Assignment 1AD, 2025

DD2434 Machine Learning, Advanced Course

Bruno Carchia
carchia@kth.se

December 9, 2025

D Section

D.1

Question 1.1.1

Starting from the definition of the KL divergence $\mathrm{KL}(q(Z)||p(Z|X))$, we want to rewrite it and identify the quantity referred as the Evidence Lower Bound (ELBO).

$$\begin{aligned}
\mathrm{KL}(q(z)||p(z|x)) &= E_q\left[\log \frac{q(z)}{p(z|x)}\right] \\
&= \int q(z) \log \frac{q(z)}{p(z|x)} dz \\
&= \int q(z) \log \frac{q(z)}{\frac{p(z,x)}{p(x)}} dz \\
&= \int q(z) \log \frac{q(z)}{p(z,x)} dz + \int q(z) \log p(x) dz \\
&= E_q\left[\log \frac{q(z)}{p(z,x)}\right] + \log p(x)
\end{aligned}$$

Knowing that the ELBO is defined as $\mathcal{L} = \mathrm{KL}(p(x,z)|q(z))$

$$\mathrm{KL}(q(z)||p(z|x)) = -\mathcal{L} + \log p(x)$$

$$\log p(x) = \mathcal{L} + \mathrm{KL}(q(z)||p(z|x))$$

The ELBO is a lower bound because the KL divergence is always non-negative. Maximizing the ELBO effectively maximizes the lower bound on $\log p(x)$ and minimizes the KL divergence.

Question 1.1.2

- A more expressive variational family can better approximate the true posterior, reducing $\mathrm{KL}(q(z)||p(z|x))$ and increasing the ELBO (a fully factorized mean field distribution produces a looser ELBO because it ignores dependencies between latent variables)
- A more expressive variational family can capture closely dependencies between latent variables, yielding $q(z)$ closer to the true posteriori $p(z|x)$ (a fully factorized mean field distribution produces a less accurate posterior approximation)

D.2

Question 1.1.3

Considering a mean field assumption on our variation distribution $q(Z_1, Z_2, Z_3) = q(Z_1)q(Z_2)q(Z_3)$, we want to prove that $\log q_1^*(Z_1) = E_{-Z_1}[\log p(X, Z)]$

$$\begin{aligned}
\mathcal{L} &= \int \prod_i^3 q(z_i) \log \left(\frac{p(x,z)}{\prod_l^3 q(z_l)} \right) dz \\
&= \int \prod_i^3 q(z_i) \left[\log\left(p(x,z)\right) - \sum_l^3 \log q(z_l) \right] dz \\
&= \int \prod_i^3 q(z_i) \log p(x,z) dz - \int \prod_i^3 q(z_i) \sum_l^3 \log q(z_l) dz
\end{aligned}$$

We work individually on these two terms:

- First term can be decomposed in an outer component that depends on z_j and an the inner component that depends on all the other variables z_{-j}

$$\int \prod_i^3 q(z_i) \log p(x,z) dz = \int_{z_j} q(z_j) \left[\int_{z_{-j}} \prod_{i \neq j} q(z_i) \log p(x,z) dz_{-j} \right] dz_j$$

  We define $\log \tilde{p}(x, z_j) = E_{\prod_{l \neq j} q(z_l)}[\log p(x, z)]$, the first term becomes

$$\int_{z_j} q(z_j) \log \tilde{p}(x, z_j) dz_j$$

- Second term can be rewritten in the following way

$$-\int \prod_i^3 q(z_i) \sum_l^3 \log q(z_l) dz = -\int_z \sum_l^3 \log q(z_l) \prod_i^3 q(z_i) dz$$

After switching the sum and the integral, we can decompose the term in an outer (depends on all the variables excepting for z_l) and inner component (depends just on z_l)

$$= -\sum_{l}^{3} \int_{z} \log q(z_l) \prod_{i}^{3} q(z_i) dz$$
$$= -\sum_{l}^{3} \int_{z_{-l}} \prod_{i \neq l} q(z_i) \left[\int_{z_l} q(z_l) \log q(z_l) dz_l \right] dz_{-l}$$

The inner component does not depend on terms related to z_{-l} , the external integral gives 1 and we got as final result

$$= -\sum_{l}^{3} \int_{z_l} q(z_l) \log q(z_l) dz_l$$

The ELBO is

$$\mathcal{L} = \int_{z_j} q(z_j) \log \tilde{p}(x, z_j) dz_j - \sum_{l}^{3} \int_{z_l} q(z_l) \log q(z_l) dz_l$$

But our objective is to maximize the ELBO by iteratively optimizing a single variational factor $q(z_j)$ while holding all other factors $q(z_{-j})$ constant. In other words we treat as constant all the terms that do not depend on z_j

$$\mathcal{L} = \int_{z_j} q(z_j) \log \tilde{p}(x, z_j) dz_j - \int_{z_j} q(z_j) \log q(z_j) dz_j + \text{const}$$
$$= E_q(z_j) \left[\frac{\log \tilde{p}(x, z_j)}{q(z_j)} \right]$$
$$= -\text{KL}(\tilde{p}(x, z_j) || q(z_j))$$

Maxiziming the ELBO wrt a single variational factor $q(z_j)$ is equivalent to minimizing $\text{KL}(\tilde{p}(x, z_j) || q(z_j))$: the minimum occurs when $q(z_j)* = \tilde{p}(x, z_j)$

$$\log \tilde{p}(x, z_j) = \log q(z_j)* = E_{\prod_{l \neq j} q(z_l)}[\log p(x, z)]$$

In this case $j = 1$

D.3

We will analyze the model with Normal-likelihood and NormalGamma prior of 1E.3. instead of using the Coordinate Ascent Variational Inference (CAVI) algorithm, we will employ Black-Box Variational Inference (BBVI). This BBVI will utilize the RE-INFORCE gradient estimator (in its basic, high-variance form) to infer the variational distributions $q(\mu)$ and $q(\tau)$ under the mean-field assumption $q(\mu, \tau) = q(\mu)q(\tau)$

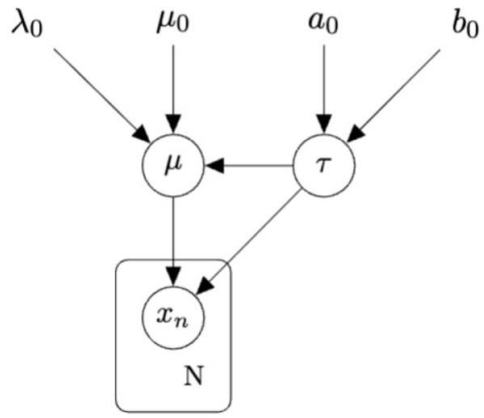

Figure 1: Bayesian network of the Normal-NormalGamma model

Question 1.2.4

We provide the final expressisions for

- Log Likelihood $\log P(D|\mu,\tau)$

$$\begin{aligned}\log P(D|\mu,\tau) &= \log\left(\prod_{n=1}^{N} f_{\tau,\mu}(x_n)\right)\\ &= \sum_{n=1}^{N}\log\left(\sqrt{\frac{\tau}{2\pi}}e^{-\frac{\tau}{2}(x_n-\mu)^2}\right)\\ &= \sum_{n=1}^{N}\left[0.5\log\tau - 0.5\log 2\pi - 0.5\tau(x_n-\mu)^2\right]\end{aligned}$$

- Log Prior $\log P(\tau,\mu)$

$$\begin{aligned}\log P(\tau,\mu) &= \log P(\tau)P(\mu|\tau)\\ &= \log\left[\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\sqrt{\frac{\lambda_0\tau}{2\pi}}\tau^{\alpha_0-1}e^{-\beta_0\tau}e^{-\frac{\lambda_0\tau}{2}(\mu-\mu_0)^2}\right]\\ &= \alpha_0\log\beta_0 - \log\Gamma(\alpha_0) + 0.5\log\lambda_0\tau - 0.5\log 2\pi + (\alpha_0-1)\log\tau - \beta_0\tau - \frac{\lambda_0\tau}{2}(\mu-\mu_0)^2\end{aligned}$$

- Log Variational Distribution $q(z[s]|\lambda)$

$$\log q(\mu,\tau|\mu_N,\lambda_N,\alpha_N,\beta_N) = \log q(\mu|\mu_N,\lambda_N) + \log q(\tau|\alpha_N,\beta_N)$$

Knowing that $q(\tau)\sim\text{Gamma}(\alpha_N,\beta_N)$ and $q(\mu)\sim\text{Normal}(\mu_N,\lambda_N^{-1})$

$$\left[0.5\log\frac{\lambda_N}{2\pi} - \frac{\lambda_N(\mu-\mu_N)^2}{2}\right] + \left[\alpha_N\log\beta_N + (\alpha_N-1)\log\tau - \log\Gamma(\alpha_N) - \beta_N\tau\right]$$

- Score function $\nabla_\lambda \log q(z[s]|\lambda)$

  $\nabla_{\mu_N,\lambda_N,\alpha_N,\beta_N} \log q(\mu,\tau|\mu_N,\lambda_N,\alpha_N,\beta_N) = \nabla_{\mu_N,\lambda_N} \log q(\mu|\mu_N,\lambda_N) + \nabla_{\alpha_N,\beta_N} \log q(\tau|\alpha_N,\beta_N)$

  We compute each term individually

  - $\nabla_{\mu_N,\lambda_N} \log q(\mu|\mu_N,\lambda_N)$
    - $\nabla_{\mu_N} \log q(\mu|\mu_N,\lambda_N) = \lambda_N(\mu-\mu_N)$
    - $\nabla_{\lambda_N} \log q(\mu|\mu_N,\lambda_N) = +0.5\frac{\frac{1}{2\pi}}{\frac{\lambda_N}{2\pi}} - 0.5(\mu-\mu_N)^2 = \frac{1}{2\lambda_N} - \frac{(\mu-\mu_N)^2}{2}$
  - $\nabla_{\alpha_N,\beta_N} \log q(\tau|\alpha_N,\beta_N)$
    - $\nabla_{\alpha_N} \log q(\tau|\alpha_N,\beta_N) = \log\beta_N + \log\tau - \psi(\lambda_N)$, where $\psi(x) = \frac{d\ln\Gamma(x)}{dx}$
    - $\nabla_{\beta_N} \log q(\tau|\alpha_N,\beta_N) = \frac{\alpha_N}{\beta_N} - \tau$
- $\log P(D, z[s]) = \log P(D,\mu_s,\tau_s) = \log P(D|\mu_s,\tau_s) + \log P(\mu_s,\tau_s)$

Question 1.2.5

We implemented algorithm 1 of the BBVI paper using Pytorch.