}} \frac{1}{N} \sum_{n=1}^{N} \Big(\underbrace{x^{(n)T} \cdot x^{(n)}}_{\text{constant}} - (U^T \cdot x^{(n)})^T \cdot U^T \cdot x^{(n)}$$

$$= \max_{U \in \mathcal{U}} \frac{1}{N} \sum_{n=1}^{N} \left(U^T \cdot x^{(n)} \right)^T \cdot \left(U^T \cdot x^{(n)} \right)$$

(zero mean)

$$= \max_{U \in \mathcal{U}} \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{d} \left(u_i^T x^{(n)} - u_i^T \bar{x} \right)^T \cdot \left(u_i^T \cdot x^{(n)} - u_i^T \bar{x} \right)$$

$$= \max_{U \in \mathcal{U}} \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{d} \left(u_i^T x^{(n)} - u_i^T \bar{x} \right) \cdot \left(u_i^T x^{(n)} - u_i^T \bar{x} \right)^T$$

$$= \max_{U \in \mathcal{U}} \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{d} u_i^T \left(x^{(n)} - \bar{x} \right) \cdot \left(x^{(n)} - \bar{x} \right)^T \cdot u_i$$

$$= \max_{U \in \mathcal{U}} \sum_{i=1}^{d} u_i^T \cdot S \cdot u_i$$

Use a Lagrange multiplier

$$\Rightarrow \max_{U \in \mathcal{U}} \sum_{i=1}^{d} u_i^T S u_i + \lambda_i (1 - u_i^T u_i) = \mathcal{L}$$

$$\Rightarrow \text{derivation} \quad \sum_{i=1}^{k} 2 S u_i - 2 \lambda_i u_i = 0$$

$$\nabla_{u_1} \mathcal{L} = 2 S u_1 - 2 \lambda u_1 = 0 \Rightarrow S u_1 = \lambda u_1$$

$$\Rightarrow \lambda_1 u_1 = \mathrm{eig}(S)$$

$$\nabla_{u_2} \mathcal{L} = 2 S u_2 - 2 \lambda u_2 = 0 \Rightarrow S u_2 = \lambda_2 u_2$$

$$\Rightarrow \lambda_2 u_2 = \mathrm{eig}(S)$$

$\vdots$

u_i are orthonormal vectors $\Rightarrow$ u_i are linearly independent

$\Rightarrow$ u_i are the eigenvectors

and λ_i are the corresponding eigenvalues.

$$\Rightarrow \min_{U \in \mathcal{U}} \frac{1}{N} \sum_{n=1}^{N} \left[x^{(n)T} \cdot x^{(n)} - (U^T x^{(n)})^T (U^T x^{(n)}) \right]$$

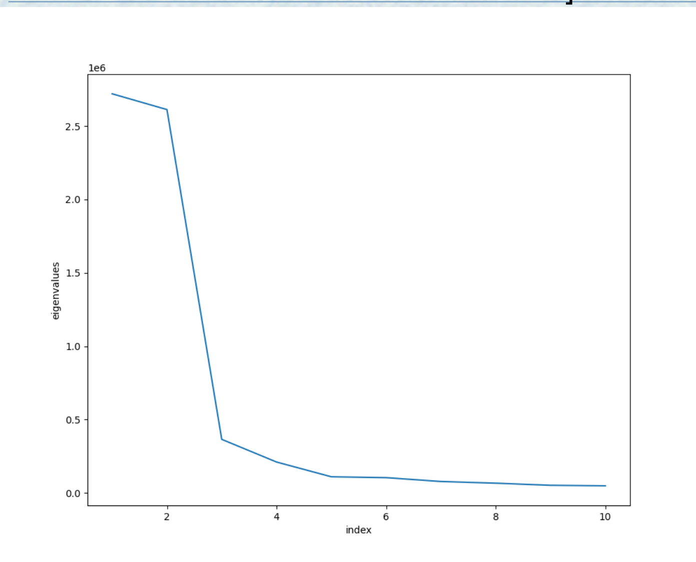
$$= \sum_{i=1}^{d} \lambda_i - \sum_{i=1}^{k} u_i^T S u_i = \sum_{i=1}^{d} \lambda_i - \sum_{i=1}^{k} \lambda_i = \sum_{i=k+1}^{d} \lambda_i$$

(b) eigenvalues

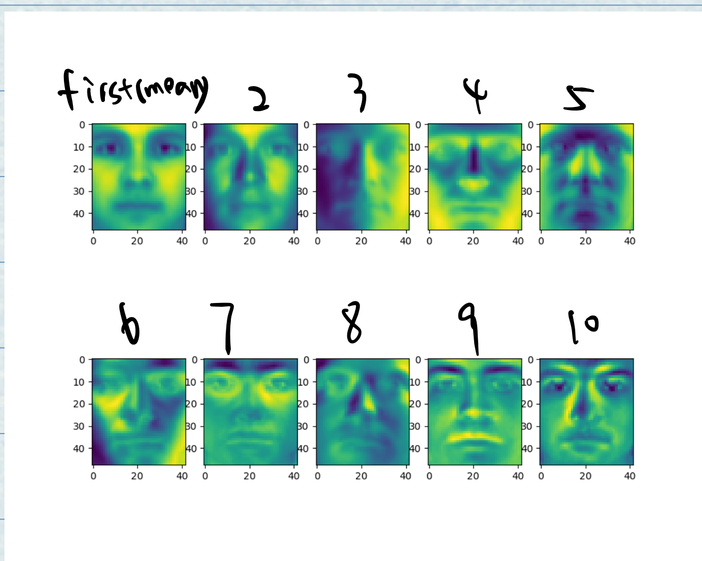
eigenvalues: [2719333.89451627 2611865.95964518 365515.29797577 211149.83657024

110603.22720593 104715.79567702   78512.9353901   67032.06261281

52592.80467658   49197.08114141]



(c)



first image: mean

third image: light from left

fifth image: have shadow

sixth image: face turn left.

(d)

42 = [0.95032236]

166 = [0.99004664]

We need 42 principal components to represent 95%. Reduce 97.9%

166 principal components to represent 99%. reduce 91.8%.

4. Expectation Maximization

(a) There will be 2 peaks, one peak is at the Y-axis and other peak is at $X = \mu + 1$, two peak with scales of ϕ and $1-\phi$ respectively.

(b)

$$P(z^{(n)}, \varepsilon^{(n)}, x^{(n)}) = \phi^{z^{(n)}} \cdot (1-\phi)^{1-z^{(n)}} \cdot \mathcal{N}(\varepsilon^{(n)} \mid \mu, \sigma)$$

$$\log\left[P(z^{(n)}, \varepsilon^{(n)}, x^{(n)})\right] = z^{(n)} \log\phi + (1-z^{(n)})\log(1-\phi) + \log\frac{1}{\sigma\sqrt{2\pi}} - \frac{(\varepsilon^{(n)}-\mu)^2}{2\sigma^2}$$

$\mathcal{L}$
$\parallel$

$$\sum_{n=1}^{N} \log\left[P(z^{(n)}, \varepsilon^{(n)}, x^{(n)})\right] = \sum_{n=1}^{N} z^{(n)} \log\phi + \sum_{n=1}^{N} (1-z^{(n)}) \cdot \log(1-\phi) + \sum_{n=1}^{N} \log\frac{1}{\sigma\sqrt{2\pi}} - \sum_{n=1}^{N} \frac{(\varepsilon^{(n)}-\mu)^2}{2\sigma^2}$$

$$\nabla_\phi \mathcal{L} = \frac{1}{\phi} \cdot \sum_{n=1}^{N} z^{(n)} - \frac{1}{1-\phi} \sum_{n=1}^{N} (1-z^{(n)}) = 0$$

$$\Rightarrow (1-\phi) \sum_{n=1}^{N} z^{(n)} = \phi \sum_{n=1}^{N} (1-z^{(n)})$$

$$\Rightarrow \phi = \frac{\sum_{n=1}^{N} z^{(n)}}{N}$$

$$\nabla_\mu \mathcal{L} = \sum_{n=1}^{N} \frac{2(\varepsilon^{(n)}-\mu)}{2\sigma^2} = 0 \Rightarrow \mu = \frac{\sum_{n=1}^{N} \varepsilon^{(n)}}{N}$$

$$\nabla_\sigma \mathcal{L} = \sum_{n=1}^{N} \frac{-1}{\sigma} + \sum_{n=1}^{N} \frac{2(\varepsilon^{(n)}-\mu)^2}{2\sigma^3} = 0$$

$$\Rightarrow \sum_{n=1}^{N} \left(\frac{(\varepsilon^{(n)}-\mu)^2}{\sigma^2} - 1 \right) = 0$$

$$\Rightarrow \frac{\sum_{n=1}^{N} (\varepsilon^{(n)}-\mu)^2}{N} = \sigma^2$$

$$\Rightarrow \sigma = \sqrt{\frac{\sum_{n=1}^{N} (\varepsilon^{(n)}-\mu)^2}{N}}$$

(c)

$$P(x \mid \mu, \sigma) = \sum_z P(x, z \mid \phi, \mu, \sigma)$$

$$\log P(x \mid \mu, \sigma) = \sum_z q(z) \cdot \log P(x \mid \phi, \mu, \sigma)$$

$$= \sum_z q(z) \log \frac{P(x, z \mid \phi, \mu, \sigma)}{q(z)} + \sum_z q(z) \cdot \log \frac{q(z)}{P(z \mid x, \sigma, \mu, \phi)}$$

$$\Rightarrow \mathcal{L}(q, \mu, \sigma) = \sum_z q(z) \log p(x, z \mid \sigma, \mu) - \sum_z q(z) \cdot \log q(z)$$

$\Rightarrow$ maximization: $\operatorname{argmax}_{\sigma, \mu, \phi} \sum_z q(z) \log p(x, z \mid \phi, \varphi, \sigma)$

Let z in $\{0,1\}^2$ be a 1-of-K random variable.

$p(z_k = 1) = \pi_k \qquad \sum_{k=1}^{2} \pi_k = 1 \qquad P(z_0 = 1) = 1 - \phi \quad P(z_1 = 1) = \phi$

$$p(x \mid z_0 = 1) = \mathcal{N}(x \mid \mu, \phi, \sigma)$$

$$p(x \mid z_1 = 1) = \mathcal{N}(x - 1 \mid \mu_1, \phi_1, \sigma_1) \quad \Rightarrow \mu_1 = \mu - 1$$

$$= \mathcal{N}(x \mid \mu - 1, \phi, \sigma)$$

$$P(x) = \sum_z P(z) \cdot p(x \mid z) = \sum_{k=1}^{2} \pi_k \mathcal{N}(x \mid \mu_k, \phi, \sigma_k)$$

$$= \phi \cdot \mathcal{N}(x \mid \mu - 1, \phi, \sigma) + (1 - \phi) \cdot \mathcal{N}(x \mid \mu, \varphi, \sigma)$$

E-step:

$$\gamma(z_{nk}) = P(z_0 = 1 \mid x^{(n)}) = \frac{(1-\phi) \cdot \mathcal{N}(x^{(n)} \mid \mu, \sigma)}{(1-\phi) \cdot \mathcal{N}(x^{(n)} \mid \mu, \sigma) + \phi \mathcal{N}(x^{(n)} \mid \mu - 1, \varphi, \sigma)} = q(z_0)$$

$$\gamma(z_{nk}) = P(z_1 = 1 \mid x^{(n)}) = \frac{\phi \mathcal{N}(x^{(n)} - 1 \mid \mu, \phi, \sigma)}{(1-\phi) \mathcal{N}(x^{(n)} \mid \mu, \varphi, \sigma) + \phi \mathcal{N}(x^{(n)} \mid \mu - 1, \phi, \sigma)} = q(z_1)$$

M-step:

$$L = \sum_{Z} q(Z) \log p(X, Z \mid \phi, \mu, \sigma)$$

$$\log p(X, Z \mid \phi, \mu, \sigma) = \sum_{n=1}^{N} \sum_{k}^{K} Z_n^k \left(\log \pi_k + \log \mathcal{N}(x^{(n)} \mid \mu_k, \sigma) \right)$$

$$\Rightarrow L = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(Z_{nk}) \left[\log \pi_k + \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2 \right]$$

$$= \sum_{n=1}^{N} \gamma(Z_0^{(n)}) \cdot \left[\log(1-\phi) + \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right]$$

$$+ \sum_{n=1}^{N} \gamma(Z_1^{(n)}) \cdot \left[\log \phi + \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2}\left(\frac{x-(\mu-1)}{\sigma}\right)^2 \right]$$

$$\nabla_\phi L = -\sum_{n=1}^{N} \gamma(Z_0^{(n)}) \cdot \frac{1}{1-\phi} + \sum_{n=1}^{N} \gamma(Z_1^{(n)}) \cdot \frac{1}{\phi} = 0$$

$$\Rightarrow \text{since } \gamma(Z_0^{(n)}) + \gamma(Z_1^{(n)}) = 1$$

$$\sum_{n=1}^{N} (1-\gamma_1^{(n)}) \phi = \sum_{n=1}^{N} \gamma_1^{(n)} (1-\phi)$$

$$\Rightarrow N\phi - \cancel{\phi \sum_{n=1}^{N} \gamma_1^{(n)}} = \sum_{n=1}^{N} \gamma_1^{(n)} - \cancel{\phi \cdot \sum_{n=1}^{N} \gamma_1^{(n)}}$$

$$\boxed{\Rightarrow \phi^{new} = \frac{\sum_{n=1}^{N} \gamma_1^{(n)}}{N}}$$

$$\nabla_\mu L = \sum_{n=1}^{N} \gamma(Z_0^{(n)}) \cdot \frac{x^{(n)}-\mu}{\sigma} + \sum_{n=1}^{N} \gamma(Z_1^{(n)}) \cdot \frac{x^{(n)}-\mu+1}{\sigma} = 0$$

$$= \sum_{n=1}^{N} (1-\gamma_1^{(n)}) \cdot x^{(n)} - \mu \cdot \sum_{n=1}^{N} (1-\gamma_1^{(n)}) + \sum_{n=1}^{N} \gamma_1^{(n)} x^{(n)} - \mu \sum_{n=1}^{N} \gamma_1^{(n)} + \sum_{n=1}^{N} \gamma_1^{(n)}$$

$$= \sum_{n=1}^{N} x^{(n)} - \cancel{\sum_{n=1}^{N} \gamma_1^{(n)} x^{(n)}} - N\mu + \cancel{\mu \sum_{n=1}^{N} \gamma_1^{(n)}} + \cancel{\sum_{n=1}^{N} \gamma_1^{(n)} x^{(n)}} - \cancel{\mu \sum_{n=1}^{N} \gamma_1^{(n)}} + \sum_{n=1}^{N} \gamma_1^{(n)} = 0$$

$$\Rightarrow \boxed{\mu^{new} = \frac{\sum_{n=1}^{N} \gamma_1^{(n)} + \sum_{n=1}^{N} x^{(n)}}{N} = \frac{N_k + \sum_{n=1}^{N} x^{(n)}}{N}, \text{ where } N_k = \sum_{n=1}^{N} \gamma_1^{(n)}}$$

$$\nabla_\sigma L = \sum_{n=1}^{N} \gamma(z_0^{(n)}) \left[\left(-\frac{1}{\sigma}\right) + \frac{(x-\mu)^2}{\sigma^3} \right] + \sum_{n=1}^{N} \gamma(z_1^{(n)}) \left[\left(-\frac{1}{\sigma}\right) + \frac{(x-\mu+1)^2}{\sigma^3} \right] = 0$$

$$\Rightarrow \sum_{n=1}^{N} (1-\gamma_1^{(n)}) \cdot \left((x-\mu)^2 - \sigma^2 \right) + \sum_{n=1}^{N} \gamma_1^{(n)} \cdot \left((x-\mu+1)^2 - \sigma^2 \right) = 0$$

$$\Rightarrow \sum_{n=1}^{N} (x-\mu)^2 - \sum_{n=1}^{N} \gamma_1^{(n)} (x-\mu)^2 - N\sigma^2 + \cancel{\sum_{n=1}^{N} \gamma_1^{(n)} \sigma^2} + \sum_{n=1}^{N} \gamma_1^{(n)} (x-\mu+1)^2 - \cancel{\sum_{n=1}^{N} \gamma_1^{(n)} \sigma^2} = 0$$

$$\Rightarrow N\sigma^2 = \sum_{n=1}^{N} (1-\gamma_1^{(n)}) \cdot (x^{(n)} - \mu^{new})(x^{(n)} - \mu^{new})^T + \sum_{n=1}^{N} \gamma_1^{(n)} \cdot (x^{(n)} - \mu^{new} + 1)(x^{(n)} - \mu^{new} + 1)^T$$

$$\Rightarrow \boxed{\sigma^{new} = \sqrt{\frac{\sum_{n=1}^{N} (1-\gamma_1^{(n)}) \cdot (x^{(n)} - \mu^{new})(x^{(n)} - \mu^{new})^T + \sum_{n=1}^{N} \gamma_1^{(n)} (x^{(n)} - \mu^{new} + 1)(x^{(n)} - \mu^{new} + 1)^T}{N}}}$$

(d) Because of the introduction of λ, we can control the difference between μ_1 and μ_2.

5.

W matrix

[[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]]}