統計的機械学習
第四回　レポート

37-196360　森田涼介

2019 年 5 月 9 日

宿題 1

ガウス混合分布の最尤推定量が満たす関係式を求める。

ガウス混合モデルは次のように表される。ここで，m は混合数である。

$$q(\boldsymbol{x};\,\boldsymbol{\theta}) = \sum_{j=1}^{m} w_j \phi(\boldsymbol{x};\,\boldsymbol{\mu}_j, \sigma_j) \tag{1}$$

$$w_j \geq 0, \quad \sum_{j=1}^{m} w_j = 1 \tag{2}$$

$$\boldsymbol{\theta} = \begin{bmatrix} w_1 & \cdots & w_m & \boldsymbol{\mu}_1^{\mathrm{T}} & \cdots & \boldsymbol{\mu}_m^{\mathrm{T}} & \sigma_1 & \cdots & \sigma_m \end{bmatrix}^{\mathrm{T}} \tag{3}$$

$$w_j \in \mathbb{R},\ \boldsymbol{\mu}_j \in \mathbb{R}^d,\ \sigma_j \in \mathbb{R}_{>0} \tag{4}$$

$$\phi(\boldsymbol{x};\,\boldsymbol{\mu}, \sigma) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2} (\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{\mu}) \right) \tag{5}$$

いま，w_j を $\gamma_j \in \mathbb{R}$ を用いて次のように表すことで，式 (2) の w_j の拘束条件を自動的に満たすことができる。

$$w_j = \frac{\exp(\gamma_j)}{\sum_{j'=1}^{m} \exp(\gamma_{j'})} \tag{6}$$

いま，$q(\boldsymbol{x}_i;\,\boldsymbol{\theta}) = q_i$, $\phi(\boldsymbol{x}_i;\,\boldsymbol{\mu}_j, \sigma_j) = \phi_{ij}$ と表すと，対数尤度は次のように表される。

$$\begin{aligned} \log(L(\boldsymbol{\theta})) &= \sum_{i=1}^{n} \log q_i & (7) \\ &= \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} w_j \phi_{ij} \right\} & (8) \end{aligned}$$

最尤推定量を与える $\boldsymbol{\theta}$ について，$\partial \log L / \partial \boldsymbol{\theta} = \mathbf{0}$ が成り立つから，$j = 1, \cdots, m$ について次式が成り立つ。

$$\frac{\partial \log L}{\partial \gamma_j} = 0, \quad \frac{\partial \log L}{\partial \boldsymbol{\mu}_j} = \mathbf{0}, \quad \frac{\partial \log L}{\partial \sigma_j} = 0 \tag{9}$$

また，対数尤度の，$\boldsymbol{\theta}$ のうち一部のパラメータ $\boldsymbol{\psi}$ による偏微分は次のようになる。

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\psi}} (\log(L(\boldsymbol{\theta}))) &= \frac{\partial}{\partial \boldsymbol{\psi}} \left(\sum_{i=1}^{n} \log q_i \right) & (10) \\ &= \sum_{i=1}^{n} \frac{1}{q_i} \frac{\partial q_i}{\partial \boldsymbol{\psi}} & (11) \end{aligned}$$

まず，対数尤度の γ_j による偏微分を考える。いま，式 (6) から，

$$\begin{aligned}
\frac{\partial w_j}{\partial \gamma_j} &= \frac{\partial}{\partial \gamma_j}\left(\frac{\exp(\gamma_j)}{\sum_{j'=1}^{m}\exp(\gamma_{j'})}\right) && (12)\\
&= \exp(\gamma_j)\cdot\frac{1}{\sum_{j'=1}^{m}\exp(\gamma_{j'})} + \exp(\gamma_j)\cdot\frac{-\exp(\gamma_j)}{\left(\sum_{j'=1}^{m}\exp(\gamma_{j'})\right)^2} && (13)\\
&= \frac{\exp(\gamma_j)}{\sum_{j'=1}^{m}\exp(\gamma_{j'})} - \left(\frac{\exp(\gamma_j)}{\sum_{j'=1}^{m}\exp(\gamma_{j'})}\right)^2 && (14)\\
&= w_j - {w_j}^2 && (15)\\
&= w_j(1-w_j) && (16)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial w_k}{\partial \gamma_j} &= \frac{\partial}{\partial \gamma_j}\left(\frac{\exp(\gamma_k)}{\sum_{j'=1}^{m}\exp(\gamma_{j'})}\right) && (17)\\
&= \exp(\gamma_k)\left(\frac{-\exp(\gamma_j)}{\left(\sum_{j'=1}^{m}\exp(\gamma_{j'})\right)^2}\right) && (18)\\
&= -\frac{\exp(\gamma_j)}{\sum_{j'=1}^{m}\exp(\gamma_{j'})}\cdot\frac{\exp(\gamma_k)}{\sum_{j'=1}^{m}\exp(\gamma_{j'})} && (19)\\
&= -w_j w_k && (20)
\end{aligned}$$

よって，q_i の γ_j による偏微分は，

$$\begin{aligned}
\frac{\partial q_i}{\partial \gamma_j} &= \frac{\partial}{\partial \gamma_j}\left(\sum_{j=1}^{m} w_j\phi_{ij}\right) && (21)\\
&= \sum_{k=1,k\neq j}^{m}(-w_j w_k)\phi_{ik} + w_j(1-w_j)\phi_{ij} && (22)\\
&= -w_j\sum_{k=1}^{m}(w_k\phi_{ik}) + w_j\phi_{ij} && (23)\\
&= w_j\phi_{ij} - w_j q_i && (24)
\end{aligned}$$

従って，式 (11) を用いることで，対数尤度の γ_j による偏微分は次のようになる。

$$\begin{aligned}
\frac{\partial}{\partial \gamma_j}(\log(L(\boldsymbol{\theta}))) &= \sum_{i=1}^{n}\frac{1}{q_i}\frac{\partial q_i}{\partial \gamma_j} && (25)\\
&= \sum_{i=1}^{n}\frac{1}{q_i}(w_j\phi_{ij} - w_j q_i) && (26)\\
&= \sum_{i=1}^{n}\left(\frac{w_j\phi_{ij}}{q_i} - w_j\right) && (27)
\end{aligned}$$

ここで，

$$\eta_{ij} = \frac{w_j \phi_{ij}}{q_i} = \frac{w_j \phi(\boldsymbol{x}_i;\ \boldsymbol{\mu}_j, \sigma_j)}{q(\boldsymbol{x}_i;\ \boldsymbol{\theta})} \tag{28}$$

$$= \frac{w_j \phi(\boldsymbol{x}_i;\ \boldsymbol{\mu}_j, \sigma_j)}{\sum_{j'=1}^{b} w_{j'} \phi(\boldsymbol{x}_i;\ \boldsymbol{\mu}_{j'}, \sigma_{j'})} \tag{29}$$

なる η_{ij} を用いると，結局，

$$\frac{\partial}{\partial \gamma_j}(\log(L(\boldsymbol{\theta}))) = \sum_{i=1}^{n} \eta_{ij} - n w_j \tag{30}$$

これが 0 となるときに最尤推定量となるので，

$$\hat{w}_j = \frac{1}{n} \sum_{i=1}^{n} \hat{\eta}_{ij} \tag{31}$$

を得る。

次に，対数尤度の $\boldsymbol{\mu}_j$ による偏微分を考える。ϕ_{ij} の $\boldsymbol{\mu}_j$ による偏微分は次のようになる。

$$\frac{\partial \phi_{ij}}{\partial \boldsymbol{\mu}_j} = \frac{\partial}{\partial \boldsymbol{\mu}_j}\left(\frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)\right) \tag{32}$$

$$= \frac{1}{(2\pi\sigma_j^2)^{d/2}} \cdot \left(-\frac{1}{2\sigma_j^2}\right) \cdot 2(\boldsymbol{\mu}_j - \boldsymbol{x}_i) \cdot \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right) \tag{33}$$

$$= \frac{1}{\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\phi_{ij} \tag{34}$$

よって，q_i の $\boldsymbol{\mu}_j$ による偏微分は，

$$\frac{\partial q_i}{\partial \boldsymbol{\mu}_j} = \frac{\partial}{\partial \boldsymbol{\mu}_j}\left(\sum_{j=1}^{m} w_j \phi_{ij}\right) \tag{35}$$

$$= w_j \frac{\partial \phi_{ij}}{\partial \boldsymbol{\mu}_j} \tag{36}$$

$$= w_j \frac{1}{\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\phi_{ij} \tag{37}$$

$$= w_j \phi_{ij} \cdot \frac{1}{\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) \tag{38}$$

式 (11) から，対数尤度の $\boldsymbol{\mu}_j$ による偏微分は次のようになる。

$$\frac{\partial}{\partial \boldsymbol{\mu}_j}(\log(L(\boldsymbol{\theta}))) = \sum_{i=1}^{n} \frac{1}{q_i} \frac{\partial q_i}{\partial \boldsymbol{\mu}_j} \tag{39}$$

$$= \sum_{i=1}^{n} \frac{1}{q_i}\left(w_j \phi_{ij} \cdot \frac{1}{\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right) \tag{40}$$

$$= \frac{1}{\sigma_j^2}\left(\sum_{i=1}^{n} \frac{w_j \phi_{ij}}{q_i}\boldsymbol{x}_i - \boldsymbol{\mu}_j \sum_{i=1}^{n} \frac{w_j \phi_{ij}}{q_i}\right) \tag{41}$$

$$= \frac{1}{\sigma_j^2}\left(\sum_{i=1}^{n} \eta_{ij}\boldsymbol{x}_i - \boldsymbol{\mu}_j \sum_{i=1}^{n} \eta_{ij}\right) \tag{42}$$

これが $\mathbf{0}$ となるときに最尤推定量となるので，

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{n} \hat{\eta}_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{n} \hat{\eta}_{ij}} \tag{43}$$

を得る。

最後に，対数尤度の σ_j による偏微分を考える。ϕ_{ij} の σ_j による偏微分は次のようになる。

$$\frac{\partial \phi_{ij}}{\partial \sigma_j} = \frac{\partial}{\partial \sigma_j}\left(\frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)\right) \tag{44}$$

$$= \frac{1}{(2\pi)^{d/2}} \frac{-d}{{\sigma_j}^{d+1}} \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right) \tag{45}$$

$$+ \frac{1}{(2\pi)^{d/2}} \frac{1}{{\sigma_j}^{d}} \left\{-\frac{-2}{2{\sigma_j}^3}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\} \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right) \tag{46}$$

$$= \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)\left\{-\frac{d}{\sigma_j} + \frac{1}{{\sigma_j}^3}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\} \tag{47}$$

$$= \phi_{ij}\left\{-\frac{d}{\sigma_j} + \frac{1}{{\sigma_j}^3}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\} \tag{48}$$

よって，q_i の σ_j による偏微分は，

$$\frac{\partial q_i}{\partial \sigma_j} = \frac{\partial}{\partial \sigma_j}\left(\sum_{j=1}^{m} w_j \phi_{ij}\right) \tag{49}$$

$$= w_j \frac{\partial \phi_{ij}}{\partial \sigma_j} \tag{50}$$

$$= w_j \phi_{ij}\left\{-\frac{d}{\sigma_j} + \frac{1}{{\sigma_j}^3}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\} \tag{51}$$

式 (11) から，対数尤度の σ_j による偏微分は次のようになる。

$$\frac{\partial}{\partial \sigma_j}(\log(L(\boldsymbol{\theta}))) = \sum_{i=1}^{n} \frac{1}{q_i} \frac{\partial q_i}{\partial \sigma_j} \tag{52}$$

$$= \sum_{i=1}^{n} \frac{1}{q_i} w_j \phi_{ij}\left\{-\frac{d}{\sigma_j} + \frac{1}{{\sigma_j}^3}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\} \tag{53}$$

$$= \frac{1}{{\sigma_j}^3} \sum_{i=1}^{n} \frac{w_j \phi_{ij}}{q_i}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) - \frac{d}{\sigma_j} \sum_{i=1}^{n} \frac{w_j \phi_{ij}}{q_i} \tag{54}$$

$$= \frac{1}{{\sigma_j}^3} \sum_{i=1}^{n} \eta_{ij}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) - \frac{d}{\sigma_j} \sum_{i=1}^{n} \eta_{ij} \tag{55}$$

これが $\mathbf{0}$ となるときに最尤推定量となるので，

$$\hat{\sigma}_j^2 = \frac{1}{d} \frac{\sum_{i=1}^{n} \hat{\eta}_{ij}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)}{\sum_{i=1}^{n} \hat{\eta}_{ij}} \tag{56}$$

$$\hat{\sigma}_j = \sqrt{\frac{1}{d} \frac{\sum_{i=1}^{n} \hat{\eta}_{ij}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)}{\sum_{i=1}^{n} \hat{\eta}_{ij}}} \tag{57}$$

を得る。

以上，式 (29)，(31)，(43)，(57) が，ガウス混合分布の最尤推定量の関係式である。

宿題 2

ガウス混合モデルの EM アルゴリズムを実装し，適当な一次元の確率密度関数を推定する。

今回は，平均 0・分散 1 の正規分布からのサンプリングのうち，7 割に 2 を足し，3 割に -2 を足した分布を用いる。サンプル数 1000 点，および 10000 点について実験を行った。この分布のサンプル数 1000 のときのヒストグラムを図 1 に示す。なお，このヒストグラムは正規化されていることに注意されたい。

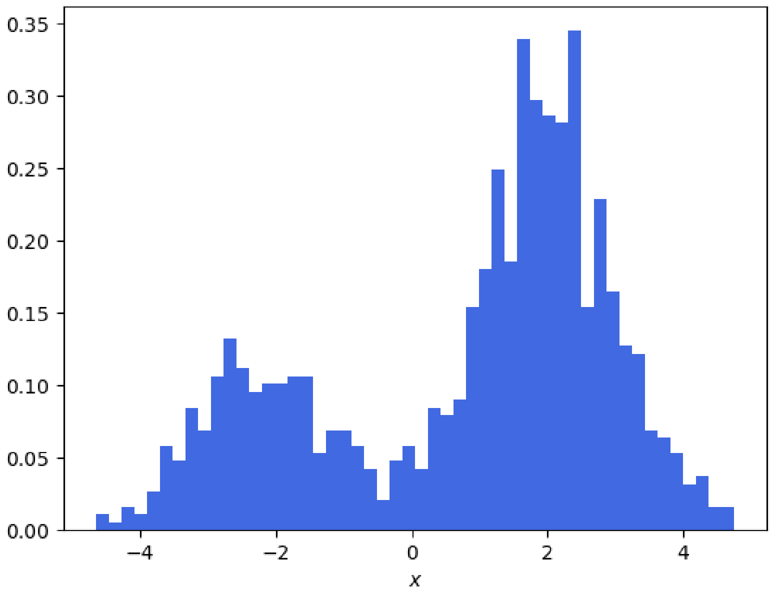

図 1: 実データのヒストグラム（サンプル数 1000）

パラメータの初期化は一様分布により行った。w_j は総和が 1 となるようにし，$\boldsymbol{\mu}$ は $[-1.5, 1.5)$，$\boldsymbol{\sigma}$ は $[0, 1)$ の一様分布を用いた。また，実データを見ると，2 つの正規分布で近似するのが妥当だと考えられるため，$m = 2$ とした。

結果は図 2，3 に示した。また，収束時のパラメータを表 1 に示す。データの作り方から考えて，平均 2・分散 1 の正規分布と平均 -2・分散 1 の正規分布が $7:3$ で重ね合わせられている分布が真の分布であると考えられる。表 1 を見ると，特に 10000 点の場合は，実際そのようなパラメータに近くなっていることがわかる。

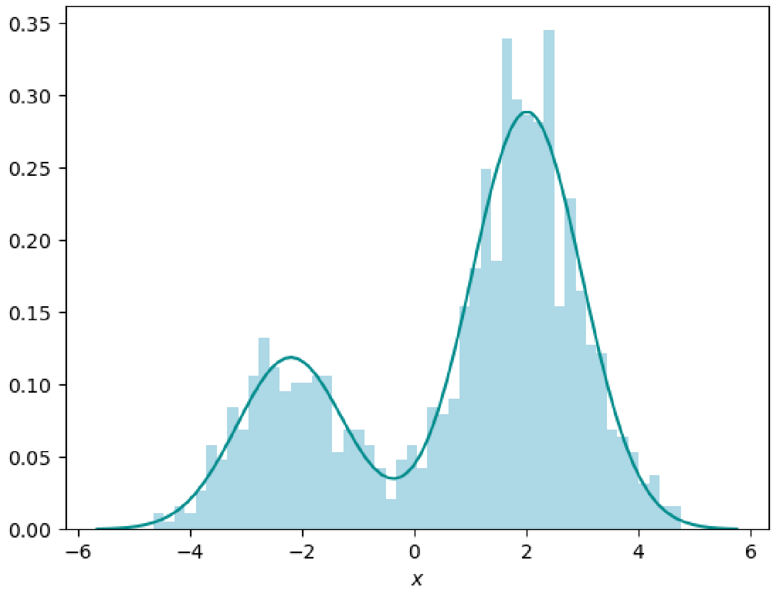

図 2: サンプル数 1000 のときの実データのヒストグラムとガウス混合モデル

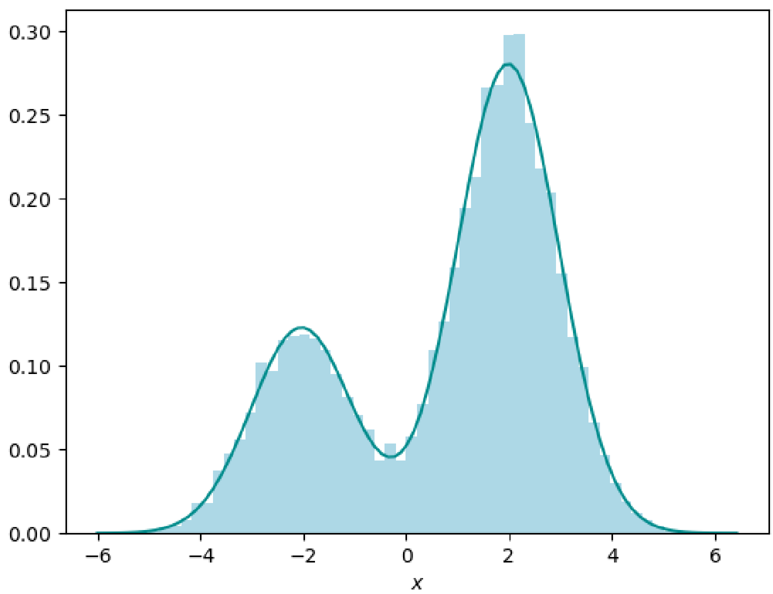

図 3: サンプル数 10 000 のときの実データのヒストグラムとガウス混合モデル

表 1: 収束時のパラメータと Q の値

n	Q	Q/n	w_1	w_2	μ_1	μ_2	σ_1	σ_2
1000	-1996	-1.996	0.7138	0.2862	2.013	-2.201	0.9054	0.9614
10000	-20194	-2.019	0.3020	0.6980	-2.041	1.980	0.9797	0.9916

プログラム

実行環境と用いた言語・ライブラリを以下の表 2 に示す。

表 2: プログラムの実行環境

OS	:	Microsoft Windows 10 Pro (64bit)
CPU	:	Intel(R) Core(TM) i5-4300U
RAM	:	4.00 GB
使用言語	:	Python3.6
可視化	:	matplotlib ライブラリ

Listings 1: assignment2.py

```
# -*- coding: utf-8 -*-


import numpy as np
import matplotlib.pyplot as plt


def generate_sample(n):
    sample = (np.random.randn(n) + np.where(np.random.rand(n) > 0.3, 2.,
    -2.))
    return sample


def gauss_dist(x, mu, sigma):
    d = mu.shape[0]
    phi = (1/(2 * np.pi * sigma**2)**(d/2)) * np.exp(-(1/(2*sigma**2) * (x -
    mu)**2))
    return phi


def calc_w_phi(x, w, mu, sigma):
    phi_list = []
    m = mu.shape[0]
    for j_dash in range(m):
        phi_list.append(gauss_dist(x, mu[j_dash], sigma[j_dash]))
    phi_list = np.array(phi_list).T
    w_phi = w * phi_list
    return w_phi


def gaussian_mixture_model(x, w, mu, sigma):
    w_phi = calc_w_phi(x, w, mu, sigma)
```

```
    q = np.sum(w_phi, axis=1)
    return q


def calc_eta(x, w, mu, sigma):
    w_phi = calc_w_phi(x, w, mu, sigma)
    w_phi_sum = w_phi.sum(axis=1)
    eta = w_phi
    for j in range(eta.shape[1]):
        eta[:, j] /= w_phi_sum
    return eta


def update_w(j, eta):
    w_new = np.mean(eta[:, j])
    return w_new


def update_mu(j, eta, x):
    mu_new = (np.sum(eta[:, j] * x)) / np.sum(eta[:, j])
    return mu_new


def update_sigma(j, eta, x, mu):
    d = mu.shape[1]
    sigma_new_2 = (1/d) * (np.sum(eta[:, j] * (x - mu[j])**2))/
    np.sum(eta[:, j])
    sigma_new = np.sqrt(sigma_new_2)
    return sigma_new


def update(x, w, mu, sigma):
    m = w.shape[0]
    eta = calc_eta(x, w, mu, sigma)
    w_new = np.empty_like(w)
    mu_new = np.empty_like(mu)
    sigma_new = np.empty_like(sigma)
    for j in range(m):
        w_new[j] = update_w(j, eta)
        mu_new[j] = update_mu(j, eta, x)
        sigma_new[j] = update_sigma(j, eta, x, mu)
    return w_new, mu_new, sigma_new


def calc_Q(x, w, mu, sigma):
    w_phi = calc_w_phi(x, w, mu, sigma)
    eta = calc_eta(x, w, mu, sigma)
```

```
    Q = np.sum(np.sum(eta * np.log(w_phi), axis=1))
    return Q


def train(x, w, mu, sigma, eps=1e-4, n_converge=5, max_iter=100, show_log=True,):
    n = x.shape[0]
    QbyN_list = []
    for idx_iter in range(max_iter):
        w, mu, sigma = update(x, w, mu, sigma)
        Q = calc_Q(x, w, mu, sigma)
        QbyN = Q/n
        if show_log:
            print('Iter: {} \t Q: {:.1f} \t Q/n: {:.4f}'.format(idx_iter+1, Q, QbyN))
        if len(QbyN_list) < n_converge:
            QbyN_list.append(QbyN)
        else:
            QbyN_list = QbyN_list[1:] + [QbyN]
            if (max(QbyN_list) - min(QbyN_list)) < eps:
                break
    n_iter = idx_iter + 1
    return w, mu, sigma, n_iter


def main():
    # settings
    n = 10000
    m = 2
    d = 1
    result_path = '../figures/assignment2_result_n{}.png'.format(n)
    offset = 1.0
    np.random.seed(0)

    # data
    sample = generate_sample(n)

    # init params
    w = np.random.rand(m)
    w = w / w.sum()
    mu = (np.random.rand(m, d) - 0.5) * 6
    sigma = np.random.rand(m)

    print('Init')
    print('w: {}'.format(w))
    print('sigma: {}'.format(sigma))
    print('mu: \n{}'.format(mu))
```

```
    print()

    # train
    w, mu, sigma, n_iter = train(
        sample, w, mu, sigma,
        eps=1e-4, n_converge=5, max_iter=100,
        show_log=True,
        )
    Q = calc_Q(sample, w, mu, sigma)

    # result
    print()
    print('Result')
    print('n: {} \t Iter: {}'.format(n, n_iter))
    print('Q: {} \t Q/n: {}'.format(Q, Q/n))
    print('w: {} (sum = {})'.format(w, w.sum()))
    print('sigma: {}'.format(sigma))
    print('mu: \n{}'.format(mu))
    print()

    # plot
    x_axis = np.linspace(sample.min()-offset, sample.max()+offset, 100)
    q = gaussian_mixture_model(x_axis, w, mu, sigma)
    plt.plot(x_axis, q, color='darkcyan')
    plt.hist(sample, bins=50, normed=True, color='lightblue')
    plt.xlabel('$x$')
    plt.savefig(result_path)
    plt.show()


if __name__ == '__main__':
    main()
```