ECE521: Inference Algorithms and Machine Learning
University of Toronto

Solution to Assignment 3:
Unsupervised Learning and Probabilistic Models

Renjie Liao

March 27, 2017

1 K-means

1.1 Learning K-means [8 pt.]

1, [3 pt.] The loss function $\mathcal{L}(\boldsymbol{\mu})$ is non-convex in $\boldsymbol{\mu}$. We use contradiction to prove the statement. Assuming $\mathcal{L}(\boldsymbol{\mu})$ is convex in $\boldsymbol{\mu}$. We consider a special case where $B = 1$, $K = 2$, $D = 2$, $\mathbf{x} = [x_1, x_2]$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2]$ and $\mathbf{x} \neq \mathbf{0}$. We can rewrite the loss function as $\mathcal{L}(\boldsymbol{\mu}) = \min(\|\mathbf{x} - \boldsymbol{\mu}_1\|_2^2, \|\mathbf{x} - \boldsymbol{\mu}_2\|_2^2)$. Now we will show that for two specific constructions $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2]$ and $\boldsymbol{\mu}' = [\boldsymbol{\mu}_1', \boldsymbol{\mu}_2']$ and any $\theta \in (0, 1)$, we have $\mathcal{L}(\theta\boldsymbol{\mu} + (1-\theta)\boldsymbol{\mu}') > \theta\mathcal{L}(\boldsymbol{\mu}) + (1-\theta)\mathcal{L}(\boldsymbol{\mu}')$. In particular, let $\boldsymbol{\mu}_1 = [x_1, x_2]$, $\boldsymbol{\mu}_2 = [0, 0]$, $\boldsymbol{\mu}_1' = [0, 0]$ and $\boldsymbol{\mu}_2' = [x_1, x_2]$. We have that $\mathcal{L}(\boldsymbol{\mu}) = 0$ and $\mathcal{L}(\boldsymbol{\mu}') = 0$. Moreover, we have,

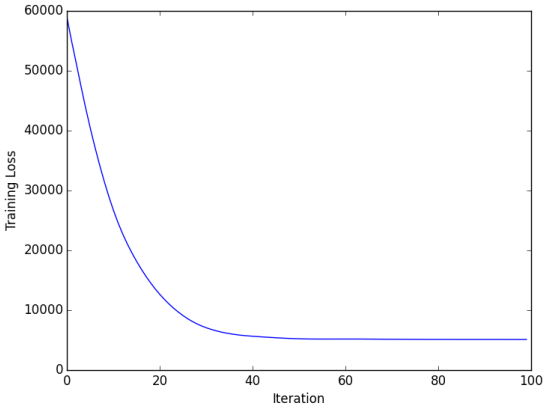
$$\begin{aligned}\mathcal{L}(\theta\boldsymbol{\mu} + (1-\theta)\boldsymbol{\mu}') &= \min(\|\mathbf{x} - \theta\boldsymbol{\mu}_1 - (1-\theta)\boldsymbol{\mu}_1'\|_2^2, \|\mathbf{x} - \theta\boldsymbol{\mu}_2 - (1-\theta)\boldsymbol{\mu}_2'\|_2^2) \\ &= \min((1-\theta)^2\|\mathbf{x}\|^2, \theta^2\|\mathbf{x}\|^2). \end{aligned} \tag{1}$$

It is thus clear that $\mathcal{L}(\theta\boldsymbol{\mu} + (1-\theta)\boldsymbol{\mu}') > \theta\mathcal{L}(\boldsymbol{\mu}) + (1-\theta)\mathcal{L}(\boldsymbol{\mu}') = 0$ for any $\theta \in (0, 1)$. Hence, we have the contradiction.

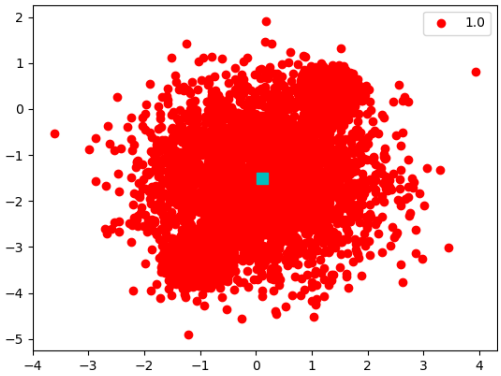
2, [2 pt.] Referring to Fig. 1a.

3, [3 pt.] Referring to Fig. 1b - 1f. K = 5 might be the best as it gives the lowest loss function value.

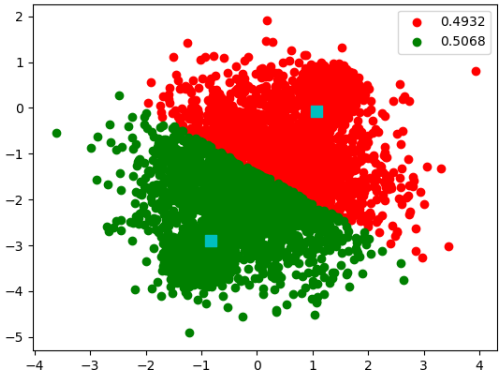
4, [2 pt.] The validation loss values are 12856.2089844, 2982.43554688, 1659.61975098, 1095.4954834, 921.618164062 when $K = 1, 2, 3, 4, 5$ respectively. Therefore, $K = 5$ is the best.



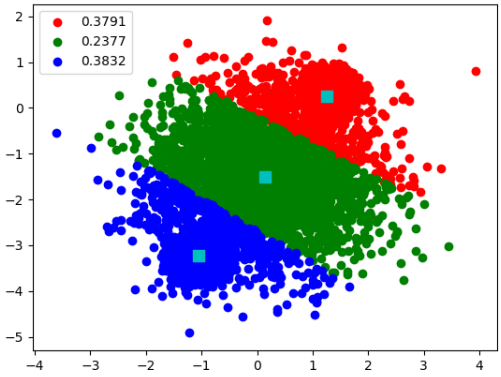
(a) Kmeans, training loss



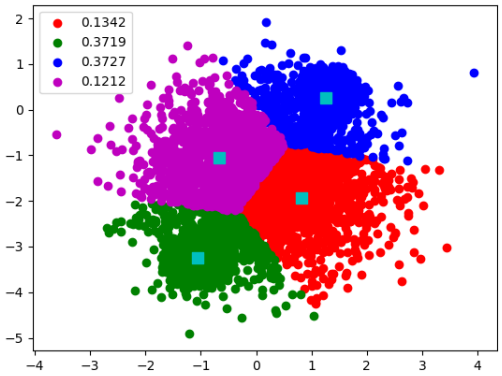
(b) Kmeans, K = 1



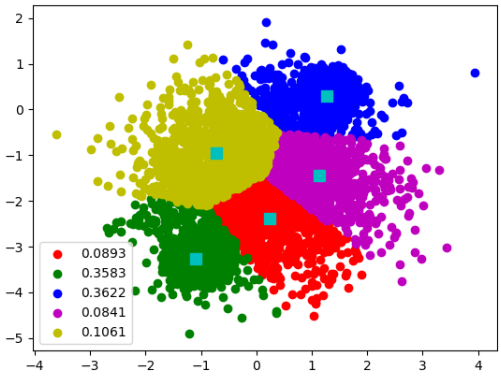
(c) Kmeans, K = 2



(d) Kmeans, K = 3



(e) Kmeans, K = 4



(f) Kmeans, K = 5

```python
def pdist(X, Y):
    """ Compute pair-wise distances between X and Y

    Args:
      X: size M X D
      Y: size N X D
    Returns:
      Dist: size M X N
    """

    YT = tf.transpose(Y)
    Y2 = tf.reduce_sum(tf.square(YT), 0, keep_dims=True)
    X2 = tf.reduce_sum(tf.square(X), 1, keep_dims=True)
    XY = tf.matmul(X, YT)

    return X2 - 2.0 * XY + Y2
```

(a) Distance function.

```python
def log_pdf_mix_gaussian(X, mu, sigma, log_pi):
    """ log pdf of mixture gaussian with covariance sigma * I

      Args:
        X: B X D
        mu: K X D
        sigma: K X 1
        log_pi: K X 1

      Returns:
        log likelihood
    """

    Pi = tf.constant(float(np.pi))
    sigma_2 = tf.transpose(tf.square(sigma))  # K X 1
    diff = ut.pdist(X, mu)  # B X K

    log_likelihood = diff / sigma_2  # B X K
    log_likelihood += DIM * tf.log(2 * Pi)
    log_likelihood += DIM * tf.log(sigma_2)
    log_likelihood *= -0.5
    log_likelihood += tf.transpose(log_pi)
    log_likelihood = ut.reduce_logsumexp(log_likelihood)  # B x 1

    return tf.reduce_sum(log_likelihood)
```

(b) Log probability.

2 Mixtures of Gaussians

2.1 The Gaussian cluster model

1, [3 pt.]

$$P(z|\mathbf{x}) = \frac{\pi^k \left(\sigma^k\right)^{-D} \exp\left\{-\frac{1}{2\left(\sigma^k\right)^2}(\mathbf{x}-\boldsymbol{\mu}^k)^\top(\mathbf{x}-\boldsymbol{\mu}^k)\right\}}{\sum_{j=1}^{K} \pi^j \left(\sigma^j\right)^{-D} \exp\left\{-\frac{1}{2(\sigma^j)^2}(\mathbf{x}-\boldsymbol{\mu}^j)^\top(\mathbf{x}-\boldsymbol{\mu}^j)\right\}} \tag{2}$$

2, [2 pt.] Referring to Fig. 2a.

3, [3 pt.] Referring to Fig. 2b. Using log-sum-exp is numerically more accurate and stable and will help avoid issues like underflow or overflow.

	GMM	Kmeans
k=3	651434.875	1418950.625
k=5	352855.46875	1091327.0
k=10	211011.671875	1304746.25
k=15	209351.46875	486612.15625
k=20	207494.59375	484491.625
k=30	204230.015625	485255.78125

Table 1: Loss values on 100D dataset.

2.2 Learning the MoG

1, [2 pt.]

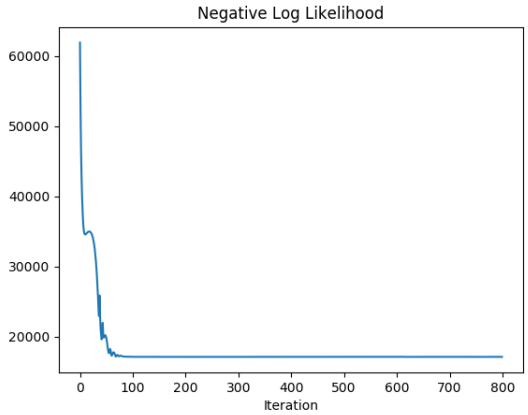
$$\begin{aligned}
\nabla_{\boldsymbol{\mu}} \log P(\mathbf{x}) &= \nabla_{\boldsymbol{\mu}} \log \left(\sum_{k=1}^{K} P(\mathbf{x}, z=k) \right) \\
&= \frac{1}{\sum_{k=1}^{K} P(\mathbf{x}, z=k)} \sum_{k=1}^{K} \nabla_{\boldsymbol{\mu}} P(\mathbf{x}, z=k) \\
&= \frac{1}{\sum_{k=1}^{K} P(\mathbf{x}, z=k)} \sum_{k=1}^{K} P(\mathbf{x}, z=k) \nabla_{\boldsymbol{\mu}} \log P(\mathbf{x}, z=k) \\
&= \sum_{k=1}^{K} P(z=k|\mathbf{x}) \nabla_{\boldsymbol{\mu}} \log P(\mathbf{x}, z=k), \qquad (3)
\end{aligned}$$

where we use the fact $f(x)\nabla_x \log(f(x)) = \nabla_x f(x)$.

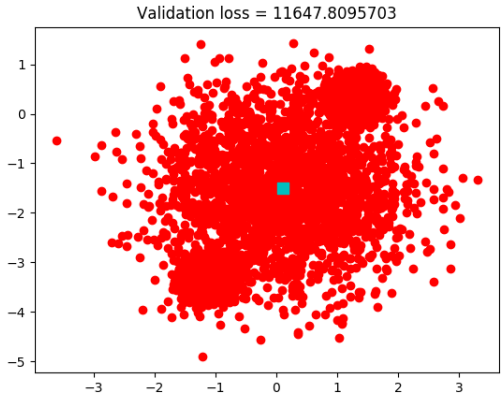
2, [6 pt.] Referring to Fig. 3a. The best model parameters are: $\mu_1 = [-1.09925008 - 3.30307055]$, $\mu_2 = [1.299408670.30630079]$, $\mu_1 = [0.09809401 - 1.5353744]$, $\sigma_1 = 0.19823763$, $\sigma_2 = 0.19667165$, $\sigma_3 = 0.99854809$, $\pi_1 = 0.32968238$, $\pi_2 = 0.33281365$ and $\pi_3 = 0.33750394$.

3, [2 pt.] Referring to Fig. 3b - 3f. $K = 3$ is the best as the validation loss is the minimum.

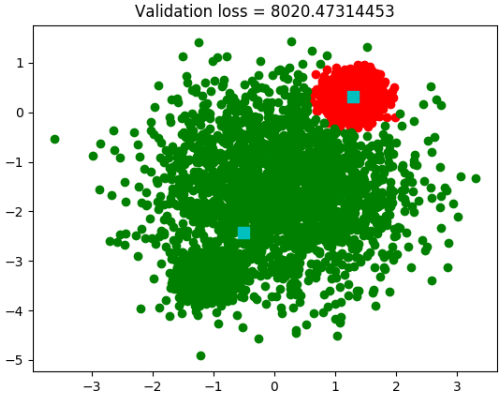
4, [2 pt.] The loss function values of Kmeans and MoG with different K are listed in the table 1. From the current trials, it seems that the number of clusters should be around 20. We visualize the cluster assignments and cluster centers in terms of the first 2 dimensions of the 100D dataset in Fig. 4a and Fig. 4b. It is hard to draw any conclusion from the learned results. It seems that some clusters of GMM are collapsed while the ones of Kmeans are not.



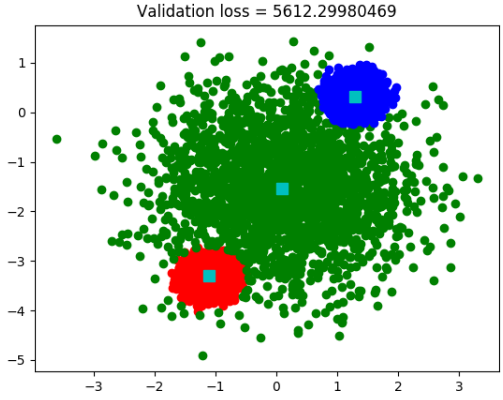
(a) GMM training loss



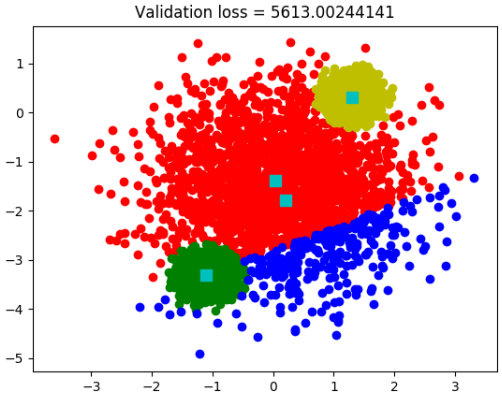
(b) GMM, K = 1



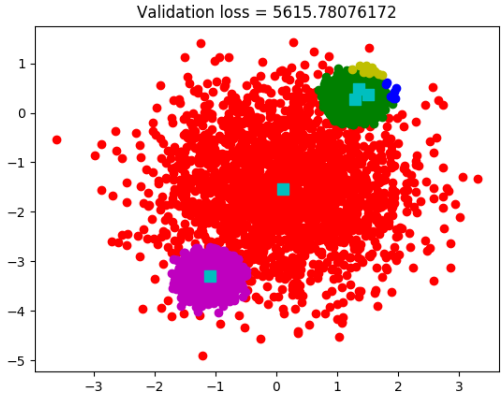
(c) GMM, K = 2



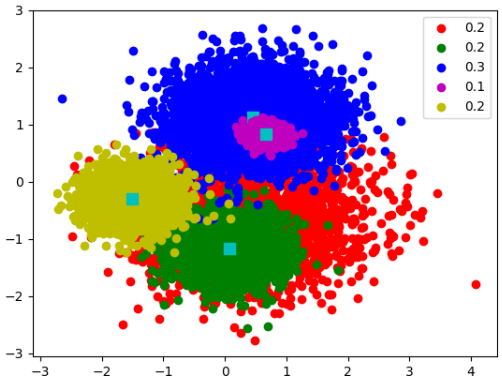
(d) GMM, K = 3



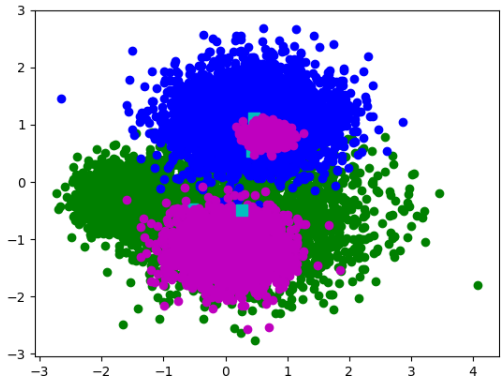
(e) GMM, K = 4



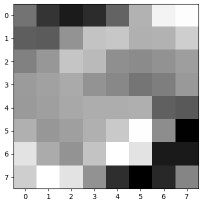
(f) GMM, K = 5



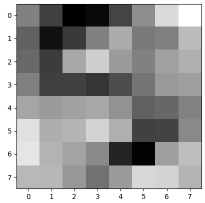
(a) K = 5, 100D, Kmeans



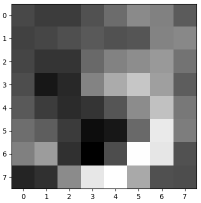
(b) K = 5, 100D, GMM



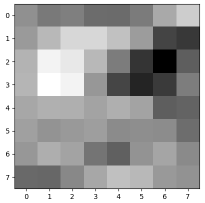
(a) Weight 1



(b) Weight 2



(c) Weight 3



(d) Weight 4

3 Discover Latent Dimensions

3.1 Factor Analysis

1, [2 pt.] First, we have,

$$P(\mathbf{x}) = \int_{\mathbf{z}} P(\mathbf{x}|\mathbf{z})P(\mathbf{z})d\mathbf{z} \tag{4}$$

From the multivariate results appended in the assignment, we have,

$$\begin{aligned} P(\mathbf{z}) &= \mathcal{N}(\mathbf{z}|0, I) \\ P(\mathbf{x}|\mathbf{z}) &= \mathcal{N}(z|W\mathbf{z} + \boldsymbol{\mu}, \Psi) \\ P(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Psi + WW^\top) \end{aligned} \tag{5}$$

2, [3 pt.] The training, validation and testing marginal likelihoods are 8399.542, 1238.54980469 and 4644.74414062 respectively. The visualization of weights are in Fig. 5a - 5d. You can see that the weight capture the some parts of digits. For example, Fig. 5a and 5c captures the shape of "5" and Fig. 5b and 5d captures "3".

3, [3 pt.]

The first component of PCA is $[-2.72359396e^{-04}, -2.61620560e^{-04}, 9.99999929e^{-01}]$ which approximately corresponds to the maximum variance direction, i.e., x_3.

The conditional distribution of $\mathbf{s}$ (posterior) of a new data point $\mathbf{x}^*$ in a FA is: $p(\mathbf{s}|\mathbf{x}^*) = \mathcal{N}\big(\sim; \Sigma W^T \Psi^{-1}(\mathbf{x}^* - \boldsymbol{\mu}), \Sigma\big)$ (Eq(4) of Multivariate Gaussian Results) where $\Sigma = (I + W^T \Psi^{-1} W)^{-1}$. Now let us definite $W_{proj} \triangleq \Sigma W^T \Psi^{-1} = (I + W^T \Psi^{-1} W)^{-1} W^T \Psi^{-1}$. Therefore, W_{proj} is the "principle component" matrix that transforms the observation data to the latent space of FA instead of the W matrix.

The first component of W_{proj} is $[3.48705414e-01, 5.76612226e-01, 4.26386575e-08]$ which is the maximum correlation direction, i.e. $x_1 + x_2$