DD2434 - Machine Learning, Advanced Course
Assignment 1A

Tristan Perrot
tristanp@kth.se

Étienne Riguet
riguet@kth.se

November 2023

Tristan Perrot
Étienne Riguet

Contents

Tristan Perrot
Étienne Riguet

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 <u>Poisson distribution of parameter λ</u>.

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 <u>Gamma distribution of parameters α and β</u>.

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 <u>Normal distribution of parameters μ and σ^2</u>.

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

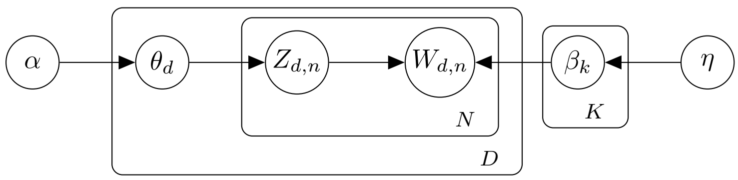
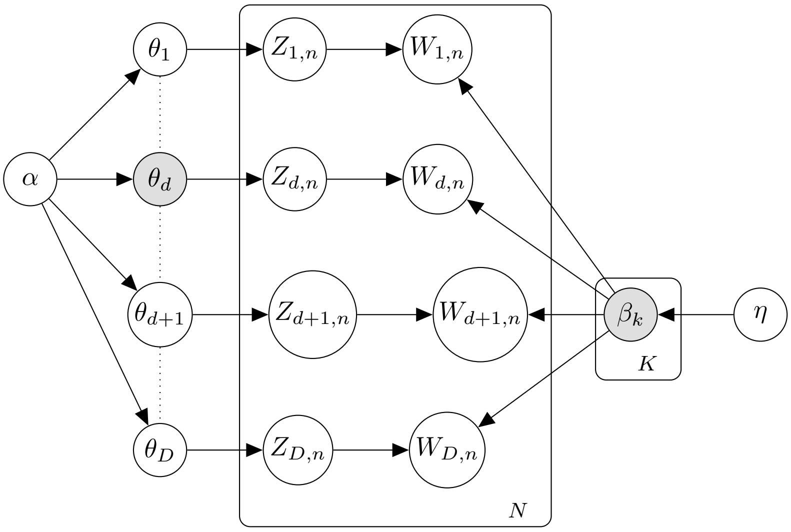
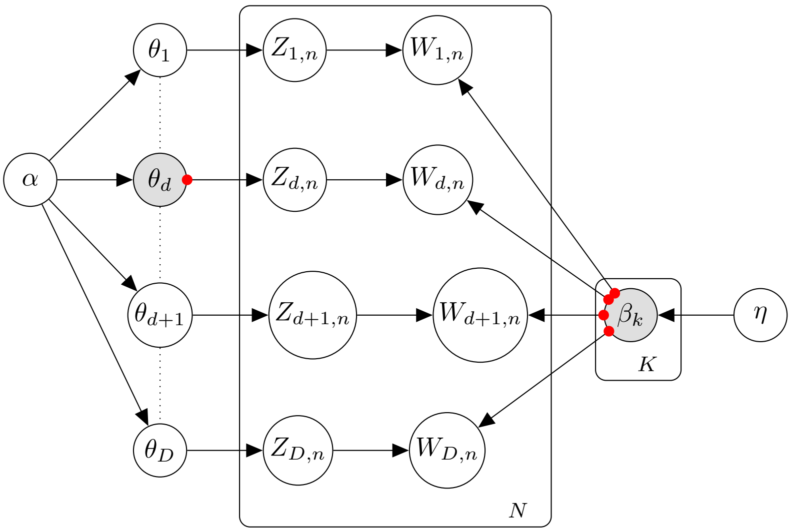


Figure 1: Graphical model of smooth LDA.

The Bayes net take this form :



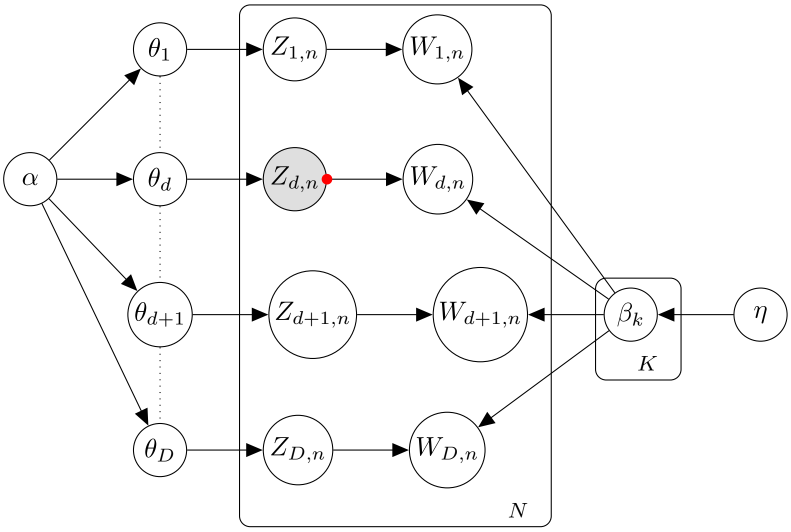
Then, if we use the method using the d-separation, we obtain this :



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

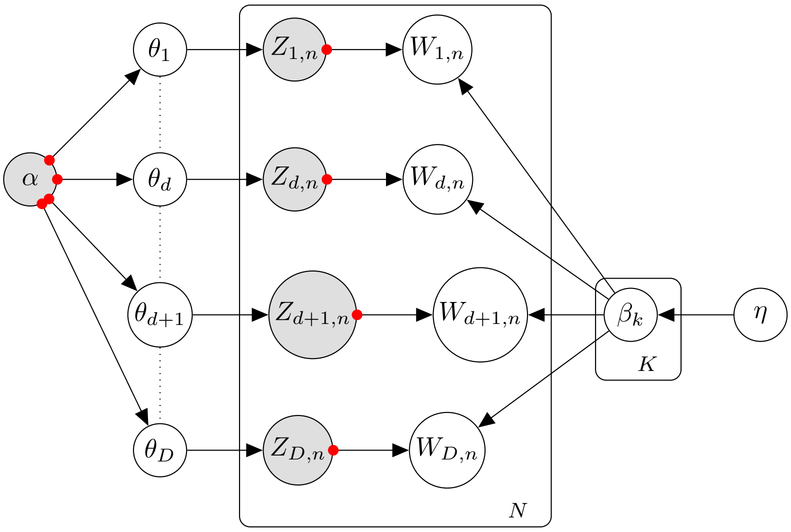
The Bayes net take this form (with d-separation marks) :



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

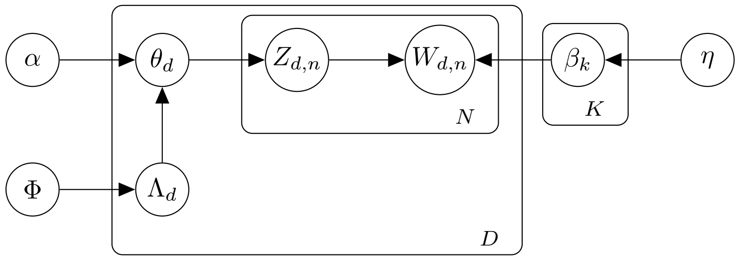
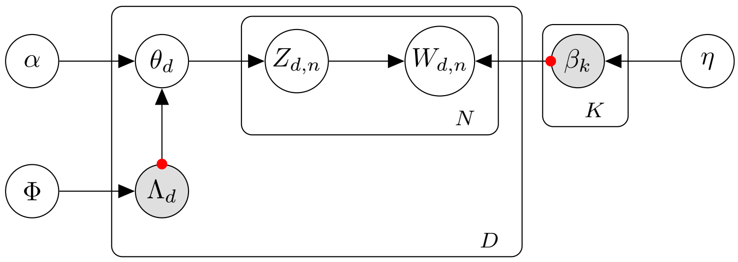


Figure 2: Graphical model of Labeled LDA.

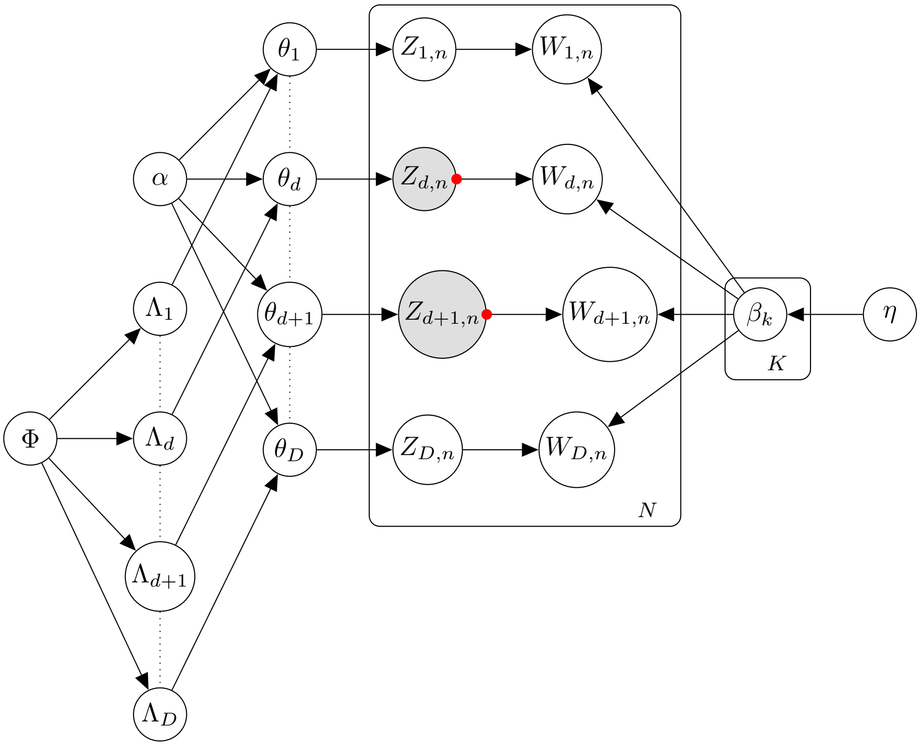
The Bayes net take this form (with d-separation marks) :



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

Question 2.10

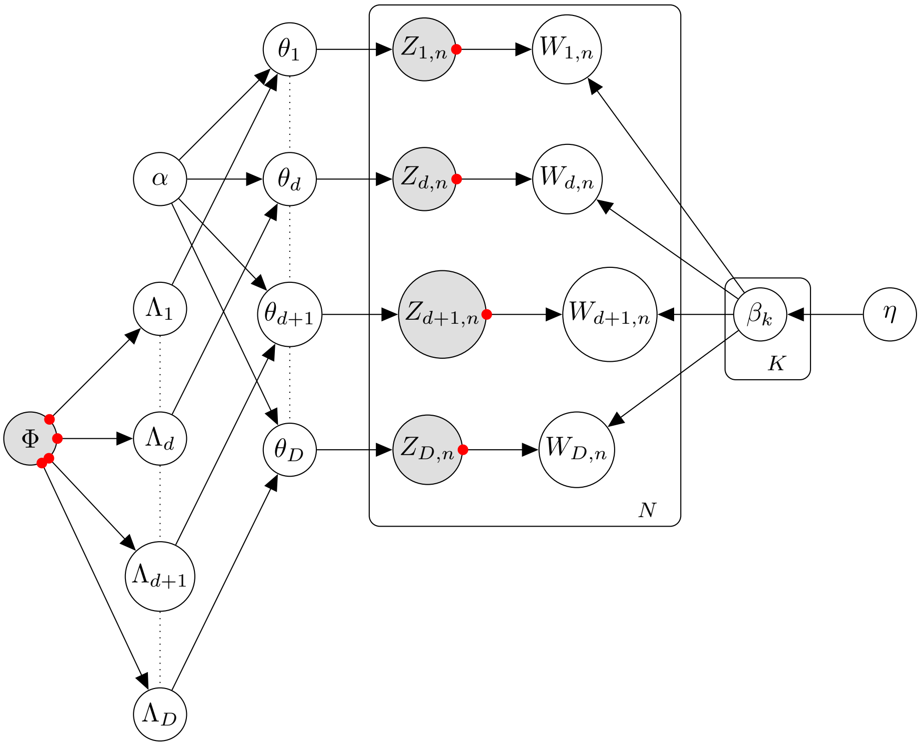
The Bayes net take this form (with d-separation marks) :



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

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 false.

3 CAVI

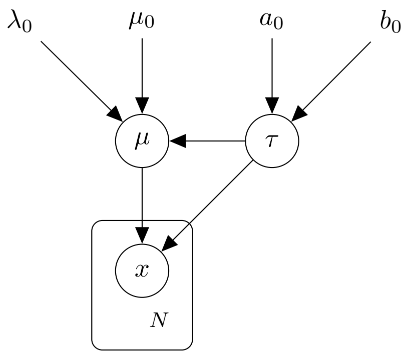


Figure 3: DGM

Question 3.12

In the bishop book, we can see that :

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

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

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

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

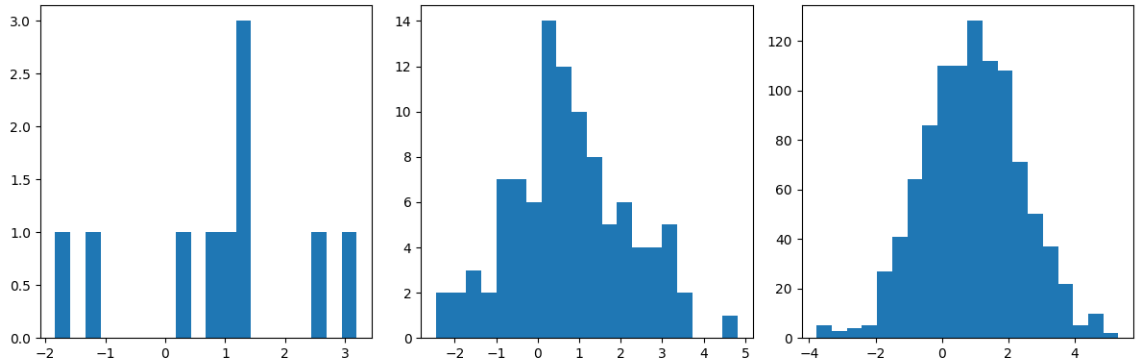


Figure 4: Generated Data

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 :

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

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 :

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

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:

$$
\begin{aligned}
\frac{1}{\mathbb{E}[\tau]} = \frac{N}{N+1} \left(\overline{x^2} - 2\overline{x}^2 + \frac{1}{N\mathbb{E}[\tau]} + \overline{x}^2 \right) &\Leftrightarrow \frac{1}{\mathbb{E}[\tau]} - \frac{1}{(N+1)\mathbb{E}[\tau]} = \frac{N}{N+1} \left(\overline{x^2} - \overline{x}^2 \right) \\
&\Leftrightarrow \frac{N+1-1}{(N+1)\mathbb{E}[\tau]} = \frac{N}{N+1} \left(\overline{x^2} - \overline{x}^2 \right) \\
&\Leftrightarrow \frac{1}{\mathbb{E}[\tau]} = \left(\overline{x^2} - \overline{x}^2 \right) \\
&\Leftrightarrow \frac{1}{\mathbb{E}[\tau]} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \overline{x})^2
\end{aligned}
\tag{19}
$$

Which define the ML estimates. The implementation is in the code in appendix A.1.

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 :

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

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

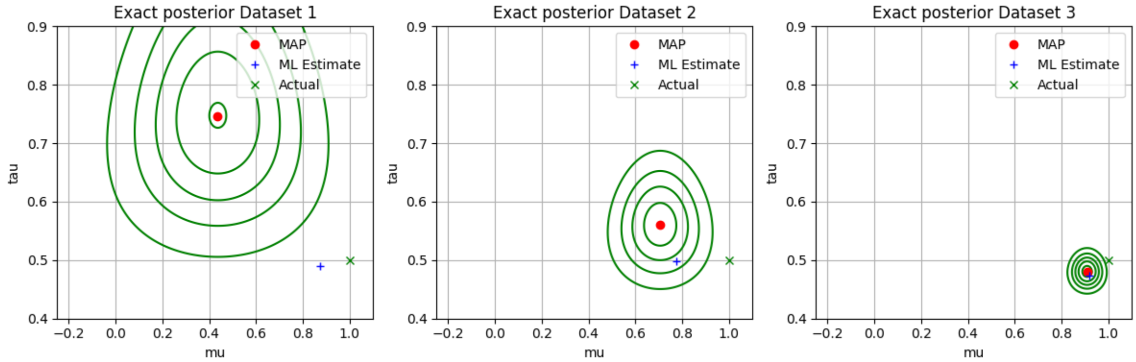


Figure 5: Contours of exact posteriors by datasets

The rest of the answer is in the code in appendix A.1.

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 :

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

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 :

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

If we compute the first term we have:

$$\begin{aligned}
\mathbb{E}_{q(\mu),q(\tau)}[\log p(X|\mu,\tau)] &= \mathbb{E}_{q(\mu),q(\tau)}\left[\frac{N}{2}\log\left(\frac{\tau}{2\pi}\right) - \frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right] \\
&= \frac{N}{2}\left(\mathbb{E}_{q(\tau)}[\log\tau] - \log(2\pi)\right) \\
&\quad - \frac{\mathbb{E}_{q(\tau)}[\tau]}{2}\left(\sum_{n=1}^{N}x_n^2 - 2\mathbb{E}_{q(\mu)}[\mu]N\overline{x_n} + N\mathbb{E}_{q(\mu)}[\mu^2]\right) \\
&\stackrel{+}{=} \frac{N}{2}\mathbb{E}_{q(\tau)}[\log\tau] - \frac{\mathbb{E}_{q(\tau)}[\tau]}{2}\left(\sum_{n=1}^{N}x_n^2 - 2\mathbb{E}_{q(\mu)}[\mu]N\overline{x_n} + N\mathbb{E}_{q(\mu)}[\mu^2]\right)
\end{aligned} \tag{26}$$

And the second one is:

$$\begin{aligned}
\mathbb{E}_{q(\mu),q(\tau)}[\log p(\mu,\tau)] &= \mathbb{E}_{q(\mu),q(\tau)}\left[\log\left(\frac{b_0^{a_0}\sqrt{\lambda_0}}{\Gamma(a_0)\sqrt{2\pi}}\right) + (a_0-\frac{1}{2})\log\tau - b_0\tau - \frac{\lambda_0\tau(\mu-\mu_0)^2}{2}\right] \\
&= \log\left(\frac{b_0^{a_0}\sqrt{\lambda_0}}{\Gamma(a_0)\sqrt{2\pi}}\right) + (a_0-\frac{1}{2})\mathbb{E}_{q(\tau)}[\log\tau] - b_0\mathbb{E}_{q(\tau)}[\tau] \\
&\quad - \frac{\lambda_0\mathbb{E}_{q(\tau)}[\tau]}{2}\left(\mathbb{E}_{q(\mu)}[\mu^2] - 2\mu_0\mathbb{E}_{q(\mu)}[\mu] + \mu_0^2\right) \\
&\stackrel{+}{=} (a_0-\frac{1}{2})\mathbb{E}_{q(\tau)}[\log\tau] - b_0\mathbb{E}_{q(\tau)}[\tau] - \frac{\lambda_0\mathbb{E}_{q(\tau)}[\tau]}{2}\left(\mathbb{E}_{q(\mu)}[\mu^2] - 2\mu_0\mathbb{E}_{q(\mu)}[\mu] + \mu_0^2\right)
\end{aligned} \tag{27}$$

And we can compute all of that because entropies are known and we have the following expectations:

$$\begin{aligned}
\mathbb{E}_{q(\mu)}[\mu] &= \mu_N \\
\mathbb{E}_{q(\mu)}[\mu^2] &= \frac{1}{\lambda_N} + \mu_N^2 \\
\mathbb{E}_{q(\tau)}[\tau] &= \frac{a_N}{b_N} \\
\mathbb{E}_{q(\tau)}[\log\tau] &= \psi(a_N) - \log b_N
\end{aligned} \tag{28}$$

Then we obtain this result :

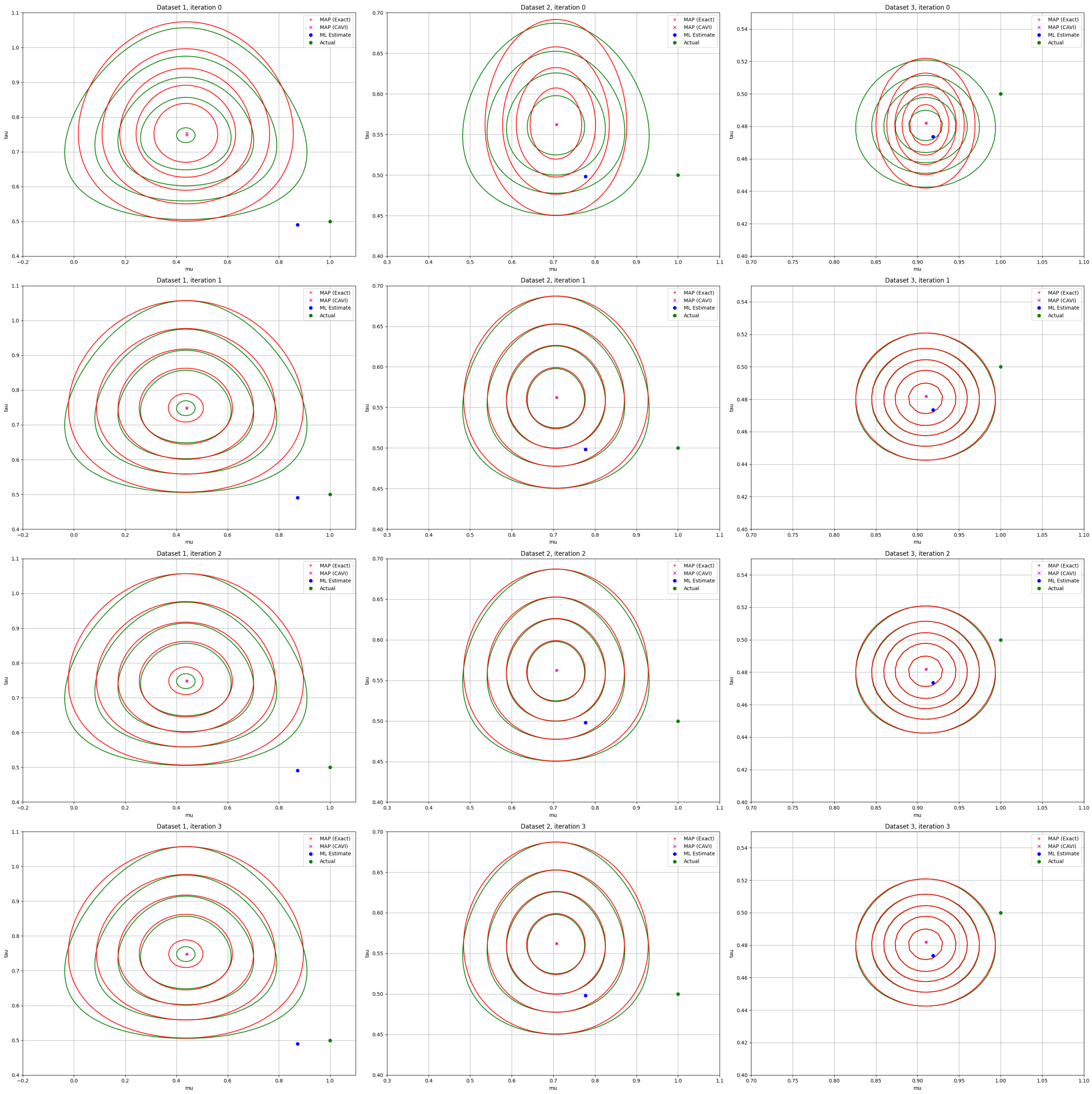


Figure 6: Contours of the approximations by VI and the exact posterior by datasets, by iterations

And we obtain an elbo plot :

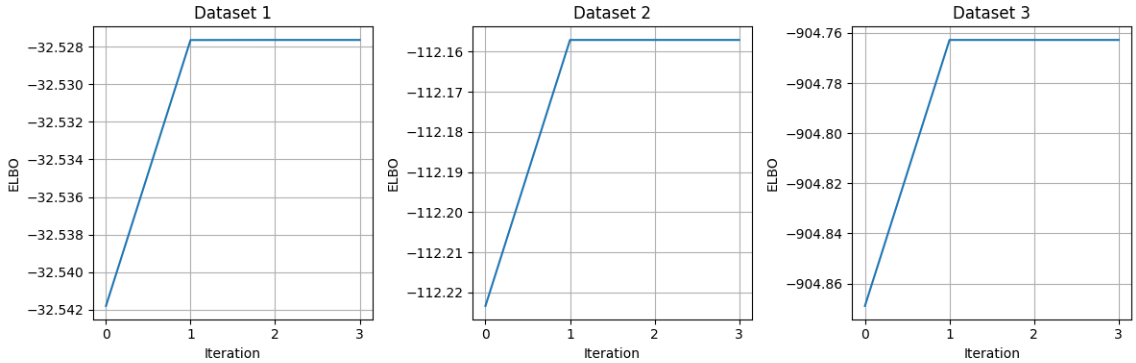


Figure 7: ELBO plot by datasets

The code is in appendix A.1.

4 SVI - LDA

Question 4.16

According to the Hoffman paper, the local hidden variables are defined by the model where the distribution of each observation x_n only depends on its corresponding local variable z_n and the global variables $\beta_{1:K}$. Therefore, we can write:

$$p(x, z, \beta | alpha) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta) \tag{29}$$

Because:

$$p(x_n, z_n|x_{-n}, z_{-n}, \beta, \alpha) = p(x_n, z_n|\beta, \alpha) \tag{30}$$

Question 4.17

In this figure the global hidden variables are the topics $\beta_{1:K}$ and the local hidden variables are the topic proportions θ_d and the topic assignments $z_{d,1:N}$.

Question 4.18

The ELBO formula is:

$$\mathcal{L}(q) = \mathbb{E}_{q(\theta,z,\beta)}[\log p(w, \theta, z, \beta)] - \mathbb{E}_{q(\theta,z,\beta)}[\log q(\theta, z, \beta)] \tag{31}$$

And here we recall that we have:

$$p(w, z, \theta, \beta) = \prod_{k=1}^{K} p(\beta_k) \prod_{d=1}^{D} p(\theta_d) \prod_{d=1}^{D} \prod_{n=1}^{N} p(z_{dn}|\theta_d)p(w_{dn}|\beta_{z_{dn}}) \tag{32}$$

Therefore we have:

$$\begin{aligned}
&\mathbb{E}_{q(\theta,z,\beta)}[\log p(w,\theta,z,\beta)] \\
&= \mathbb{E}_{q(\theta,z,\beta)}\left[\sum_{k=1}^{K}\log p(\beta_k) + \sum_{d=1}^{D}\log p(\theta_d) + \sum_{d=1}^{D}\sum_{n=1}^{N}\log p(z_{dn}|\theta_d)p(w_{dn}|\beta_{z_{dn}})\right] \\
&\qquad - \mathbb{E}_{q(\theta,z,\beta)}\left[\sum_{k=1}^{K}\log q(\beta_k) + \sum_{d=1}^{D}\log q(\theta_d) + \sum_{d=1}^{D}\sum_{n=1}^{N}\log q(z_{dn}|\theta_d) + \log q(w_{dn}|\beta_{z_{dn}})\right] \\
&= \sum_{k=1}^{K}\mathbb{E}_{q(\beta_k)}[\log p(\beta_k)] + \sum_{d=1}^{D}\mathbb{E}_{q(\theta_d)}[\log p(\theta_d)] + \sum_{d=1}^{D}\sum_{n=1}^{N}\mathbb{E}_{q(z_{dn}),q(\theta_d)}[\log p(z_{dn}|\theta_d)] \\
&\qquad + \sum_{d=1}^{D}\sum_{n=1}^{N}\mathbb{E}_{q(z_{dn}),q(\beta_{z_{dn}})}[\log p(w_{dn}|\beta_{z_{dn}})] \\
&\qquad + \sum_{k=1}^{K}\mathbb{H}_q[\beta_k] + \sum_{d=1}^{D}\mathbb{H}_q[\theta_d] + \sum_{d=1}^{D}\sum_{n=1}^{N}\mathbb{H}_q[z_{dn}|\theta_d] + \mathbb{H}_q[w_{dn}|\beta_{z_{dn}}]
\end{aligned} \tag{33}$$

Using Hoffman updates equations, we can write:

$$\begin{aligned}
\lambda_k &= \eta + \sum_{d=1}^{D}\sum_{n=1}^{N} z_{dn}^{k} w_{dn} \\
\gamma_d &= \alpha + \sum_{n=1}^{N} z_{dn} \\
\phi_{dn} &= \log \beta_{k,w_{dn}} + \log \theta_{dk}
\end{aligned} \tag{34}$$

And by using the expectations given in the Hoffman paper, we can write:

$$\begin{aligned}
\sum_{k=1}^{K}\mathbb{E}_{q(\beta_k)}[\log p(\beta_k)] &= \sum_{k=1}^{K}\sum_{\nu=1}^{W}(\eta-1)\mathbb{E}_{q(\beta_{k\nu})}[\log \beta_{k\nu}] \\
&= \sum_{k=1}^{K}\sum_{\nu=1}^{W}(\eta-1)\left(\Psi(\lambda_{k\nu}) - \Psi\left(\sum_{y=1}^{W}\lambda_{ky}\right)\right)
\end{aligned} \tag{35}$$

$$\begin{aligned}
\sum_{d=1}^{D}\mathbb{E}_{q(\theta_d)}[\log p(\theta_d)] &= \sum_{d=1}^{D}\sum_{k=1}^{K}(\alpha-1)\mathbb{E}_{q(\theta_{dk})}[\log \theta_{dk}] \\
&= \sum_{d=1}^{D}\sum_{k=1}^{K}(\alpha-1)\left(\Psi(\gamma_{dk}) - \Psi\left(\sum_{v=1}^{K}\gamma_{dv}\right)\right)
\end{aligned} \tag{36}$$

$$\begin{aligned}
\sum_{d=1}^{D}\sum_{n=1}^{N}\mathbb{E}_{q(z_{dn}),q(\theta_d)}[\log p(z_{dn}|\theta_d)] &= \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q(z_{dn})}[z_{dn}^{k}]\mathbb{E}_{q(\theta_{dk})}[\log \theta_{dk}] \\
&= \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}\phi_{dn}^{k}\left(\Psi(\gamma_{dk}) - \Psi\left(\sum_{v=1}^{K}\gamma_{dv}\right)\right)
\end{aligned} \tag{37}$$

$$\begin{aligned}\sum_{d=1}^{D}\sum_{n=1}^{N}\mathbb{E}_{q(z_{dn}),q(\beta_{z_{dn}})}[\log p(w_{dn}|\beta_{z_{dn}})] &= \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\nu=1}^{W} w_{dn}^{k}\mathbb{E}_{q(z_{dn})}[z_{dn}^{k}]\mathbb{E}_{q(\beta_{k\nu})}[\log \beta_{k\nu}] \\ &= \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\nu=1}^{W} w_{dn}^{k}\phi_{dn}^{k}\left(\Psi(\lambda_{k\nu}) - \Psi\left(\sum_{y=1}^{W}\lambda_{ky}\right)\right)\end{aligned} \tag{38}$$

And the entropies can be found on Wikipedia.

Question 4.19

The code is in appendix A.2.

5 BBVI

Question 5.20

We have the simple model:

$$\begin{aligned} X|\theta &\sim \mathcal{N}(\theta, \sigma^2) \\ \theta &\sim Gamma(\alpha, \beta)\end{aligned} \tag{39}$$

With α, β and σ^2 known. Now, we will derive the gradient estimate w.r.t. ν without Rao-Blackwellization using one sample $z \sim q_\nu(\theta)$, $q_\nu(\theta) = LogNormal(\nu, \epsilon^2)$. We recall the formula:

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S}\sum_{s=1}^{S}\nabla_\lambda \log q(z_s|\lambda)\left(\log p(x, z_s) - \log q(z_s|\lambda)\right) \tag{40}$$

Where $z_s \sim q(z|\lambda)$. Therefore, here we have:

$$\begin{aligned}\nabla_\nu \mathcal{L} &\approx \nabla_\nu \log q(z|\nu)\left(\log p(x,z) - \log q(z|\nu)\right) \\ &\approx \nabla_\nu \log\left(\frac{\exp\left(-\frac{(\log\theta-\nu)^2}{2\epsilon^2}\right)}{\theta\epsilon\sqrt{2\pi}}\right)\left(\log\left(\frac{\exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}}\right)\right. \\ &\qquad \left. + \log\left(\frac{\beta^\alpha\theta^{\alpha-1}e^{-\beta\theta}}{\Gamma(\alpha)}\right) - \log\left(\frac{\exp\left(-\frac{(\log\theta-\nu)^2}{2\epsilon^2}\right)}{\theta\epsilon\sqrt{2\pi}}\right)\right) \\ &\approx \nabla_\nu\left(-\frac{(\log\theta-\nu)^2}{2\epsilon^2}\right)\left(-\frac{(x-\theta)^2}{2\sigma^2} - \log\left(\sigma\sqrt{2\pi}\right)\right. \\ &\qquad \left. -\beta\theta + (\alpha-1)\log\theta + \log\left(\frac{\beta^\alpha}{\Gamma(\alpha)}\right) + \frac{(\log\theta-\nu)^2}{2\epsilon^2} + \log\left(\theta\epsilon\sqrt{2\pi}\right)\right) \\ &\approx \frac{\log\theta-\nu}{\epsilon^2}\left(\frac{(\sigma(\log\theta-\nu))^2 - (\epsilon(x-\theta))^2}{2\sigma^2\epsilon^2} + \log\left(\frac{\epsilon}{\sigma}\right)\right. \\ &\qquad \left. -\beta\theta + \alpha\log\theta + \log\left(\frac{\beta^\alpha}{\Gamma(\alpha)}\right)\right)\end{aligned} \tag{41}$$

A Appendix

A.1 CAVI

1A-3-CAVI

December 5, 2023

1 *Assignment 1.3 - CAVI*

Consider the model defined by Equation (10.21)-(10-23) in Bishop, for which DGM is presented below:

1.0.1 *Question 1.3.12:*

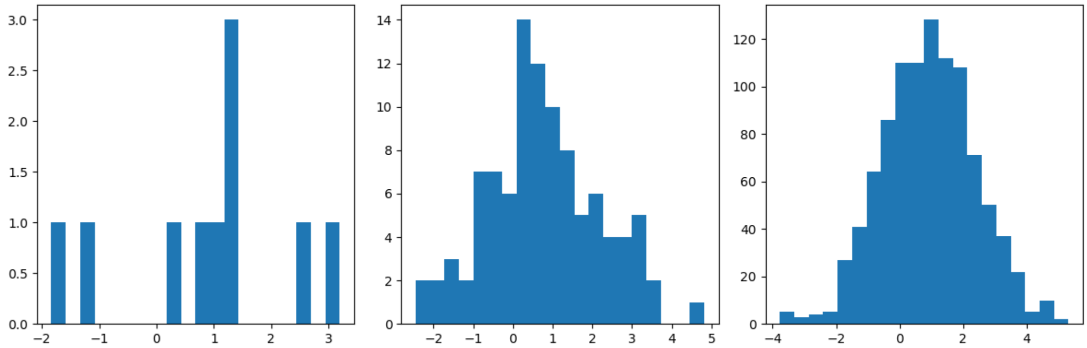
Implement a function that generates data points for the given model.

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

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

  return D
```

Set = 1, = 0.5 and generate datasets with size N=10,100,1000. Plot the histogram for each of 3 datasets you generated.

[ ]:
```
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
import matplotlib.pyplot as plt
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('../images/12_data.png')
plt.show()
```



1.0.2 *Question 1.3.13:*

Find ML estimates of the variables and

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

         tau_ml = 1 / x_var
         mu_ml = x_mean

         return mu_ml, tau_ml
```

1.0.3 *Question 1.3.14:*

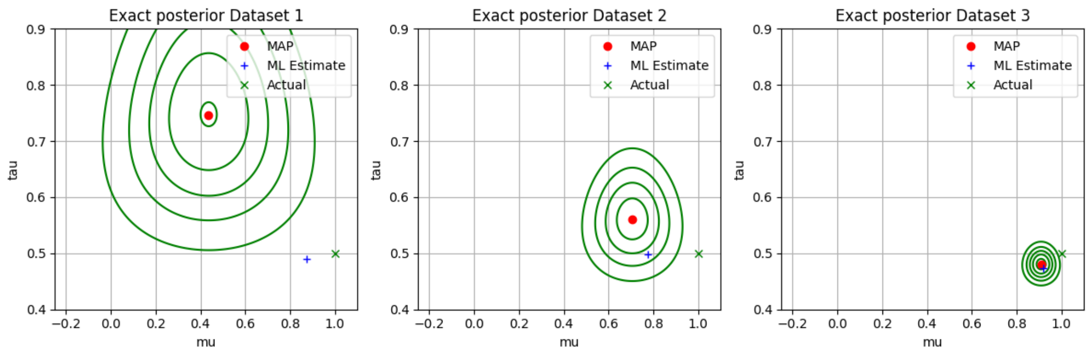
What is the exact posterior? First derive it in closed form, and then implement a function that computes it for the given parameters:

```
[ ]: 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-1)/ 2
         b_prime = b_0 + 0.5 * (np.sum(D**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(-0.25, 1.1, 200)
taus = np.linspace(0.4, 0.9, 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_exact = mus[np.argmax(np.max(Z_exact, axis=1))]
  tau_max_exact = 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_exact, tau_max_exact, '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.tight_layout()
plt.savefig('../images/14_contours.png')
plt.show()
```



1.0.4 *Question 1.3.15:*

You will implement the VI algorithm for the variational distribution in Equation (10.24) in Bishop. Start with introducing the prior parameters:

Continue with a helper function that computes ELBO:

[ ]:
```
def compute_E_tau(a_N, b_N):
    E_tau = a_N / b_N

    return E_tau

def compute_E_mu_2(mu_N, lambda_N):
    E_mu_2 = mu_N**2 + 1/lambda_N

    return E_mu_2

def compute_E_log_tau(a_N, b_N):
    E_log_tau = psi(a_N) - np.log(b_N)

    return E_log_tau
```

[ ]:
```
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)
    E_tau = compute_E_tau(a_N, b_N)
    E_mu_2 = compute_E_mu_2(mu_N, lambda_N)
    E_log_tau = compute_E_log_tau(a_N, b_N)

    # compute the elbo
    # E[log p(D|mu, tau)]
    E_log_p_D = N/2 * E_log_tau - 0.5*E_tau * (x_2_sum - 2*N*x_mean*mu_N +␣
↪N*E_mu_2)
```

```
    # E[log p(mu, tau)]
    E_log_p_mu_tau = (a_0-0.5)*E_log_tau - b_0*E_tau - 0.
↪5*lambda_0*E_tau*(E_mu_2 + mu_0**2 - 2*mu_0*mu_N)

    # Entropy of mu
    entropy_mu = norm.entropy(loc=mu_N, scale=1/np.sqrt(lambda_N))
    # Entropy of tau
    entropy_tau = gamma.entropy(a=a_N, scale=1/b_N)

    elbo = E_log_p_D + E_log_p_mu_tau + entropy_mu + entropy_tau

    return elbo
```

Now, implement the CAVI algorithm:

```
[ ]: def CAVI(D, a_0, b_0, mu_0, lambda_0, iter=5):
       # make an initial guess for the expected value of tau
       E_tau = 1

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

       # Constants
       a_N = a_0 + (N+1) / 2
       mu_N = (lambda_0 * mu_0 + N * x_mean) / (lambda_0 + N)
       E_mu = mu_N

       # Variables
       b_Ns = []
       lambda_Ns = []

       # ELBO
       elbos = []

       # CAVI iterations ...
       for i in range(iter):
         # update the values for the variational parameters
         lambda_N = (lambda_0 + N) * E_tau

         E_mu_2 = compute_E_mu_2(mu_N, lambda_N)
         b_N = b_0 + 0.5 * (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))

         E_tau = compute_E_tau(a_N, b_N)
```

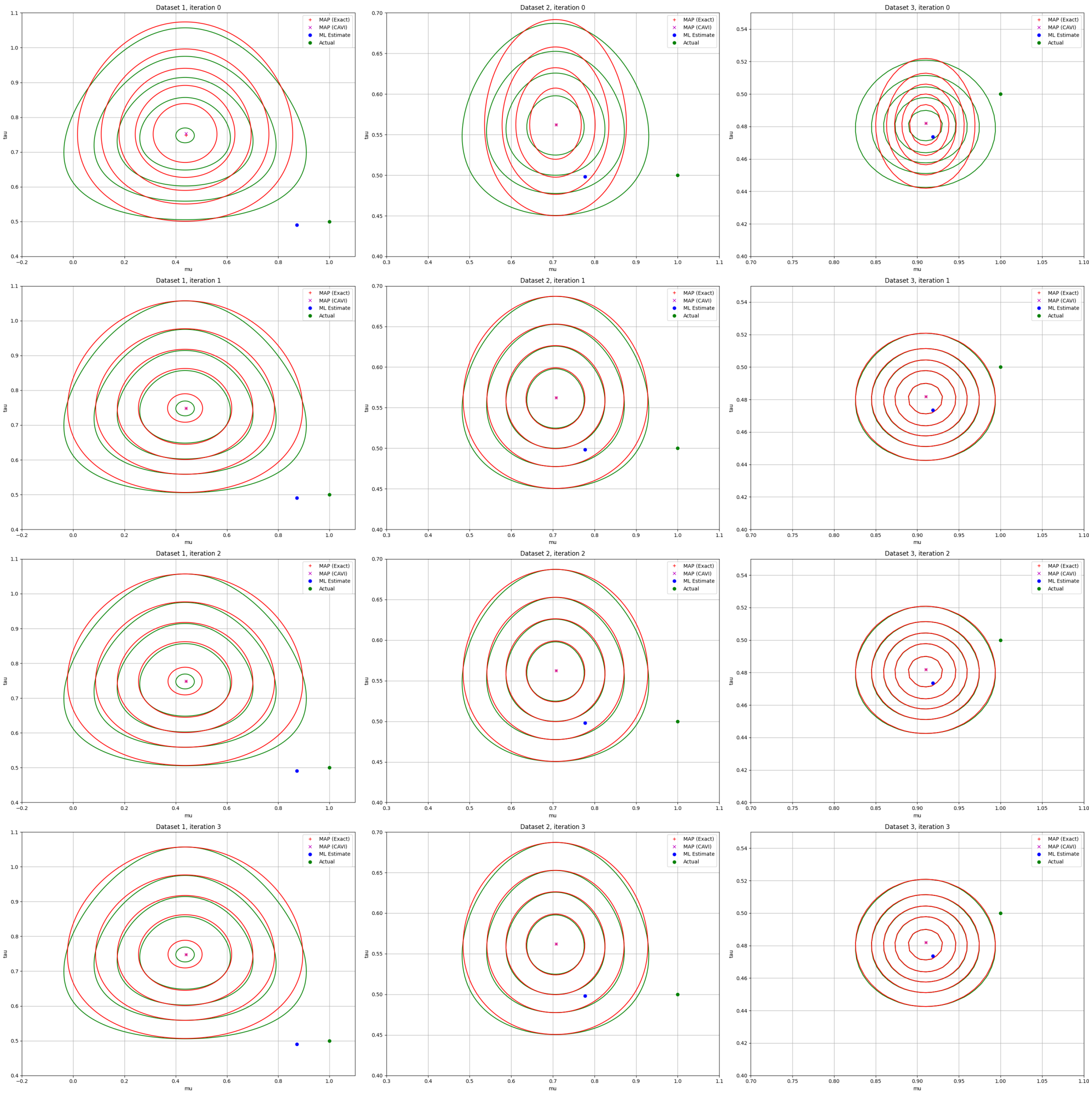
```
    b_Ns.append(b_N)
    lambda_Ns.append(lambda_N)
    # save ELBO for each iteration, plot them afterwards to show convergence
    elbos.append(compute_elbo(D, a_0, b_0, mu_0, lambda_0, a_N , b_N, mu_N,␣
↪lambda_N))

  return a_N, b_N, mu_N, lambda_N, elbos, b_Ns, lambda_Ns
```

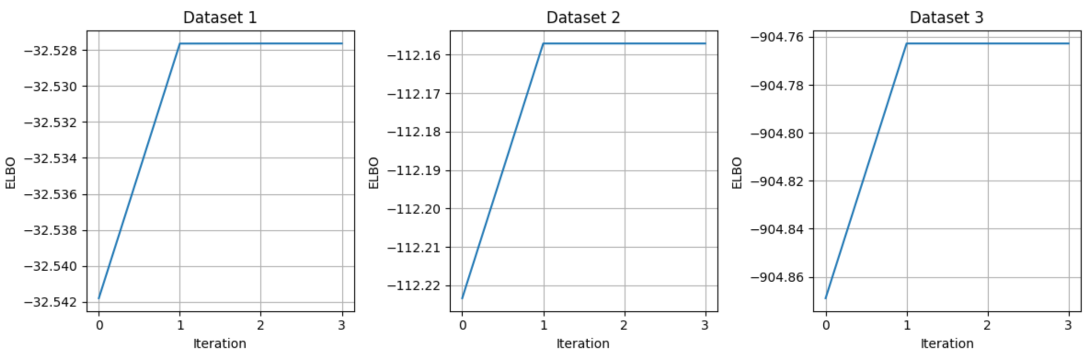
Run the VI algorithm on the datasets. Compare the inferred variational distribution with the exact posterior and the ML estimate. Visualize the results and discuss your findings.

```
[ ]: def compute_z_exact(mus, taus, a_, b_, mu_, lambda_):
         z = np.zeros((len(mus), len(taus)))
         pTau = gamma(a=a_, loc=0, scale=1/b_)
         for j, tau in enumerate(taus):
             pMu = norm(loc=mu_, scale=1/np.sqrt(lambda_*tau))
             z[:, j] = pMu.pdf(mus) * pTau.pdf(tau)

         return z

     def compute_z_cavi(mus, taus, a_, b_, mu_, lambda_):
         pTau = gamma(a=a_, loc=0, scale=1/b_)
         pMu = norm(loc=mu_, scale=1/np.sqrt(lambda_))
         z = np.outer(pMu.pdf(mus), pTau.pdf(taus))
         return z
```

```
[ ]: iter = 4 # number of iterations for CAVI
     mus = np.linspace(-0.2, 1.1, 200)
     taus = np.linspace(0.1, 1.1, 200)

     xlims = [[-0.2, 1.1], [0.3, 1.1], [0.7, 1.1]]
     ylims = [[0.4, 1.1], [0.4, 0.7], [0.4, 0.55]]

     elbos_list = []

     fig, axs = plt.subplots(iter, 3, figsize=(30, 30))
     for i, dataset in enumerate([dataset_1, dataset_2, dataset_3]):
       mu_ml, tau_ml = ML_est(dataset)
       a_N, b_N, mu_N, lambda_N, elbos, b_Ns, lambda_Ns = CAVI(dataset, a_0, b_0,␣
     ↪mu_0, lambda_0, iter=iter)
       a_T, b_T, mu_T, lambda_T = compute_exact_posterior(
           dataset, a_0, b_0, mu_0, lambda_0)

       elbos_list.append(elbos)

       for j in range(iter):
         Z_exact = compute_z_exact(mus, taus, a_T, b_T, mu_T, lambda_T)
```

```python
    Z_cavi = compute_z_cavi(mus, taus, a_N, b_Ns[j], mu_N, lambda_Ns[j])
    # Finding the maximum of the exact posterior
    mu_max_exact = mus[np.argmax(np.max(Z_exact, axis=1))]
    tau_max_exact = taus[np.argmax(np.max(Z_exact, axis=0))]
    # Finding the maximum of the CAVI approximation
    mu_max_cavi = mus[np.argmax(np.max(Z_cavi, axis=1))]
    tau_max_cavi = taus[np.argmax(np.max(Z_cavi, axis=0))]
    # Plotting the results
    axs[j, i].contour(*np.meshgrid(mus, taus), Z_exact.T,
                  levels=5, colors=['green'])
    axs[j, i].contour(*np.meshgrid(mus, taus), Z_cavi.T,
                    levels=5, colors=['red'])
    axs[j, i].plot(mu_max_exact, tau_max_exact, 'r+', label='MAP (Exact)')
    axs[j, i].plot(mu_max_cavi, tau_max_cavi, 'mx', label='MAP (CAVI)')
    axs[j, i].plot(mu_ml, tau_ml, 'bo', label='ML Estimate')
    axs[j, i].plot(MU, TAU, 'go', label='Actual')
    axs[j, i].legend()
    axs[j, i].grid()
    axs[j, i].set_xlabel('mu')
    axs[j, i].set_ylabel('tau')
    axs[j, i].set_title(f'Dataset {i+1}, iteration {j}')
    axs[j, i].set_xlim(xlims[i])
    axs[j, i].set_ylim(ylims[i])
plt.tight_layout()
plt.savefig('../images/15_contours.png')
plt.show()

# Plot ELBOs
fig, axs = plt.subplots(1, 3, figsize=(12, 4))
for i in range(3):
  axs[i].plot(elbos_list[i])
  axs[i].set_xlabel('Iteration')
  axs[i].set_ylabel('ELBO')
  axs[i].set_title(f'Dataset {i+1}')
  axs[i].grid()
plt.tight_layout()
plt.savefig('../images/15_elbo.png')
plt.show()
```


Dataset 1, iteration 0
Dataset 2, iteration 0
Dataset 3, iteration 0
Dataset 1, iteration 1
Dataset 2, iteration 1
Dataset 3, iteration 1
Dataset 1, iteration 2
Dataset 2, iteration 2
Dataset 3, iteration 2
Dataset 1, iteration 3
Dataset 2, iteration 3
Dataset 3, iteration 3
MAP (Exact)
MAP (CAVI)
ML Estimate
Actual
mu
tau


Dataset 1
Dataset 2
Dataset 3
ELBO
Iteration
−32.528
−32.530
−32.532
−32.534
−32.536
−32.538
−32.540
−32.542
−112.16
−112.17
−112.18
−112.19
−112.20
−112.21
−112.22
−904.76
−904.78
−904.80
−904.82
−904.84
−904.86
0
1
2
3

A.2 SVI