

統計的機械学習レポート

工学部電子情報工学科3年 03190449 堀 紡希

5月14日

宿題1 ガウス混合分布の尤度方程式

$$\frac{d \log L(\boldsymbol{\theta})}{d \boldsymbol{\theta}} = 0$$

を解く。 μ_j について偏微分すると

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \mu_j} = \sum \frac{\hat{w}_j \phi(x_i; \hat{\mu}_j, \hat{\sigma}_j)}{\sum w_{j'} \phi(x_i; \hat{\mu}_{j'}, \hat{\sigma}_{j'})} \frac{x_i - \hat{\mu}_j}{\sigma^2} = 0$$

$$\hat{\eta}_{i,j} = \frac{\hat{w}_j \phi(x_i; \hat{\mu}_j, \hat{\sigma}_j)}{\sum \phi(x_i; \hat{\mu}_{j'}, \hat{\sigma}_{j'})}$$

と置くと

$$\hat{\mu}_j = \frac{\sum_{i=1}^n \hat{\eta}_{i,j} x_i}{\sum_{i'=1}^n \hat{\eta}_{i',j}}$$

次に σ で偏微分して整理すると先程と同様に

$$\hat{\sigma}_j = \sqrt{\frac{1}{d} \frac{\sum_{i=1}^n \hat{\eta}_{i,j} (x_i - \hat{\mu}_j)^T (x_i - \hat{\mu}_j)}{\sum_{i'=1}^n \hat{\eta}_{i',j}}}$$

が得られる。

\hat{w}_j の導出はわかりませんでした。

宿題2 Python で EM アルゴリズムを実装した。

```
import numpy as np
import math
import matplotlib.pyplot as plt
np.random.seed(0)
def data_generate(n):
    return (np.random.randn(n)
            + np.where(np.random.rand(n) > 0.3, 2., -2.))
    def gauss(x, u, s):
    return math.exp(-1*((x-u)**2)/(2*(s**2)))/math.sqrt((2*math.pi*(s**2)))
```

```

def tau(x, w, u, s, i, j):
    m = len(u)
    sumt = w[0]*gauss(data[i], u[0], s[0])
    for k in range(1, m):
        sumt += w[k]*gauss(data[i], u[k], s[k])
    return (w[j]*gauss(data[i], u[j], s[j]))/sumt

def Gauss_EM(data, m, study):
    n = len(data)
    w = []
    u = []
    s = []
    tw = []
    tu = []
    ts = []
    for i in range(0, m):
        w.append(1+i)
        tw.append(1.)
        u.append(1+i)
        tu.append(1.)
        s.append(1+i)
        ts.append(1.)
    for t in range(0, study):
        Q = 0
        #Estep
        for i in range(0, n):
            for j in range(0, m):
                taut = tau(data, w, u, s, i, j)
                Q += taut*(math.log(w[j]) + math.log(gauss(data[i], u[j], s[j])))
        #Mstep
        for i in range(0, m):
            tw[i] = w[i]
            tu[i] = u[i]
            ts[i] = s[i]
            w[i] = 0
            u[i] = 0
            s[i] = 0
        for j in range(0, m):
            for i in range(0, n):
                w[j] += tau(data, tw, tu, ts, i, j)
                u[j] += tau(data, tw, tu, ts, i, j)*data[i]
                s[j] += tau(data, tw, tu, ts, i, j)*((data[i]-tu[j])**2)
            w[j] *= 1/n
            sum1 = 0
            for i in range(0, n):
                sum1 += tau(data, tw, tu, ts, i, j)
            u[j] *= 1/sum1
            s[j] *= 1/sum1
            s[j] = math.sqrt(s[j])
    return (w, u, s)

```

```

data = data_generate(1000)
theta = Gauss_EM(data, 3, 30)
print(theta)
plt.hist(data, bins = 30)
plt.show()
x = np.linspace(-4, 6, 1000)
X = []
Y = []
for s in x:
    X.append(s)
    Y.append((theta[0][0]*gauss(s, theta[1][0], theta[2][0])+theta[0][1]*gauss(s, the
plt.scatter(X, Y, c = 'red')
plt.show()

```

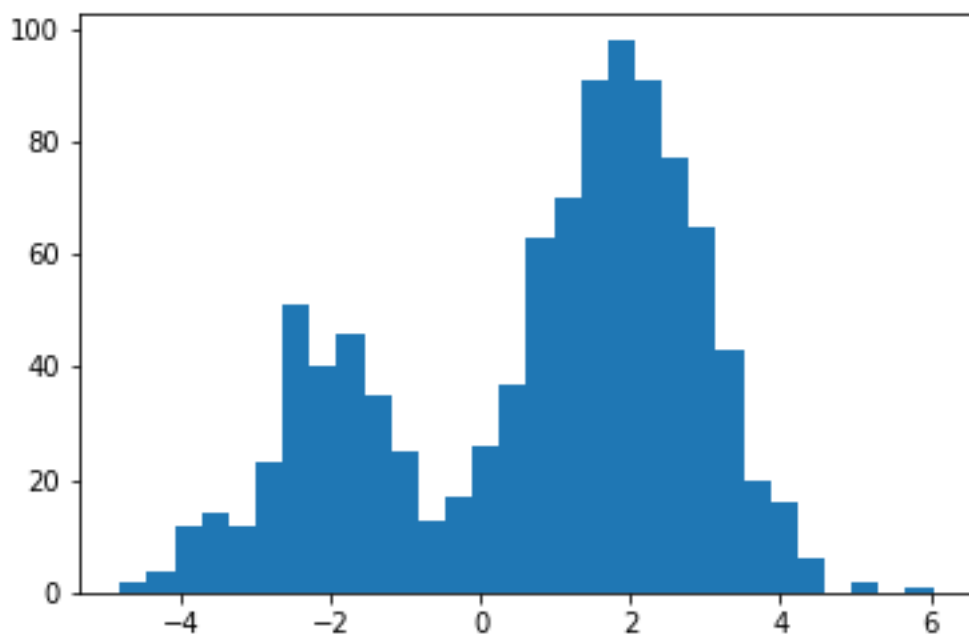


Figure 1: 学習データ

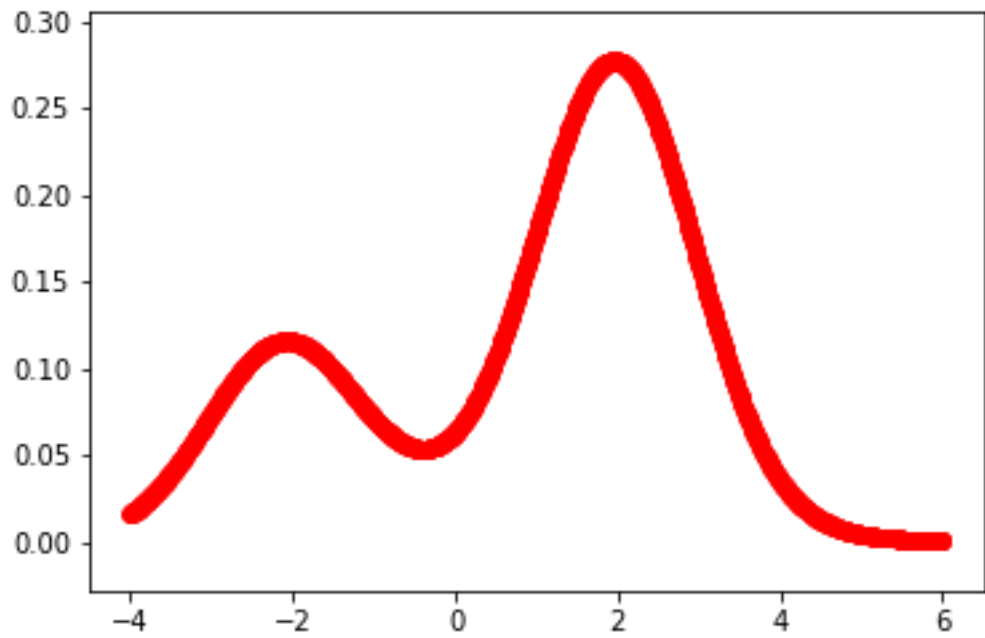


Figure 2: 学習結果

結果 実装するとこのようになった。今回は3つのガウス分布の和をモデルとしたが、確かにガウス分布の和で学習データをうまく表現していると言えるだろう。