DD2434 - Machine Learning, Advanced Course
Assignment 1A

Tristan Perrot
tristanp@kth.se

Étienne Riguet
riguet@kth.se

November 2023

Contents

1 Exponential Family **3**

2 Dependencies in a Directed Graphical Model **4**

3 CAVI **9**

A Appendix **16**

A.1 Question 3.12 16
A.2 Question 3.13 16
A.3 Question 3.14 16
A.4 Question 3.15 18

1 Exponential Family

Question 1.1

$$\begin{aligned}
p(x|\theta) &= h(x)\exp(\eta(\theta)\cdot T(x) - A(\eta)) \\
&= h(x)\exp(\eta(\lambda)\cdot T(x) - A(\eta(\lambda))) \\
&= h(x)\exp(\log\lambda\cdot x - A(\log\lambda)) \\
&= h(x)\exp(\log\lambda\cdot x - \lambda) \\
&= h(x)\exp(\log\lambda\cdot x)\exp(-\lambda) \\
&= e^{-\lambda}\frac{\lambda^x}{x!}
\end{aligned} \tag{1}$$

We can see that the distribution correspond to a Poisson distribution of parameter λ.

Question 1.2

$$\begin{aligned}
p(x|\theta) &= h(x)\exp(\eta(\theta)\cdot T(x) - A(\eta)) \\
&= \exp(\eta([\alpha,\beta])\cdot[\log x, x] - A(\alpha-1,-\beta)) \\
&= \exp([\alpha-1,-\beta]\cdot[\log x, x] - \log\Gamma(\alpha) + \alpha\log(\beta)) \\
&= \exp((\alpha-1)\log x - \beta x - \log\Gamma(\alpha) + \alpha\log(\beta)) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}
\end{aligned} \tag{2}$$

We can see that the distribution correspond to a Gamma distribution of parameters α and β.

Question 1.3

$$\begin{aligned}
p(x|\theta) &= h(x)\exp(\eta(\theta)\cdot T(x) - A(\eta)) \\
&= \frac{\exp(\eta([\mu,\sigma^2])\cdot[x,x^2] - A(\eta([\mu,\sigma^2])))}{\sqrt{2\pi}} \\
&= \frac{\exp([\frac{\mu}{\sigma^2},-\frac{1}{2\sigma^2}]\cdot[x,x^2] - A([\frac{\mu}{\sigma^2},-\frac{1}{2\sigma^2}]))}{\sqrt{2\pi}} \\
&= \frac{\exp(\frac{\mu x}{\sigma^2} - \frac{x^2}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} - \log\sigma)}{\sqrt{2\pi}} \\
&= \frac{\exp(-\frac{(x-\mu)^2}{2\sigma^2})}{\sigma\sqrt{2\pi}}
\end{aligned} \tag{3}$$

We can see that the distribution correspond to a Normal distribution of parameters μ and σ^2.

Question 1.4

$$\begin{aligned} p(x|\theta) &= h(x)\exp(\eta(\theta)\cdot T(x) - A(\eta)) \\ &= 2\exp(\eta(\lambda)\cdot x - A(\eta(\lambda))) \\ &= 2\exp(-\lambda x - A(-\lambda)) \\ &= 2\exp\left(-\lambda x + \log\left(\frac{\lambda}{2}\right)\right) \\ &= \lambda e^{-\lambda x} \end{aligned} \tag{4}$$

We can see that the distribution correspond to a Exponential distribution of parameter λ.

Question 1.5

$$\begin{aligned} p(x|\theta) &= h(x)\exp(\eta(\theta)\cdot T(x) - A(\eta)) \\ &= \exp(\eta([\psi_1,\psi_2])\cdot[\log x, \log(1-x)] - A(\eta([\psi_1,\psi_2]))) \\ &= \exp([\psi_1 - 1, \psi_2 - 1]\cdot[\log x, \log(1-x)] - A([\psi_1 - 1, \psi_2 - 1])) \\ &= \exp((\psi_1 - 1)\log x + (\psi_2 - 1)\log(1-x) - \log\Gamma(\psi_1) - \log\Gamma(\psi_2) + \log\Gamma(\psi_1+\psi_2)) \\ &= \frac{\Gamma(\psi_1+\psi_2)}{\Gamma(\psi_1)\Gamma(\psi_2)} x^{\psi_1 - 1}(1-x)^{\psi_2 - 1} \end{aligned} \tag{5}$$

We can see that the distribution correspond to a Beta distribution of parameters ψ_1 and ψ_2.

2 Dependencies in a Directed Graphical Model

Question 2.6

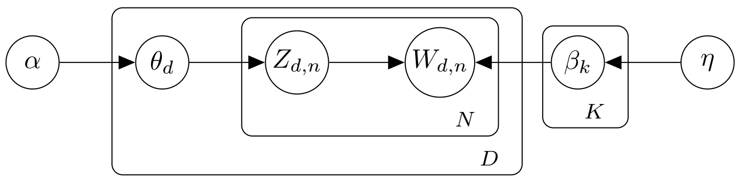
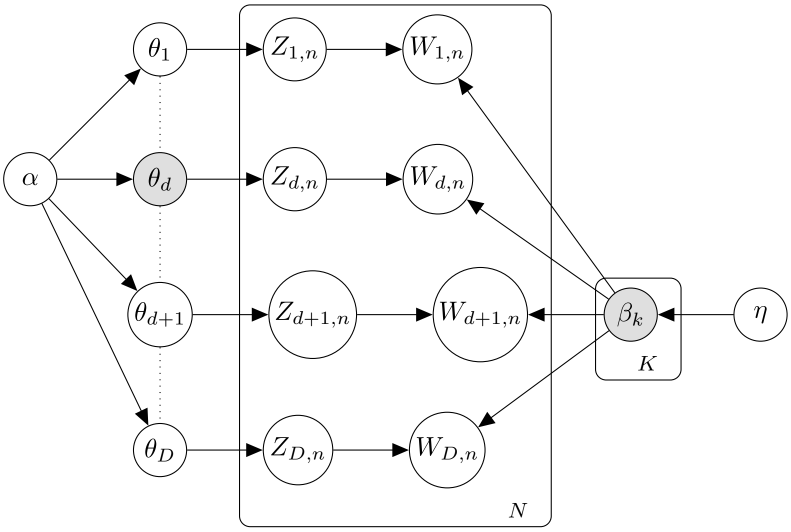


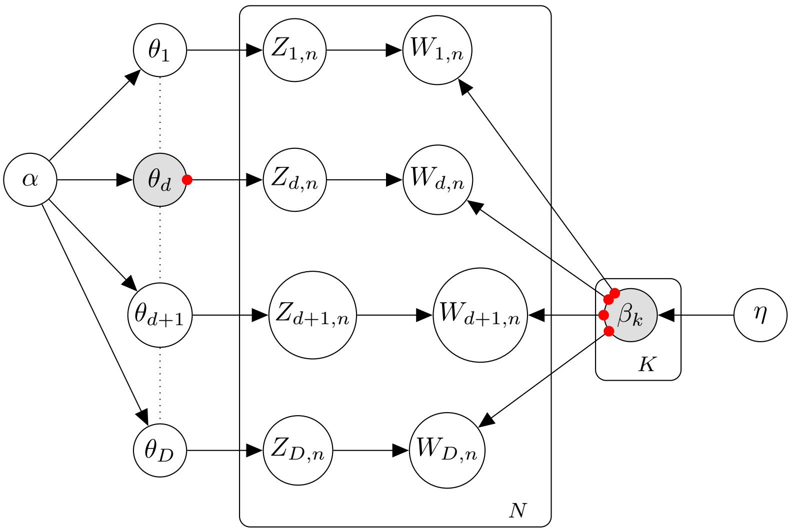
Figure 1: Graphical model of smooth LDA.

The Bayes net take this form :

Tristan Perrot
Étienne Riguet



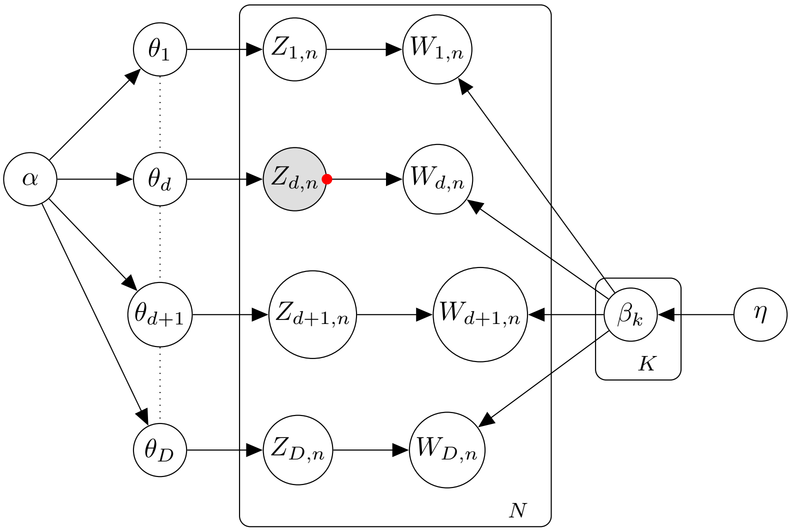
Then, if we use the method using the d-separation, we obtain this :



Therefore, we can see that $W_{d,n} \perp W_{d,n+1} | \theta_d, \beta_{1:K}$ is <u>true</u>.

Question 2.7

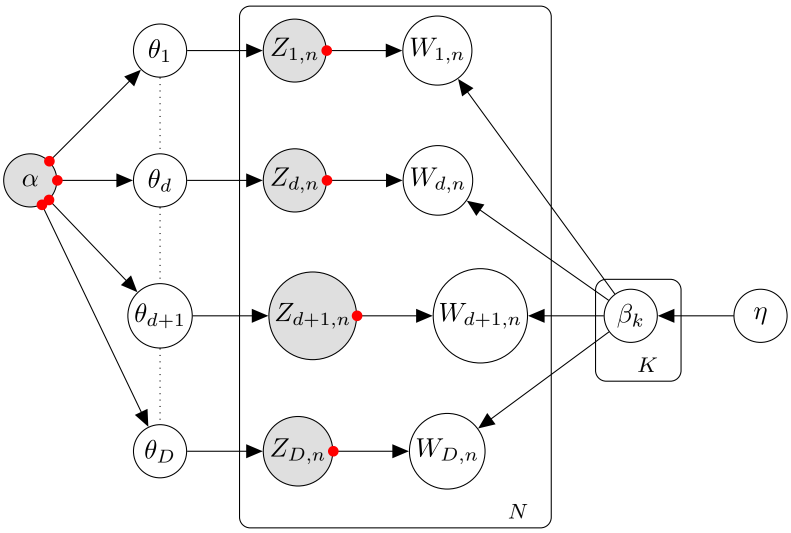
The Bayes net take this form (with d-separation marks) :



Therefore, we can see that $\theta_d \perp \theta_{d+1} | Z_{d,1:N}$ is <u>false</u>.

Question 2.8

The Bayes net take this form (with d-separation marks) :



Therefore, we can see that $\theta_d \perp \theta_{d+1} | \alpha, Z_{1:D,1:N}$ is <u>true</u>.

Question 2.9

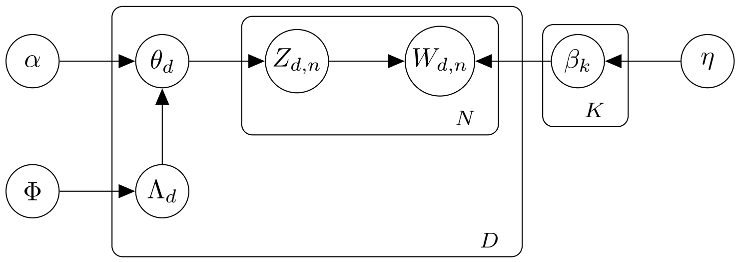
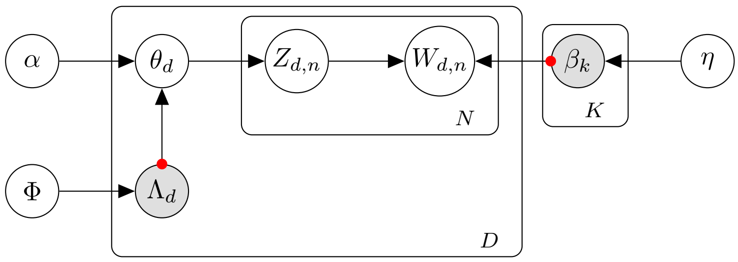


Figure 2: Graphical model of Labeled LDA.

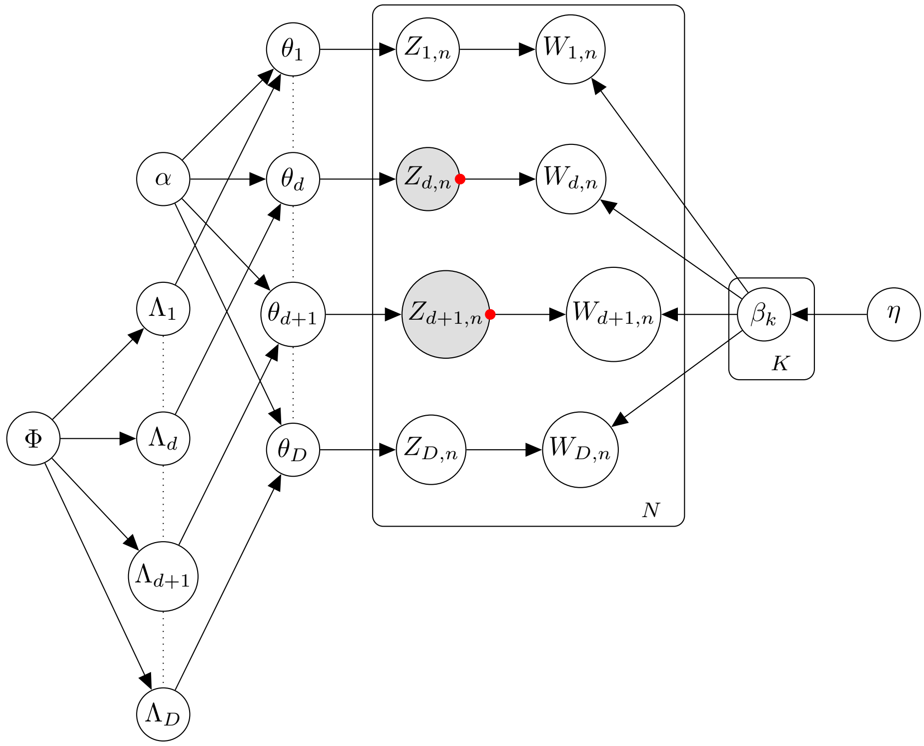
The Bayes net take this form (with d-separation marks) :



Therefore, we can see that $W_{d,n} \perp W_{d,n+1} | \Lambda_d, \beta_{1:K}$ is false.

Question 2.10

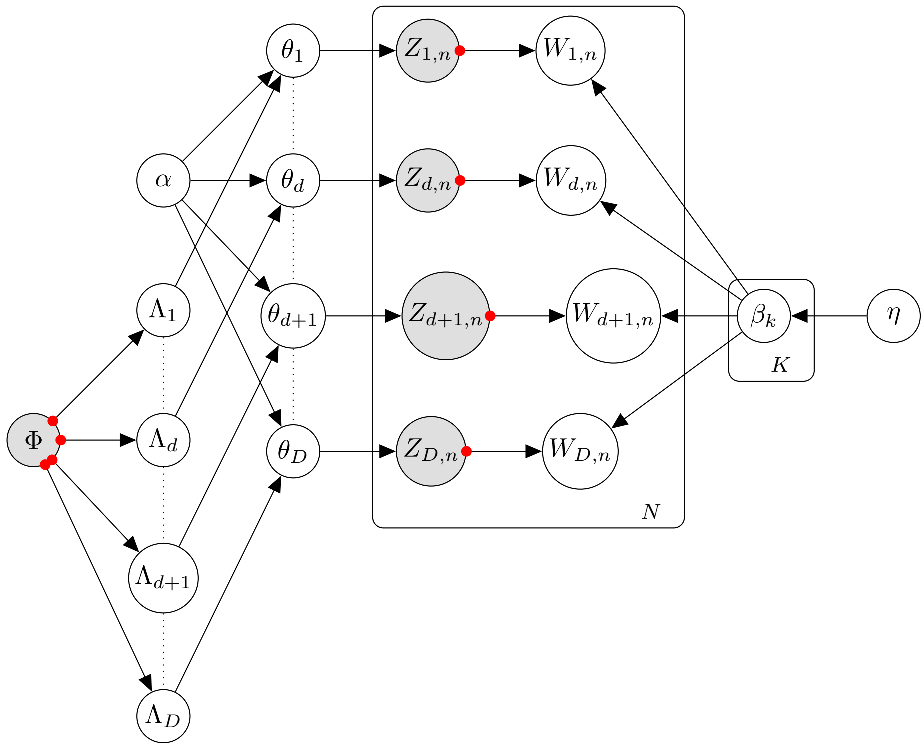
The Bayes net take this form (with d-separation marks) :



Therefore, we can see that $\theta_d \perp \theta_{d+1} | Z_{d,1:N}, Z_{d+1,1:N}$ is false.

Question 2.11

The Bayes net take this form (with d-separation marks) :



Therefore, we can see that $\Lambda_d \perp \Lambda_{d+1} | \Phi, Z_{1:D,1:N}$ is <u>false</u>.

3 CAVI

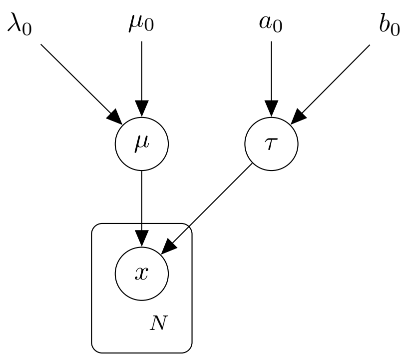


Figure 3: DGM

Question 3.12

In the bishop book, we can see that :

$$p(X|\mu,\tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \tag{6}$$

$$p(\mu|\tau) = \mathcal{N}(\mu|\mu_0, (\lambda_0\tau)^{-1}) \tag{7}$$

$$p(\tau) = \text{Gam}(\tau|a_0, b_0) \tag{8}$$

Then, by using the code in appendix A.1, we obtain :

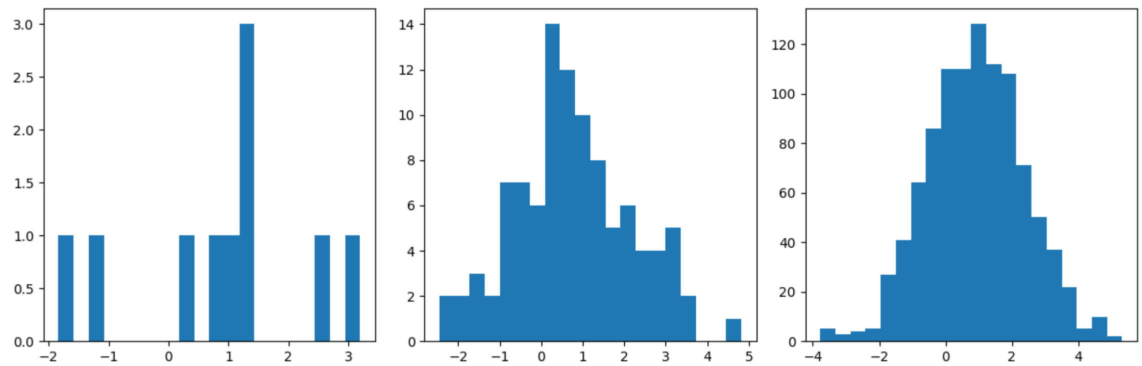


Figure 4: Generated Data.

Question 3.13

Let's find the ML estimates of μ and τ. We know that $\log(q^*(\mu)) = \mathbb{E}_{-\mu}[\log p(X,\mu,\tau)]$. Then, we can write :

$$\begin{aligned}
\log(q^*(\mu)) &= \mathbb{E}_{-\mu}[\log p(X,\mu,\tau)] \\
&\overset{+}{=} \mathbb{E}_\tau[\log p(X|\mu,\tau) + \log p(\mu|\tau)] \\
&= \mathbb{E}_\tau\left[\frac{N}{2}\log\left(\frac{\tau}{2\pi}\right) + \frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2 + \frac{1}{2}\log\left(\frac{\lambda_0\tau}{2\pi}\right) + \frac{\lambda_0\tau}{2}(\mu-\mu_0)\right] \\
&\overset{+}{=} \frac{\mathbb{E}_\tau[\tau]}{2}\left(\lambda_0(\mu-\mu_0) + \sum_{n=1}^{N}(x_n-\mu)^2\right) \\
&= -\frac{\mathbb{E}_\tau[\tau]}{2}\left(\lambda_0\mu^2 - 2\lambda_0\mu\mu_0 + \lambda_0\mu_0^2 + \sum_{n=1}^{N}x_n^2 - 2\mu\sum_{n=1}^{N}x_n + N\mu^2\right) \\
&= -\frac{\mathbb{E}_\tau[\tau]}{2}\left((\lambda_0+N)\mu^2 - 2(\lambda_0\mu_0 + \sum_{n=1}^{N}x_n)\mu + \lambda_0\mu_0^2 + \sum_{n=1}^{N}x_n^2\right) \\
&= -\frac{\mathbb{E}_\tau[\tau](\lambda_0+N)}{2}\left(\mu^2 - 2\mu\frac{\lambda_0\mu_0 + \sum_{n=1}^{N}x_n}{\lambda_0+N} + \frac{\lambda_0\mu_0^2 + \sum_{n=1}^{N}x_n^2}{\lambda_0+N}\right)
\end{aligned} \tag{9}$$

Therefore we can conclude that $q^*(\mu) = \mathcal{N}(\mu|\mu_N, \lambda_N^{-1})$ with :

$$\mu_N = \frac{\lambda_0\mu_0 + \sum_{n=1}^N x_n}{\lambda_0 + N} \tag{10}$$

$$\lambda_N = (\lambda_0 + N)\mathbb{E}[\tau] \tag{11}$$

And for τ we have :

$$\begin{aligned}
\log(q^*(\tau)) &= \mathbb{E}_{-\tau}[\log p(X, \mu, \tau)] \\
&\overset{\pm}{=} \mathbb{E}_\mu[\log p(X|\mu, \tau) + \log p(\mu|\tau)] + \log p(\tau) \\
&\overset{\pm}{=} (a_0 - 1)\log\tau - b_0\tau + \frac{N+1}{2}\log\tau - \frac{\tau}{2}\mathbb{E}_\mu\left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2\right] \\
&= (a_0 + \frac{N+1}{2} - 1)\log\tau - \left(b_0 + \frac{1}{2}\mathbb{E}_\mu\left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2\right]\right)\tau
\end{aligned} \tag{12}$$

Therefore we can conclude that $q^*(\tau) = \mathrm{Gam}(\tau|a_N, b_N)$ with :

$$a_N = a_0 + \frac{N+1}{2} \tag{13}$$

$$\begin{aligned}
b_N &= b_0 + \frac{1}{2}\mathbb{E}_\mu\left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2\right] \\
b_N &= b_0 + \frac{1}{2}\left(\sum_{n=1}^N x_n^2 + N\mathbb{E}_\mu[\mu^2] - 2\mathbb{E}_\mu[\mu]\sum_{n=1}^N x_n + \lambda_0\left(\mathbb{E}_\mu[\mu^2] + \mu_0^2 - 2\mu_0\mathbb{E}_\mu[\mu]\right)\right)
\end{aligned} \tag{14}$$

With :

$$\begin{aligned}
\mathbb{E}_{q(\mu)}[\mu] &= \mu_N \\
\mathbb{E}_{q(\mu)}[\mu^2] &= \frac{1}{\lambda_N} + \mu_N^2 \\
\mathbb{E}_{q(\tau)}[\tau] &= \frac{a_N}{b_N}
\end{aligned} \tag{15}$$

If we take non-informative priors then $a_0 = b_0 = \mu_0 = \lambda_0 = 0$, then we have :

$$\begin{aligned}
\mu_N &= \overline{x} \\
\lambda_N &= N\mathbb{E}[\tau] \\
a_N &= \frac{N+1}{2} \\
b_N &= \frac{1}{2}\mathbb{E}_\mu\left[\sum_{n=1}^N (x_n - \mu)^2\right]
\end{aligned} \tag{16}$$

And by using $\mathbb{E}[\tau] = \frac{a_N}{b_N}$ we obtain :

$$\begin{aligned}
\frac{1}{\mathbb{E}[\tau]} &= \frac{b_N}{a_N} \\
\frac{1}{\mathbb{E}[\tau]} &= \frac{2}{2(N+1)}\mathbb{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2\right] \\
\frac{1}{\mathbb{E}[\tau]} &= \frac{N}{N+1}\left(\overline{x^2} - 2\overline{x}\mathbb{E}[\mu] + \mathbb{E}[\mu^2]\right)
\end{aligned} \tag{17}$$

And, with the fact that $\mathbb{E}[\mu] = \mu_N$ and $\mathbb{E}[\mu^2] = \frac{1}{\lambda_N} + \mu_N^2$, we obtain :

$$\begin{aligned}
\mathbb{E}[\mu] &= \overline{x} \\
\mathbb{E}[\mu^2] &= \frac{1}{N\mathbb{E}[\tau]} + \overline{x}^2
\end{aligned} \tag{18}$$

And therefore:

$$\frac{1}{\mathbb{E}[\tau]} = \overline{x^2} - \overline{x}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \overline{x})^2 \tag{19}$$

Which define the ML estimates. The implementation is in the code in appendix A.2.

Question 3.14

The posterior is defined as $p(\mu, \tau|x)$. Then, we can write :

$$\begin{aligned}
p(\mu,\tau|x) &= \frac{p(x|\mu,\tau)p(\mu,\tau)}{p(x)} \\
&\propto p(x|\mu,\tau)p(\mu,\tau)
\end{aligned} \tag{20}$$

Where $x|\mu,\tau \sim \mathcal{N}(\mu|\mu,\tau^{-1})$ and $\mu,\tau \sim NormalGamma(\mu_0,\lambda_0,a_0,b_0)$. Therefore, as we saw in the question 1.3 in the Module 1 exercise, we have $\mu,\tau|x \sim NormalGamma(\mu',\lambda',a',b')$, where :

$$\begin{aligned}
\mu' &= \frac{N\overline{x} + \mu_0\lambda_0}{N+\lambda_0} \\
\lambda' &= N + \lambda_0 \\
a' &= a_0 + \frac{N}{2} \\
b' &= b_0 + \frac{1}{2}\left(\sum_{n=1}^{N} x_n^2 + \lambda_0\mu_0^2 - \frac{(N\overline{x}+\mu_0\lambda_0)^2}{N+\lambda_0}\right)
\end{aligned} \tag{21}$$

Therefore, if we plot the contour for each datasets we obtain :

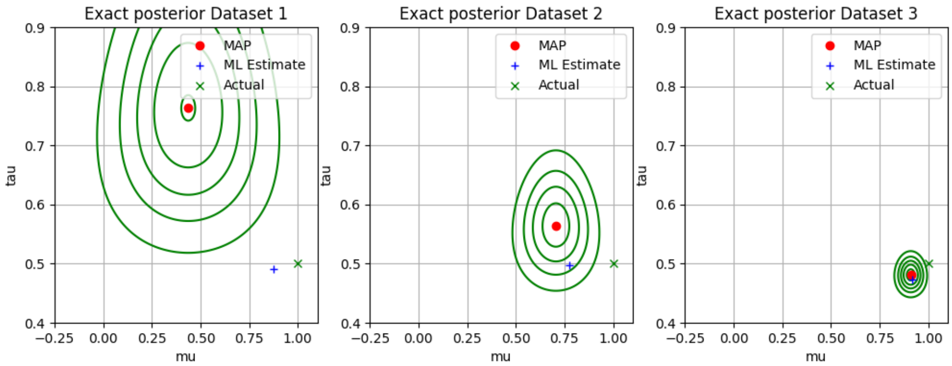


Figure 5: Contours of exact posteriors.

The rest of the answer is in the code in appendix A.3.

Question 3.15

The equation (10.24) in the Bishop is the mean-field approximation which is :

$$q(\mu, \tau) = q(\mu)q(\tau) \tag{22}$$

This time, we take the result of the question 3.13 without setting the priors to 0. Then, we have :

$$\begin{aligned} q(\mu) &= \mathcal{N}(\mu|\mu_N, \lambda_N^{-1}) \\ q(\tau) &= \text{Gam}(\tau|a_N, b_N) \end{aligned} \tag{23}$$

with updates equations in the cavi algorithm described by :

$$\begin{aligned} \mu_N &= \frac{\lambda_0\mu_0 + N\overline{x}}{\lambda_0 + N} \\ \lambda_N &= (\lambda_0 + N)\mathbb{E}[\tau] \\ a_N &= a_0 + \frac{N+1}{2} \\ b_N &= b_0 + \frac{1}{2}\left(\sum_{n=1}^{N} x_n^2 + N\mathbb{E}_\mu[\mu^2] - 2\mathbb{E}_\mu[\mu]\sum_{n=1}^{N} x_n + \lambda_0\left(\mathbb{E}_\mu[\mu^2] + \mu_0^2 - 2\mu_0\mathbb{E}_\mu[\mu]\right)\right) \end{aligned} \tag{24}$$

and the expectations are the ones described in the equation (15).
Now, we need to find the ELBO formula :

$$\begin{aligned} \mathcal{L}(q) &= \mathbb{E}_{q(\mu),q(\tau)}[\log p(X, \mu, \tau)] - \mathbb{E}_{q(\mu),q(\tau)}[\log q(\mu, \tau)] \\ &= \mathbb{E}_{q(\mu),q(\tau)}[\log p(X|\mu, \tau) + \log p(\mu, \tau)] - \mathbb{E}_{q(\mu),q(\tau)}[\log q(\mu) + \log q(\tau)] \\ &= \mathbb{E}_{q(\mu),q(\tau)}[\log p(X|\mu, \tau)] + \mathbb{E}_{q(\mu),q(\tau)}[\log p(\mu, \tau)] - \mathbb{E}_{q(\mu)}[\log q(\mu)] - \mathbb{E}_{q(\tau)}[\log q(\tau)] \\ &= \mathbb{E}_{q(\mu),q(\tau)}[\log p(X|\mu, \tau)] + \mathbb{E}_{q(\mu),q(\tau)}[\log p(\mu, \tau)] + \mathbb{H}_q[\mu] + \mathbb{H}_q[\tau] \end{aligned} \tag{25}$$

Then we obtain this result :

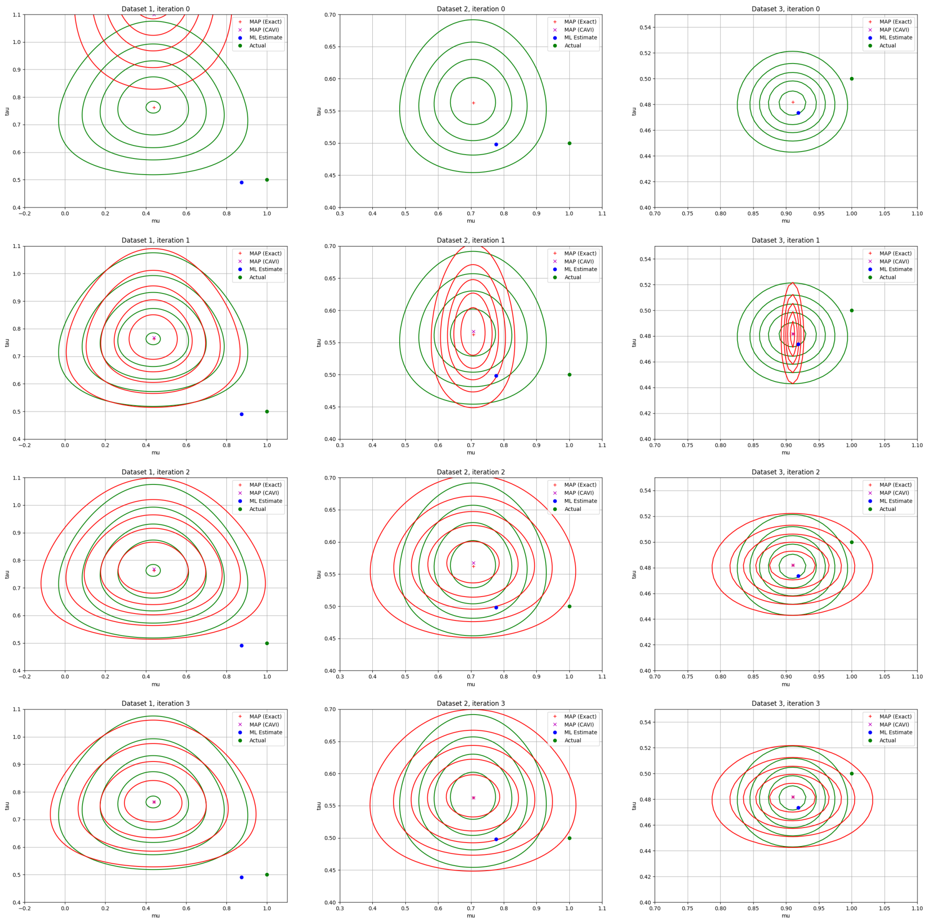


Figure 6: Contours of the approximations by VI.

And we obtain an elbo plot :

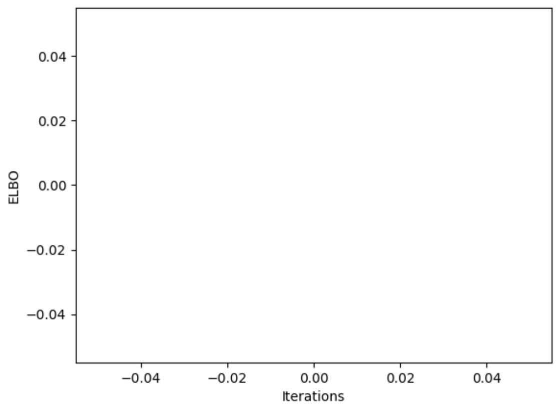


Figure 7: ELBO plot.

The code is in appendix A.4.

A Appendix

A.1 Question 3.12

```
import numpy as np
from scipy.stats import gamma, norm
from scipy.special import psi
np.random.seed(14)


def generate_data(mu, tau, N):
  # Insert your code here
  D = np.random.normal(mu, 1/tau, N)

  return D


mu = 1
tau = 0.5

dataset_1 = generate_data(mu, tau, 10)
dataset_2 = generate_data(mu, tau, 100)
dataset_3 = generate_data(mu, tau, 1000)

# Visulaize the datasets via histograms
# Insert your code here
fig, axs = plt.subplots(1, 3, figsize=(12, 4))
axs[0].hist(dataset_1, bins=20)
axs[1].hist(dataset_2, bins=20)
axs[2].hist(dataset_3, bins=20)
plt.tight_layout()
plt.savefig('12_data.png')
plt.show()
```

A.2 Question 3.13

```
def ML_est(data):
    # insert your code
    N = len(data)
    x_mean = np.mean(data)
    x_var = np.var(data, ddof=1)

    tau_ml = 1 / x_var
    mu_ml = x_mean

    return mu_ml, tau_ml
```

A.3 Question 3.14

```
def compute_exact_posterior(D, a_0, b_0, mu_0, lambda_0):
    # your implementation
    x_mean = np.mean(D)
    N = len(D)

    mu_prime = (lambda_0 * mu_0 + N * x_mean) / (lambda_0 + N)
    lambda_prime = lambda_0 + N
    a_prime = a_0 + N / 2
    b_prime = b_0 + 1 / 2 * (np.sum((D - x_mean)**2) +
                             lambda_0 * mu_0**2 - lambda_prime * mu_prime
                             **2)

    exact_post_distribution = (a_prime, b_prime, mu_prime, lambda_prime)

    return exact_post_distribution


# prior parameters
mu_0 = 0
lambda_0 = 10
a_0 = 20
b_0 = 20

mus = np.linspace(0.3, 1.2, 200)
taus = np.linspace(0.1, 1, 200)

fig, axs = plt.subplots(1, 3, figsize=(12, 4))
for i, dataset in enumerate([dataset_1, dataset_2, dataset_3]):
    mu_ml, tau_ml = ML_est(dataset)

    a_T, b_T, mu_T, lambda_T = compute_exact_posterior(
        dataset, a_0, b_0, mu_0, lambda_0)

    Z_exact = np.zeros((len(mus), len(taus)))
    pTau = gamma(a=a_T, loc=0, scale=1/b_T)
    for j, tau in enumerate(taus):
        pMu = norm(loc=mu_T, scale=1/np.sqrt(lambda_T*tau))
        Z_exact[:, j] = pMu.pdf(mus) * pTau.pdf(tau)
    # Finding the maximum of the exact posterior
    mu_max = mus[np.argmax(np.max(Z_exact, axis=1))]
    tau_max = taus[np.argmax(np.max(Z_exact, axis=0))]
    # Plotting the results
    axs[i].contour(*np.meshgrid(mus, taus), Z_exact.T,
                   levels=5, colors=['green'])
    axs[i].plot(mu_max, tau_max, 'ro', label='MAP')
    axs[i].plot(mu_ml, tau_ml, 'b+', label='ML Estimate')
    axs[i].plot(MU, TAU, 'gx', label='Actual')
    axs[i].legend()
    axs[i].grid()
    axs[i].set_xlabel('mu')
    axs[i].set_ylabel('tau')
```

```
    axs[i].set_title('Exact posterior Dataset {}'.format(i+1))

plt.savefig('14_contours.png')
plt.show()
```

A.4 Question 3.15

```
eps = 1e-6


def compute_elbo(D, a_0, b_0, mu_0, lambda_0, a_N, b_N, mu_N, lambda_N):
    N = len(D)
    x_mean = np.mean(D)
    x_2_sum = np.sum(D**2)

    # Expected log likelihood
    expected_log_likelihood = 0.5 * N * (np.log(lambda_N) - np.log(
        2 * np.pi)) - 0.5 * lambda_N * (x_2_sum - 2 * x_mean * mu_N * N + N
            * mu_N**2)

    # Expected log prior for mu
    expected_log_prior_mu = 0.5 * (np.log(lambda_0) - np.log(2 * np.pi)) - \
        0.5 * lambda_0 * (mu_N**2 - 2 * mu_N * mu_0 + mu_0**2)

    # Expected log prior for tau
    expected_log_prior_tau = a_0 * np.log(b_0) - np.log(gamma(a_0).pdf(0)) +
         (
        a_0 - 1) * (psi(a_N) - np.log(b_N)) - b_0 * (a_N / b_N)

    # Entropy of variational distribution for mu
    entropy_mu = 0.5 * (1 + np.log(2 * np.pi)) + 0.5 * np.log(1 / lambda_N)

    # Entropy of variational distribution for tau
    entropy_tau = a_N - np.log(b_N) + \
        np.log(gamma(a_N).pdf(0) + eps) + (1 - a_N) * psi(a_N)

    # Compute ELBO
    elbo = expected_log_likelihood + expected_log_prior_mu + \
        expected_log_prior_tau + entropy_mu + entropy_tau

    return elbo


def CAVI(D, a_0, b_0, mu_0, lambda_0, iter=10):
    # make an initial guess for the expected value of tau
    initial_guess_exp_tau = 1

    N = len(D)
    x_mean = np.mean(D)
    x_2_sum = np.sum(D**2)
```

```
    # Constants
    a_N = a_0 + N / 2
    mu_N = (lambda_0 * mu_0 + N * x_mean) / (lambda_0 + N)
    E_mu = mu_N

    # Variational parameters
    b_N = b_0
    lambda_N = lambda_0

    # ELBO
    elbos = []

    # CAVI iterations ...
    for i in range(iter):
        # update the values for the variational parameters
        E_tau = a_N / b_N
        E_mu_2 = 1 / lambda_N + mu_N**2

        lambda_N = (lambda_0 + N) * E_tau
        b_N = b_0 + 1 / 2 * (x_2_sum + N*E_mu_2 - 2*N*E_mu *
                             x_mean + lambda_0*(E_mu_2 - 2*E_mu*mu_0 + mu_0
                                 **2))
        # save ELBO for each iteration, plot them afterwards to show
            convergence
        elbos.append(compute_elbo(D, a_0, b_0, mu_0,
                     lambda_0, a_N, b_N, mu_N, lambda_N))

    return a_N, b_N, mu_N, lambda_N, elbos


# Insert your main code here
mu_ml, tau_ml = ML_est(dataset_2)
a_N, b_N, mu_N, lambda_N, elbos = CAVI(dataset_2, a_0, b_0, mu_0, lambda_0)
exact_post_dist = compute_exact_posterior(dataset_2, a_0, b_0, mu_0,
    lambda_0)


mu_vals = np.linspace(-1, 1, 100)
tau_vals = np.linspace(0, 2, 100)
mu_grid, tau_grid = np.meshgrid(mu_vals, tau_vals)

# Calculate the Z values for contours


def calculate_density(mu, tau, mu_0, lambda_0, a_0, b_0):
    mu_prior = norm(mu_0, 1/np.sqrt(lambda_0)).pdf(mu)
    tau_prior = gamma(a_0, scale=1/b_0).pdf(tau)
    return mu_prior * tau_prior


Z = np.zeros_like(mu_grid)
for i in range(mu_grid.shape[0]):
```

```
    for j in range(mu_grid.shape[1]):
        Z[i, j] = calculate_density(
            mu_grid[i, j], tau_grid[i, j], mu_N, lambda_N, a_N, b_N)

# Plot the contour
plt.figure(figsize=(10, 6))
contour = plt.contour(mu_grid, tau_grid, Z, colors='k')
plt.clabel(contour, inline=1, fontsize=10)
plt.xlabel('mu')
plt.ylabel('tau')
plt.title('Contour plot for the variational posterior')
plt.show()


print('mu_ml: ', mu_ml)
print('tau_ml: ', tau_ml)
print('a_N: ', a_N)
print('b_N: ', b_N)
print('mu_N: ', mu_N)
print('lambda_N: ', lambda_N)
print('expected tau: ', a_N / b_N)
print('exact_post_dist (a, b, mu, lambda): ', exact_post_dist)
print('exact expected tau: ', exact_post_dist[0] / exact_post_dist[1])

# Plot ELBOs
plt.plot(elbos)
plt.xlabel('Iterations')
plt.ylabel('ELBO')
# plt.savefig('elbo.png')
plt.show()

# Example flow for dataset_2:

# mu_ml, tau_ml = ML_est(dataset_2)
# a_N, b_N, mu_N, lambda_N, elbos = CAVI(dataset_2, a_0, b_0, mu_0, lambda_0
    )
# plot elbos, show convergence
# exact_post_dist = compute_exact_posterior(dataset_2, a_0, b_0, mu_0,
    lambda_0)
# compare exact_post_dist with the CAVI result ( = q(a_N, b_N, mu_N,
    lambda_N) ) using for ex. contour plots, show also ML estimate on this
    plot
```