Homework 3

1. Normal-normal-normal.
(a) We can write the log posterior as follows:

$$\log P\left(\mu, \sigma^2 | x\right) = \frac{N}{2}\log\sigma^2 - \frac{\sum_{i=1}^{N}\left(y_i-\mu\right)^2}{2\sigma^2} - \frac{\left(\mu-\mu_0\right)^2}{2\sigma_0^2} - (a+1)\log\sigma^2 - \frac{\beta}{\sigma^2} + \text{const.}$$

To further get the MAP esitmators, we calculate the corresponding gradient w.r.t μ and σ^2 as follows:

$$\begin{aligned}\frac{\partial \log P}{\partial \mu} &= -\frac{\sum_{i=1}^{N}\left(\mu-y_i\right)}{\sigma^2} - \frac{\left(\mu-\mu_0\right)}{\sigma_0^2}\\ \frac{\partial \log P}{\partial \sigma^2} &= -(\frac{n}{2}+\alpha+1)(\sigma^2)^{-1} + (\frac{\sum_{i=1}^{N}\left(y_i-\mu\right)^2}{2}+\beta)(\sigma^2)^{-2}\end{aligned} \tag{1}$$

The exact analytical solution is hard to get, so in the coding part, I leverage a numerical solver `SymPy` (`https://docs.sympy.org/`) in Python to solve for $\hat{\mu}$ and $\hat{\sigma^2}$ by equating (1) to zero. Also, optimization methods like gradient descent or Newton's methods could serve as alternatives to solve the MAP problem. The estimation is as follows:

$$\begin{aligned}\hat{\mu} &= 49.63\\ \hat{\sigma^2} &= 9.06\end{aligned} \tag{2}$$

The we can calculate the second order derivatives (Hessian) as follows:

$$\begin{aligned}\frac{\partial^2 \log P}{\partial \mu^2} &= -\frac{N}{\sigma^2} - \frac{1}{\sigma_0^2}\\ \frac{\partial^2 \log P}{\partial \mu\sigma^2} &= \frac{\sum_{i=1}^{N}(\mu-y_i)}{\sigma^4}\\ \frac{\partial^2 \log P}{\partial (\sigma^2)^2} &= (\frac{n}{2}+\alpha+1)(\sigma^2)^{-2} - (\frac{\sum_{i=1}^{N}\left(y_i-\mu\right)^2}{2}+\beta)(\sigma^2)^{-3}\end{aligned} \tag{3}$$

By calculating the negative inverse using Python, we can get the following covariance matrix:

$$\Sigma = \begin{bmatrix} 0.0906 & -0.0077 \\ -0.0077 & 1.550 \end{bmatrix} \tag{4}$$

Finally, we create the coutour plots via `Matplotlib` (`https://matplotlib.org/`) in Python.

The pseudo code is as follows (simple, since the derivation has been shown above):

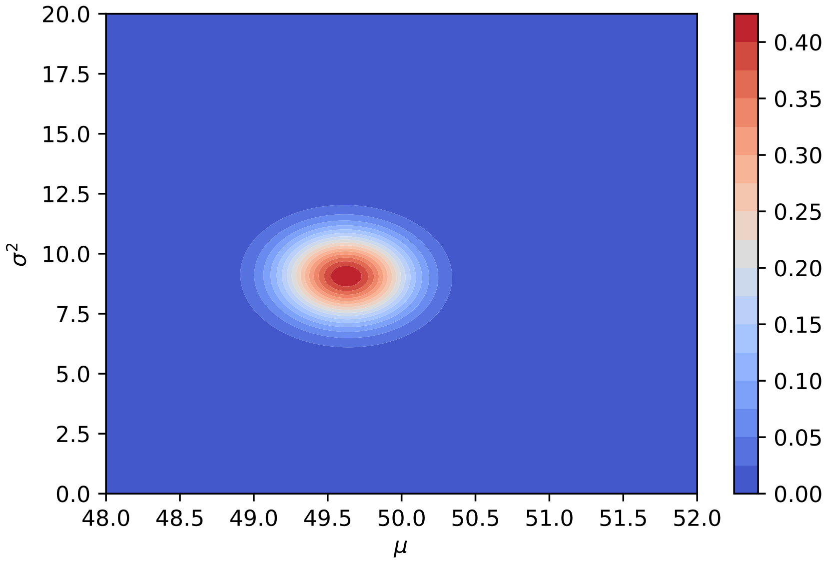

Figure 1: 2D contour of the estimated Gaussian via MAP.

0 Sample N=100 sample points from the ground truth distribution.

1 Derive the MAP estimators in (2).

2 Derive the Hessian of log posterior in (3).

3 Calculate the negative inverse of the Hessian and evaluate the value at MAP in (4).

4 Plot 2D contour of the estimated Gaussian in Figure 1:

  - Directly draw from 2D Gaussian in a proper range.
  - Plot the 2D contour.

(2) For mean field variational inference, the derivation has been shown in Babak's note `https://ucla-biostats-202c.github.io/reading/VI.pdf`. So I just show the pseudo code below:

0 Reuse generated samples from (a).

1 Initialize $m = m_0$, $v = \sigma_0$, $b = \beta$ (my personal choice).

2 Step total iteration steps $L = 1000$.

- **for** $\ell = 1$ to L, do:

$$
\begin{aligned}
m^{(\ell+1)} &= \frac{\sum_{i=1}^{N} y_i + \frac{b}{a}\frac{\mu_0}{\sigma_0^2}}{\left(N + \frac{b}{a}\frac{1}{\sigma_0^2}\right)} \\
v^{(\ell+1)} &= \sqrt{\frac{\frac{b^{(\ell)}}{a}}{N + \frac{b^{(\ell)}}{a}\frac{1}{\sigma_0^2}}} \\
b^{(\ell+1)} &= \frac{a\left[n\left[v^{(\ell+1)}\right]^2 + \sum_{i=1}^{N}\left(y_i - m^{(\ell+1)}\right)^2 + 2\beta\right]}{2\alpha + N}
\end{aligned}
\tag{5}
$$

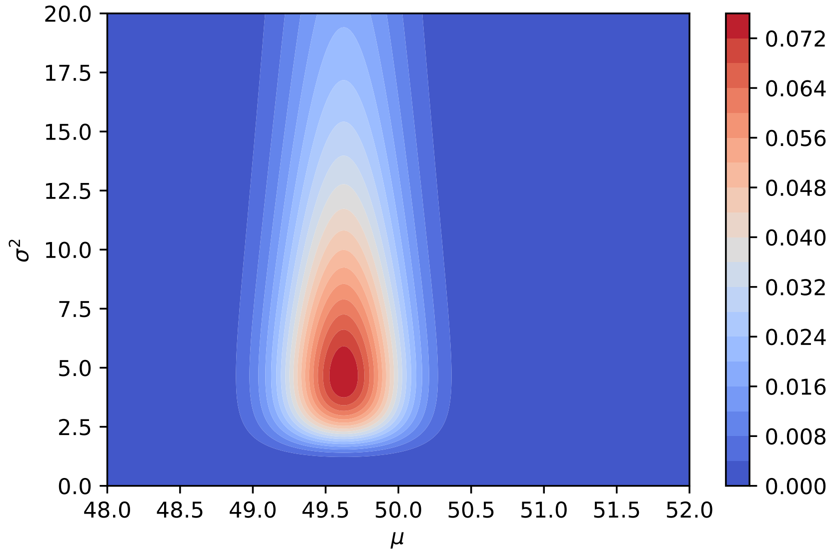

Figure 2: 2D contour of the estimated mean field distribution from variational inference.

3 Plot 2D contour of mean field distribution in Figure 2:

- Independently draw samples from Normal and Gamma in a proper range, respectively.
- Calculate the corresponding products of the probability densities.
- Plot the 2D contour.

2. 2D Clutter problem.
(a) Assumed density filtering.
The derivation has been shown in the lecture note (`https://ucla-biostats-202c.github.io/reading/Notes2.pdf`). The pseudo code is as follows:

0 Sample $N = 200$ sample points from the mixture distribution.

- Generate $w_1, \ldots, w_{200}$ from Bernoulli(0.5).
- **For** $n = 1$ to N:
  if $w_n == 1$: Generate y_n from the first mixture component.
  else: Generate y_n from the second mixture component.

1 Initialize $\mathbf{m}_\theta = \mathbf{0}, v_\theta^{-n} = 100, s = 1$.

2 **For** $n = 1, \ldots, N$, do the following update:

$$s = s^{-n} z_n$$

$$\pi_n = 1 - \frac{w}{z_n} \mathrm{N}\left(\mathbf{y}_n \mid \mathbf{0}, 10\mathbf{I}_D\right)$$

$$\mathbf{m}_\theta = \mathbf{m}_\theta^{-n} + v_\theta^{-n} \pi_n \frac{\mathbf{y}_n - \mathbf{m}_\theta^{-n}}{v_\theta^{-n} + 1}$$

$$v_\theta = v_\theta^{-n} - \pi_n \frac{\left(v_\theta^{-n}\right)^2}{v_\theta^{-n} + 1} + \pi_n \left(1 - \pi_n\right) \frac{\left(v_\theta^{-n}\right)^2 \left\|\mathbf{y}_n - \mathbf{m}_\theta^{-n}\right\|^2}{D\left(v_\theta^{-n} + 1\right)^2}$$

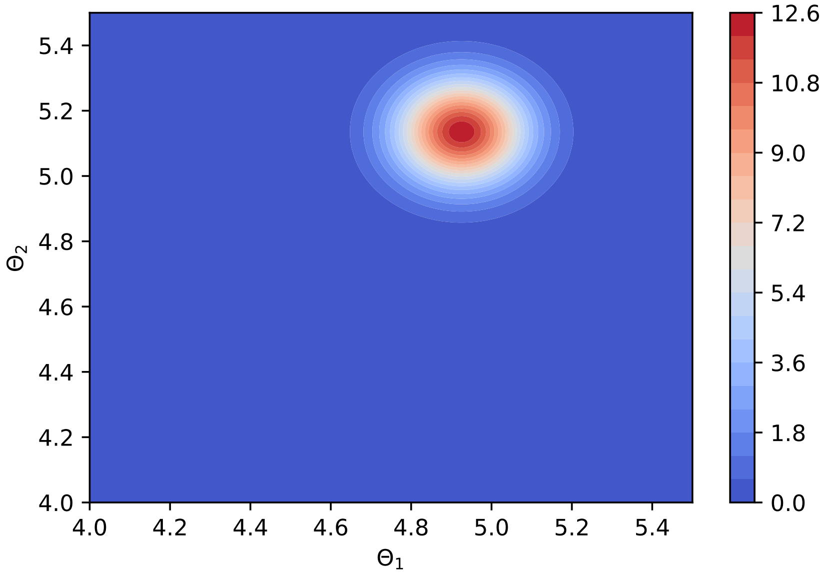

Figure 3: 2D contour of the Gaussian approximation via ADF.

3 Plot the 2D coutour of $q(\boldsymbol{\theta}) = \mathrm{N}\left(\mathbf{m}_\theta, v_\theta \mathbf{I}_D\right)$ in Figure 3.

(b) Expectation propagation.
The derivation has also been shown in the lecture note (`https://ucla-biostats-202c.github.io/reading/Notes2.pdf`). The pseudo code is as follows:

0 Reuse generated samples from (a).

1 Initialize $v_0 = 100, \mathbf{m}_0 = \mathbf{0}, s_0 = (2\pi v_0)^{-D/2}$ and $v_n = \infty, \mathbf{m}_n = \mathbf{0}$ and $s_n = 1$.

2 Initialize $\mathbf{m}_\theta = \mathbf{m}_0, v_\theta = v_0$.

3 Loop until convergence (I set convergence criteria to 10^{-4}):

- **For** $n = 1, \ldots, N$, do the following update:
  * Remove r_n to get old posterior:

$$\begin{aligned}
\left(v_\theta^{-n}\right)^{-1} &= v_\theta^{-1} - v_n^{-1} \\
\mathbf{m}_\theta^{-n} &= \mathbf{m}_\theta + \frac{v_\theta^{-n}}{v_n}\left(\mathbf{m}_\theta - \mathbf{m}_n\right)
\end{aligned}$$

  * Recompute $(\mathbf{m}_\theta, v_\theta, z_n)$ from $\left(\mathbf{m}_\theta^{-n}, \left(v_\theta^{-n}\right)^{-1}\right)$ following the same ADF steps:

$$\begin{aligned}
z_n &= (1-w)\mathrm{N}\left(\mathbf{y}_n \mid \mathbf{m}_\theta^{-n}, \left(v_\theta^{-n}+1\right)\mathbf{I}_D\right) + w\,\mathrm{N}\left(\mathbf{y}_n \mid \mathbf{0}, 10\mathbf{I}_D\right) \\
\pi_n &= 1 - \frac{w}{z_n}\mathrm{N}\left(\mathbf{y}_n \mid \mathbf{0}, 10\mathbf{I}_D\right) \\
\mathbf{m}_\theta &= \mathbf{m}_\theta^{-n} + v_\theta^{-n}\pi_n \frac{\mathbf{y}_n - \mathbf{m}_\theta^{-n}}{v_\theta^{-n}+1} \\
v_\theta &= v_\theta^{-n} - \pi_n \frac{\left(v_\theta^{-n}\right)^2}{v_\theta^{-n}+1} + \pi_n\left(1-\pi_n\right)\frac{\left(v_\theta^{-n}\right)^2 \left\|\mathbf{y}_n - \mathbf{m}_\theta^{-n}\right\|^2}{D\left(v_\theta^{-n}+1\right)^2}
\end{aligned}$$

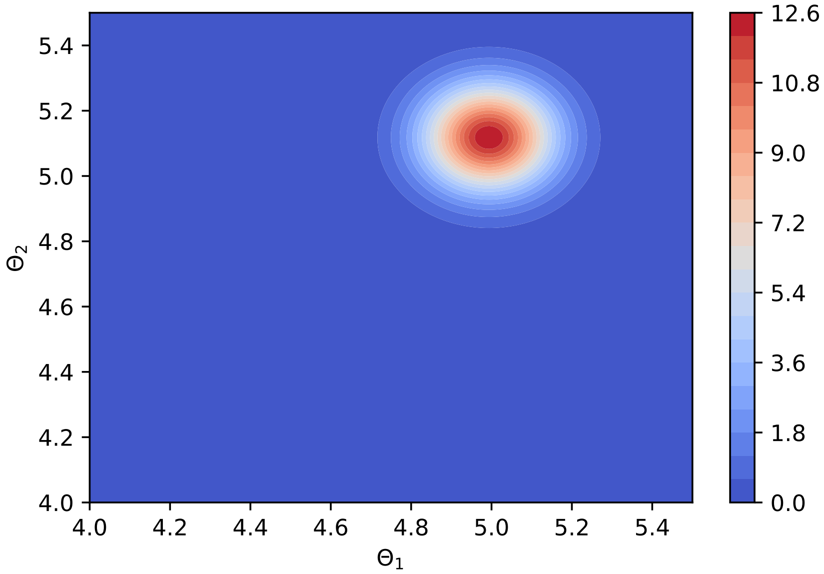

Figure 4: 2D contour of the Gaussian approximation via expectation propagation.

- Update r_n:

$$v_n^{-1} = v_\theta^{-1} - \left(v_\theta^{-n}\right)^{-1}$$
$$\mathbf{m}_n = \mathbf{m}_\theta^{-n} + \frac{\left(v_n + v_\theta^{-n}\right)}{v_\theta^{-n}} \left(\mathbf{m}_\theta - \mathbf{m}_\theta^{-n}\right)$$
$$s_n = \frac{z_n}{\left(2\pi v_n\right)^{D/2} \ \mathrm{N}\left(\mathbf{m}_n \mid \mathbf{m}_\theta^{-n}, \left(v_n + v_\theta^{-n}\right)\mathbf{I}\right)}$$

4 Plot the 2D contour of $q(\boldsymbol{\theta}) = \mathrm{N}\left(\mathbf{m}_\theta, v_\theta \mathbf{I}_D\right)$ in Figure 4.

Note that in the implementation, the algorithm sometimes has convergence issue, which has been described in [1]. So I set v_n to a large number (10^{10}) when it becomes negative or infinity. The final result produced by EP is a bit better than ADP, since it is closer to the ground truth $(5, 5)$.