Università degli Studi di Padova

Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in

Scienze Statistiche

Variable selection for Poisson regression model via mean field variational Bayes

Relatore: Prof. Mauro Bernardi
Università degli Studi di Padova,
Dipartimento di Scienze Statistiche

Correlatore: Dott. Nicolas Bianco
Universitat Pompeu Fabra,
Department of Economics and Business

Laureando: Daniele Cugnigni
Matricola N. 2054519

Anno Accademico 2022/2023

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

One of the most challenging issues in modern Statistics, especially within the framework of Bayesian Statistics, concerns approximate inference for complex statistical models. Indeed, the amount of data we have to deal with has dramatically grown, requiring suitable techniques to compute many models in a reasonable amount of time. For this reason, the leading paradigm of Markov chain Monte Carlo (Gelfand and Smith, 1990; Hastings, 1970) has sometimes left the place to faster techniques, such as variational inference. The latter is a set of methods from machine learning which aims to approximate probability densities through optimization rather than sampling (Blei et al., 2017). On the other hand, this new set of methods may suffer of a limited accuracy compared to Markov chain Monte Carlo (MCMC), which can provide more accurate estimates through the increasing of the Monte Carlo sample size (Ormerod and Wand, 2010).

Variational approximations consist in finding the densities which optimize a lower bound, introduced with the aim to provide a more tractable computational problem, and giving an approximate solution to the original problem based on the likelihood function. The most famous variational inference technique is probably variational Bayes, which consists in minimizing the Kullback-Leibler divergence (Kullback and Leibler, 1951) between a proposed density $q(\cdot)$ and the true density $p(\cdot)$. However, alternative divergences have been proposed (Dieng et al., 2017; Minka, 2005). Finally, convergence diagnostics that go beyond the investigation of the behaviour of the lower bound are presented by Yao et al. (2018).

Although many recent papers focus on variational inference, there is still uncovered area of research. In this thesis we present a variational Bayes framework to deal with Poisson regression models and variable selection.

The thesis is organized as follows. *Chapter 1* presents Bayesian variational inference in general, focusing in particular on both non-parametric and semi-parametric mean field variational Bayes (MFVB). The methodological discussion is accompanied by illustrative examples involving the multivariate Gaussian distribution and the Bayesian Poisson regression model with non-informative prior. In these two examples, the similar results in terms of inference accuracy between MFVB and MCMC, and the advantage of MFVB with respect to MCMC in terms of computational cost are assessed.

Chapter 2 tackles the variable selection problem in the Bayesian Poisson regression model. Three solutions are considered: a continuos shrinkage prior, that is the horseshoe prior (Carvalho et al., 2010), the spike-and-slab prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993, 1997) and the Bernoulli-Gaussian prior (Ormerod et al., 2017; Bernardi et al., 2023). The estimation is carried out within a semi-parametric mean field variational Bayes framework.

An extensive simulation study is implemented in order to evaluate the different models in terms of inference and quality of variable selection, and understand their limits. The proposed algorithms are tested with respect to popular methods, such as Poisson lasso, Gaussian lasso (Tibshirani, 1996) and EM variable selection (EMVS) approach (Ročková and George, 2014). The metrics used are the *mean squared error* (MSE) for the inference accuracy, and *F1-score* and *classification accuracy* for the variable selection. In addition, the *area under the curve* (AUC) for EMVS, Poisson model with spike-and-slab and Bernoulli-Gaussian prior is computed. Finally, a comparison in terms of computational cost is provided.

Chapter 3 deals with an application to real data. The dataset considered is *Football 2022-2023*, which contains informations on the football player performances in the major european leagues. All the implemented models are tested against Poisson lasso, generalized linear model (GLM) Poisson, Gaussian lasso and EMVS, both in terms of in-sample estimates and out-of-sample forecasting accuracy.

Chapter 1

Approximate Bayesian inference

The approximate inference is one of the possible solutions to solve the problem of increasing statistical models complexity, especially in Bayesian framework thanks to the development of several approximate inference algorithms. Let $\mathbf{y} = (y_1, \dots, y_n)^\intercal$ be the observed data vector from a random variable Y and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p) \in \Theta \subseteq \mathbb{R}^p$ a set of parameters. The main goal in Bayesian inference is to study the properties of the *posterior density* $p(\boldsymbol{\theta}|\mathbf{y})$, which is obtained updating a prior belief with the evidence of the observed data through the Bayes' theorem:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_\Theta p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}, \tag{1.1}$$

where

- $p(\boldsymbol{\theta})$ is the *prior* distribution on $\boldsymbol{\theta} \in \Theta$;
- $p(\mathbf{y}|\boldsymbol{\theta})$ is the *likelihood* of the data given the set of parameters;
- $p(\mathbf{y})$ is the *marginal likelihood*, also known as model evidence in machine learning literature.

However, in many situations the evaluation of $p(\mathbf{y})$ is intractable, either because the required integration has not a closed form or because its computation is time-consuming. A solution to this problem is to exploit approximate inference techniques. The latter can be divided in two big families: stochastic and deterministic approximations.

The most widespread stochastic method is probably Markov chain Monte Carlo (MCMC) (Robert and Casella, 2004, 2011), which involves the construction of an ergodic Markov chain on $\boldsymbol{\theta}$, whose stationary distribution converges to the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$. Sampling R values $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_R^*)$ from the posterior distribution, one can evaluate the properties of $\boldsymbol{\theta}|\mathbf{y}$, such as mean, median, variance and credibility intervals, based on $\boldsymbol{\theta}^*$. The most appealing feature of the stochastic methods is that it is possible to improve their accuracy by increasing the sample size R. In fact, in the limiting case $R \to \infty$, the approximation is exact. On the other hand, the main drawback of these methods is that they are computationally demanding and assessing their convergence is not trivial. For this reason they are mainly used in problems with a small number of observations and/or number of variables.

On the other hand, deterministic approximations rely on optimization rather than sampling. As a consequence, these methods allow for substantial gains in terms of computational cost, but their approximation accuracy is bounded. In particular, we focus on *variational approximations*, which has been used in a wide range of applications, ranging from statistics (Rustagi, 1976) to quantum mechanics (Sakurai, 1994), statistical mechanics (Parisi, 1988), machine learning (Hinton and van Camp, 1993) and then generalized to many probabilistic models, taking advantage of the graphical models' representation (Jordan et al., 1999). They have applications in both frequentist and Bayesian inference, but their use had a greater impact in the Bayesian literature due to presence of intractable calculus and computational issues in high-dimensional and complex models.

1.1 Variational inference

Variational inference (VI) relies on a density transform approach. The latter involves the approximation of a posterior density $p(\boldsymbol{\theta}|\mathbf{y})$ by another one which has the advantage of being more tractable. Let $\mathcal{Q}$ be a family of densities, with single element $q(\boldsymbol{\theta})$ called variational density. The goal of variational inference is to find the optimal density $q^*(\boldsymbol{\theta})$ that minimizes a divergence measure D between the variational density $q(\boldsymbol{\theta})$ and the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$. Given the couple $(\mathcal{Q}, D)$ we fall into different paradigms. Probably the mostly adopted one assumes the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951):

$$q^*(\boldsymbol{\theta}) = \underset{q(\boldsymbol{\theta}) \in \mathcal{Q}}{\operatorname{argmin}} \, KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})), \tag{1.2}$$

where

$$KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})) = \int_\Theta q(\boldsymbol{\theta}) \log \left\{ \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})} \right\} d\boldsymbol{\theta}. \tag{1.3}$$

This setting is called variational Bayes (VB). The main properties of KL divergence are:

- non-negativity: the KL divergence is always non-negative. For any two probability distributions $p(\cdot)$ and $q(\cdot)$, the KL divergence $KL(p||q) \geq 0$;
- asymmetry: the KL divergence is not symmetric. In other words, $KL(p||q) \neq KL(q||p)$. This property arises due to the logarithmic nature of the KL divergence.

The minimization problem defined in (1.2) requires the true posterior distribution, which, however, is unknown. This makes infeasible the direct solution of (1.2). An alternative formulation of the original problem can be stated leveraging on a manipulation of the logarithm of the marginal likelihood:

$$\begin{aligned}
\log p(\mathbf{y}) &= \log p(\mathbf{y}) \int q(\boldsymbol{\theta}) d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta}) \log p(\mathbf{y}) d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{\theta}; \mathbf{y})/q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})/q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{\theta}; \mathbf{y})}{q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta} \\
&\quad + \int q(\boldsymbol{\theta}) \log \left\{ \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})} \right\} d\boldsymbol{\theta} \\
&\geq \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{\theta}; \mathbf{y})}{q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta} = \log \underline{p}(\mathbf{y}; q),
\end{aligned} \tag{1.4}$$

following that $p(\mathbf{y}) \geq \underline{p}(\mathbf{y}; q)$. The logarithm of marginal likelihood can be decomposed in two terms:

- the lower bound on the log-marginal likelihood $\log \underline{p}(\mathbf{y}; q)$, also called the evidence lower bound (ELBO) in the machine learning literature;
- the Kullback-Leibler divergence between the variational density $q(\boldsymbol{\theta})$ and the true posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$, $KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y}))$.

From (1.4), notice that maximizing the lower bound $\underline{p}(\mathbf{y}; q)$ is equivalent to minimizing the KL divergence $KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y}))$. This is useful because all the quantities used to solve the new optimization problem are known.

Summarizing, the essence of the variational inference (Ormerod and Wand, 2010) is the approximation of the posterior density $p(\boldsymbol{\theta}|\mathbf{y})$ by a density $q(\boldsymbol{\theta})$ for which $\underline{p}(\mathbf{y}; q)$

is more tractable than $p(\mathbf{y})$. Tractability is achieved by restricting $q(\boldsymbol{\theta})$ to belong to a more manageable family of densities. Two common assumptions on $\mathcal{Q}$ are:

1. mean field (MF): $q(\boldsymbol{\theta}) = \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i)$, for some partition $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ of $\boldsymbol{\theta}$;
2. $q(\boldsymbol{\theta})$ is member of a parametric family of density functions.

Depending on the Bayesian model considered, both restrictions can have a different impact on the inference. The first restriction is non-parametric, in the sense that non-parametric distributions are pre-speficified for $q(\boldsymbol{\theta})$ and the only assumption is the independence as the factorisation of the joint distribution.

1.2 Non-parametric mean field approximation

In the case of non-parametric mean field approximation, the lower bound can be written in the following way:

$$\begin{aligned}\log \underline{p}(\mathbf{y}; q) &= \int \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i) \Big\{ \log p(\boldsymbol{\theta}; \mathbf{y}) - \sum_{i=1}^{M} \log q_i(\boldsymbol{\theta}_i) \Big\} d\boldsymbol{\theta}_1 \ldots d\boldsymbol{\theta}_M \\ &\propto \int q_1(\boldsymbol{\theta}_1) \Big\{ \int \big(\log p(\boldsymbol{\theta}; \mathbf{y}) q_2(\boldsymbol{\theta}_2) \ldots q_M(\boldsymbol{\theta}_M)\big) d\boldsymbol{\theta}_2 \ldots d\boldsymbol{\theta}_M \Big\} d\boldsymbol{\theta}_1 \\ &\quad - \int q_1(\boldsymbol{\theta}_1) \log q_1(\boldsymbol{\theta}_1) d\boldsymbol{\theta}_1.\end{aligned} \tag{1.5}$$

A new joint density $\tilde{p}(\boldsymbol{\theta}_1; \mathbf{y})$ can now be defined

$$\tilde{p}(\boldsymbol{\theta}_1; \mathbf{y}) = \frac{\exp \int \log p(\boldsymbol{\theta}; \mathbf{y}) q_2(\boldsymbol{\theta}_2) \ldots q_M(\boldsymbol{\theta}_M) d\boldsymbol{\theta}_2 \ldots d\boldsymbol{\theta}_M}{\iint \left\{ \exp \int \log p(\boldsymbol{\theta}; \mathbf{y}) q_2(\boldsymbol{\theta}_2) \ldots q_M(\boldsymbol{\theta}_M) d\boldsymbol{\theta}_2 \ldots d\boldsymbol{\theta}_M \right\} d\boldsymbol{\theta}_1 d\mathbf{y}}, \tag{1.6}$$

and the lower bound is then equal to

$$\log \underline{p}(\boldsymbol{\theta}; \mathbf{y}) = \int q_1(\boldsymbol{\theta}_1) \log \left\{ \frac{\tilde{p}(\boldsymbol{\theta}_1; \mathbf{y})}{q_1(\boldsymbol{\theta}_1)} \right\} d\boldsymbol{\theta}_1 + \text{terms not involving } q_1(\boldsymbol{\theta}_1). \tag{1.7}$$

From (1.4) it follows that

$$\begin{aligned} q_1^*(\boldsymbol{\theta}_1) &= \underset{q_1(\boldsymbol{\theta}_1) \in Q}{\operatorname{argmax}} \log \underline{p}(\mathbf{y}; q) = \tilde{p}(\boldsymbol{\theta}_1 | \mathbf{y}) = \frac{\tilde{p}(\boldsymbol{\theta}_1; \mathbf{y})}{\int \tilde{p}(\boldsymbol{\theta}_1; \mathbf{y}) d\boldsymbol{\theta}_1} \\ &\propto \exp \left\{ \int \log p(\boldsymbol{\theta}; \mathbf{y}) q_2(\boldsymbol{\theta}_2) \ldots q_M(\boldsymbol{\theta}_M) d\boldsymbol{\theta}_2 \ldots d\boldsymbol{\theta}_M \right\} \\ &= \exp \left\{ \mathbb{E}_{-\theta_1} \left[\log p(\boldsymbol{\theta}; \mathbf{y}) \right] \right\}, \end{aligned} \tag{1.8}$$

where $\mathbb{E}_{-\theta_1}$ represents the expected value over $\prod_{j \neq 1} q_j(\boldsymbol{\theta}_j)$. If we consider the same procedure for the maximization of $\log \underline{p}(\boldsymbol{\theta}; \mathbf{y})$ over each $q_2(\boldsymbol{\theta}_2), \ldots, q_M(\boldsymbol{\theta}_M)$, we get

Algorithm 1: CAVI for non-parametric MFVB.

```
Initialize: $q_1^*(\boldsymbol{\theta}_1),\ q_2^*(\boldsymbol{\theta}_2), \dots, q_M^*(\boldsymbol{\theta}_M)$
while increase in $\log \underline{p}(\mathbf{y}; q)$ is greater than $\varepsilon$ do
    for $i = 1, \dots, M$ do
        $q_i^*(\boldsymbol{\theta}_i) \leftarrow \frac{\exp\{\mathbb{E}_{-\theta_i}[\log p(\boldsymbol{\theta}; \mathbf{y})]\}}{\int \exp\{\mathbb{E}_{-\theta_i}[\log p(\boldsymbol{\theta}; \mathbf{y})]\}\, d\theta_i}$
    end
    compute $\log \underline{p}(\mathbf{y}; q)$;
end
```

the general solution

$$\begin{aligned} q_i^*(\boldsymbol{\theta}_i) &\propto \exp\left\{\mathbb{E}_{-\theta_i}\left[\log p(\boldsymbol{\theta}; \mathbf{y})\right] \right\} \\ &\propto \exp\left\{\mathbb{E}_{-\theta_i}\left[\log p(\boldsymbol{\theta}_i | \boldsymbol{\theta}_{-i}, \mathbf{y})\right] \right\}, \quad i = 1, ..., M. \end{aligned} \tag{1.9}$$

This result suggests the iterative procedure shown in Algorithm 1, also known as coordinate-ascent variational inference (CAVI) in the machine learning literature (Blei et al., 2017), which allows to obtain the optimal variational densities $q_i^*(\boldsymbol{\theta}_i)$.

From (1.9), we can see the strong connection between non-parametric mean field approximation and Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990). The densities $p(\boldsymbol{\theta}_i | \boldsymbol{\theta}_{-i}, \mathbf{y})$, $i = 1, \dots, M$, are called full conditionals distributions, which represent the key ingredient to implement the Gibbs sampler. Computing the non-parametric mean field variational Bayes (MFVB) solution requires only one more step, that is the computation of the expected value of the full conditionals in logarithmic scale.

It is important to pay attention about the partition $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M\}$ of $\boldsymbol{\theta}$, because different partitions lead to different inference, and a trade-off between tractability and accuracy of the approximation should be considered (Wand et al., 2011).

For example, suppose that the joint density $p(\boldsymbol{\theta}) = p(\theta_1, \theta_2, \theta_3)$ has to be approximated. The four possible factorizations for $q(\boldsymbol{\theta})$ are

$$q(\boldsymbol{\theta}) = q(\theta_1, \theta_2, \theta_3) = \begin{cases} q(\theta_1)q(\theta_2)q(\theta_3) \\ q(\theta_1, \theta_2)q(\theta_3) \\ q(\theta_1)q(\theta_2, \theta_3) \\ q(\theta_1, \theta_3)q(\theta_2). \end{cases} \tag{1.10}$$

If there is strong correlation between any of the parameters in $\boldsymbol{\theta}$, choosing the first factorization in (1.10), which is commonly known as naïve mean field approximation, will result in a weak approximation of the joint density.

1.2.1 Illustrative example: the multivariate Normal distribution

This section is devoted to show how MFVB works within a simple context, that is multivariate Normal distribution. Let:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{pmatrix} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,p} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & \cdots & y_{n,p} \end{pmatrix} \tag{1.11}$$

be a $n \times p$ matrix of observed values, where each row is a realization of a multivariate Gaussian distribution

$$\mathbf{Y}_i | \boldsymbol{\mu}, \boldsymbol{\Omega} \sim \mathsf{N}_p(\boldsymbol{\mu}, \boldsymbol{\Omega}^{-1}), \quad i = 1, ..., n, \quad \mathbf{Y}_i \perp \mathbf{Y}_j, \quad \forall i \neq j, \tag{1.12}$$

so that the likelihood for the model above is equal to

$$p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Omega}) = (2\pi)^{-pn/2} |\boldsymbol{\Omega}|^{n/2} \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} (\mathbf{y}_i - \boldsymbol{\mu})^\intercal \boldsymbol{\Omega} (\mathbf{y}_i - \boldsymbol{\mu}) \right\}. \tag{1.13}$$

The Bayesian paradigm requires the choice of the prior distribution for $\boldsymbol{\mu}$ and $\boldsymbol{\Omega}$. In particular, one possible choice is given by the conjugate distributions for both $\boldsymbol{\mu}$ and $\boldsymbol{\Omega}$:

$$\begin{aligned} \boldsymbol{\mu} &\sim \mathsf{N}_p(\boldsymbol{\mu}_0, \boldsymbol{\Omega}_0^{-1}), \\ \boldsymbol{\Omega} &\sim \mathsf{W}(\nu, \mathbf{V}), \end{aligned} \tag{1.14}$$

which leads to the joint distribution of the data and parameters:

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\Omega}; \mathbf{y}) = {} & (2\pi)^{-pn/2} |\boldsymbol{\Omega}|^{n/2} \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} (\mathbf{y}_i - \boldsymbol{\mu})^\intercal \boldsymbol{\Omega} (\mathbf{y}_i - \boldsymbol{\mu}) \right\} \\ & \times (2\pi)^{-p/2} |\boldsymbol{\Omega}_0|^{1/2} \exp\left\{ -\frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Omega}_0 (\boldsymbol{\mu} - \boldsymbol{\mu}_0) \right\} \\ & \times \frac{2^{-\nu p/2} |\mathbf{V}|^{-\nu/2}}{\Gamma_p(\nu/2)} |\boldsymbol{\Omega}|^{(\nu - p - 1)/2} \exp\left\{ -\frac{1}{2} \mathsf{tr}\left\{ \mathbf{V}^{-1} \boldsymbol{\Omega} \right\} \right\}. \end{aligned} \tag{1.15}$$

We now present the estimation of the model both via MFVB and MCMC in order to compare their performances.

Gibbs sampler approach. MCMC is the most famous approach to make Bayesian inference. An interesting case is the Gibbs sampler, which can be implemented when the full conditionals $p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{-i}, \mathbf{y})$ can be traced back to known distributions. Given the complete data likelihood, that is the joint distribution of the data and parameters $p(\mathbf{y}, \theta)$, it is possible to find the full conditonal for a given parameter simply applying Bayes' theorem.

In this example the full conditionals have a known distribution, as provided by the following propositions.

Proposition 1.1. *The full conditional distribution for* $\boldsymbol{\Omega}$ *is* $p(\boldsymbol{\Omega}|\boldsymbol{\mu}, \mathbf{y}) \sim \mathsf{W}(\nu^*, \mathbf{V}^*)$ *with*

$$\nu^* = \nu + n, \qquad \mathbf{V}^* = \left(\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})^\intercal + \mathbf{V}^{-1}\right)^{-1}. \tag{1.16}$$

Proof. Since the full conditional distribution for $\boldsymbol{\Omega}$ is $p(\boldsymbol{\Omega}|rest) = p(\boldsymbol{\Omega}|\boldsymbol{\mu}, \mathbf{y})$,

$$\begin{aligned} p(\boldsymbol{\Omega}|\boldsymbol{\mu}, \mathbf{y}) \propto |\boldsymbol{\Omega}|^{n/2} \exp&\left\{-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})^\intercal\boldsymbol{\Omega}(\mathbf{y}_i - \boldsymbol{\mu})\right\} \\ &\times |\boldsymbol{\Omega}|^{(\nu-p-1)/2} \exp\left\{ -\frac{1}{2}\mathsf{tr}\left\{\mathbf{V}^{-1}\boldsymbol{\Omega}\right\}\right\} \\ &= |\boldsymbol{\Omega}|^{(n+\nu-p-1)/2} \exp\left\{-\frac{1}{2}\mathsf{tr}\left\{\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})^\intercal\boldsymbol{\Omega}\right\}\right\} \\ &\times \exp\left\{ -\frac{1}{2}\mathsf{tr}\left\{\mathbf{V}^{-1}\boldsymbol{\Omega}\right\}\right\} \\ &= |\boldsymbol{\Omega}|^{(n+\nu-p-1)/2} \exp\left\{-\frac{1}{2}\mathsf{tr}\left\{\left[\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})^\intercal + \mathbf{V}^{-1}\right]\boldsymbol{\Omega}\right\}\right\}. \end{aligned}$$

The latter is the kernel of a Wishart distribution with parameters as defined in Proposition 1.1. □

Proposition 1.2. *The full conditional distribution for* $\boldsymbol{\mu}$ *is* $p(\boldsymbol{\mu}|\boldsymbol{\Omega}, \mathbf{y}) \sim \mathsf{N}_p(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ *with*

$$\boldsymbol{\mu}^* = (n\boldsymbol{\Omega} + \boldsymbol{\Omega}_0)^{-1}\left(\boldsymbol{\Omega}\sum_{i=1}^{n}\mathbf{y}_i + \boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right), \qquad \boldsymbol{\Sigma}^* = (n\boldsymbol{\Omega} + \boldsymbol{\Omega}_0)^{-1}. \tag{1.17}$$

Proof. Since the full conditional distribution for $\boldsymbol{\mu}$ is $p(\boldsymbol{\mu}|rest) = p(\boldsymbol{\mu}|\boldsymbol{\Omega}, \mathbf{y})$,

$$\begin{aligned} p(\boldsymbol{\mu}|\boldsymbol{\Omega}; \mathbf{y}) &\propto \exp\left\{ -\frac{1}{2}\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})^\intercal\boldsymbol{\Omega}(\mathbf{y}_i - \boldsymbol{\mu})\right\} - \exp\left\{\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\intercal\boldsymbol{\Omega}_0(\boldsymbol{\mu} - \boldsymbol{\mu}_0)\right\} \\ &\propto \exp\left\{ -\frac{1}{2}\left(n\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}\boldsymbol{\mu} - 2\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}\sum_{i=1}^{n}\mathbf{y}_i + \boldsymbol{\mu}^\intercal\boldsymbol{\Omega}_0\boldsymbol{\mu} - 2\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right)\right\} \\ &= \exp\left\{ -\frac{1}{2}\left(\boldsymbol{\mu}^\intercal(n\boldsymbol{\Omega} + \boldsymbol{\Omega}_0)\boldsymbol{\mu} - 2\boldsymbol{\mu}^\intercal\left(\boldsymbol{\Omega}\sum_{i=1}^{n}\mathbf{y}_i + \boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right)\right)\right\}. \end{aligned}$$

The latter is the kernel of a multivariate Gaussian distribution with parameters as defined in Proposition 1.2. □

The Gibbs sampling method consists in drawing values from each full conditional distribution sequentially until reaching a sample of a fixed size R for the parameters. Notice that, according to the value of R, it is possible to make the inference arbitrarially accurate at the cost of increasing the computational effort. A Gibbs sampler algorithm for the estimation and inference about $(\boldsymbol{\mu}, \boldsymbol{\Omega})$ is presented in Algorithm 2.

Algorithm 2: Gibbs sampling for multivariate Gaussian model.

```
Initialize: $\boldsymbol{\mu}^{*(0)}$, $\boldsymbol{\Omega}^{*(0)}$, $R$
Compute $\nu^* \leftarrow \nu + n$
while $r < R$ do
    Compute $\mathbf{V}^* \leftarrow \left( \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu}^{*(r-1)})(\mathbf{y}_i - \boldsymbol{\mu}^{*(r-1)})^\intercal + \mathbf{V}^{-1} \right)^{-1}$
    Sample $\boldsymbol{\Omega}^{*(r)} \sim \mathsf{W}(\nu^*, \mathbf{V}^*)$
    Compute $\boldsymbol{\Sigma}^* \leftarrow (n\boldsymbol{\Omega}^{*(r)} + \boldsymbol{\Omega}_0)^{-1}$
    Compute $\boldsymbol{\mu}^* \leftarrow \boldsymbol{\Sigma}^* \left( \boldsymbol{\Omega}^{*(r)} \sum_{i=1}^n \mathbf{y}_i + \boldsymbol{\Omega}_0 \boldsymbol{\mu}_0 \right)$
    Sample $\boldsymbol{\mu}^{*(r)} \sim \mathsf{N}_p(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$
end
```

Mean field variational Bayes approach. In order to compute the optimal variational densities, a factorization must be defined. A tractable solution arises for the two component product:

$$q(\boldsymbol{\mu}, \boldsymbol{\Omega}) = q(\boldsymbol{\mu}) q(\boldsymbol{\Omega}). \tag{1.18}$$

Following the mean field approximation paradigm in (1.9), the optimal densities are provided by the next two propositions.

Proposition 1.3. *The optimal density for* $\boldsymbol{\Omega}$ *is* $q^*(\boldsymbol{\Omega}) \sim \mathsf{W}(\nu_{q(\Omega)}, \mathbf{V}_{q(\Omega)})$ *with*

$$\nu_{q(\Omega)} = \nu + n, \qquad \mathbf{V}_{q(\Omega)} = \left(\sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu}_{q(\mu)})(\mathbf{y}_i - \boldsymbol{\mu}_{q(\mu)})^\intercal + n\boldsymbol{\Sigma}_{q(\mu)} + \mathbf{V}^{-1} \right)^{-1}. \tag{1.19}$$

Furthermore, $\boldsymbol{\mu}_{q(\Omega)} = \nu_{q(\Omega)} \mathbf{V}_{q(\Omega)}$.

Proof. Since $q^*(\boldsymbol{\Omega}) \propto \exp\Big\{ \mathbb{E}_{-\boldsymbol{\Omega}} \left[\log p\left(\boldsymbol{\Omega} | \boldsymbol{\mu}; \mathbf{y}\right) \right] \Big\}$,

$$\log q^*(\boldsymbol{\Omega}) \propto \mathbb{E}_{-\boldsymbol{\Omega}} \left[\frac{n}{2} \log |\boldsymbol{\Omega}| - \frac{1}{2} \mathsf{tr} \left\{ \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})^\intercal \boldsymbol{\Omega} \right\} \right]$$

$$
\begin{aligned}
&+\mathbb{E}_{-\boldsymbol{\Omega}}\left[\frac{\nu-p-1}{2}\log|\boldsymbol{\Omega}|-\frac{1}{2}\mathsf{tr}\left\{\mathbf{V}^{-1}\boldsymbol{\Omega}\right\}\right]\\
&=\frac{n}{2}\log|\boldsymbol{\Omega}|-\frac{1}{2}\mathsf{tr}\left\{\left(\sum_{i=1}^{n}(\mathbf{y}_i-\boldsymbol{\mu}_{q(\mu)})(\mathbf{y}_i-\boldsymbol{\mu}_{q(\mu)})^\intercal+n\boldsymbol{\Sigma}_{q(\mu)}\right)\boldsymbol{\Omega}\right\}\\
&+\frac{\nu-p-1}{2}\log|\boldsymbol{\Omega}|-\frac{1}{2}\mathsf{tr}\left\{\mathbf{V}^{-1}\boldsymbol{\Omega}\right\}\\
&=\frac{(\nu+n)-p-1}{2}\log|\boldsymbol{\Omega}|\\
&-\frac{1}{2}\mathsf{tr}\left\{\left(\sum_{i=1}^{n}(\mathbf{y}_i-\boldsymbol{\mu}_{q(\mu)})(\mathbf{y}_i-\boldsymbol{\mu}_{q(\mu)})^\intercal+n\boldsymbol{\Sigma}_{q(\mu)}+\mathbf{V}^{-1}\right)\boldsymbol{\Omega}\right\}.
\end{aligned}
$$

Take the exponential and notice that it coincides with the kernel of a Wishart distribution with parameters as in Proposition 1.3. □

Proposition 1.4. *The optimal density for* $\boldsymbol{\mu}$ *is* $q^*(\boldsymbol{\mu})\sim\mathsf{N}_p(\boldsymbol{\mu}_{q(\mu)},\boldsymbol{\Sigma}_{q(\mu)})$ *with*

$$
\boldsymbol{\mu}_{q(\mu)}=(n\boldsymbol{\mu}_{q(\Omega)}+\boldsymbol{\Omega}_0)^{-1}\left(\boldsymbol{\mu}_{q(\Omega)}\sum_{i=1}^{n}\mathbf{y}_i+\boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right),\qquad\boldsymbol{\Sigma}_{q(\mu)}=(n\boldsymbol{\mu}_{q(\Omega)}+\boldsymbol{\Omega}_0)^{-1}.\tag{1.20}
$$

Proof. Since $q^*(\boldsymbol{\mu})\propto\exp\left\{\mathbb{E}_{-\boldsymbol{\mu}}\left[\log p\left(\boldsymbol{\mu}|\boldsymbol{\Omega};\mathbf{y}\right)\right]\right\}$,

$$
\begin{aligned}
\log q^*(\boldsymbol{\mu})&\propto\mathbb{E}_{-\boldsymbol{\mu}}\left[-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{y}_i-\boldsymbol{\mu})^\intercal\boldsymbol{\Omega}(\mathbf{y}_i-\boldsymbol{\mu})-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\intercal\boldsymbol{\Omega}_0(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\right]\\
&\propto\mathbb{E}_{-\boldsymbol{\mu}}\left[-\frac{1}{2}\left(n\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}\boldsymbol{\mu}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}\sum_{i=1}^{n}\mathbf{y}_i+\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}_0\boldsymbol{\mu}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right)\right]\\
&=-\frac{1}{2}\left(n\boldsymbol{\mu}^\intercal\boldsymbol{\mu}_{q(\Omega)}\boldsymbol{\mu}-2\boldsymbol{\mu}^\intercal\boldsymbol{\mu}_{q(\Omega)}\sum_{i=1}^{n}\mathbf{y}_i+\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}_0\boldsymbol{\mu}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right)\\
&=-\frac{1}{2}\left(\boldsymbol{\mu}^\intercal(n\boldsymbol{\mu}_{q(\Omega)}+\boldsymbol{\Omega}_0)\boldsymbol{\mu}-2\boldsymbol{\mu}^\intercal\left(\boldsymbol{\mu}_{q(\Omega)}\sum_{i=1}^{n}\mathbf{y}_i+\boldsymbol{\Omega}_0\boldsymbol{\mu}_0\right)\right).
\end{aligned}
$$

Take the exponential and notice that it coincides with the kernel of a multivariate Gaussian distribution with parameters as in Proposition 1.4. □

The last step for the derivation of the MFVB algorithm requires the computation of the ELBO, which can be expressed in a closed form and it is provided in the next proposition.

Proposition 1.5. *The lower bound* $\log\underline{p}(\mathbf{y};q)$ *for the multivariate Gaussian model in (1.12) and (1.14), and associated to the variational density factorized as* $q(\boldsymbol{\mu},\boldsymbol{\Omega})=$

$q(\boldsymbol{\mu})q(\boldsymbol{\Omega})$, given a matrix of data $\mathbf{Y}$, can be expressed in a closed form:

$$\begin{aligned}\log \underline{p}(\mathbf{y}; q) &= \iint q(\boldsymbol{\mu}, \boldsymbol{\Omega}) \log \frac{p(\boldsymbol{\mu}, \boldsymbol{\Omega}; \mathbf{y})}{q(\boldsymbol{\mu}, \boldsymbol{\Omega})} \, d\boldsymbol{\mu} \, d\boldsymbol{\Omega} \\ &= \mathbb{E}_q(\log p(\boldsymbol{\mu}, \boldsymbol{\Omega}; \mathbf{y})) - \mathbb{E}_q(\log q(\boldsymbol{\mu}, \boldsymbol{\Omega})) \\ &= -\frac{np}{2}\log 2\pi - \frac{1}{2}(\boldsymbol{\mu}_{q(\mu)} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Omega}_0 (\boldsymbol{\mu}_{q(\mu)} - \boldsymbol{\mu}_0) + \frac{1}{2}\log|\boldsymbol{\Omega}_0| \qquad (1.21) \\ &+ \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\mu)}| - \frac{1}{2}\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\mu)}\boldsymbol{\Omega}_0\right\} + \frac{p}{2} - \frac{\nu p}{2}\log 2 - \frac{\nu}{2}\log|\mathbf{V}| \\ &- \log\Gamma_p(\nu/2) + \frac{\nu_{q(\Omega)}p}{2}\log 2 + \frac{\nu_{q(\Omega)}}{2}\log|\mathbf{V}_{q(\Omega)}| + \log\Gamma_p(\nu_{q(\Omega)}/2).\end{aligned}$$

Proof. The first term is

$$\begin{aligned}\mathbb{E}_q(\log p(\boldsymbol{\mu}, \boldsymbol{\Omega}; \mathbf{y})) &= -\frac{np}{2}\log 2\pi + \frac{n}{2}\boldsymbol{\mu}_{q(\log|\Omega|)} \\ &- \frac{1}{2}\mathsf{tr}\left\{ \left(\sum_{i=1}^{n} (\mathbf{y}_i - \boldsymbol{\mu}_{q(\mu)})(\mathbf{y}_i - \boldsymbol{\mu}_{q(\mu)})^\intercal + n\boldsymbol{\Sigma}_{q(\mu)} \right) \boldsymbol{\mu}_{q(\Omega)} \right\} \\ &- \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Omega}_0| - \frac{1}{2}(\boldsymbol{\mu}_{q(\mu)} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Omega}_0 (\boldsymbol{\mu}_{q(\mu)} - \boldsymbol{\mu}_0) \\ &- \frac{1}{2}\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\mu)}\boldsymbol{\Omega}_0\right\} - \frac{\nu p}{2}\log 2 - \frac{\nu}{2}\log|\mathbf{V}| - \log\Gamma_p(\nu/2) \\ &+ \frac{\nu - p - 1}{2}\boldsymbol{\mu}_{q(\log|\Omega|)} - \frac{1}{2}\mathsf{tr}\left\{\mathbf{V}^{-1}\boldsymbol{\mu}_{q(\Omega)}\right\} \\ &= -\frac{np}{2}\log 2\pi + \frac{\nu_{q(\Omega)} - p - 1}{2}\boldsymbol{\mu}_{q(\log|\Omega|)} - \frac{1}{2}\mathsf{tr}\left\{\mathbf{V}_{q(\Omega)}^{-1}\boldsymbol{\mu}_{q(\Omega)}\right\} \\ &- \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Omega}_0| \\ &- \frac{1}{2}(\boldsymbol{\mu}_{q(\mu)} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Omega}_0 (\boldsymbol{\mu}_{q(\mu)} - \boldsymbol{\mu}_0) - \frac{1}{2}\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\mu)}\boldsymbol{\Omega}_0\right\} \\ &- \frac{\nu p}{2}\log 2 - \frac{\nu}{2}\log|\mathbf{V}| - \log\Gamma_p(\nu/2),\end{aligned}$$

while the second term is

$$\begin{aligned}\mathbb{E}_q(\log q(\boldsymbol{\mu}, \boldsymbol{\Omega})) &= -\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\mu)}| - \frac{p}{2} - \frac{\nu_{q(\Omega)}p}{2}\log 2 - \frac{\nu_{q(\Omega)}}{2}\log|\mathbf{V}_{q(\Omega)}| \\ &- \log\Gamma_p(\nu_{q(\Omega)}/2) + \frac{\nu_{q(\Omega)} - p - 1}{2}\boldsymbol{\mu}_{q(\log|\Omega|)} - \frac{1}{2}\mathsf{tr}\left\{\mathbf{V}_{q(\Omega)}^{-1}\boldsymbol{\mu}_{q(\Omega)}\right\}.\end{aligned}$$

After some simplification we obtain the result in Proposition 1.5. □

In order to prove Proposition 1.5 we used the following result, which will be useful also later.

Result 1.1. *Let $\mathbf{x}$ be a p-dimensional Gaussian random vector with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{x} \sim \mathsf{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The expectation of the quadratic form $(\mathbf{x} - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ is equal to p.*

Proof.

$$\begin{aligned}\mathbb{E}_q\left[(\mathbf{x}-\boldsymbol{\mu})^\intercal\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] &= \mathbb{E}_q\left[\mathbf{x}^\intercal\boldsymbol{\Sigma}^{-1}\mathbf{x}\right]+\boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-2\mathbb{E}_q\left[\mathbf{x}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right]\\ &= \boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\mathsf{tr}\left\{\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\right\}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\\ &= 2\boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\mathsf{tr}\left\{\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\right\}-2\boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\\ &= \mathsf{tr}\left\{\mathbf{I}_p\right\}=p.\end{aligned}$$

□

In order to make approximate Bayesian inference on the multivariate Gaussian model with a non-parametric mean field variational Bayes approach, we can implement the iterative Algorithm 3.

Algorithm 3: MFVB for multivariate Gaussian model.

```
Initialize: $q^*(\boldsymbol{\mu})$, $q^*(\boldsymbol{\Omega})$, $\varepsilon$
$\nu_{q(\Omega)} \leftarrow \nu + n$
while convergence not reached do
    $\boldsymbol{\Sigma}_{q(\mu)} \leftarrow (n\boldsymbol{\mu}_{q(\Omega)} + \boldsymbol{\Omega}_0)^{-1}$
    $\boldsymbol{\mu}_{q(\mu)} \leftarrow \boldsymbol{\Sigma}_{q(\mu)}(\boldsymbol{\mu}_{q(\Omega)}\sum_{i=1}^n \mathbf{y}_i + \boldsymbol{\Omega}_0\boldsymbol{\mu}_0)$
    $\mathbf{V}_{q(\Omega)} \leftarrow \left(\sum_{i=1}^n(\mathbf{y}_i - \boldsymbol{\mu}_{q(\mu)})(\mathbf{y}_i - \boldsymbol{\mu}_{q(\mu)})^\intercal + n\boldsymbol{\Sigma}_{q(\mu)} + \mathbf{V}^{-1}\right)^{-1}$
    $\boldsymbol{\mu}_{q(\Omega)} \leftarrow \nu_{q(\Omega)}\mathbf{V}_{q(\Omega)}$
    compute $\log \underline{p}(\mathbf{y}; q)^{(iter)}$;
    evaluate $|\log \underline{p}(\mathbf{y}; q)^{(iter)} - \log \underline{p}(\mathbf{y}; q)^{(iter-1)}| < \varepsilon$;
end
```

Application to simulated dataset. We simulate $n = 300$ observations from a 3-dimensional Normal distribution with mean vector $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{\Omega}$ defined as

$$\boldsymbol{\mu} = \begin{pmatrix}0\\0\\0\end{pmatrix}, \quad \boldsymbol{\Omega} = \begin{pmatrix}2.67 & -1.33 & -2\\ -1.33 & 1.17 & 1\\ -2 & 1 & 2\end{pmatrix}. \tag{1.22}$$

As concerns the hyperparameters' setting, we fix $\boldsymbol{\mu}_0 = \mathbf{0}_p$, $\boldsymbol{\Omega}_0 = 0.01\mathbf{I}_p$, $\nu = p+1$, $\mathbf{V} = (p+1)\mathbf{I}_p$. Thus we have:

$$\begin{aligned}\boldsymbol{\mu} &\sim \mathsf{N}_p(\mathbf{0}_p, 100\mathbf{I}_p),\\ \boldsymbol{\Omega} &\sim \mathsf{W}(p+1, (p+1)\mathbf{I}_p).\end{aligned} \tag{1.23}$$

Algorithm 3 converged after 4 iterations (with increase in lower bound less than 1e-5) and we get the following point estimates with MFVB approach

$$\boldsymbol{\mu}_{q(\mu)} = \begin{pmatrix} -0.02 \\ -0.01 \\ -0.02 \end{pmatrix}, \qquad \boldsymbol{\mu}_{q(\Omega)} = \begin{pmatrix} 2.536 & -1.322 & -1.912 \\ -1.322 & 1.256 & 1.006 \\ -1.912 & 1.006 & 1.981 \end{pmatrix}, \tag{1.24}$$

showing their closeness to the true parameters.

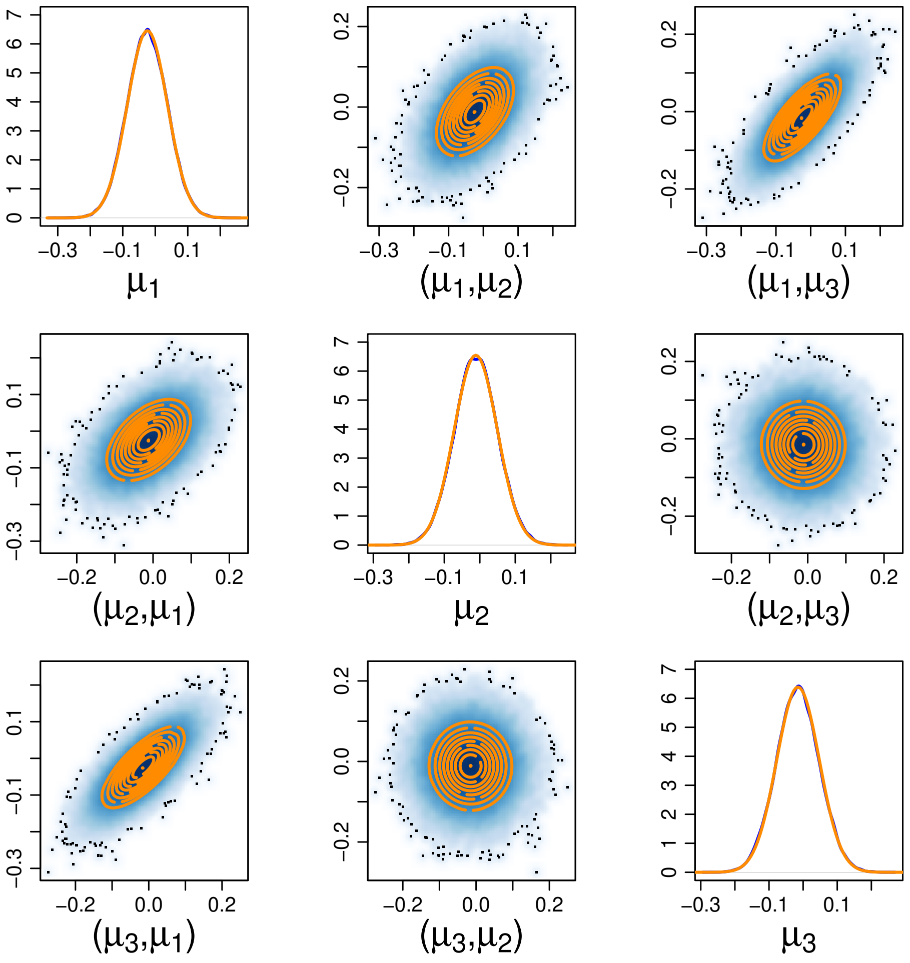


Figure 1.1: Marginal and bivariate posterior distributions for $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$ estimated both via MFVB (orange) and MCMC (blue).

Furthermore, a comparison of the posterior distribution obtained through MFVB and MCMC is carried out. In the MCMC, we ran five parallel chains (with $R = 50000$ values and a burn-in of 25000 in each of them) in order to assess that the convergence leads to satisfying diagnostics results, e.g. the multivariate scale reduction factor (Gelman and Rubin, 1992) equal to 1. Figure 1.1 shows estimated densities of $\boldsymbol{\mu}$, while Figure 1.2 shows estimated densities relative to $\boldsymbol{\Omega}$: in both cases the differences between the two methods are negligibles.

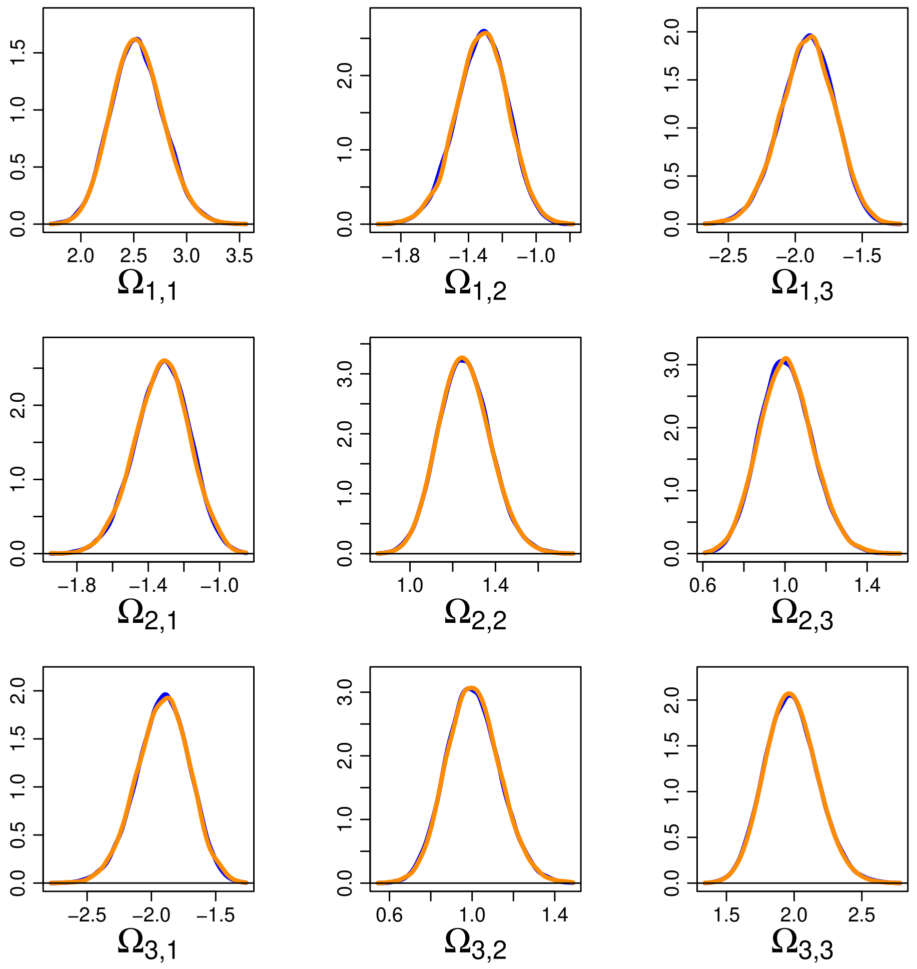


Figure 1.2: Marginal posterior distributions for the elements of $\mathbf{\Omega}$ estimated both via MFVB (orange) and MCMC (blue).

After that the similar results between MFVB and MCMC in terms of posterior inference accuracy have been tested, we are interested in assessing the effective advantage of MFVB with respect to MCMC in terms of computational cost. In particular, we inspect the running time of the two algorithms in two circumstances: the first concerns the increase of the number of observations with a fixed number of variables ($p = 3$); the second considers the progressive increase of the number of variables with a fixed number of observations ($n = 300$). As concerns the priors, we consider the same as in (1.23). Figure 1.3 depicts the results. We notice an higher speed of the MFVB with respect to MCMC approach in both circumstances. In particular, the computational cost of MFVB algorithm is constant and close to 0 both increasing the number of observations and variables. On the other hand, MCMC shows a linear growth in the first case and an exponential growth in the second scenario.

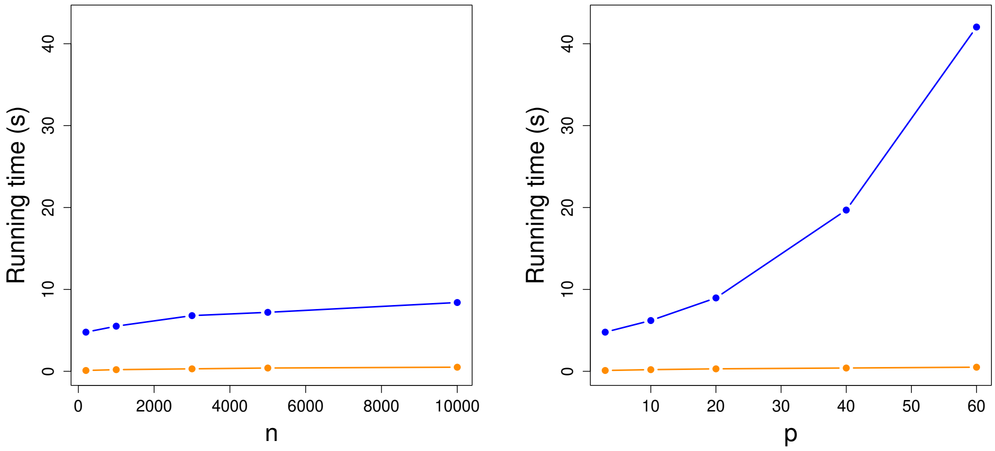


Figure 1.3: MFVB (orange) and MCMC (blue) running time in multivariate Gaussian model for different number of observations with $p = 3$ (left) and for different number of variables with $n = 300$ (right).

1.3 Semi-parametric mean field approximation

An alternative to non-parametric mean field approximation for Bayesian inference is the semi-parametric approach. In this paradigm some density functions in the factorization are pre-specified to belong to a family of convenient parametric densities (Rohde and Wand, 2016). In order to have a better comprehension of the semi-parametric MFVB, suppose we have the data $\mathbf{y}$, the parameter $\boldsymbol{\theta}$, and a further d-dimensional parameter $\boldsymbol{\phi} \in \boldsymbol{\Phi} \subseteq \mathbb{R}^d$. Semi-parametric MFVB proceeds in two steps. First, it requires to define a mean field product restriction of the variational density $q(\boldsymbol{\theta}, \boldsymbol{\phi})$, for example:

$$q(\boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{i=1}^{M} q(\boldsymbol{\theta}_i) q(\boldsymbol{\phi}). \tag{1.25}$$

The second step concerns the parametric approximation. Assume that $q(\boldsymbol{\phi})$ belongs to a parametric family of density functions with parameter vector $\boldsymbol{\xi} \in \boldsymbol{\Xi} \subseteq \mathbb{R}^z$. The restriction in (1.25) becomes

$$q(\boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{i=1}^{M} q(\boldsymbol{\theta}_i) q(\boldsymbol{\phi}; \boldsymbol{\xi}). \tag{1.26}$$

In this framework, a useful graphical representation that provides a way to represent a joint probability distribution with a specific choice of the mean field restriction is given by the factor graph. Suppose we have the joint density $p(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{y}) =$

$p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_M, \boldsymbol{\phi}; \mathbf{y})$ that is factorized in N factor nodes, p_j, $j = 1, \dots, N$, with N based on the Bayesian model specification. For example, we can have $N = M$ and the following factorization

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{y}) = p_1(\boldsymbol{\theta}_1) p_2(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) p_3(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \dots p_N(\boldsymbol{\theta}_{M-1}, \boldsymbol{\theta}_M, \boldsymbol{\phi}), \tag{1.27}$$

with some of these factors that depend on the data vector $\mathbf{y}$. Assuming $N = M = 4$, Figure 1.4 shows the factor graph for the model (1.27) with mean field product restriction (1.26).

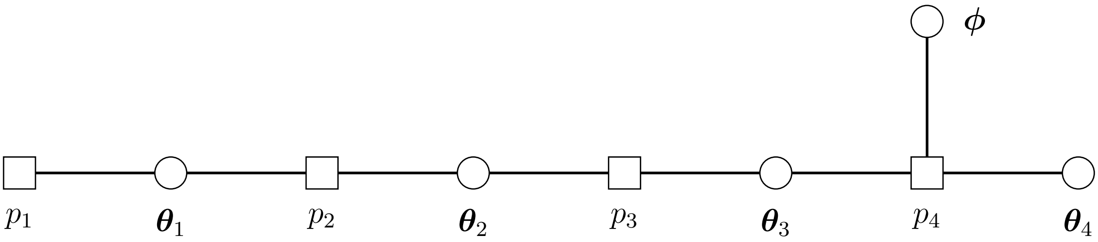


Figure 1.4: Factor graph corresponding to the model (1.27) and density product restriction (1.26).

In addition, considering the general case of joint density $p(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{y})$ with N factors and semi-parametric mean field restriction (1.26), the lower bound can be written in terms of the components of the corresponding factor graph:

$$\begin{aligned} \log \underline{p}(\mathbf{y}; q) &= \int q(\boldsymbol{\theta}, \boldsymbol{\phi}) \left\{ \log p(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{y}) - \sum_{i=1}^{M} \log q_i(\boldsymbol{\theta}_i) - \log q(\boldsymbol{\phi}; \boldsymbol{\xi}) \right\} d\boldsymbol{\theta}_1 \dots d\boldsymbol{\theta}_M d\boldsymbol{\phi} \\ &= \sum_{j=1}^{N} \mathbb{E}_q \left\{ \log p_j \right\} + \sum_{i=1}^{M} \mathbb{E}_q \left[- \log q_i(\boldsymbol{\theta}_i) \right] + \mathbb{E}_q \left[- \log q(\boldsymbol{\phi}; \boldsymbol{\xi}) \right] \\ &= \sum_{j=1}^{N} \mathbb{E}_q \left\{ \log p_j \right\} + \sum_{i=1}^{M} \mathrm{H} \left\{ q_i(\boldsymbol{\theta}_i) \right\} + \mathrm{H} \left\{ q(\boldsymbol{\phi}; \boldsymbol{\xi}) \right\}, \end{aligned} \tag{1.28}$$

where if X is a random variable with density function $p(x)$, then the corresponding entropy is given by

$$\mathrm{H}\{p(x)\} = \mathbb{E}_p \big[- \log p(x) \big]. \tag{1.29}$$

As concerns the maximization of (1.28) over each $q(\boldsymbol{\theta}_1), \dots, q(\boldsymbol{\theta}_M), q(\boldsymbol{\phi}; \boldsymbol{\xi})$, the optimal variational densities $q^*(\boldsymbol{\theta}_1), \dots, q^*(\boldsymbol{\theta}_M)$ are still given by the general solution in (1.9), while for $q^*(\boldsymbol{\phi}; \boldsymbol{\xi})$ is prudent to maximize the $\boldsymbol{\phi}$-localized component of $\log \underline{p}(\mathbf{y}; q)$, which we denote by $\log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\phi}]}$, over $\boldsymbol{\xi}$ (Rohde and Wand, 2016). The last quantity is defined as

$$\begin{aligned}\log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\phi}]} &= \mathrm{H}\left\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\right\} + NonEntropy\left\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\right\} \\ &= \mathrm{H}\left\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\right\} + \overline{\mathrm{H}}\left\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\right\} \\ &= \mathrm{H}\left\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\right\} + \sum_{j \in \text{neighbors}(\boldsymbol{\phi})} \mathbb{E}_q\left[\log p_j\right],\end{aligned} \tag{1.30}$$

where

$$\begin{aligned}\text{neighbors}(\boldsymbol{\phi}) = &\ \{1 \leq j \leq N : p_j \text{ is a neighbor of } \boldsymbol{\phi} \text{ on the factor graph }\} \\ = &\ \{1 \leq j \leq N : p_j \text{ involves } \boldsymbol{\phi}\}.\end{aligned}$$

For example, in the factor graph showed in Figure 1.4, neighbors($\boldsymbol{\phi}$) = {4}, thus we have:

$$\log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\phi}]} = \mathrm{H}\left\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\right\} + \mathbb{E}_q\big[\log p_4\big].$$

The iterative procedure that generalizes the coordinate-ascent algorithm (Algorithm 1) in the case of semi-parametric mean field approximation is shown in Algorithm 4.

Algorithm 4: CAVI for semi-parametric MFVB.

```
Initialize: $q^*(\boldsymbol{\theta}_1)$, $q^*(\boldsymbol{\theta}_2), \dots, q^*(\boldsymbol{\theta}_M), \boldsymbol{\xi}^*$
while increase in $\log \underline{p}(\mathbf{y}; q)$ is greater than $\varepsilon$ do
    for $i = 1, \dots, M$ do
        $q^*(\boldsymbol{\theta}_i) \leftarrow \frac{\exp\{\mathbb{E}_{-\theta_i}[\log p(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{y})]\}}{\int \exp\{\mathbb{E}_{-\theta_i}[\log p(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{y})]\}\, d\theta_i}$
    end
    while convergence not reached do
        $\boldsymbol{\xi}^* \leftarrow \underset{\boldsymbol{\xi}}{\operatorname{argmax}} \log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\phi}]}$
    end
    compute $\log \underline{p}(\mathbf{y}; q)$;
end
```

1.3.1 Illustrative example: the Poisson regression model

The goal of this section is to show how semi-parametric MFVB works with an example, that is the Poisson regression model. Let:

$$Y_i | \boldsymbol{\beta} \sim \mathsf{Poi}(\exp\{\mathbf{x}_i^\intercal \boldsymbol{\beta}\}), \quad i = 1, ..., n, \quad Y_i \perp Y_j \ \forall i \neq j, \tag{1.31}$$

where $\mathbf{x}_i$ is a p-dimensional set of known covariates, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^\intercal$ a p-dimensional vector of coefficients and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\intercal$ a n-dimensional vector where each element is a realization of a Poisson random variable.

The likelihood for the model above is equal to

$$p(\mathbf{y}|\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{e^{-\exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\beta}\right\}} e^{\mathbf{x}_i^\intercal \boldsymbol{\beta} y_i}}{y_i!} = \prod_{i=1}^{n} \frac{e^{\mathbf{x}_i^\intercal \boldsymbol{\beta} y_i - \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\beta}\right\}}}{y_i!}. \tag{1.32}$$

In order to make Bayesian inference, we have to choose a prior distribution for $\boldsymbol{\beta}$. In this example, we choose a hierarchical specification. We assume, conditionally to σ^2, a multivariate Normal distribution with mean equal to $\mathbf{0}_p$ and covariance matrix equal to $\sigma^2 \mathbf{I}_p$ for $\boldsymbol{\beta}$, and an Inverse-Gamma distribution for σ^2:

$$\boldsymbol{\beta}|\sigma^2 \sim \mathsf{N}_p(\mathbf{0}_p, \sigma^2 \mathbf{I}_p), \quad \sigma^2 \sim \mathsf{InvGa}(\alpha, \delta). \tag{1.33}$$

This specification leads to the following joint posterior distribution of the data and parameters:

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2; \mathbf{y}) = & \frac{\delta^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-(\alpha+1)} \exp\left\{-\frac{\delta}{\sigma^2}\right\} \\ & \times \frac{1}{(2\pi)^{\frac{p}{2}}} (\sigma^2)^{-\frac{p}{2}} \exp\left\{-\frac{1}{2\sigma^2} \boldsymbol{\beta}^\intercal \boldsymbol{\beta}\right\} \\ & \times \prod_{i=1}^{n} \frac{e^{\mathbf{x}_i^\intercal \boldsymbol{\beta} y_i - \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\beta}\right\}}}{y_i!}. \end{aligned} \tag{1.34}$$

At this point, as we did in the previous section, we fit the model both via MFVB and MCMC in order to compare their performances.

Markov chain Monte Carlo approach. In the case of MCMC approach, in order to be able to implement the Gibbs sampler, the full conditionals should have a known distribution. Unfortunately, in this example only the full conditional for σ^2, $p(\sigma^2|\boldsymbol{\beta}, \mathbf{y})$, is known, as shown by the following two propositions.

Proposition 1.6. *The full conditional distribution for σ^2 is $p(\sigma^2|\boldsymbol{\beta}, \mathbf{y}) \sim \mathsf{InvGa}\left(\alpha^*, \delta^*\right)$, with*

$$\alpha^* = \alpha + \frac{p}{2}, \qquad \delta^* = \delta + \frac{\boldsymbol{\beta}^\intercal \boldsymbol{\beta}}{2}. \tag{1.35}$$

Proof. Since the full conditional distribution for σ^2 is $p(\sigma^2|rest) = p(\sigma^2|\boldsymbol{\beta}, \mathbf{y})$,

$$\begin{aligned} p(\sigma^2|\boldsymbol{\beta}, \mathbf{y}) & \propto (\sigma^2)^{-(\alpha+1)} \exp\left\{-\frac{\delta}{\sigma^2}\right\} (\sigma^2)^{-\frac{p}{2}} \exp\left\{-\frac{1}{2\sigma^2} \boldsymbol{\beta}^\intercal \boldsymbol{\beta}\right\} \\ & = (\sigma^2)^{-(\alpha+\frac{p}{2}+1)} \exp\left\{-\frac{1}{\sigma^2}\left[\delta + \frac{\boldsymbol{\beta}^\intercal \boldsymbol{\beta}}{2}\right]\right\}. \end{aligned}$$

The latter is the kernel of an Inverse-Gamma distribution with parameters as defined in Proposition 1.6. □

Proposition 1.7. *The full conditional distribution for $\boldsymbol{\beta}$ is not known.*

Proof. Since the full conditional distribution for $\boldsymbol{\beta}$ is $p(\boldsymbol{\beta}|rest) = p(\boldsymbol{\beta}|\sigma^2, \mathbf{y})$,

$$\begin{aligned} p(\boldsymbol{\beta}|\sigma^2, \mathbf{y}) &\propto \exp\left\{-\frac{\boldsymbol{\beta}^\intercal\boldsymbol{\beta}}{2\sigma^2}\right\}\prod_{i=1}^{n}\exp\left\{\mathbf{x}_i^\intercal\boldsymbol{\beta}y_i - \exp\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\}\right\} \\ &= \exp\left\{-\frac{\boldsymbol{\beta}^\intercal\boldsymbol{\beta}}{2\sigma^2} + \sum_{i=1}^{n}\left(\mathbf{x}_i^\intercal\boldsymbol{\beta}y_i - \exp\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\}\right)\right\}, \end{aligned}$$

and we do not recognise the kernel of any known distribution. □

Because the full conditional distribution for $\boldsymbol{\beta}$, $p(\boldsymbol{\beta}|\sigma^2, \mathbf{y})$, has not explicit form, it is not possible to use the Gibbs sampler. One of the possible alternatives is the implementation of an hybrid MCMC, where we sample σ^2 from its full conditional and $\boldsymbol{\beta}$ with a Normal random walk Metropolis-Hastings step, until reaching a sample of size R. This procedure is described in Algorithm 5.

Algorithm 5: Hybrid MCMC for Poisson distribution with prior for the variance of $\boldsymbol{\beta}$.

```
Initialize: $\boldsymbol{\beta}^{*(0)}$, $(\sigma^2)^{*(0)}$, R
Compute $\alpha^* \leftarrow \alpha + \frac{p}{2}$
while $r < R$ do
    Compute $\delta^* \leftarrow \delta + \frac{\boldsymbol{\beta}^{*(r-1)\intercal}\boldsymbol{\beta}^{*(r-1)}}{2}$
    Sample $(\sigma^2)^{*(r)} \sim \mathsf{InvGa}(\alpha^*, \delta^*)$
    Sample $\boldsymbol{\beta}^{*(r)}$ with Normal Random Walk Metropolis-Hastings step
end
```

Mean field variational Bayes approach. In the case of MFVB approach, the first step concerns the choice of the factorization for the approximation of the posterior density $p(\boldsymbol{\beta}, \sigma^2|\mathbf{y})$. In this case a tractable solution is the following:

$$q(\boldsymbol{\beta}, \sigma^2) = q(\boldsymbol{\beta})q(\sigma^2). \tag{1.36}$$

The next step of the mean field variational paradigm is the search of the optimal variational densities, which are provided by the next two propositions.

Proposition 1.8. *The optimal density for σ^2 is $q^*(\sigma^2) \sim \mathsf{InvGa}\left(\alpha_{q(\sigma^2)}, \delta_{q(\sigma^2)}\right)$ with*

$$\alpha_{q(\sigma^2)} = \alpha + \frac{p}{2}, \qquad \delta_{q(\sigma^2)} = \delta + \frac{1}{2}\bigg(\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\beta)}\right\} + \boldsymbol{\mu}_{q(\beta)}^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}\bigg). \tag{1.37}$$

Furthermore, $\mu_{q(1/\sigma^2)} = \alpha_{q(\sigma^2)}/\delta_{q(\sigma^2)}$.

Proof. Since $q^*(\sigma^2) \propto \exp\bigg\{\mathbb{E}_{-\sigma^2}\bigg[\log p(\boldsymbol{\beta}, \sigma^2, \mathbf{y})\bigg]\bigg\} = \exp\bigg\{\mathbb{E}_{\boldsymbol{\beta}}\bigg[\log p(\boldsymbol{\beta}, \sigma^2, \mathbf{y})\bigg]\bigg\}$,

$$\begin{aligned}\log q^*(\sigma^2) &\propto \mathbb{E}_{\boldsymbol{\beta}}\bigg[-(\alpha+1)\log\sigma^2 - \frac{\delta}{\sigma^2} - \frac{p}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta}\bigg] \\ &= -(\alpha + \frac{p}{2} + 1)\log\sigma^2 - \frac{1}{\sigma^2}\bigg[\delta + \frac{1}{2}\bigg(\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\beta)}\right\} + \boldsymbol{\mu}_{q(\beta)}^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}\bigg)\bigg].\end{aligned}$$

Take the exponential and notice that it coincides with the kernel of an Inverse-Gamma distribution with parameters as in Proposition 1.8. □

Proposition 1.9. *The optimal density for $\boldsymbol{\beta}$, $q^*(\boldsymbol{\beta})$, is not a standard form.*

Proof. Since $q^*(\boldsymbol{\beta}) \propto \exp\bigg\{\mathbb{E}_{-\boldsymbol{\beta}}\bigg[\log p(\boldsymbol{\beta}, \sigma^2; \mathbf{y})\bigg]\bigg\} = \exp\bigg\{\mathbb{E}_{\sigma^2}\bigg[\log p(\boldsymbol{\beta}, \sigma^2; \mathbf{y})\bigg]\bigg\}$,

$$\begin{aligned}\log q^*(\boldsymbol{\beta}) &\propto \mathbb{E}_{\sigma^2}\bigg[-\frac{1}{2\sigma^2}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta} + \sum_{i=1}^{n}\bigg(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}y_i - \exp\left\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\right\}\bigg)\bigg] \\ &= -\frac{1}{2}\mu_{q(1/\sigma^2)}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta} + \sum_{i=1}^{n}\bigg(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}y_i - \exp\left\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\right\}\bigg),\end{aligned}$$

and we do not recognise the kernel of any known distribution. □

Since $q^*(\boldsymbol{\beta})$ is not a standard form, we use a semi-parametric mean field variational Bayes approach to obtain the optimal variational densities $q^*(\boldsymbol{\beta})$ and $q^*(\sigma^2)$. In particular, we have to pre-specify a parametric family of density functions for $q(\boldsymbol{\beta})$. We follow Rohde and Wand (2016) and we choose the multivariate Normal density function in $\boldsymbol{\beta}$, $\boldsymbol{\beta} \sim N_p(\boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})$,

$$q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)}) = (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}_{q(\beta)}|^{-\frac{1}{2}}\exp\bigg\{-\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_{q(\beta)}^{-1}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)}\right)\bigg\}. \tag{1.38}$$

As a consequence, the mean field variational approximation in (1.36) takes the form of

$$q(\boldsymbol{\beta}, \sigma^2) = q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})q(\sigma^2), \tag{1.39}$$

leading to the factor graph in Figure 1.5 and to a closed form of the ELBO.

Proposition 1.10. *The lower bound* $\log \underline{p}(\mathbf{y}; q)$ *for the Poisson regression model in (1.31) and (1.33), and associated to the variational density factorized as* $q(\boldsymbol{\beta}, \sigma^2) = q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})q(\sigma^2)$*, given a vector of realizations of the dependent variable,* $\mathbf{y}$*, and design matrix,* $\mathbf{X}$*, can be expressed in a closed form:*

$$
\begin{aligned}
\log \underline{p}(\mathbf{y}; q) &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\big] + \mathbb{E}_q\big[-\log q(\sigma^2)\big] + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] \\
&\quad + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\sigma^2)\big] + \mathbb{E}_q\big[\log p(\sigma^2)\big] \\
&= \mathrm{H}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} + \mathrm{H}\left\{q(\sigma^2)\right\} + \overline{\mathrm{H}}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} \\
&\quad + \mathbb{E}_q\big[\log p(\sigma^2)\big] \\
&= \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2} + \alpha_{q(\sigma^2)} + \log\Gamma(\alpha_{q(\sigma^2)}) - \alpha_{q(\sigma^2)}\psi(\alpha_{q(\sigma^2)}) \\
&\quad + \sum_{i=1}^{n}\left(\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} y_i - \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}\right\} - \log(y_i!)\right) \\
&\quad - \frac{p}{2}\big[\log(\delta_{q(\sigma^2)}) - \psi(\alpha_{q(\sigma^2)})\big] - \frac{1}{2}\mu_{q(1/\sigma^2)}\left[\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\beta)}\right\} + \boldsymbol{\mu}_{q(\beta)}^\intercal \boldsymbol{\mu}_{q(\beta)}\right] \\
&\quad + \alpha_0 \log\delta_0 - \log\Gamma(\alpha_0) - \alpha_0\log\delta_{q(\sigma^2)} + \alpha_0\psi(\alpha_{q(\sigma^2)}) - \delta\mu_{q(1/\sigma^2)}.
\end{aligned}
\tag{1.40}
$$

Proof. The first term is

$$
\begin{aligned}
\mathrm{H}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\big] \\
&= \mathbb{E}_q\bigg[-\bigg(-\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| \\
&\qquad - \frac{1}{2}\Big((\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)})^\intercal \boldsymbol{\Sigma}_{q(\beta)}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)})\Big)\bigg)\bigg] \\
&= \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2},
\end{aligned}
$$

the second term is

$$
\begin{aligned}
\mathrm{H}\left\{q(\sigma^2)\right\} &= \mathbb{E}_q\big[-\log q(\sigma^2)\big] \\
&= \mathbb{E}_q\left[-\alpha_{q(\sigma^2)}\log\delta_{q(\sigma^2)} + \log\Gamma(\alpha_{q(\sigma^2)}) + \big(\alpha_{q(\sigma^2)} + 1\big)\log\sigma^2 + \frac{\delta_{q(\sigma^2)}}{\sigma^2}\right] \\
&= -\alpha_{q(\sigma^2)}\log\delta_{q(\sigma^2)} + \log\Gamma(\alpha_{q(\sigma^2)}) + \big(\alpha_{q(\sigma^2)} + 1\big)\log\delta_{q(\sigma^2)} \\
&\quad - \big(\alpha_{q(\sigma^2)} + 1\big)\psi(\alpha_{q(\sigma^2)}) + \alpha_{q(\sigma^2)} \\
&= \alpha_{q(\sigma^2)} + \log\big(\delta_{q(\sigma^2)}\Gamma(\alpha_{q(\sigma^2)})\big) - \big(\alpha_{q(\sigma^2)} + 1\big)\psi(\alpha_{q(\sigma^2)}),
\end{aligned}
$$

the third term is

$$
\overline{\mathrm{H}}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} = \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\sigma^2)\big]
$$

$$
\begin{aligned}
&= \mathbb{E}_q\Bigg[\sum_{i=1}^{n}\Big(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}y_i - \exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\} - \log(y_i!)\Big) \\
&\quad - \frac{p}{2}\log 2\pi - \frac{p}{2}\log\sigma^2 - \frac{1}{2}\frac{\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta}}{\sigma^2}\Bigg] \\
&= \sum_{i=1}^{n}\Big(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}y_i - \exp\Big\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}\Big\} - \log(y_i!)\Big) \\
&\quad - \frac{p}{2}\log 2\pi - \frac{p}{2}\big[\log(\delta_{q(\sigma^2)}) - \psi(\alpha_{q(\sigma^2)})\big] \\
&\quad - \frac{1}{2}\mu_{q(1/\sigma^2)}\Big[\mathsf{tr}\,\big\{\boldsymbol{\Sigma}_{q(\beta)}\big\} + \boldsymbol{\mu}_{q(\beta)}^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}\Big],
\end{aligned}
$$

the fourth term is

$$
\begin{aligned}
\mathbb{E}_q\big[\log p(\sigma^2)\big] &= \mathbb{E}_q\Big[\alpha_0\log\delta_0 - \log\Gamma(\alpha_0) - (\alpha_0+1)\log\sigma^2 - \frac{\delta}{\sigma^2}\Big] \\
&= \alpha_0\log\delta_0 - \log\Gamma(\alpha_0) - (\alpha_0+1)\log\delta_{q(\sigma^2)} \\
&\quad + (\alpha_0+1)\psi(\alpha_{q(\sigma^2)}) - \delta\mu_{q(1/\sigma^2)}.
\end{aligned}
$$

After some simplification we obtain the result in Proposition 1.10. □

Following Rohde and Wand (2016), a possibile way to obtain the optimal $\boldsymbol{\mu}_{q(\beta)}$ and $\boldsymbol{\Sigma}_{q(\beta)}$ is based on the maximization of the $\boldsymbol{\beta}$-localized approximate log-likelihood, $\log \underline{p}(y;q)^{[\boldsymbol{\beta}]}$. The red line box in Figure 1.5 highlights the neighboring factors of $\boldsymbol{\beta}$: they are $p(\mathbf{y}|\boldsymbol{\beta})$ and $p(\boldsymbol{\beta}|\sigma^2)$ and they allow to have the following proposition.

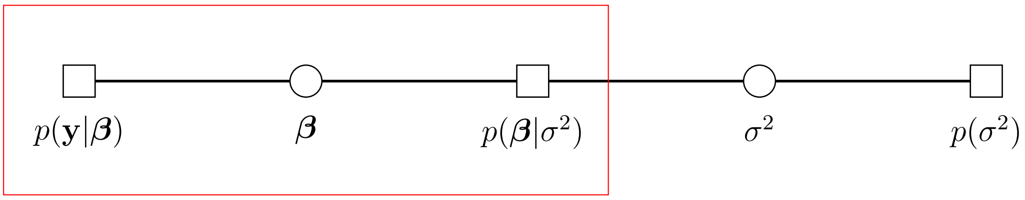

$p(\mathbf{y}|\boldsymbol{\beta})$
$\boldsymbol{\beta}$
$p(\boldsymbol{\beta}|\sigma^2)$
σ^2
$p(\sigma^2)$

Figure 1.5: Factor graph for the model (1.31) and (1.33) with stochastic nodes corresponding to the mean field restriction (1.39).

Proposition 1.11. *The $\boldsymbol{\beta}$-localized component of lower bound $\log \underline{p}(\mathbf{y};q)$ can be ex-*

pressed in a closed form:

$$
\begin{aligned}
\log \underline{p}(\mathbf{y};q)^{[\beta]} &= \mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} + \overline{\mathrm{H}}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} \\
&= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\big] + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\sigma^2)\big] \\
&= \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2} + \sum_{i=1}^{n}\Bigg(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}y_i \\
&\quad - \log(y_i!)\Bigg) - \frac{p}{2}\big(\log(\delta_{q(\sigma^2)}) - \psi(\alpha_{q(\sigma^2))})\big) \\
&\quad - \frac{1}{2}\mu_{q(1/\sigma^2}\Big[\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\beta)}\right\} + \boldsymbol{\mu}_{q(\beta)}^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}\Big].
\end{aligned}
\tag{1.41}
$$

Proof. The first term is

$$
\begin{aligned}
\mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta})\big] \\
&= \mathbb{E}_q\Bigg[-\Bigg(-\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| \\
&\quad - \frac{1}{2}\Big((\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\beta)})^{\mathsf{T}}\boldsymbol{\Sigma}_{q(\beta)}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\beta)})\Big)\Bigg)\Bigg] \\
&= \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2},
\end{aligned}
$$

while the second term is

$$
\begin{aligned}
\overline{\mathrm{H}}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\sigma^2)\big] \\
&= \mathbb{E}_q\Bigg[\sum_{i=1}^{n}\Big(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}y_i - \exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\} - \log(y_i!)\Big)\Bigg] - \frac{p}{2}\log 2\pi \\
&\quad - \frac{p}{2}\log\sigma^2 - \frac{1}{2}\frac{\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta}}{\sigma^2} \\
&= \sum_{i=1}^{n}\Bigg(\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}y_i - \exp\left\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}\right\} - \log(y_i!)\Bigg) \\
&\quad - \frac{p}{2}\log 2\pi - \frac{p}{2}\big(\log(\delta_{q(\sigma^2)}) - \psi(\alpha_{q(\sigma^2)})\big) \\
&\quad - \frac{1}{2}\mu_{q(1/\sigma^2)}\Big[\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\beta)}\right\} + \boldsymbol{\mu}_{q(\beta)}^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)}\Big].
\end{aligned}
$$

After some simplification we obtain the result. □

From a computational point of view, we consider the natural fixed-point iteration algorithm for the maximization of $\log \underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]}$. The reason is that in the case of $q(\boldsymbol{\phi};\boldsymbol{\xi}) = N(\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$, the natural fixed-point iteration is equivalent to the following updating scheme (Rohde and Wand, 2016):

$$\begin{cases} \boldsymbol{\nu}_{q(\beta)} \leftarrow \dfrac{d}{d\boldsymbol{\mu}_{q(\beta)}} \overline{\mathrm{H}}\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\}^{\mathsf{T}} \\ \boldsymbol{\Sigma}_{q(\beta)} \leftarrow -\left\{ -\dfrac{d^2}{d\boldsymbol{\mu}_{q(\beta)}^{\mathsf{T}} d\boldsymbol{\mu}_{q(\beta)}} \overline{\mathrm{H}}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} \right\}^{-1} \\ \boldsymbol{\mu}_{q(\beta)} \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)} \boldsymbol{\nu}_{q(\beta)}. \end{cases} \tag{1.42}$$

This updating scheme has the advantage that requires only the first and second derivatives of $\overline{\mathrm{H}}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\}$ with respect to $\boldsymbol{\mu}_{q(\beta)}$. In this example, the updating scheme in (1.42) corresponds to:

$$\begin{cases} \boldsymbol{\nu}_{q(\beta)} \leftarrow -\mu_{q(1/\sigma^2)} \boldsymbol{\mu}_{q(\beta)} + \sum_{i=1}^n \mathbf{x}_i y_i - \sum_{i=1}^n \mathbf{x}_i \exp\left\{ \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\mu}_{q(\beta)} + \dfrac{1}{2} \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{q(\beta)} \mathbf{x}_i \right\} \\ \boldsymbol{\Sigma}_{q(\beta)} \leftarrow -\left\{ -\mu_{q(1/\sigma^2)} I_p - \sum_{i=i}^n \mathbf{x}_i \exp\left\{ \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\mu}_{q(\beta)} + \dfrac{1}{2} \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\Sigma}_{q(\beta)} \mathbf{x}_i \right\} \mathbf{x}_i^{\mathsf{T}} \right\}^{-1} \\ \boldsymbol{\mu}_{q(\beta)} \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)} \boldsymbol{\nu}_{q(\beta)}. \end{cases} \tag{1.43}$$

Thus, the natural application of Algorithm 4 in order to obtain the optimal variational densities $q^*(\boldsymbol{\beta})$ and $q^*(\sigma^2)$ is provided in Algorithm 6.

Application to simulated dataset. We simulate $n = 300$ independent observations from a Poisson distribution, $Y_i|\boldsymbol{\beta} \sim \mathsf{Poi}(\exp\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}\})$, $i = 1, \ldots, 300$, where

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 0.00 \\ 0.19 \\ 0.26 \end{pmatrix} \tag{1.44}$$

and the covariates $\mathbf{x}_i$ are generated from a standard Normal distribution.

As concerns the hyperparameters' setting, we fix $\alpha = \dfrac{1}{2}$ and $\delta = 2$ in order to have a non-informative Inverse-Gamma distribution for σ^2, i.e. $\sigma^2 \sim \mathsf{InvGa}\left(\dfrac{1}{2}, 2\right)$. Running the MFVB algorithm required only 4 iterations, after which the change in the global lower bound is negligible (i.e. less than 1e-5). We get the vector

$$\boldsymbol{\mu}_{q(\beta)} = \begin{pmatrix} -0.01 \\ 0.22 \\ 0.27 \end{pmatrix} \tag{1.45}$$

as point estimates for $\boldsymbol{\beta}$ and $\mu_{q(\sigma^2)} = 2.07$ as point estimate for σ^2.

Also in this case we can compare the results obtained from MCMC and MFVB. Figure 1.6 and Figure 1.7 depicts respectively the results about $\boldsymbol{\beta}$ and σ^2.

Algorithm 6: Semi-parametric MFVB for Poisson regression model with prior for the variance of $\boldsymbol{\beta}$.

```
Initialize: $q^*(\boldsymbol{\beta})$, $q^*(\sigma^2)$, $\varepsilon_\beta$, $\varepsilon_{global}$
while convergence not reached do
    while convergence not reached do
        $\boldsymbol{\nu}_{q(\beta)} \leftarrow -\mu_{q(1/\sigma^2)}\boldsymbol{\mu}_{q(\beta)} + \sum_{i=1}^n \mathbf{x}_i y_i - \sum_{i=1}^n \mathbf{x}_i \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\}$
        $\boldsymbol{\Sigma}_{q(\beta)} \leftarrow -\left\{ -\mu_{q(1/\sigma^2)} I_p - \sum_{i=i}^n \mathbf{x}_i \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\} \mathbf{x}_i^\intercal \right\}^{-1}$
        $\boldsymbol{\mu}_{q(\beta)} \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)}\boldsymbol{\nu}_{q(\beta)}$
        compute $\log \underline{p}(y;q)^{[\boldsymbol{\beta}](z)}$;
        evaluate $\log \underline{p}(y;q)^{[\boldsymbol{\beta}](z)} - \log \underline{p}(y;q)^{[\boldsymbol{\beta}](z-1)} < \varepsilon_\beta$;
    end
    $\alpha_{q(\sigma^2)} \leftarrow \alpha + \frac{p}{2}$
    $\delta_{q(\sigma^2)} \leftarrow \delta + \frac{1}{2}\left[\mathsf{tr}\left\{\boldsymbol{\Sigma}_{q(\beta)}\right\} + \boldsymbol{\mu}_{q(\beta)}^\intercal \boldsymbol{\mu}_{q(\beta)}\right]$
    $\mu_{q(1/\sigma^2)} \leftarrow \frac{\alpha_{q(\sigma^2)}}{\delta_{q(\sigma^2)}}$
    compute $\log \underline{p}(y;q)^{(iter)}$;
    evaluate $|\log \underline{p}(y;q)^{(iter)} - \log \underline{p}(y;q)^{(iter-1)}| < \varepsilon_{global}$;
end
```

We observe that the MFVB approximate posterior distributions of $\boldsymbol{\beta}$ and σ^2 do not suffer of relevant problems compared to the MCMC benchmark. In the latter, we ran five parallel chains, with $R = 50000$ values and a burn-in of 25000 in each of them in order to have the multivariate scale reduction factor equal to 1.

After that the comparable results of MFVB and MCMC in terms of posterior inference accuracy have been tested, also in this framework we assess the advantage of MFVB with respect to MCMC in terms of computational burden. Thus we inspect the running time of both algorithms in two different situations. In the first we increase progressively the number of observations for a fixed number of variables ($p = 3$). In the second we increase the number of variables for a fixed number of observations ($n = 300$). As concerns the priors, we always consider $\boldsymbol{\beta}|\sigma^2 \sim \mathsf{N}_p(\mathbf{0}_p, \sigma^2\mathbf{I}_p)$ and $\sigma^2 \sim \mathsf{InvGa}\left(\frac{1}{2}, 2\right)$. Figure 1.8 shows the results and it highlights an higher speed of the MFVB with respect to MCMC approach also in this framework.

This illustrative example has the role to introduce the Bayesian Poisson regression model. In the next Chapter, we develop the Poisson regression model with three

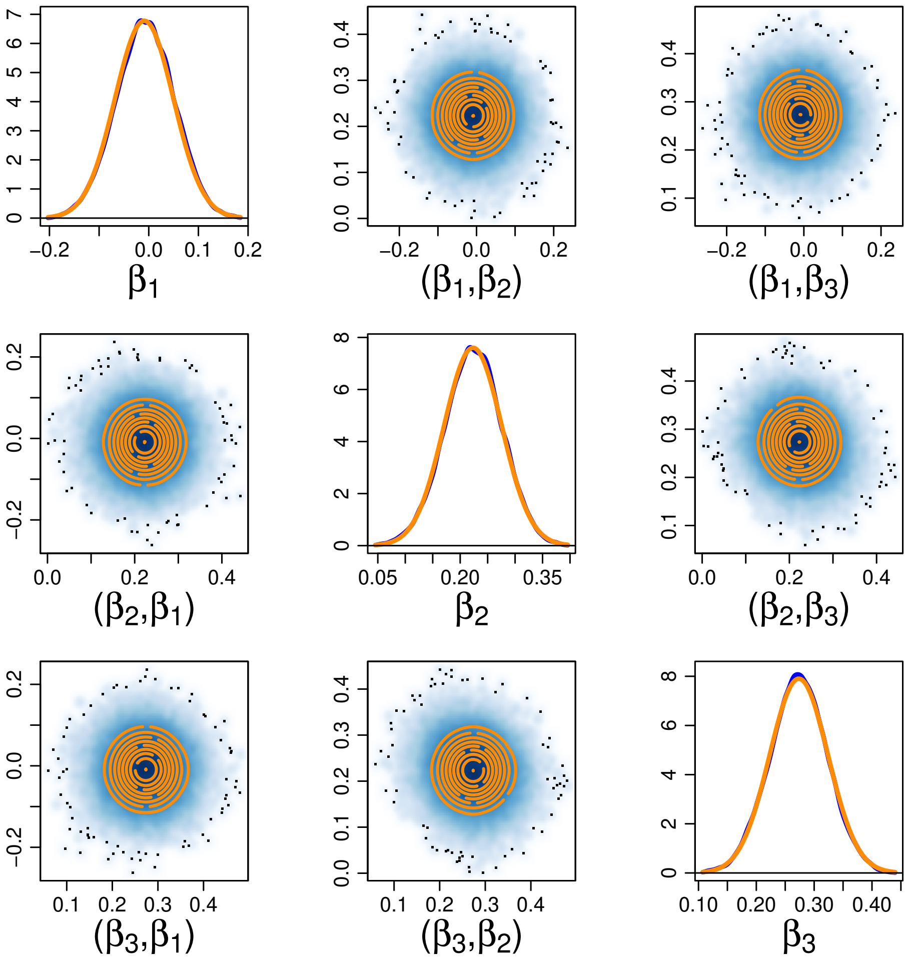


Figure 1.6: Marginal and bivariate posterior distributions for $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$ estimated both via MFVB (orange) and MCMC (blue).

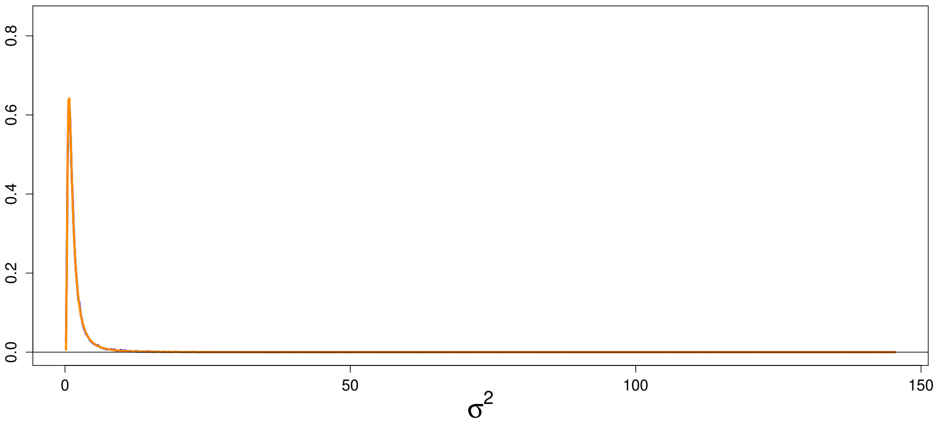


Figure 1.7: Posterior distributions for σ^2 estimated both via MFVB (orange) and MCMC (blue).

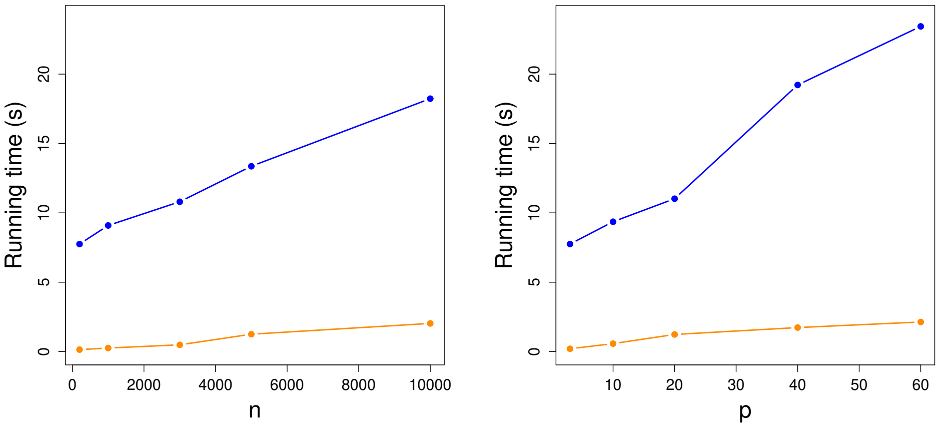


Figure 1.8: MFVB (orange) and MCMC (blue) running time in Poisson regression model for different number of observations with $p = 3$ (left) and for different number of variables with $n = 300$ (right).

different priors in order to induce sparsity and to perform variable selection.

Chapter 2

Variable selection for Poisson regression model

The Bayesian Poisson regression model considered in the previous Chapter allows to make Bayesian inference on the $\boldsymbol{\beta}$ vector-parameter in the classical way, i.e. through HPD credibility intervals. On the other hand, it does not allow to perform variable selection. Indeed, there is not element that brings information about the inclusion or exclusion of a variable from the model. However, the variable selection is rising in importance in lots of modern applications because the number of variables to work with is growing more and more and the standard methods seems to be inappropriate in lots of these cases. The problem of working with many covariates is even more important in the case of Poisson regression model with canonic link because the explosion of the predictor could lead to computational issues. A common approach to deal with this problem consists in perform shrinkage methods, which push towards 0 the unsignificant parameters.

In order to group the independent variables into signals and nulls, in this Chapter we develop, implement and compare Bayesian Poisson regression model with three different options: horseshoe prior, spike-and-slab prior and Bernoulli-Gaussian prior.

2.1 Horseshoe prior

In the Bayesian context, one of the most used continuos shrinkage prior is the horseshoe (HS) prior (Carvalho et al., 2010). The main idea of this prior is to perform shrinkage of regression coefficients which do not have effect on the response variable without affecting the largest ones. In contrast with the Bayesian lasso (Park and

Casella, 2008) and Bayesian ridge, where the shrinkage is uniform across all regression coefficients, its is a global-local shrinkage. This means that there is a common parameter responsible of the overall level of shrinkage and a specific parameter for each covariate responsible of the local shrinkage.

Following Wand et al. (2011), we use a scale mixture representation of the half-Cauchy distribution in order to have, except for the regression coefficients, all the full-conditionals recognized as known density functions.

Bayesian model specification. The Poisson regression model with horseshoe prior that we consider is the following:

$$\begin{aligned}
&Y_i|\boldsymbol{\beta} \sim \mathsf{Poi}(\exp\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\}), \quad i=1,...,n, \quad Y_i \perp Y_j \quad \forall i \neq j,\\
&\boldsymbol{\beta}|\boldsymbol{\lambda}^2,\tau^2 \sim \mathsf{N}_p(\mathbf{0}_p,\tau^2\boldsymbol{\Sigma}_\beta), \quad \boldsymbol{\Sigma}_\beta = \mathsf{diag}\left\{\lambda_1^2,\dots,\lambda_p^2\right\},\\
&\lambda_j^2|\nu_j \sim \mathsf{InvGa}\left(\frac{1}{2},\frac{1}{\nu_j}\right), \quad \nu_j \sim \mathsf{InvGa}\left(\frac{1}{2},1\right), \quad j=1,\dots,p,\\
&\tau^2|\eta \sim \mathsf{InvGa}\left(\frac{1}{2},\frac{1}{\eta}\right), \quad \eta \sim \mathsf{InvGa}\left(\frac{1}{2},1\right),
\end{aligned} \tag{2.1}$$

which is a hierarchical horseshoe prior for the vector of coefficients $\boldsymbol{\beta}$. Conditionally on $\boldsymbol{\lambda}^2$ and τ^2, the prior distribution of $\boldsymbol{\beta}$ is Gaussian with p-dimensional mean vector $\mathbf{0}_p$ and diagonal variance-covariance matrix $\tau^2\boldsymbol{\Sigma}_\beta$. In addition, the property of global-local shrinkage is guaranteed by the presence of τ^2 and λ_j^2, which determine global and local shrinkage respectively.

The joint prior distribution specified in (2.1) leads to have $(\boldsymbol{\beta},\boldsymbol{\lambda}^2,\tau^2,\eta,\boldsymbol{\nu})$ as set of parameters. Notice that, differently from similar methods, this prior does not require the choice of hyperparameters. The model specification in (2.1) leads to the following joint posterior

$$\begin{aligned}
p(\boldsymbol{\beta},\boldsymbol{\lambda}^2,\tau^2,\eta,\boldsymbol{\nu};\mathbf{y}) = &\prod_{i=1}^n \left\{\frac{e^{\mathbf{x}_i^\intercal\boldsymbol{\beta}y_i - \exp\left\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\right\}}}{y_i!}\right\}(2\pi)^{-\frac{p}{2}}(\tau^2)^{-\frac{p}{2}}\\
&\times\left\{\prod_{j=1}^p(\lambda_j^2)^{-\frac{1}{2}}\exp\left\{-\frac{\beta_j^2}{2\lambda_j^2\tau^2}\right\}\right\}\left(\frac{1}{\eta}\right)^{\frac{1}{2}}\frac{1}{\Gamma(\frac{1}{2})}(\tau^2)^{-\frac{3}{2}}\\
&\times\exp\left\{-\frac{1}{\eta\tau^2}\right\}\frac{1}{\Gamma(\frac{1}{2})}\eta^{-\frac{3}{2}}\exp\left\{-\frac{1}{\eta}\right\}\\
&\times\prod_{j=1}^p\left\{\left(\frac{1}{\nu_j}\right)^{\frac{1}{2}}\frac{1}{\Gamma(\frac{1}{2})}(\lambda_j^2)^{-\frac{3}{2}}\exp\left\{-\frac{1}{\nu_j\lambda_j^2}\right\}\right\}\\
&\times\prod_{j=1}^p\left\{\frac{1}{\Gamma(\frac{1}{2})}\nu_j^{-\frac{3}{2}}\exp\left\{-\frac{1}{\nu_j}\right\}\right\},
\end{aligned} \tag{2.2}$$

that plays a central role to fit the model via mean field variational Bayes.

Mean field variational Bayes approach. Let the density $q(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu})$ be a mean field approximation to the true posterior density of the form

$$q(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}) = q(\boldsymbol{\beta}) \prod_{j=1}^{p} q(\lambda_j^2) q(\tau^2) q(\eta) \prod_{j=1}^{p} q(\nu_j). \tag{2.3}$$

According to (1.9), the optimal densities for each element of the joint variational distribution are given by:

$$\begin{aligned}
q^*(\boldsymbol{\beta}) &\propto \exp\left\{\mathbb{E}_{-\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\},\\
q^*(\lambda_j^2) &\propto \exp\left\{\mathbb{E}_{-\lambda_j^2}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\}, \quad j = 1, \ldots, p,\\
q^*(\tau^2) &\propto \exp\left\{\mathbb{E}_{-\tau^2}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\},\\
q^*(\eta) &\propto \exp\left\{\mathbb{E}_{-\eta}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\},\\
q^*(\nu_j) &\propto \exp\left\{\mathbb{E}_{-\nu_j}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\}, \quad j = 1, \ldots, p.
\end{aligned} \tag{2.4}$$

Some computations highlight that all the optimal densities, except for $q^*(\boldsymbol{\beta})$, are available in closed form, as showed by the next propositions.

Proposition 2.1. *The optimal density for $\boldsymbol{\beta}$, $q^*(\boldsymbol{\beta})$, is not a standard form.*

Proof. Since $q^*(\boldsymbol{\beta}) \propto \exp\left\{\mathbb{E}_{-\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\}$,

$$\begin{aligned}
\log q^*(\boldsymbol{\beta}) &\propto \mathbb{E}_{-\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\\
&\propto \mathbb{E}_{-\boldsymbol{\beta}}\left[\sum_{i=1}^{n}\left\{\mathbf{x}_i^\intercal \boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^\intercal \boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\left\{\frac{\beta_j^2}{2\lambda_j^2\tau^2}\right\}\right]\\
&= \sum_{i=1}^{n}\left\{\mathbf{x}_i^\intercal \boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^\intercal \boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\left\{\frac{1}{2}\beta_j^2 \mu_{q(1/\lambda_j^2)} \mu_{q(1/\tau^2)}\right\}
\end{aligned}$$

and we do not recognise the kernel of any known distribution. □

Proposition 2.2. *The optimal density for λ_j^2 is $q^*(\lambda_j^2) \sim \mathsf{InvGa}\left(a_{q(\lambda_j^2)}, b_{q(\lambda_j^2)}\right)$, with*

$$a_{q(\lambda_j^2)} = 1, \qquad b_{q(\lambda_j^2)} = \mu_{q(1/\nu_j)} + \frac{1}{2}(\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)})\mu_{q(1/\tau^2)}. \tag{2.5}$$

Furthermore, $\mu_{q(1/\lambda_j^2)} = a_{q(\lambda_j^2)}/b_{q(\lambda_j^2)}$ and $\mu_{q(\log \lambda_j^2)} = \log b(_{q(\lambda_j^2)}) - \psi(a_{q(\lambda_j^2)})$.

Proof. Since $q^*(\lambda_j^2) \propto \exp\left\{\mathbb{E}_{-\lambda_j^2}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]\right\}$,

$$\log q^*(\lambda_j^2) \propto \mathbb{E}_{-\lambda_j^2}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\right]$$

$$\begin{aligned}
&\propto \mathbb{E}_{-\lambda_j^2}\left[-\frac{1}{2}\log\lambda_j^2-\frac{\beta_j^2}{2\lambda_j^2\tau^2}-\frac{1}{\nu_j}\frac{1}{\lambda_j^2}-\frac{3}{2}\log\lambda_j^2\right]\\
&=-2\log\lambda_j^2-\frac{1}{\lambda_j^2}\left[\mu_{q(1/\nu_j)}+\frac{1}{2}\mu_{q(\beta_j^2)}\mu_{q(1/\tau^2)}\right]\\
&=-2\log\lambda_j^2-\frac{1}{\lambda_j^2}\left[\mu_{q(1/\nu_j)}+\frac{1}{2}(\sigma^2_{q(\beta_j)}+\mu^2_{q(\beta_j)})\mu_{q(1/\tau^2)}\right].
\end{aligned}$$

Take the exponential and notice that it coincides with the kernel of an Inverse-Gamma distribution with parameters as in Proposition 2.2. □

Proposition 2.3. *The optimal density for τ^2 is $q^*(\tau^2)\sim \mathsf{InvGa}\left(a_{q(\tau^2)},b_{q(\tau^2)}\right)$, with*

$$a_{q(\tau^2)}=\frac{p+1}{2},\qquad b_{q(\tau^2)}=\mu_{q(1/\eta)}+\frac{1}{2}\sum_{j=1}^{p}(\sigma^2_{q(\beta_j)}+\mu^2_{q(\beta_j)})\mu_{q(1/\lambda_j^2)}. \tag{2.6}$$

Furthermore, $\mu_{q(1/\tau^2)}=a_{q(\tau^2)}/b_{q(\tau^2)}$ and $\mu_{q(\log\tau^2)}=\log b(_{q(\tau^2)})-\psi(a_{q(\tau^2)})$.

Proof. Since $q^*(\tau^2)\propto\exp\left\{\mathbb{E}_{-\tau^2}\left[\log p(\boldsymbol{\beta},\boldsymbol{\lambda}^2,\tau^2,\eta,\boldsymbol{\nu};\mathbf{y})\right]\right\}$,

$$\begin{aligned}
\log q^*(\tau^2)&\propto\mathbb{E}_{-\tau^2}\left[\log p(\boldsymbol{\beta},\boldsymbol{\lambda}^2,\tau^2,\eta,\boldsymbol{\nu};\mathbf{y})\right]\\
&\propto\mathbb{E}_{-\tau^2}\left[-\frac{p}{2}\log\tau^2-\frac{1}{2\tau^2}\sum_{j=1}^{p}\frac{\beta_j^2}{\lambda_j^2}-\frac{1}{\eta}\frac{1}{\tau^2}-\frac{3}{2}\log\tau^2\right]\\
&=-\left(\frac{p+3}{2}\right)\log\tau^2-\frac{1}{\tau^2}\left[\mu_{q(1/\eta)}+\frac{1}{2}\sum_{j=1}^{p}\mu_{q(\beta_j^2)}\mu_{q(1/\lambda_j^2)}\right]\\
&=-\left(\frac{p+3}{2}\right)\log\tau^2-\frac{1}{\tau^2}\left[\mu_{q(1/\eta)}+\frac{1}{2}\sum_{j=1}^{p}(\sigma^2_{q(\beta_j)}+\mu^2_{q(\beta_j)})\mu_{q(1/\lambda_j^2)}\right].
\end{aligned}$$

Take the exponential and notice that it coincide with the kernel of an Inverse-Gamma distribution with parameters as in Proposition 2.3. □

Proposition 2.4. *The optimal density for η is $q^*(\eta)\sim\mathsf{InvGa}\left(a_{q(\eta)},b_{q(\eta)}\right)$, with*

$$a_{q(\eta)}=1,\qquad b_{q(\eta)}=1+\mu_{q(1/\tau^2)}. \tag{2.7}$$

Furthermore, $\mu_{q(1/\eta)}=a_{q(\eta)}/b_{q(\eta)}$ and $\mu_{q(\log\eta)}=\log b(_{q(\eta)})-\psi(a_{q(\eta)})$.

Proof. Since $q^*(\eta)\propto\exp\left\{\mathbb{E}_{-\eta}\left[\log p(\boldsymbol{\beta},\boldsymbol{\lambda}^2,\tau^2,\eta,\boldsymbol{\nu};\mathbf{y})\right]\right\}$,

$$\begin{aligned}
\log q^*(\eta)&\propto\mathbb{E}_{-\eta}\left[\log p(\boldsymbol{\beta},\boldsymbol{\lambda}^2,\tau^2,\eta,\boldsymbol{\nu};\mathbf{y})\right]\\
&\propto\mathbb{E}_{-\eta}\left[-\frac{1}{2}\log\eta-\frac{1}{\eta}\frac{1}{\tau^2}-\frac{3}{2}\log\eta-\frac{1}{\eta}\right]
\end{aligned}$$

$$= -2\log\eta - \frac{1}{\eta}\Bigg[\mu_{q(1/\tau^2)} + 1\Bigg].$$

Take the exponential and notice that it coincides with the kernel of an Inverse-Gamma distribution with parameters as in Proposition 2.4. □

Proposition 2.5. *The optimal density for ν_j is $q^*(\nu_j) \sim \mathsf{InvGa}\left(a_{q(\nu_j)}, b_{q(\nu_j)}\right)$, with*

$$a_{q(\nu_j)} = 1, \qquad b_{q(\nu_j)} = 1 + \mu_{q(1/\lambda_j^2)}. \tag{2.8}$$

Furthermore, $\mu_{q(1/\nu_j)} = a_{q(\nu_j)}/b_{q(\nu_j)}$ and $\mu_{q(\log\nu_j)} = \log b(_{q(\nu_j)}) - \psi(a_{q(\nu_j)})$.

Proof. Since $q^*(\nu_j) \propto \exp\Bigg\{\mathbb{E}_{-\nu_j}\Bigg[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\Bigg]\Bigg\}$,

$$\begin{aligned}
\log q^*(\nu_j) &\propto \mathbb{E}_{-\nu_j}\Bigg[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}; \mathbf{y})\Bigg] \\
&\propto \mathbb{E}_{-\nu_j}\Bigg[-\frac{1}{2}\log\nu_j - \frac{1}{\nu_j}\frac{1}{\lambda_j^2} - \frac{3}{2}\log\nu_j - \frac{1}{\nu_j}\Bigg] \\
&= -2\log\nu_j - \frac{1}{\nu_j}\Bigg[\mu_{q(1/\lambda_j^2)} + 1\Bigg].
\end{aligned}$$

Take the exponential and notice that it coincides with the kernel of an Inverse-Gamma distribution with parameters as in Proposition 2.5. □

As we have already seen in the example of Section 1.3.1, since $q^*(\boldsymbol{\beta})$ is not a standard form, we use a semi-parametric mean field variational Bayes approach to obtain the variational Bayes densities. In particular, we have to pre-specify parametric family of density functions for $q(\boldsymbol{\beta})$. Also in this case we choose the multivariate normal, $\boldsymbol{\beta} \sim \mathsf{N}_p(\boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})$, so that:

$$q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)}) = (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}_{q(\beta)}|^{-\frac{1}{2}}\exp\Bigg\{-\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)}\right)^{\intercal}\boldsymbol{\Sigma}_{q(\beta)}^{-1}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)}\right)\Bigg\}. \tag{2.9}$$

At this point, the mean field variational approximation in (2.3) takes the following form:

$$q(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}) = q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\prod_{j=1}^{p} q(\lambda_j^2) q(\tau^2) q(\eta)\prod_{j=1}^{p} q(\nu_j), \tag{2.10}$$

leading to the factor graph in Figure 2.1 and to a closed form of the lower bound.

Proposition 2.6. *The lower bound $\log\underline{p}(\mathbf{y}; q)$ for the Poisson regression model with horseshoe prior in (2.1), and associated to the variational density factorized as*

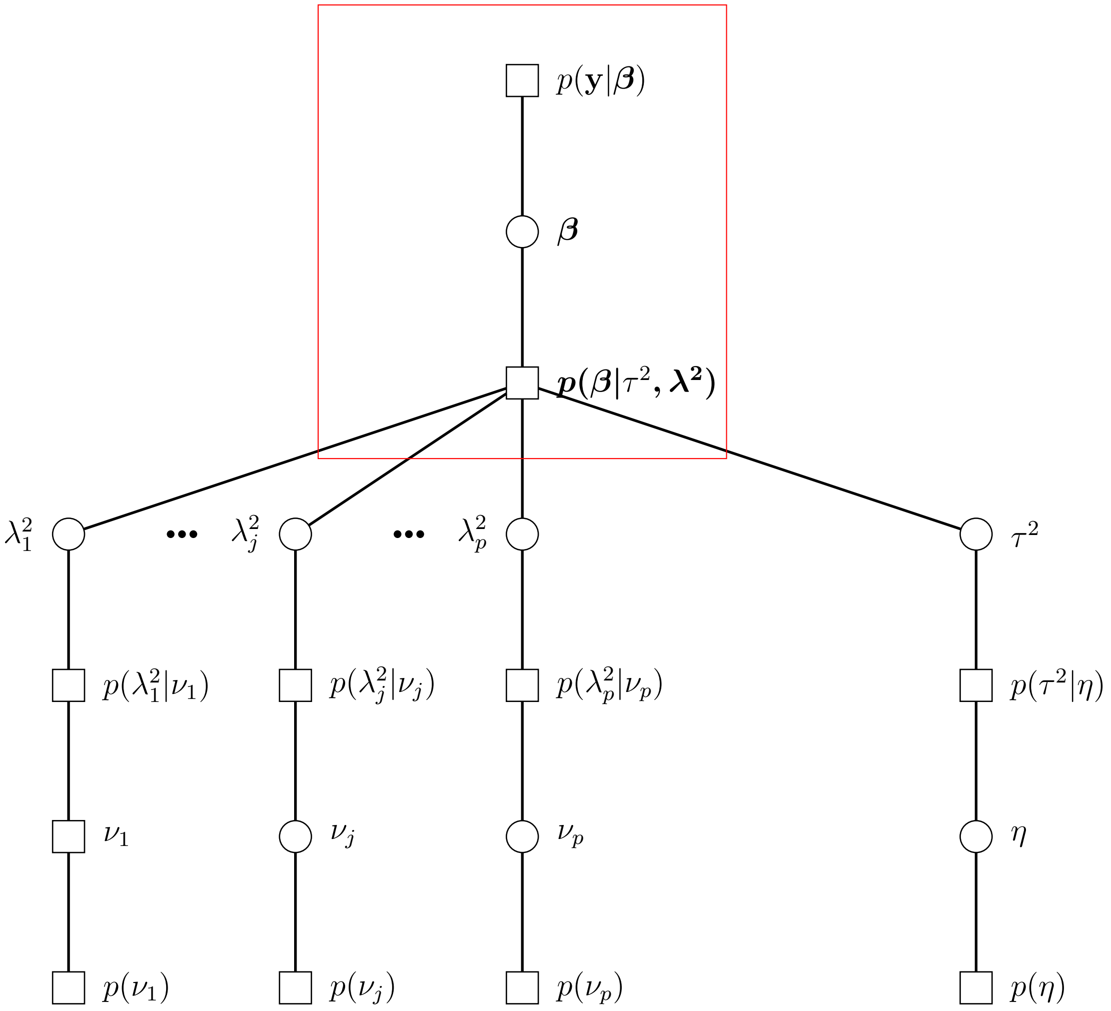


Figure 2.1: Factor graph for the model (2.1) with stochastic nodes corresponding to the mean field restriction (2.10).

$q(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \eta, \boldsymbol{\nu}) = q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)}) \prod_{j=1}^{p} q(\lambda_j^2) q(\tau^2) q(\eta) \prod_{j=1}^{p} q(\nu_j)$, *given a vector of realizations of dependent variable,* $\mathbf{y}$, *and design matrix,* $\mathbf{X}$, *can be expressed in a closed form:*

$$\begin{aligned}
\log \underline{p}(\mathbf{y}; q) = {} & \frac{1}{2} \log |\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2} + \sum_{j=1}^{p} \Big\{ a_{q(\lambda_j^2)} + \log(b_{q(\lambda_j^2)} \Gamma(a_{q(\lambda_j^2)})) \\
& - (a_{q(\lambda_j^2)} + 1) \psi(a_{q(\lambda_j^2)}) \Big\} + a_{q(\tau^2)} + \log(b_{q(\tau^2)} \Gamma(a_{q(\tau^2)})) \\
& - (a_{q(\tau^2)} + 1) \psi(a_{q(\tau^2)}) + a_{q(\eta)} + \log(b_{q(\eta)} \Gamma(a_{q(\eta)})) \\
& - (a_{q(\eta)} + 1) \psi(a_{q(\eta)}) + \sum_{j=1}^{p} \Big\{ a_{q(\nu_j)} + \log(b_{q(\nu_j)} \Gamma(a_{q(\nu_j)})) \\
& - (a_{q(\nu_j)} + 1) \psi(a_{q(\nu_j)}) \Big\} - \frac{p}{2} \mu_{q(\log \tau^2)} - \frac{1}{2} \sum_{j=1}^{p} \mu_{q(\log \lambda_j^2)} \\
& + \sum_{i=1}^{n} \Big\{ \mathbf{x}_i^{\intercal} \boldsymbol{\mu}_{q(\beta)} y_i - \exp\Big\{ \mathbf{x}_i^{\intercal} \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2} \mathbf{x}_i^{\intercal} \boldsymbol{\Sigma}_{q(\beta)} \mathbf{x}_i \Big\} - \log(y_i!) \Big\} \qquad (2.11) \\
& - \sum_{j=1}^{p} \Big\{ \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2} \mu_{q(1/\lambda_j^2)} \mu_{q(1/\tau^2)} \Big\} - p \log \Gamma\left(\frac{1}{2}\right) \\
& - \sum_{j=1}^{p} \Big\{ \frac{1}{2} \mu_{q(\log \nu_j)} + \frac{3}{2} \mu_{q(\log \lambda_j^2)} + \mu_{q(1/\nu_j)} \mu_{q(1/\lambda_j^2)} \Big\} \\
& - \frac{1}{2} \mu_{q(\log \eta)} - \log \Gamma\left(\frac{1}{2}\right) - \frac{3}{2} \mu_{q(\log \tau^2)} - \mu_{q(1/\eta)} \mu_{q(1/\tau^2)} \\
& - \log \Gamma\left(\frac{1}{2}\right) - \frac{3}{2} \mu_{q(\log \eta)} - \mu_{q(1/\eta)} - p \log \Gamma\left(\frac{1}{2}\right) \\
& - \sum_{j=1}^{p} \Big\{ \frac{3}{2} \mu_{q(\log \nu_j)} + \mu_{q(1/\nu_j)} \Big\}.
\end{aligned}$$

Proof. The lower bound can be expressed in terms of the components of the factor graph in Figure 2.1:

$$\begin{aligned}
\log \underline{p}(\mathbf{y}; q) = {} & \mathrm{H}\left\{ q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)}) \right\} + \sum_{j=1}^{p} \mathrm{H}\left\{ q(\lambda_j^2) \right\} \\
& + \mathrm{H}\left\{ q(\tau^2) \right\} + \mathrm{H}\left\{ q(\eta) \right\} + \sum_{j=1}^{p} \mathrm{H}\left\{ q(\nu_j) \right\} \\
& + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta}) \big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\lambda}^2, \tau^2) \big] + \sum_{j=1}^{p} \mathbb{E}_q\big[\log p(\lambda_j^2|\nu_j) \big] \qquad (2.12) \\
& + \mathbb{E}_q\big[\log p(\tau^2|\eta) \big] + \mathbb{E}_q\big[\log p(\eta) \big] + \sum_{j=1}^{p} \mathbb{E}_q\big[p(\nu_j) \big].
\end{aligned}$$

Moreover, the first term is

$$\begin{aligned}
\mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\big] \\
&= \mathbb{E}_q\Bigg[-\Bigg[-\frac{p}{2}\log 2\pi-\frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| \\
&\quad -\frac{1}{2}\bigg((\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\beta)})^{\intercal}\boldsymbol{\Sigma}_{q(\beta)}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\beta)})\bigg)\Bigg]\Bigg] \\
&= \frac{p}{2}\log 2\pi+\frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}|+\frac{p}{2},
\end{aligned}$$

the second term is

$$\sum_{j=1}^{p}\mathrm{H}\left\{q(\lambda_j^2)\right\}=\sum_{j=1}^{p}\bigg\{a_{q(\lambda_j^2)}+\log(b_{q(\lambda_j^2)}\Gamma(a_{q(\lambda_j^2)}))-(a_{q(\lambda_j^2)}+1)\psi(a_{q(\lambda_j^2)})\bigg\},$$

the third term is

$$\mathrm{H}\left\{q(\tau^2)\right\}=a_{q(\tau^2)}+\log(b_{q(\tau^2)}\Gamma(a_{q(\tau^2)}))-(a_{q(\tau^2)}+1)\psi(a_{q(\tau^2)}),$$

the fourth term is

$$\mathrm{H}\left\{q(\eta)\right\}=a_{q(\eta)}+\log(b_{q(\eta)}\Gamma(a_{q(\eta)}))-(a_{q(\eta)}+1)\psi(a_{q(\eta)}),$$

the fifth term is

$$\sum_{j=1}^{p}\mathrm{H}\left\{q(\nu_j)\right\}=\sum_{j=1}^{p}\bigg\{a_{q(\nu_j)}+\log(b_{q(\nu_j)}\Gamma(a_{q(\nu_j)}))-(a_{q(\nu_j)}+1)\psi(a_{q(\nu_j)})\bigg\},$$

the sixth term is

$$\begin{aligned}
\mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] &= \mathbb{E}_q\Bigg[\sum_{i=1}^{n}\bigg[\mathbf{x}_i^{\intercal}\boldsymbol{\beta}y_i-\exp\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\right\}-\log(y_i!)\bigg]\Bigg] \\
&= \sum_{i=1}^{n}\bigg\{\mathbf{x}_i^{\intercal}\boldsymbol{\mu}_{q(\beta)}y_i-\exp\bigg\{\mathbf{x}_i^{\intercal}\boldsymbol{\mu}_{q(\beta)}+\frac{1}{2}\mathbf{x}_i^{\intercal}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\bigg\}-\log(y_i!)\bigg\},
\end{aligned}$$

the seventh term is

$$\begin{aligned}
\mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\lambda}^2,\tau^2)\big] &= \mathbb{E}_q\Bigg[-\frac{p}{2}\log 2\pi-\frac{p}{2}\log\tau^2+\sum_{j=1}^{p}\bigg\{-\frac{1}{2}\log\lambda_j^2-\frac{\beta_j^2}{2\lambda_j^2\tau^2}\bigg\}\Bigg] \\
&= -\frac{p}{2}\log 2\pi-\frac{p}{2}\mu_{q(\log\tau^2)}+\sum_{j=1}^{p}\bigg\{-\frac{1}{2}\mu_{q(\log\lambda_j^2)} \\
&\quad -\frac{\mu_{q(\beta_j^2)}}{2}\mu_{q(1/\lambda_j^2)}\mu_{q(1/\tau^2)}\bigg\} \\
&= -\frac{p}{2}\log 2\pi-\frac{p}{2}\mu_{q(\log\tau^2)}-\frac{1}{2}\sum_{j=1}^{p}\mu_{q(\log\lambda_j^2)}
\end{aligned}$$

$$-\sum_{j=1}^{p}\left\{\frac{\sigma^2_{q(\beta_j)}+\mu^2_{q(\beta_j)}}{2}\mu_{q(1/\lambda_j^2)}\mu_{q(1/\tau^2)}\right\},$$

the eighth term is

$$\begin{aligned}\sum_{j=1}^{p}\mathbb{E}_q\big[\log p(\lambda_j^2|\nu_j)\big] &= \sum_{j=1}^{p}\mathbb{E}_q\bigg[-\frac{1}{2}\log\nu_j-\log\Gamma\bigg(\frac{1}{2}\bigg)-\frac{3}{2}\log\lambda_j^2-\frac{1}{\nu_j\lambda_j^2}\bigg]\\ &= -p\log\Gamma\bigg(\frac{1}{2}\bigg)-\sum_{j=1}^{p}\left\{\frac{1}{2}\mu_{q(\log\nu_j)}+\frac{3}{2}\mu_{q(\log\lambda_j^2)}+\mu_{q(1/\nu_j)}\mu_{q(1/\lambda_j^2)}\right\},\end{aligned}$$

the ninth term is

$$\begin{aligned}\mathbb{E}_q\big[\log p(\tau^2|\eta)\big] &= \mathbb{E}_q\bigg[-\frac{1}{2}\log\eta-\log\Gamma\bigg(\frac{1}{2}\bigg)-\frac{3}{2}\log\tau^2-\frac{1}{\eta\tau^2}\bigg]\\ &= -\frac{1}{2}\mu_{q(\log\eta)}-\log\Gamma\bigg(\frac{1}{2}\bigg)-\frac{3}{2}\mu_{q(\log\tau^2)}-\mu_{q(1/\eta)}\mu_{q(1/\tau^2)},\end{aligned}$$

the tenth term is

$$\begin{aligned}\mathbb{E}_q\big[\log p(\eta)\big] &= \mathbb{E}_q\bigg[-\log\Gamma\bigg(\frac{1}{2}\bigg)-\frac{3}{2}\log\eta-\frac{1}{\eta}\bigg]\\ &= -\log\Gamma\bigg(\frac{1}{2}\bigg)-\frac{3}{2}\mu_{q(\log\eta)}-\mu_{q(1/\eta)},\end{aligned}$$

the eleventh term is

$$\begin{aligned}\sum_{j=1}^{p}\mathbb{E}_q\big[p(\nu_j)\big] &= \sum_{j=1}^{p}\mathbb{E}_q\bigg[-\log\Gamma\bigg(\frac{1}{2}\bigg)-\frac{3}{2}\log\nu_j-\frac{1}{\nu_j}\bigg]\\ &= -p\log\Gamma\bigg(\frac{1}{2}\bigg)-\sum_{j=1}^{p}\left\{\frac{3}{2}\mu_{q(\log\nu_j)}+\mu_{q(1/\nu_j)}\right\}.\end{aligned}$$

Substituting in (2.12) and after the simplification of $\frac{p}{2}\log 2\pi$ in $\mathrm{H}\big\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\big\}$ with $-\frac{p}{2}\log 2\pi$ in $\mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\lambda}^2,\tau^2)\big]$, we obtain the result in Proposition 2.6. □

Finally, the last step for the derivation of the MFVB algorithm in the case of horseshoe prior is the update of the optimal parameters $(\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$ within a coordinate ascent scheme. For this purpose we maximize $\log\underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]}$, the $\boldsymbol{\beta}$-localized component of lower bound $\log\underline{p}(\mathbf{y};q)$, over $(\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$. The red line box in Figure 2.1 shows that the neighbours of $\boldsymbol{\beta}$ are $p(\mathbf{y}|\boldsymbol{\beta})$ and $p(\boldsymbol{\beta}|\boldsymbol{\gamma}^2,\boldsymbol{\lambda}^2)$, which lead to a closed form of $\log\underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]}$.

Proposition 2.7. *The $\boldsymbol{\beta}$-localized component of lower bound $\log \underline{p}(\mathbf{y}; q)$ can be expressed in a closed form and it is equal to:*

$$\begin{aligned}\log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\beta}]} &= \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2} + \sum_{i=1}^{n}\Bigg\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} y_i - \exp\Bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} \\ &+ \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\Bigg\} - \log(y_i!)\Bigg\} - \frac{p}{2}\mu_{q(\log\tau^2)} + \sum_{j=1}^{p}\Bigg\{ - \frac{1}{2}\mu_{q(\log\lambda_j^2)} \\ &- \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2}\mu_{q(1/\lambda_j^2)}\mu_{q(1/\tau^2)}\Bigg\}.\end{aligned} \tag{2.13}$$

Proof. Since the $\boldsymbol{\beta}$-localized component of $\log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\beta}]}$ is equal to

$$\begin{aligned}\log \underline{p}(\mathbf{y}; q)^{[\boldsymbol{\beta}]} &= \mathrm{H}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} + \overline{\mathrm{H}}\left\{q(\boldsymbol{\beta})\right\} \\ &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\big] + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\lambda_j^2, \tau^2)\big],\end{aligned} \tag{2.14}$$

the first term is

$$\begin{aligned}\mathrm{H}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\big] \\ &= \mathbb{E}_q\Bigg[-\Bigg[-\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| \\ &- \frac{1}{2}\Big((\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)})^\intercal \boldsymbol{\Sigma}^{-1}_{q(\beta)}(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)})\Big)\Bigg]\Bigg] \\ &= \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2},\end{aligned}$$

while the second term is

$$\begin{aligned}\overline{\mathrm{H}}\left\{q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\lambda_j^2, \tau^2)\big] \\ &= \mathbb{E}_q\Bigg[\sum_{i=1}^{n}\Big[\mathbf{x}_i^\intercal\boldsymbol{\beta} y_i - \exp\left\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\right\} - \log(y_i!)\Big]\Bigg] \\ &+ \mathbb{E}_q\Bigg[-\frac{p}{2}\log 2\pi - \frac{p}{2}\log\tau^2 + \sum_{j=1}^{p}\Bigg\{-\frac{1}{2}\log\lambda_j^2 - \frac{\beta_j^2}{2\lambda_j^2\tau^2}\Bigg\}\Bigg] \\ &= \sum_{i=1}^{n}\Bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} y_i - \exp\Bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\Bigg\} \\ &- \log(y_i!)\Bigg\} - \frac{p}{2}\log 2\pi - \frac{p}{2}\mu_{q(\log\tau^2)} \\ &+ \sum_{j=1}^{p}\Bigg\{-\frac{1}{2}\mu_{q(\log\lambda_j^2)} - \frac{\mu_{q(\beta_j^2)}}{2}\mu_{q(1/\lambda_j^2)}\mu_{q(1/\tau^2)}\Bigg\} \\ &= \sum_{i=1}^{n}\Bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} y_i - \exp\Bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\Bigg\}\end{aligned}$$

$$
\begin{aligned}
&- \log(y_i!)\Big\} - \frac{p}{2}\log 2\pi - \frac{p}{2}\mu_{q(\log\tau^2)} \\
&+ \sum_{j=1}^{p}\Bigg\{ - \frac{1}{2}\mu_{q(\log\lambda_j^2)} - \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2}\mu_{q(1/\lambda_j^2)}\mu_{q(1/\tau^2)}\Bigg\}.
\end{aligned}
$$

Susbstituting in (2.14) and after the simplification of $\frac{p}{2}\log 2\pi$ in $\mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\}$ with $-\frac{p}{2}\log 2\pi$ in $\overline{\mathrm{H}}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\}$, we obtain the result in Proposition 2.7. □

For the reason explained in the example of Section 1.3.1, we use the natural fixed-point iteration as numerical method to obtain the optimal density $\mathsf{N}_p(\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$. In the case of the horseshoe prior it is equivalent to the following updating scheme for $\boldsymbol{\mu}_{q(\beta)}$ and $\boldsymbol{\Sigma}_{q(\beta)}$:

$$
\begin{cases}
\boldsymbol{\nu}_{q(\beta)} & \leftarrow \sum_{i=1}^{n}\left\{\mathbf{x}_i y_i - \mathbf{x}_i \exp\left\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\}\right\} - \boldsymbol{\mu}_{q(\beta)}\boldsymbol{\mu}_{q(1/\lambda^2)}\mu_{q(1/\tau^2)} \\
\boldsymbol{\Sigma}_{q(\beta)} & \leftarrow \left[\sum_{i=1}^{n}\mathbf{x}_i \exp\left\{\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\}\mathbf{x}_i^{\mathsf{T}} + \boldsymbol{\mu}_{q(1/\lambda^2)}\mu_{q(1/\tau^2)}\right]^{-1} \\
\boldsymbol{\mu}_{q(\beta)} & \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)}\boldsymbol{\nu}_{q(\beta)},
\end{cases}
\tag{2.15}
$$

and at each iteration the convergence is assessed by checking the increase in the $\boldsymbol{\beta}$-localized component of lower bound $\log \underline{p}(\mathbf{y};q)$.

The MFVB scheme for the fit of the Poisson regression model with horseshoe prior is provided in Algorithm 7, with the convergence of all parameters assessed by checking the increase in the lower bound $\log \underline{p}(\mathbf{y};q)$.

Notice that the horseshoe prior does not group the variables into signals and nulls, but it induces sparsity in the regression coefficients allowing a subset of them to be heavily shrunk towards zero. An immediate consequence is that the posterior distribution of $\boldsymbol{\beta}|\mathbf{y}$ is non-sparse with probability one. Thus, in order to perform variable selection, we use the Signal Adaptive Variable Selector (SAVS) approach, introduced by Ray and Bhattacharya (2018). The main advantage of this approach is the full automation because there is not need to specify any tuning parameter. In particular, the SAVS algorithm (Algorithm 8) takes as input the design matrix $\mathbf{X}$ and the non-sparse point estimate $\boldsymbol{\mu}_{q(\beta)}$ obtained with the Algorithm 7, and returns a sparse estimate $\boldsymbol{\mu}^*_{q(\beta)}$, which we call SAVS estimator and that can be used for variable

Algorithm 7: Semi-parametric MFVB for Poisson regression model with horseshoe prior.

```
Initialize: $q^*(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})$, $q^*(\eta)$, $q^*(\tau^2)$, $q^*(\nu_1)$, $\ldots$,$q^*(\nu_p)$, $q^*(\lambda_1^2)$ ,$\ldots$,
            $q^*(\lambda_p^2)$, $\varepsilon_\beta$, $\varepsilon_{global}$
$a_{q(\eta)} \leftarrow 1$
$a_{q(\nu_1)} = \ldots = a_{q(\nu_p)} \leftarrow 1$
$a_{q(\tau^2)} \leftarrow \frac{p+1}{2}$
$a_{q(\lambda_1^2)} = \ldots = a_{q(\lambda_p^2)} \leftarrow 1$
while convergence not reached do
    while convergence not reached do
        $\boldsymbol{\nu}_{q(\beta)} \leftarrow \sum_{i=1}^n \left\{ \mathbf{x}_i y_i - \mathbf{x}_i \exp\left\{ \mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2} \mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)} \mathbf{x}_i \right\} \right\}$
                $-\boldsymbol{\mu}_{q(\beta)} \boldsymbol{\mu}_{q(1/\lambda^2)} \mu_{q(1/\tau^2)}$
        $\boldsymbol{\Sigma}_{q(\beta)} \leftarrow \left[ \sum_{i=1}^n \mathbf{x}_i \exp\left\{ \mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2} \mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)} \mathbf{x}_i \right\} \mathbf{x}_i^\intercal + \boldsymbol{\mu}_{q(1/\lambda^2)} \mu_{q(1/\tau^2)} \right]^{-1}$
        $\boldsymbol{\mu}_{q(\beta)} \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)} \boldsymbol{\nu}_{q(\beta)}$
        compute $\log \underline{p}(y; q)^{[\boldsymbol{\beta}](z)}$;
        evaluate $|\log \underline{p}(y; q)^{[\boldsymbol{\beta}](z)} - \log \underline{p}(y; q)^{[\boldsymbol{\beta}](z-1)}| < \varepsilon_\beta$;
    end
    $b_{q(\eta)} \leftarrow 1 + \mu_{q(1/\tau^2)}$
    $\mu_{q(1/\eta)} \leftarrow \frac{a_{q(\eta)}}{b_{q(\eta)}}$
    $\mu_{q(\log \eta)} \leftarrow \log b_{q(\eta)} - \psi(a_{q(\eta)})$
    for $j = 1, \ldots, p$ do
        $b_{q(\nu_j)} \leftarrow 1 + \mu_{q(1/\lambda_j^2)}$
        $\mu_{q(1/\nu_j)} \leftarrow \frac{a_{q(\nu_j)}}{b_{q(\nu_j)}}$
        $\mu_{q(\log \nu_j)} \leftarrow \log b_{q(\nu_j)} - \psi(a_{q(\nu_j)})$
    end
    $b_{q(\tau^2)} \leftarrow \mu_{q(1/\eta)} + \frac{1}{2} \sum_{j=1}^p (\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}) \mu_{q(1/\lambda_j^2)}$
    $\mu_{q(1/\tau^2)} \leftarrow \frac{a_{q(\tau^2)}}{b_{q(\tau^2)}}$
    $\mu_{q(\log \tau^2)} \leftarrow \log b_{q(\tau^2)} - \psi(a_{q(\tau^2)})$
    for $j = 1, \ldots, p$ do
        $b_{q(\lambda_j^2)} \leftarrow \mu_{q(1/\nu_j)} + \frac{1}{2} (\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}) \mu_{q(1/\tau^2)}$
        $\mu_{q(1/\lambda_j^2)} \leftarrow \frac{a_{q(\lambda_j^2)}}{b_{q(\lambda_j^2)}}$
        $\mu_{q(\log \lambda_j^2)} \leftarrow \log b_{q(\lambda_j^2)} - \psi(a_{q(\lambda_j^2)})$
    end
    compute $\log \underline{p}(\mathbf{y}; q)^{(iter)}$;
    evaluate $|\log \underline{p}(\mathbf{y}; q)^{(iter)} - \log \underline{p}(\mathbf{y}; q)^{(iter-1)}| < \varepsilon_{global}$;
end
```

Algorithm 8: SAVS Algorithm.

```
Input: Posterior mean $\boldsymbol{\mu}_{q(\beta)}$, design matrix $\mathbf{X}$
for $j = 1, \dots, p$ do
    $\lambda_j \leftarrow \frac{1}{|\mu_{q(\beta_j)}|^2}$;
    if $|\mu_{q(\beta_j)}| \cdot ||\mathbf{X}_j||^2 \leq \lambda_j$ then
        $\mu^*_{q(\beta_j)} \leftarrow 0$;
    else
        $\mu^*_{q(\beta_j)} \leftarrow sign(\mu_{q(\beta_j)})||\mathbf{X}_j||^{-2}(|\mu_{q(\beta_j)}| \cdot ||\mathbf{X}_j||^2 - \lambda_j)$;
end
Output: A sparse estimate $\boldsymbol{\mu}^*_{q(\beta)}$
```

selection. This estimator is obtained solving the optimization problem

$$\boldsymbol{\mu}^*_{q(\beta)} = \underset{\boldsymbol{\beta}}{\text{argmin}} \left\{ \frac{1}{2} ||\mathbf{X}\boldsymbol{\mu}_{q(\beta)} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \sum_{j=1}^{p} \lambda_j |\beta_j| \right\}, \tag{2.16}$$

with the parameter $\lambda_j \geq 0$ that control the amount of penalization for the j-th variable. Exploiting that the horseshoe prior aggressively shrinks the noise components towards zero and retain the larger signals, λ_j is set to $\lambda_j = \frac{1}{|\mu_{q(\beta_j)}|^2}$, so that the penalties for the variables are ranked in inverse-squared order of the intensity of the corresponding coefficient. Finally, the sparse estimate $\boldsymbol{\mu}^*_{q(\beta)}$ is obtained using the coordinate descent algorithm (Friedman et al., 2007), with initial value $\boldsymbol{\mu}_{q(\beta)}$ and stopping the algorithm at the first iteration.

2.2 Spike-and-slab prior

The spike-and-slab prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993, 1997) is widely used in Bayesian variable selection and it represents an alternative to the horseshoe prior. In this thesis we consider the continuos version of the spike-and-slab prior, which is a finite mixture of two zero-mean Gaussian distribution with different variances. The first element is the *spike* component, which has smaller variance, while the second element is the *slab* component, which has greater variance. The main idea is to assign the regression coefficients equal to zero to the *spike* component and to *slab* otherwise.

Bayesian model specification. Let $\boldsymbol{\gamma}$ be the p-dimensional vector where $\gamma_j = 1$ if the j-th covariate, x_j, is included in the regression model and $\gamma_j = 0$ otherwise, and let $\boldsymbol{\lambda}_1$ be the p-dimensional vector where λ_{1j} is the prior variance of the *slab* component for the j-th regression coefficient, β_j. We consider the following Poisson model with spike-and-slab prior:

$$\begin{aligned}
&Y_i|\boldsymbol{\beta} \sim \mathsf{Poi}(\exp\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\}), \quad i=1,...,n, \quad Y_i \perp Y_j \quad \forall i \neq j,\\
&\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda}_1 \sim \mathsf{N}_p(\mathbf{0}_p,\boldsymbol{\Sigma}_\beta), \quad \boldsymbol{\Sigma}_\beta = \mathsf{diag}\{b_1,b_2,\dots,b_p\},\\
&b_j = \lambda_0^{1-\gamma_j}\lambda_{1j}^{\gamma_j}, \quad 0 \leq \lambda_0 < \lambda_{1j}, \quad j=1,\dots,p\\
&\lambda_{1j} \sim \mathsf{InvGa}(r,\delta), \quad j=1,\dots,p\\
&\gamma_j|\theta \sim \mathsf{Ber}(\theta), \quad j=1,\dots,p\\
&\theta \sim \mathsf{Be}(a,b),
\end{aligned} \tag{2.17}$$

which is a hierarchical spike-and-slab prior for the vector of coefficients $\boldsymbol{\beta}$. In detail, the joint prior distribution of $\boldsymbol{\beta}$, conditionally to $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}_1$, is multivariate Gaussian of dimension p with mean vector equal to $\mathbf{0}_p$ and diagonal variance-covariance matrix $\boldsymbol{\Sigma}_\beta$. The identification of the mixture components through the vector of prior variances $(\lambda_0,\lambda_{1j}), j=1,\dots,p$, is allowed by the variance-covariance matrix structure in third and fourth equation of (2.17). In addition, an Inverse-Gamma distribution with shape parameter r and scale parameter δ is assumed for $\lambda_{1j}, j=1,\dots,p$, with r and d independent from j. In conclusion, conditionally to θ, a Bernoulli distribution with parameter θ is assumed for each γ_j, and a Beta distribution with hyper-parameters a and b is assumed for θ. This specification leads to $(\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\lambda_1},\theta)$ as set of parameters and (λ_0,r,δ,a,b) as vector of hyper-parameters, with the latter that should be chosen by the user in order to perform the estimation. In particular, in the literature it is common practice to set λ_0 in the spike distribution to zero (Brown et al., 2002; Panagiotelis and Smith, 2008). However we follow Ročková and George (2014) and we consider small and positive values for λ_0 in order to exclude unimportant nonzero effects.

The model specification in (2.17) leads to

$$\begin{aligned}
p(\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\lambda}_1,\theta;\mathbf{y}) = &\prod_{i=1}^n \frac{e^{\mathbf{x}_i^\intercal\boldsymbol{\beta}y_i-\exp\left\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\right\}}}{y_i!}\\
&\times (2\pi)^{-\frac{p}{2}}\prod_{j=1}^p\left\{\lambda_0^{1-\gamma_j}\lambda_{1j}^{\gamma_j}\right\}^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\boldsymbol{\beta}^\intercal\boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\beta}\right\}\\
&\times\prod_{j=1}^p\left\{\frac{\delta^r}{\Gamma(r)}\frac{\exp\{-\delta/\lambda_{1j}\}}{\lambda_{1j}^{r+1}}\right\}\prod_{j=1}^p\left\{\theta^{\gamma_j}(1-\theta)^{1-\gamma_j}\right\}\\
&\times\frac{\theta^{a-1}(1-\theta)^{\beta-1}}{B(a,b)}
\end{aligned} \tag{2.18}$$

as joint distribution of the data and parameters.

Mean field variational Bayes approach. Let the density $q(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta)$ be factorizable in the following way

$$q(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta) = q(\boldsymbol{\beta}) \prod_{j=1}^{p} q(\gamma_j) \prod_{j=1}^{p} q(\lambda_{1j}) q(\theta), \tag{2.19}$$

thus the optimal densities are given by

$$\begin{aligned}
q^*(\boldsymbol{\beta}) &\propto \exp\left\{\mathbb{E}_{-\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}, \\
q^*(\gamma_j) &\propto \exp\left\{\mathbb{E}_{-\gamma_j}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}, \quad j = 1, \ldots, p, \\
q^*(\lambda_{1j}) &\propto \exp\left\{\mathbb{E}_{-\lambda_{1j}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}, \quad j = 1, \ldots, p, \\
q^*(\theta) &\propto \exp\left\{\mathbb{E}_{-\theta}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}.
\end{aligned} \tag{2.20}$$

At this point, we can obtain the optimal variational densities, which are provided by the next propositions.

Proposition 2.8. *The optimal density for $\boldsymbol{\beta}$ is not a standard form.*

Proof. Since $q^*(\boldsymbol{\beta}) \propto \exp\left\{\mathbb{E}_{-\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}$,

$$\begin{aligned}
\log q^*(\boldsymbol{\beta}) &\propto \mathbb{E}_{-\boldsymbol{\beta}}\left[\sum_{i=1}^{n}\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\frac{\beta_j^2 b_j^{-1}}{2}\right] \\
&= \sum_{i=1}^{n}\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\left\{\frac{\beta_j^2}{2}\mathbb{E}_{-\beta_j}\left[\frac{1}{\lambda_0^{1-\gamma_j}\lambda_{1j}^{\gamma_j}}\right]\right\} \\
&= \sum_{i=1}^{n}\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\left\{\frac{\beta_j^2}{2}\mathbb{E}_{-\beta_j}\left[\frac{1}{(1-\gamma_j)\lambda_0 + \gamma_j\lambda_{1j}}\right]\right\} \\
&= \sum_{i=1}^{n}\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\left\{\frac{\beta_j^2}{2}\left\{\mathbb{E}_{-\beta_j}\left[\frac{(1-\gamma_j)}{\lambda_0}\right] + \mathbb{E}_{-\beta_j}\left[\frac{\gamma_j}{\lambda_{1j}}\right]\right\}\right\} \\
&= \sum_{i=1}^{n}\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\}\right\} - \sum_{j=1}^{p}\left\{\frac{\beta_j^2}{2}\left\{\frac{(1-\mu_{q(\gamma_j)})}{\lambda_0} + \mu_{q(\gamma_j)}\mu_{q(1/\lambda_{1j})}\right\}\right\}
\end{aligned}$$

and we do not recognise the kernel of any known distribution. □

Proposition 2.9. *The optimal density for γ_j is $q^*(\gamma_j) \sim \mathsf{Ber}\big(\mu_{q(\gamma_j)}\big)$, with*

$$\begin{aligned}
\log\left(\frac{\mu_{q(\gamma_j)}}{1-\mu_{q(\gamma_j)}}\right) = \mu_{q(\log\theta)} - \mu_{q(\log(1-\theta))} - \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2}\left(\mu_{q(1/\lambda_{1j})} - \frac{1}{\lambda_0}\right) \\
- \frac{1}{2}\left(\mu_{q(\log\lambda_{1j})} - \log\lambda_0\right).
\end{aligned}$$

Proof. Since $q^*(\gamma_j) \propto \exp\left\{\mathbb{E}_{-\gamma_j}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}$,

$$\begin{aligned}
\log q^*(\gamma_j) &\propto \mathbb{E}_{-\gamma_j}\Bigg[-\frac{1}{2}(1-\gamma_j)\log\lambda_0 - \frac{1}{2}\gamma_j\log\lambda_{1j} \\
&\qquad + \gamma_j\log\theta + (1-\gamma_j)\log(1-\theta) - \frac{1}{2}\boldsymbol{\beta}^\intercal\boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\beta}\Bigg] \\
&\propto -(1-\gamma_j)\frac{1}{2}\log\lambda_0 - \frac{1}{2}\gamma_j\mu_{q(\log\lambda_{1j})} \\
&\qquad + \gamma_j\mu_{q(\log\theta)} + (1-\gamma_j)\mu_{q(\log(1-\theta)} - \frac{1}{2}\mathbb{E}_{-\gamma_j}\left[\frac{\beta_j^2}{\lambda_0^{1-\gamma_j}\lambda_{1j}^{\gamma_j}}\right] \\
&= -(1-\gamma_j)\frac{1}{2}\log\lambda_0 - \gamma_j\frac{\mu_{q(\log\lambda_{1j})}}{2} + \gamma_j\mu_{q(\log\theta)} \\
&\qquad + (1-\gamma_j)\mu_{q(\log(1-\theta)} - (1-\gamma_j)\frac{\mu_{q(\beta_j^2)}}{2\lambda_0} - \gamma_j\frac{\mu_{q(\beta_j^2)}\mu_{q(1/\lambda_{1j})}}{2} \\
&= (1-\gamma_j)\left[\mu_{q(\log(1-\theta))} - \frac{1}{2}\log\lambda_0 - \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2\lambda_0}\right] \\
&\qquad + \gamma_j\left[\mu_{q(\log\theta)} - \frac{(\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)})\mu_{q(1/\lambda_{1j})}}{2} - \frac{1}{2}\mu_{q(\log\lambda_{1j})}\right] \\
&\propto \gamma_j\Bigg[\mu_{q(\log\theta)} - \mu_{q(\log(1-\theta))} - \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2}\left(\mu_{q(1/\lambda_{1j})} - \frac{1}{\lambda_0}\right) \\
&\qquad - \frac{1}{2}\left(\mu_{q(\log\lambda_{1j})} - \log\lambda_0\right)\Bigg].
\end{aligned}$$

Take the exponential and notice that it coincides with the kernel of a Bernoulli distribution with parameter as in Proposition 2.9. □

Proposition 2.10. *The optimal density for λ_{1j} is $q^*(\lambda_{1j}) \sim \mathsf{InvGa}\left(r_{q(\lambda_{1j})}, \delta_{q(\lambda_{1j})}\right)$, with*

$$r_{q(\lambda_{1j})} = r + \frac{\mu_{q(\gamma_j)}}{2}, \qquad \delta_{q(\lambda_{1j})} = \delta + \frac{\mu_{q(\gamma_j)}(\mu^2_{q(\beta_j)} + \sigma^2_{q(\beta_j)})}{2}. \tag{2.21}$$

Furthermore, $\mu_{q(1/\lambda_{1j})} = r_{q(\lambda_{1j})}/\delta_{q(\lambda_{1j})}$ *and* $\mu_{q(\log\lambda_{1j})} = \log\delta_{q(\lambda_{1j})} - \psi(r_{q(\lambda_{1j})})$.

Proof. Since $q^*(\lambda_{1j}) \propto \exp\left\{\mathbb{E}_{-\lambda_{1j}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}$,

$$\begin{aligned}
\log q^*(\lambda_{1j}) &\propto \mathbb{E}_{-\lambda_{1j}}\left[-\frac{1}{2}\gamma_j\log\lambda_{1j} - \frac{1}{2}\boldsymbol{\beta}^\intercal\boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\beta} - \frac{\delta}{\lambda_{1j}} - (r+1)\log\lambda_{1j}\right] \\
&\propto -\left[\frac{\mu_{q(\gamma_j)}}{2} + r + 1\right]\log\lambda_{1j} - \frac{1}{\lambda_{1j}}\left[\delta + \frac{\mu_{q(\gamma_j)}\mu_{q(\beta_j^2)}}{2}\right] \\
&= -\left[\frac{\mu_{q(\gamma_j)}}{2} + r + 1\right]\log\lambda_{1j} - \frac{1}{\lambda_{1j}}\left[\delta + \frac{\mu_{q(\gamma_j)}(\mu^2_{q(\beta_j)} + \sigma^2_{q(\beta_j)})}{2}\right].
\end{aligned}$$

Take the exponential and notice that it coincides with the kernel of an Inverse-Gamma distribution with parameters as in Proposition 2.10. □

Proposition 2.11. *The optimal density for θ is $q^*(\theta) \sim \mathsf{Be}\left(a_{q(\theta)}, b_{q(\theta)}\right)$, with*

$$a_{q(\theta)} = a + \sum_{j=1}^{p} \mu_{q(\gamma_j)}, \qquad b_{q(\theta)} = b + p - \sum_{j=1}^{p} \mu_{q(\gamma_j)}. \tag{2.22}$$

Furthermore,

$$\begin{aligned} \mu_{q(\log\theta)} &= \psi(a_{q(\theta)}) - \psi(a_{q(\theta)} + b_{q(\theta)}), \\ \mu_{q(\log(1-\theta))} &= \psi(b_{q(\theta)}) - \psi(a_{q(\theta)} + b_{q(\theta)}). \end{aligned} \tag{2.23}$$

Proof. Since $q^*(\theta) \propto \exp\left\{\mathbb{E}_{-\theta}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta; \mathbf{y})\right]\right\}$,

$$\begin{aligned} \log q^*(\theta) \propto \mathbb{E}_{-\theta}\Bigg[&\sum_{j=1}^{p} \gamma_j \log\theta + \left(p - \sum_{j=1}^{p}\gamma_j\right)\log(1-\theta) \\ &+ (a-1)\log\theta + (b-1)\log(1-\theta)\Bigg] \\ = \Bigg(&\sum_{j=1}^{p} \mu_{q(\gamma_j)} + a - 1\Bigg)\log\theta + \left(p - \sum_{j=1}^{p}\mu_{q(\gamma_j)} + b - 1\right)\log(1-\theta). \end{aligned}$$

Take the exponential and notice that it coincides with the kernel of a Beta distribution with parameters as in Proposition 2.11. □

Also in this case, we have that all the optimal variational densities in (2.20), except the first, are recognized to be well known distribution functions. Since $q^*(\boldsymbol{\beta})$ is not a standard form, we use a semi-parametric mean field variational Bayes approach in order to obtain the variational Bayes estimates and, as usual, we choose the multivariate normal, $\boldsymbol{\beta} \sim \mathsf{N}_p(\boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})$, which means:

$$q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)}) = (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}_{q(\beta)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)}\right)^{\intercal} \boldsymbol{\Sigma}_{q(\beta)}^{-1}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)}\right)\right\}. \tag{2.24}$$

As a consequence, the mean field variational approximation in (2.19) becomes

$$q(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \theta) = q(\boldsymbol{\beta}; \boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)}) \prod_{j=1}^{p} q(\gamma_j) \prod_{j=1}^{p} q(\lambda_{1j}) q(\theta), \tag{2.25}$$

and the factor graph corresponding to the model (2.17) with q-density product restriction in (2.25) is depicted in Figure 2.2.

Furthermore, the lower bound associated to the variational density factorized as in (2.25) has a closed form and it is provided by the next proposition.

Proposition 2.12. *The lower bound* $\log \underline{p}(\mathbf{y};q)$ *for the Poisson regression model with spike-and-slab prior in (2.17), and associated to the variational density factorized as* $q(\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\lambda}_1,\theta) = q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\prod_{j=1}^{p} q(\gamma_j)\prod_{j=1}^{p} q(\lambda_{1j})q(\theta)$*, given a vector of realizations of the dependent variable,* $\mathbf{y}$*, and design matrix,* $\mathbf{X}$*, can be expressed in a closed form:*

$$
\begin{aligned}
\log \underline{p}(y;q) = {}& \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2} - \sum_{j=1}^{p}\Bigl\{\mu_{q(\gamma_j)}\log\mu_{q(\gamma_j)} + (1-\mu_{q(\gamma_j)})\log(1-\mu_{q(\gamma_j)})\Bigr\} \\
& + \sum_{j=1}^{p}\Bigl\{ - r_{q(\lambda_{1j})}\log\delta_{q(\lambda_{1j})} + \log\Gamma(r_{q(\lambda_{1j})}) + \delta_{q(\lambda_{1j})}\mu_{q(1/\lambda_{1j})} \\
& + (r_{q(\lambda_{1j})}+1)\mu_{q(\log\lambda_{1j})}\Bigr\} - (a_{q(\theta)}-1)\mu_{q(\log\theta)} + (a-1)\mu_{q(\log\theta)} \\
& - (b_{q(\theta)}-1)\mu_{q(\log(1-\theta))} + \log B(a_{q(\theta)},b_{q(\theta)}) - \log B(a,b) + pr\log\delta \\
& + \sum_{i=1}^{n}\Bigl\{\mathbf{x}_i^{\intercal}\boldsymbol{\mu}_{q(\beta)}y_i - \exp\Bigl\{\mathbf{x}_i^{\intercal}\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^{\intercal}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\Bigr\} - \log(y_i!)\Bigr\} \\
& + \sum_{j=1}^{p}\Bigl\{ - (1-\mu_{q(\gamma_j)})\Bigl[\frac{1}{2}\log\lambda_0 + \frac{\mu^2_{q(\beta_j)}+\sigma^2_{q(\beta_j)}}{2\lambda_0}\Bigr] \\
& - \mu_{q(\gamma_j)}\Bigl[\frac{1}{2}\mu_{q(\log\lambda_{1j})} + \frac{(\mu^2_{q(\beta_j)}+\sigma^2_{q(\beta_j)})\mu_{q(1/\lambda_{1j})}}{2}\Bigr]\Bigr\} - p\log\Gamma(r) \\
& + \sum_{j=1}^{p}\Bigl\{(1-\mu_{q(\gamma_j)})\mu_{q(\log(1-\theta))} + \mu_{q(\gamma_j)}\mu_{q(\log\theta)}\Bigr\} + (b-1)\mu_{q(\log(1-\theta))} \\
& - \delta\sum_{j=1}^{p}\mu_{q(1/\lambda_{1j})} - (r+1)\sum_{j=1}^{p}\mu_{q(\log\lambda_{1j})}.
\end{aligned}
\tag{2.26}
$$

Proof. The lower bound can be expressed in terms of the components of the factor graph in Figure 2.2:

$$
\begin{aligned}
\log \underline{p}(y;q) = {}& \mathrm{H}\bigl\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\bigr\} + \sum_{j=1}^{p}\mathrm{H}\{q(\gamma_j)\} \\
& + \sum_{j=1}^{p}\mathrm{H}\{q(\lambda_{1j})\} + \mathrm{H}\{q(\theta)\} + \mathbb{E}_q\bigl[\log p(\mathbf{y}|\boldsymbol{\beta})\bigr] \\
& + \mathbb{E}_q\bigl[\log p(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda_1})\bigr] + \sum_{j=1}^{p}\mathbb{E}_q\bigl[\log p(\gamma_j|\theta)\bigr] + \mathbb{E}_q\bigl[\log p(\theta)\bigr] \\
& + \sum_{j=1}^{p}\mathbb{E}_q\bigl[\log p(\lambda_{1j})\bigr].
\end{aligned}
\tag{2.27}
$$

Moreover, the first term is

$$
\mathrm{H}\bigl\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\bigr\} = \mathbb{E}_q\bigl[-\log q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\bigr]
$$

$$
\begin{aligned}
&= \mathbb{E}_q\bigg[- \bigg[- \frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| \\
&\quad - \frac{1}{2}\Big((\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)})^{\intercal}\boldsymbol{\Sigma}^{-1}_{q(\beta)}(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\beta)})\Big)\bigg]\bigg] \\
&= \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2},
\end{aligned}
$$

the second term is

$$
\begin{aligned}
\sum_{j=1}^{p} \mathrm{H}\left\{q(\gamma_j)\right\} &= \sum_{j=1}^{p} \mathbb{E}_q\big[- \log q(\gamma_j)\big] \\
&= \sum_{j=1}^{p} \mathbb{E}_q\bigg[- \Big(\gamma_j \log \mu_{q(\gamma_j)} + (1-\gamma_j)\log(1-\mu_{q(\gamma_j)})\Big)\bigg] \\
&= -\sum_{j=1}^{p}\Big\{ \mu_{q(\gamma_j)}\log\mu_{q(\gamma_j)} + (1-\mu_{q(\gamma_j)})\log(1-\mu_{q(\gamma_j)})\Big\},
\end{aligned}
$$

the third term is

$$
\begin{aligned}
\sum_{j=1}^{p} \mathrm{H}\left\{q(\lambda_{1j})\right\} &= \sum_{j=1}^{p} \mathbb{E}_q\big[- \log q(\lambda_{ij})\big] \\
&= \sum_{j=1}^{p} \mathbb{E}_q\bigg[- \Big(r_{q(\lambda_{1j})}\log\delta_{q(\lambda_{1j})} - \log\Gamma(r_{q(\lambda_{1j})}) - \frac{\delta_{q(\lambda_{1j})}}{\lambda_{1j}} \\
&\quad - (r_{q(\lambda_{1j})}+1)\log\lambda_{1j}\Big)\bigg] \\
&= \sum_{j=1}^{p}\bigg\{ - r_{q(\lambda_{1j})}\log\delta_{q(\lambda_{1j})} + \log\Gamma(r_{q(\lambda_{1j})}) + \delta_{q(\lambda_{1j})}\mu_{q(1/\lambda_{1j})} \\
&\quad + (r_{q(\lambda_{1j})}+1)\mu_{q(\log\lambda_{1j})}\bigg\},
\end{aligned}
$$

the fourth term is

$$
\begin{aligned}
\mathrm{H}\left\{q(\theta)\right\} &= \mathbb{E}_q\big[- \log q(\theta)\big] \\
&= \mathbb{E}_q\bigg[- \Big((a_{q(\theta)}-1)\log\theta + (b_{q(\theta)}-1)\log(1-\theta) - \log B(a_{q(\theta)}, b_{q(\theta)})\Big)\bigg] \\
&= -(a_{q(\theta)}-1)\mu_{q(\log\theta)} - (b_{q(\theta)}-1)\mu_{q(\log(1-\theta))} + \log B(a_{q(\theta)}, b_{q(\theta)}),
\end{aligned}
$$

the fifth term is

$$
\begin{aligned}
\mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] &= \mathbb{E}_q\bigg[\sum_{i=1}^{n} \Big[\mathbf{x}_i^{\intercal}\boldsymbol{\beta} y_i - \exp\left\{\mathbf{x}_i^{\intercal}\boldsymbol{\beta}\right\} - \log(y_i!)\Big]\bigg] \\
&= \sum_{i=1}^{n}\bigg\{ \mathbf{x}_i^{\intercal}\boldsymbol{\mu}_{q(\beta)} y_i - \exp\Big\{ \mathbf{x}_i^{\intercal}\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^{\intercal}\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\Big\} - \log(y_i!)\bigg\},
\end{aligned}
$$

the sixth term is

$$\begin{aligned}\mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda_1})\big] &= \mathbb{E}_q\Bigg[-\frac{p}{2}\log 2\pi+\sum_{j=1}^{p}\Bigg\{-\frac{1}{2}(1-\gamma_j)\log\lambda_0\\ &\quad -\frac{1}{2}\gamma_j\log\lambda_{1j}-\frac{1}{2}\frac{\beta_j^2}{\lambda_0^{1-\gamma_j}\lambda_{1j}^{\gamma_j}}\Bigg\}\Bigg]\\ &= -\frac{p}{2}\log 2\pi+\sum_{j=1}^{p}\Bigg\{-(1-\mu_{q(\gamma_j)})\frac{1}{2}\log\lambda_0-\mu_{q(\gamma_j)}\frac{\mu_{q(\log\lambda_{1j})}}{2}\\ &\quad -\frac{\mu^2_{q(\beta_j)}+\sigma^2_{q(\beta_j)}}{2}\left[\frac{1-\mu_{q(\gamma_j)}}{\lambda_0}+\mu_{q(\gamma_j)}\mu_{q(1/\lambda_{1j})}\right]\Bigg\}\\ &= -\frac{p}{2}\log 2\pi+\sum_{j=1}^{p}\Bigg\{-(1-\mu_{q(\gamma_j)})\left[\frac{1}{2}\log\lambda_0+\frac{\mu^2_{q(\beta_j)}+\sigma^2_{q(\beta_j)}}{2\lambda_0}\right]\\ &\quad -\mu_{q(\gamma_j)}\left[\frac{\mu_{q(\log\lambda_{1j})}}{2}+\frac{(\mu^2_{q(\beta_j)}+\sigma^2_{q(\beta_j)})\mu_{q(1/\lambda_{1j})}}{2}\right]\Bigg\},\end{aligned}$$

the seventh term is

$$\begin{aligned}\sum_{j=1}^{p}\mathbb{E}_q\big[\log p(\gamma_j|\theta)\big] &= \sum_{j=1}^{p}\mathbb{E}_q\Big[(1-\gamma_j)\log(1-\theta)+\gamma_j\log\theta\Big]\\ &= \sum_{j=1}^{p}\Big\{(1-\mu_{q(\gamma_j)})\mu_{q(\log(1-\theta)}+\mu_{q(\gamma_j)}\mu_{q(\log\theta)}\Big\},\end{aligned}$$

the eighth term is

$$\begin{aligned}\mathbb{E}_q\big[\log p(\theta)\big] &= \mathbb{E}_q\Big[(a-1)\log\theta+(b-1)\log(1-\theta)-\log B(a,b)\Big]\\ &= (a-1)\mu_{q(\log\theta)}+(b-1)\mu_{q(\log(1-\theta))}-\log B(a,b),\end{aligned}$$

the ninth term is

$$\begin{aligned}\sum_{j=1}^{p}\mathbb{E}_q\big[\log p(\lambda_{1j})\big] &= \sum_{j=1}^{p}\mathbb{E}_q\left[r\log\delta-\log\Gamma(r)-\frac{\delta}{\lambda_{1j}}-(r+1)\log\lambda_{1j}\right]\\ &= \sum_{j=1}^{p}\Big\{r\log\delta-\log\Gamma(r)-\delta\mu_{q(1/\lambda_{1j})}-(r+1)\mu_{q(\log\lambda_{1j})}\Big\}\\ &= pr\log\delta-p\log\Gamma(r)-\delta\sum_{j=1}^{p}\mu_{q(1/\lambda_{1j})}-(r+1)\sum_{j=1}^{p}\mu_{q(\log\lambda_{1j})}.\end{aligned}$$

At this point, substituting in (2.27) and after the simplification of $\frac{p}{2}\log 2\pi$ in $\mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\}$ with $-\frac{p}{2}\log 2\pi$ in $\mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda_1})\big]$, we obtain the expression in Proposition 2.12. □

At this point, we update the optimal parameters $(\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$ within a coordinate ascent scheme through the maximization of the $\boldsymbol{\beta}$-localized component of lower bound

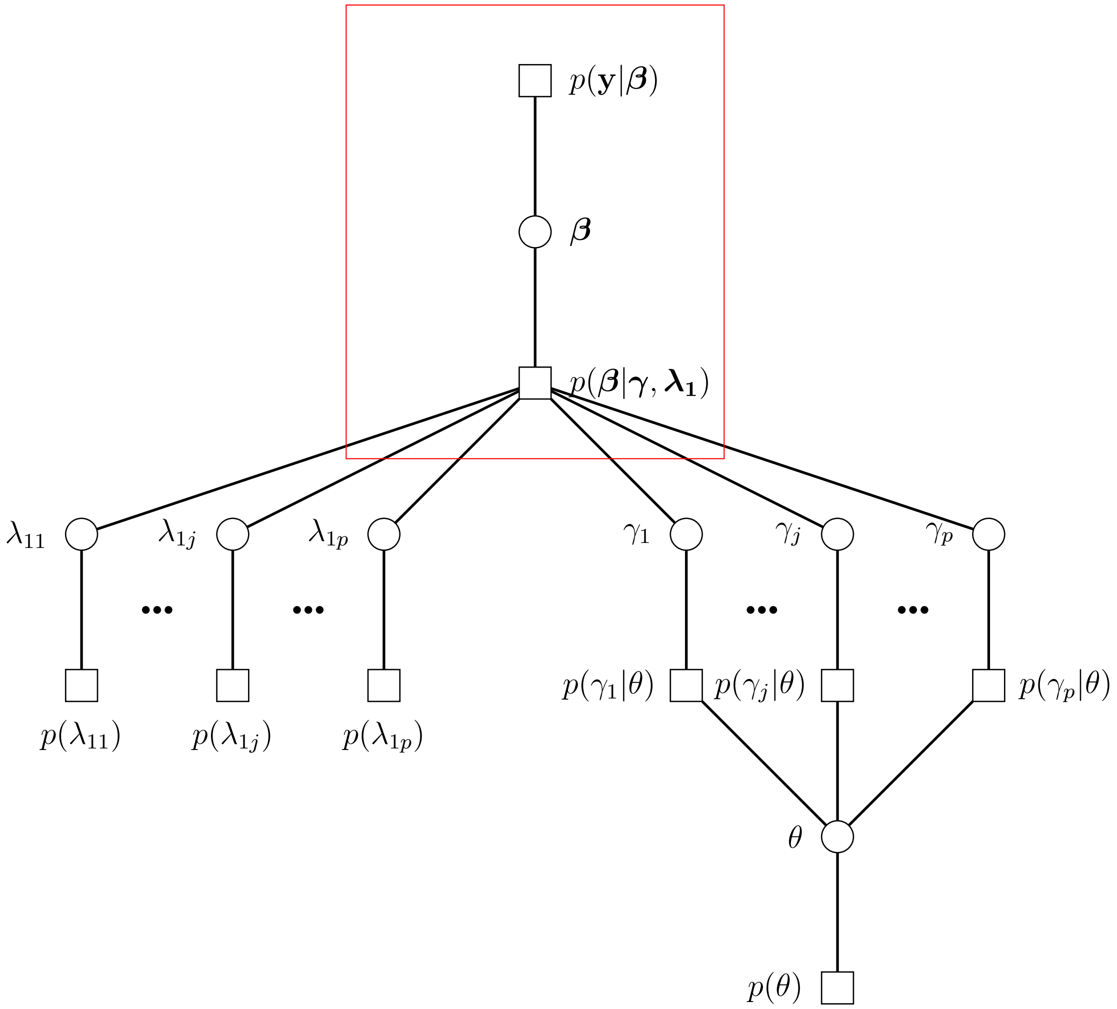


Figure 2.2: Factor graph for the model (2.17) with stochastic nodes corresponding to the mean field restriction (2.25).

$\log \underline{p}(\mathbf{y};q)$, $\log \underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]}$, over $(\boldsymbol{\mu}_{q(\beta)}, \boldsymbol{\Sigma}_{q(\beta)})$. The red line box in Figure 2.2 highlights the neighbours of $\boldsymbol{\beta}$, which are $p(\mathbf{y}|\boldsymbol{\beta})$ and $p(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda}_1)$ and they allow to have a closed form of $\log \underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]}$.

Proposition 2.13. *The $\boldsymbol{\beta}$-localized component of lower bound $\log \underline{p}(\mathbf{y};q)$ can be expressed in a closed form and it is equal to:*

$$\begin{aligned}\log \underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]} = \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2} + \sum_{i=1}^{n}\Bigg\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} y_i - \exp\Bigg\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\Bigg\} \\ - \log(y_i!)\Bigg\} + \sum_{j=1}^{p}\Bigg\{ - (1-\mu_{q(\gamma_j)})\left[\frac{1}{2}\log\lambda_0 + \frac{\mu^2_{q(\beta_j)} + \sigma^2_{q(\beta_j)}}{2\lambda_0}\right] \\ - \mu_{q(\gamma_j)}\left[\frac{\mu_{q(\log\lambda_{1j})}}{2} + \frac{(\mu^2_{q(\beta_j)} + \sigma^2_{q(\beta_j)})\mu_{q(1/\lambda_{1j})}}{2}\right]\Bigg\}.\end{aligned} \tag{2.28}$$

Proof. Since the $\boldsymbol{\beta}$-localized component of $\log \underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]}$ is equal to

$$\begin{aligned} \log \underline{p}(\mathbf{y};q)^{[\boldsymbol{\beta}]} &= \mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} + \overline{\mathrm{H}}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} \\ &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda_1})\big], \end{aligned} \tag{2.29}$$

the first term is

$$\begin{aligned} \mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[-\log q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\big] \\ &= \mathbb{E}_q\Bigg[-\bigg(-\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| \\ &\quad -\frac{1}{2}\Big((\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\beta)})^\intercal\boldsymbol{\Sigma}_{q(\beta)}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\beta)})\Big)\bigg)\Bigg] \\ &= \frac{p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Sigma}_{q(\beta)}| + \frac{p}{2}, \end{aligned}$$

while the second term is

$$\begin{aligned} \overline{\mathrm{H}}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\} &= \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta})\big] + \mathbb{E}_q\big[\log p(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\lambda_1})\big] \\ &= \mathbb{E}_q\Bigg[\sum_{i=1}^{n}\Big[\mathbf{x}_i^\intercal\boldsymbol{\beta}y_i - \exp\{\mathbf{x}_i^\intercal\boldsymbol{\beta}\} - \log(y_i!)\Big]\Bigg] \\ &\quad + \mathbb{E}_q\Bigg[-\frac{p}{2}\log 2\pi + \sum_{j=1}^{p}\bigg\{-\frac{1}{2}(1-\gamma_j)\log\lambda_0 \\ &\quad -\frac{1}{2}\gamma_j\log\lambda_{1j} - \frac{1}{2}\frac{\beta_j^2}{\lambda_0^{1-\gamma_j}\lambda_{1j}^{\gamma_j}}\bigg\}\Bigg] \\ &= \sum_{i=1}^{n}\bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)}y_i - \exp\bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\bigg\} \\ &\quad -\log(y_i!)\bigg\} - \frac{p}{2}\log 2\pi - \sum_{j=1}^{p}\bigg\{(1-\mu_{q(\gamma_j)})\frac{1}{2}\log\lambda_0 \\ &\quad +\mu_{q(\gamma_j)}\frac{\mu_{q(\log\lambda_{1j})}}{2} + \frac{\mu_{q(\beta_j)}^2+\sigma_{q(\beta_j)}^2}{2}\bigg[\frac{1-\mu_{q(\gamma_j)}}{\lambda_0} \\ &\quad +\mu_{q(\gamma_j)}\mu_{q(1/\lambda_{1j})}\bigg]\bigg\} \\ &= \sum_{i=1}^{n}\bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)}y_i - \exp\bigg\{\mathbf{x}_i^\intercal\boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal\boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\bigg\} \\ &\quad -\log(y_i!)\bigg\} - \frac{p}{2}\log 2\pi - \sum_{j=1}^{p}\bigg\{(1-\mu_{q(\gamma_j)}) \\ &\quad \times\bigg[\frac{1}{2}\log\lambda_0 + \frac{\mu_{q(\beta_j)}^2+\sigma_{q(\beta_j)}^2}{2\lambda_0}\bigg] + \mu_{q(\gamma_j)}\bigg[\frac{\mu_{q(\log\lambda_{1j})}}{2} \\ &\quad +\frac{(\mu_{q(\beta_j)}^2+\sigma_{q(\beta_j)}^2)\mu_{q(1/\lambda_{1j})}}{2}\bigg]\bigg\}. \end{aligned}$$

At this point, substituting in (2.29) and after the simplification of $\frac{p}{2}\log 2\pi$ in $\mathrm{H}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\}$ with $-\frac{p}{2}\log 2\pi$ in $\overline{\mathrm{H}}\left\{q(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})\right\}$, we obtain the expression in Proposition 2.13. □

In order to obtain the optimal density $\mathsf{N}_p(\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$, we use the natural fixed-point iteration update. In the case of the spike-and-slab prior this is equivalent to the following updating scheme for $\boldsymbol{\mu}_{q(\beta)}$ and $\boldsymbol{\Sigma}_{q(\beta)}$:

$$\begin{cases} \boldsymbol{\nu}_{q(\beta)} \leftarrow \sum_{i=1}^n \mathbf{x}_i y_i - \sum_{i=1}^n \mathbf{x}_i \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\} \\ \qquad\qquad -(1-\boldsymbol{\mu}_{q(\gamma)})\frac{\boldsymbol{\mu}_{q(\beta)}}{\lambda_0} - \boldsymbol{\mu}_{q(\gamma)}\left[\boldsymbol{\mu}_{q(\beta)}\boldsymbol{\mu}_{q(1/\lambda_1)}\right] \\ \boldsymbol{\Sigma}_{q(\beta)} \leftarrow \left\{ \sum_{i=i}^n \mathbf{x}_i \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\}\mathbf{x}_i^\intercal + \frac{(1-\boldsymbol{\mu}_{q(\gamma)})}{\lambda_0} + \boldsymbol{\mu}_{q(\gamma)}\boldsymbol{\mu}_{q(1/\lambda_1)}\right\}^{-1} \\ \boldsymbol{\mu}_{q(\beta)} \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)}\boldsymbol{\nu}_{q(\beta)}, \end{cases} \tag{2.30}$$

with the convergence assessed by checking the negligible increase in the $\boldsymbol{\beta}$-localized component of lower bound $\log \underline{p}(\mathbf{y};q)$ after each update.

The semi-parametric MFVB scheme that leads to the optimal variational densities $q^*(\boldsymbol{\beta})$, $q^*(\gamma_1)$, …, $q^*(\gamma_p)$, $q^*(\lambda_1)$, …, $q^*(\lambda_{1p})$, $q^*(\theta)$ is provided in Algorithm 9, with the convergence of all parameters assessed by checking the increase in the lower bound $\log \underline{p}(\mathbf{y};q)$.

Finally, in order to perform variable selection with spike-and-slab prior, we exclude the j-th variable from the model if the optimal $\mu_{q(\gamma_j)}$ is smaller than a threshold $s \in (0,1)$ and we include it otherwise.

2.3 Bernoulli-Gaussian prior

In Section 2.1 and 2.2, we considered respectively the Bayesian Poisson regression model with horseshoe and spike-and-slab prior, which share the feature of a hierarchical Bayesian model specification on the vector of coefficients $\boldsymbol{\beta}$. An other interesting alternative, that we consider in this Section, is represented by the Poisson regression model with Bernoulli-Gaussian (BG) prior (Ormerod et al., 2017; Bernardi et al., 2023). This is a model without hierarchical specification for the regression coefficients. Indeed, it is characterized by the introduction of a diagonal matrix in the linear predictor, with j-th element in the diagonal that brings information on the inclusion of j-th independent variable in the model.

Algorithm 9: Semi-parametric MFVB for Poisson regression model with spike-and-slab prior.

```
Initialize: $q^*(\boldsymbol{\beta};\boldsymbol{\mu}_{q(\beta)},\boldsymbol{\Sigma}_{q(\beta)})$, $q^*(\gamma_1)$, ..., $q^*(\gamma_p)$, $q^*(\lambda_{11})$, ..., $q^*(\lambda_{1p})$,$q^*(\theta)$,
            $\varepsilon_\beta$, $\varepsilon_{global}$
while convergence not reached do
    while convergence not reached do
        $\boldsymbol{\nu}_{q(\beta)} \leftarrow \sum_{i=1}^n \mathbf{x}_i y_i - \sum_{i=1}^n \mathbf{x}_i \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\}$
                $-(1-\boldsymbol{\mu}_{q(\gamma)})\frac{\boldsymbol{\mu}_{q(\beta)}}{\lambda_0} - \boldsymbol{\mu}_{q(\gamma)}\left[\boldsymbol{\mu}_{q(\beta)}\boldsymbol{\mu}_{q(1/\lambda_1)}\right]$
        $\boldsymbol{\Sigma}_{q(\beta)} \leftarrow \left\{ \sum_{i=i}^n \mathbf{x}_i \exp\left\{\mathbf{x}_i^\intercal \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2}\mathbf{x}_i^\intercal \boldsymbol{\Sigma}_{q(\beta)}\mathbf{x}_i\right\} \mathbf{x}_i^\intercal\right.$
                $\left. + \frac{(1-\boldsymbol{\mu}_{q(\gamma)})}{\lambda_0} + \boldsymbol{\mu}_{q(\gamma)}\boldsymbol{\mu}_{q(1/\lambda_1)}\right\}^{-1}$
        $\boldsymbol{\mu}_{q(\beta)} \leftarrow \boldsymbol{\mu}_{q(\beta)} + \boldsymbol{\Sigma}_{q(\beta)}\boldsymbol{\nu}_{q(\beta)}$
        compute $\log \underline{p}(y;q)^{[\boldsymbol{\beta}](z)}$;
        evaluate $|\log \underline{p}(y;q)^{[\boldsymbol{\beta}](z)} - \log \underline{p}(y;q)^{[\boldsymbol{\beta}](z-1)}| < \varepsilon_\beta$;
    end
    for $j = 1,\dots,p$ do
        $w_j \leftarrow \mu_{q(\log\theta)} - \mu_{q(\log(1-\theta))} - \frac{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}}{2}\left(\mu_{q(1/\lambda_{1j})} - \frac{1}{\lambda_0}\right)$
        $-\frac{1}{2}\left(\mu_{q(\log\lambda_{1j})} - \log\lambda_0\right)$
        $\mu_{q(\gamma_j)} \leftarrow \frac{1}{1+\exp(-w_j)}$
    end
    for $j = 1,\dots,p$ do
        $r_{q(\lambda_{1j})} \leftarrow r + \frac{\mu_{q(\gamma_j)}}{2}$
        $\delta_{q(\lambda_{1j})} \leftarrow \delta + \frac{\mu_{q(\gamma_j)}(\mu^2_{q(\beta_j)} + \sigma^2_{q(\beta_j)})}{2}$
        $\mu_{q(1/\lambda_{1j})} \leftarrow \frac{r_{q(\lambda_{1j})}}{\delta_{q(\lambda_{1j})}}$
        $\mu_{q(\log\lambda_{1j})} \leftarrow \log\delta_{q(\lambda_{1j})} - \psi(r_{q(\lambda_{1j})})$
    end
    $a_{q(\theta)} \leftarrow a + \sum_{j=1}^p \mu_{q(\gamma_j)}$
    $b_{q(\theta)} \leftarrow b + p - \sum_{j=1}^p \mu_{q(\gamma_j)}$
    $\mu_{q(\log\theta)} \leftarrow \psi(a_{q(\theta)}) - \psi(a_{q(\theta)} + b_{q(\theta)})$
    $\mu_{q(\log(1-\theta))} \leftarrow \psi(b_{q(\theta)}) - \psi(a_{q(\theta)} + b_{q(\theta)})$
    compute $\log \underline{p}(\mathbf{y};q)^{(iter)}$;
    evaluate $|\log \underline{p}(\mathbf{y};q)^{(iter)} - \log \underline{p}(\mathbf{y};q)^{(iter-1)}| < \varepsilon_{global}$;
end
```

Bayesian model specification. Let $\boldsymbol{\gamma}$ be the p-dimensional vector where $\gamma_j = 1$ if the j-th covariate is included in the regression model and $\gamma_j = 0$ otherwise. Differently from the models considered previously, we use a binary mask and we consider the following Bayesian Poisson regression model:

$$\begin{aligned} Y_i|\boldsymbol{\beta},\boldsymbol{\gamma} &\sim \mathsf{Poi}(\exp\{\mathbf{x}_i^\intercal \boldsymbol{\Gamma}\boldsymbol{\beta}\}), \quad i=1,...,n, \quad Y_i \perp Y_j \quad \forall i \neq j, \\ \boldsymbol{\beta} &\sim \mathsf{N}_p\left(\mathbf{0}_p, \sigma^2 \mathbf{I}_p\right), \\ \boldsymbol{\Gamma} &= \mathsf{diag}(\gamma_1,\dots,\gamma_p), \\ \gamma_j|\rho &\sim \mathsf{Ber}(\rho), \quad j=1,\dots,p, \quad \gamma_j \perp \gamma_k \; \forall j \neq k \\ \rho &\sim \mathsf{Beta}(\alpha,\delta). \end{aligned} \tag{2.31}$$

From the first equation in (2.31), it follows that, conditionally to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, in this model the likelihood is equal to

$$p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma}) = \prod_{i=1}^n \frac{e^{\mathbf{x}_i^\intercal \Gamma \boldsymbol{\beta} y_i - \exp\left\{\mathbf{x}_i^\intercal \Gamma \boldsymbol{\beta}\right\}}}{y_i!}. \tag{2.32}$$

As concerns the prior distribution of $\boldsymbol{\beta}$, this is multivariate Gaussian of dimension p with independent components, mean vector equal to $\mathbf{0}_p$ and variance equal to σ^2 for each β_j, as shown in (2.31). On the other hand, for the vector $\boldsymbol{\gamma}$, that allows to perform variable selection, we assume, conditionally to θ, a Bernoulli distribution with parameter θ for each γ_j. Finally, we assume a Beta distribution with hyper-parameters a and b for θ. The Bayesian model specification in (2.31) leads to $(\boldsymbol{\beta},\boldsymbol{\gamma},\theta)$ as set of parameters and (σ^2,α,δ) as set of hyper-parameters.

The joint distribution of the data and parameters rising from this model specification is the following:

$$\begin{aligned} p(\boldsymbol{\beta},\boldsymbol{\gamma},\rho;\mathbf{y}) = \prod_{i=1}^n & \frac{e^{\mathbf{x}_i^\intercal \Gamma \boldsymbol{\beta} y_i - \exp\left\{\mathbf{x}_i^\intercal \Gamma \boldsymbol{\beta}\right\}}}{y_i!} \\ & \times (2\pi)^{-\frac{p}{2}} (\sigma^2)^{-\frac{p}{2}} \exp\left\{-\frac{1}{2}\frac{\boldsymbol{\beta}^\intercal\boldsymbol{\beta}}{\sigma^2}\right\} \\ & \times \frac{\rho^{\alpha-1}(1-\rho)^{\delta-1}}{B(\alpha,\delta)} \\ & \times \rho^{\sum_{j=1}^p \gamma_j}(1-\rho)^{\sum_{j=1}^p (1-\gamma_j)}. \end{aligned} \tag{2.33}$$

Mean field variational Bayes approach. As usual, the first step with the mean field variational Bayes approach requires the definition of the factorization for the computation of the optimal variational densities. Although it seems natural to consider

$$q(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho) = q(\boldsymbol{\beta}) \prod_{j=1}^{p} q(\gamma_j) q(\rho), \tag{2.34}$$

this factorization leads to analytical issues in the computation of the optimal density $q(\boldsymbol{\beta})$. As a consequence, we relax the hypothesis of full posterior covariance matrix for $q(\boldsymbol{\beta})$ and we consider posterior independence between the regression coefficients. Thus, we work with the following mean field variational approximation:

$$q(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho) = \prod_{j=1}^{p} q(\beta_j) \prod_{j=1}^{p} q(\gamma_j) q(\rho), \tag{2.35}$$

which leads to the optimal variational densities

$$\begin{aligned} q^*(\beta_j) &\propto \exp\left\{ \mathbb{E}_{-\beta_j} \left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho; \mathbf{y}) \right] \right\}, \quad j = 1, \dots, p, \\ q^*(\gamma_j) &\propto \exp\left\{ \mathbb{E}_{-\gamma_j} \left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho; \mathbf{y}) \right] \right\}, \quad j = 1, \dots, p, \\ q^*(\rho) &\propto \exp\left\{ \mathbb{E}_{-\rho} \left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho; \mathbf{y}) \right] \right\}. \end{aligned} \tag{2.36}$$

The optimal densities we are looking for are given by the next propositions.

Proposition 2.14. *The optimal density for β_j, $q^*(\beta_j)$, $\ j = 1, \dots, p$, is not a standard form.*

Proof. Since $q^*(\beta_j) \propto \exp\left\{ \mathbb{E}_{-\beta_j} \left[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho; \mathbf{y}) \right] \right\}$,

$$\begin{aligned} \log q^*(\beta_j) &\propto \mathbb{E}_{-\beta_j} \left[-\frac{1}{2} \frac{\boldsymbol{\beta}^\intercal \boldsymbol{\beta}}{\sigma^2} + \sum_{i=1}^{n} \left\{ \mathbf{x}_i^\intercal \Gamma \boldsymbol{\beta} y_i - \exp\{\mathbf{x}_i^\intercal \Gamma \boldsymbol{\beta}\} \right\} \right] \\ &= -\frac{1}{2\sigma^2} \left[\beta_j^2 + \sum_{k \neq j} \left(\sigma^2_{q(\beta_k)} + \mu^2_{q(\beta_k)} \right) \right] \\ &\quad + \sum_{i=1}^{n} \left\{ \sum_{k \neq j} \left\{ x_{ik} \mu_{q(\gamma_k)} \mu_{q(\beta_k)} y_i \right\} + x_{ij} \mu_{q(\gamma_j)} \beta_j y_i - \prod_{k \neq j} \left[(1 - \mu_{q(\gamma_k)}) \right. \right. \\ &\quad \left. \left. + \mu_{q(\gamma_k)} \mu_{q(e^{x_{ik}\beta_k})} \right] \left[(1 - \mu_{q(\gamma_j)}) + \mu_{q(\gamma_j)} \exp\left\{ x_{ij} \beta_j \right\} \right] \right\} \\ &\propto -\frac{1}{2\sigma^2} \beta_j^2 + \sum_{i=1}^{n} \left\{ x_{ij} \mu_{q(\gamma_j)} \beta_j y_i - \prod_{k \neq j} \left[(1 - \mu_{q(\gamma_k)}) + \mu_{q(\gamma_k)} \mu_{q(e^{x_{ik}\beta_k})} \right] \right. \\ &\quad \left. \times \left[(1 - \mu_{q(\gamma_j)}) + \mu_{q(\gamma_j)} \exp\left\{ x_{ij} \beta_j \right\} \right] \right\}, \end{aligned}$$

and we do not recognise the kernel of any known distribution. □

Proposition 2.15. *The optimal density for γ_j, $q^*(\gamma_j)$, $\ j = 1, \dots, p$, is not a standard form.*

Proof. Since $q^*(\gamma_j) \propto \exp\Bigg\{\mathbb{E}_{-\gamma_j}\Bigg[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho; \mathbf{y})\Bigg]\Bigg\}$,

$$
\begin{aligned}
\log q^*(\gamma_j) \propto\ & \mathbb{E}_{-\gamma_j}\Bigg[\sum_{j=1}^{p}\Bigg\{\gamma_j \log\rho + (1-\gamma_j)\log(1-\rho)\Bigg\} \\
&\quad + \sum_{i=1}^{n}\Bigg\{\mathbf{x}_i^{\intercal}\Gamma\boldsymbol{\beta}y_i - \exp\left\{\mathbf{x}_i^{\intercal}\Gamma\boldsymbol{\beta}\right\}\Bigg\}\Bigg] \\
&= \gamma_j\mu_{q(\log\rho)} + (1-\gamma_j)\mu_{q(\log(1-\rho))} + \sum_{k\neq j}\Bigg\{\mu_{q(\gamma_k)}\mu_{q(\log\rho)} \\
&\quad + (1-\mu_{q(\gamma_k)})\mu_{q(\log(1-\rho))}\Bigg\} + \mathbb{E}_{-\gamma_j}\Bigg\{\sum_{i=1}^{n}\Bigg\{\mathbf{x}_i^{\intercal}\Gamma\boldsymbol{\beta}y_i - \exp\Bigg\{\mathbf{x}_i^{\intercal}\Gamma\boldsymbol{\beta}\Bigg\} - \log y_i!\Bigg\} \\
&\propto \gamma_j\Bigg[\mu_{q(\log\rho)} - \mu_{q(\log(1-\rho))}\Bigg] + \sum_{i=1}^{n}\Bigg\{x_{ij}\gamma_j\mu_{q(\beta_j)}y_i\Bigg\} - \sum_{i=1}^{n}\Bigg\{\prod_{k\neq j}\Bigg[(1-\mu_{q(\gamma_k)}) \\
&\quad + \mu_{q(\gamma_k)}\exp\Bigg\{x_{ik}\mu_{q(\beta_k)} + \frac{x_{ik}^2\sigma^2_{q(\beta_k)}}{2}\Bigg\}\Bigg]\Bigg[\exp\Bigg\{\mu_{q(\beta_j)}x_{ij}\gamma_j + \frac{\sigma^2_{q(\beta_j)}x_{ij}^2\gamma_j^2}{2}\Bigg\}\Bigg]\Bigg\} \\
&= \gamma_j\Bigg[\mu_{q(\log\rho)} - \mu_{q(\log(1-\rho))} + \sum_{i=1}^{n}\Bigg\{x_{ij}\mu_{q(\beta_j)}y_i\Bigg\}\Bigg] \\
&\quad - \sum_{i=1}^{n}\Bigg\{\prod_{k\neq j}\Bigg[(1-\mu_{q(\gamma_k)}) + \mu_{q(\gamma_k)}\exp\Bigg\{x_{ik}\mu_{q(\beta_k)} + \frac{x_{ik}^2\sigma^2_{q(\beta_k)}}{2}\Bigg\}\Bigg] \\
&\quad \times\Bigg[\exp\Bigg\{\mu_{q(\beta_j)}x_{ij}\gamma_j + \frac{\sigma^2_{q(\beta_j)}x_{ij}^2\gamma_j^2}{2}\Bigg\}\Bigg]\Bigg\},
\end{aligned}
$$

and we do not recognise the kernel of any known distribution. □

Proposition 2.16. *The optimal density for ρ is $q^*(\rho) \sim \mathsf{Beta}(\alpha_{q(\rho)}, \delta_{q(\rho)})$, with*

$$
\alpha_{q(\rho)} = \alpha + \sum_{j=1}^{p}\mu_{q(\gamma_j)}, \qquad \delta_{q(\rho)} = \delta + p - \sum_{j=1}^{p}\mu_{q(\gamma_j)}. \tag{2.37}
$$

Furthermore,

$$
\begin{aligned}
\mu_{q(\log\rho)} &= \psi(a_{q(\rho)}) - \psi(a_{q(\rho)} + \delta_{q(\rho)}), \\
\mu_{q(\log(1-\rho)))} &= \psi(\delta_{q(\rho)}) - \psi(a_{q(\rho)} + \delta_{q(\rho)}).
\end{aligned} \tag{2.38}
$$

Proof. Since $q^*(\rho) \propto \exp\Bigg\{\mathbb{E}_{-\rho}\Bigg[\log p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho; \mathbf{y})\Bigg]\Bigg\}$,

$$
\log q^*(\rho) \propto \mathbb{E}_{-\rho}\Bigg[(\alpha-1)\log\rho + (\delta-1)\log(1-\rho)
$$

$$
\begin{aligned}
&\quad + \sum_{j=1}^{p} \left\{ \gamma_j \log \rho + (1-\gamma_j) \log(1-\rho) \right\} \Bigg] \\
&= (\alpha - 1) \log \rho + (\delta - 1) \log(1-\rho) \\
&\quad + \sum_{j=1}^{p} \left\{ \mu_{q(\gamma_j)} \log \rho + (1 - \mu_{q(\gamma_j)}) \log(1-\rho) \right\} \\
&\propto \left[\alpha - 1 + \sum_{j=1}^{p} \mu_{q(\gamma_j)} \right] \log \rho + \left[\delta + p - \sum_{j=1}^{p} \mu_{q(\gamma_j)} - 1 \right] \log(1-\rho).
\end{aligned}
$$

Take the exponential and notice that it coincides with the kernel of a Beta distribution with parameters as in Proposition 2.16. □

After some computations, it turns out that only the optimal density $q^*(\theta)$ is same as well known distribution, while each $q^*(\beta_j)$ and $q^*(\gamma_j)$ are not. Thus, we use a semi-parametric mean field variational Bayes approach to obtain the variational Bayes densities. In particular, we use a non-parametric step for the update of $q^*(\theta)$, while we pre-specify a parametric family of density functions for $q(\beta_j)$ and $q(\gamma_j)$. We choose the univariate Normal density function for β_j, i.e. $\beta_j \sim N(\mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)})$,

$$
q(\beta_j; \mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)}) = \frac{1}{\sqrt{2\pi\sigma^2_{q(\beta_j)}}} \exp\left\{ - \frac{1}{2\sigma^2_{q(\beta_j)}} \left(\beta_j - \mu_{q(\beta_j)}\right)^2 \right\}, \tag{2.39}
$$

and the Bernoulli distribution for γ_j, i.e. $\gamma_j \sim Ber(\mu_{q(\gamma_j)})$,

$$
q(\gamma_j; \mu_{q(\gamma_j)}) = \mu_{q(\gamma_j)}^{\gamma_j} (1 - \mu_{q(\gamma_j)})^{1-\gamma_j}. \tag{2.40}
$$

As a consequence, the mean field variational approximation in (2.35) takes the following form:

$$
q(\boldsymbol{\beta}, \boldsymbol{\gamma}, \rho) = \prod_{j=1}^{p} q(\beta_j; \mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)}) \prod_{j=1}^{p} q(\gamma_j; \mu_{q(\gamma_j)}) q(\rho). \tag{2.41}
$$

Figure 2.3 shows the factor graph for the model in (2.31) with stochastic nodes corresponding to the mean field restriction in (2.41). In addition, we have a closed form of the lower bound.

Proposition 2.17. *The lower bound* $\log \underline{p}(\mathbf{y}; q)$ *for the Poisson regression model with Bernoulli-Gaussian prior in (2.31), and associated to the variational density factorized as* $q(\boldsymbol{\beta}, \gamma, \rho) = \prod_{j=1}^{p} q(\beta_j; \mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)}) \prod_{j=1}^{p} q(\gamma_j; \mu_{q(\gamma_j)}) q(\rho)$*, given a vector of realizations of the dependent variable,* $\mathbf{y}$*, and design matrix,* $\mathbf{X}$*, can be expressed in* *a closed form:*

$$\begin{aligned}\log \underline{p}(\mathbf{y};q) = {} & \frac{1}{2}\sum_{j=1}^{p}\log\sigma^2_{q(\beta_j)} + \frac{p}{2} - \sum_{j=1}^{p}\Big\{\mu_{q(\gamma_j)}\log\mu_{q(\gamma_j)} \\ & + (1-\mu_{q(\gamma_j)})\log(1-\mu_{q(\gamma_j)})\Big\} - (\alpha_{q(\rho)}-1)\mu_{q(\log\rho)} \\ & - (\delta_{q(\rho)}-1)\mu_{q(\log(1-\rho))} + \log B(\alpha_{q(\rho)},\delta_{q(\rho)}) \\ & + \sum_{i=1}^{n}\Bigg\{\sum_{j=1}^{p}x_{ij}\mu_{q(\gamma_j)}\mu_{q(\beta_j)}y_i - \prod_{j=1}^{p}\Bigg[1-\mu_{q(\gamma_j)} \\ & + \mu_{q(\gamma_j)}\exp\Bigg\{x_{ij}\mu_{q(\beta_j)} + \frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\Bigg\}\Bigg] - \log y_i!\Bigg\} \\ & - \frac{p}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{p}\Big\{\sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)}\Big\} \\ & + \sum_{j=1}^{p}\Big\{\mu_{q(\gamma_j)}\mu_{q(\log\rho)} + (1-\mu_{q(\gamma_j)})\mu_{q(\log(1-\rho))}\Big\} \\ & + (\alpha-1)\mu_{q(\log\rho)} + (\delta-1)\mu_{q(\log(1-\rho))} - \log B(\alpha,\delta).\end{aligned} \tag{2.42}$$

Proof. The lower bound can be expressed in terms of the components of the factor graph in Figure 2.3:

$$\begin{aligned}\log \underline{p}(\mathbf{y};q) = {} & \sum_{j=1}^{p}\mathrm{H}\Big\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\Big\} + \sum_{j=1}^{p}\mathrm{H}\left\{q(\gamma_j;\mu_{q(\gamma_j)})\right\} \\ & + \mathrm{H}\left\{q(\rho)\right\} + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta},\gamma)\big] + \sum_{j=1}^{p}\mathbb{E}_q\big[\log p(\beta_j)\big] \\ & + \sum_{j=1}^{p}\mathbb{E}_q\big[\log p(\gamma_j|\rho)\big] + \mathbb{E}_q\big[\log p(\rho)\big].\end{aligned} \tag{2.43}$$

Moreover, the first term is

$$\begin{aligned}\sum_{j=1}^{p}\mathrm{H}\Big\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\Big\} &= \sum_{j=1}^{p}\mathbb{E}_q\Big[-\log q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\Big] \\ &= \sum_{j=1}^{p}\mathbb{E}_q\Bigg[\frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2_{q(\beta_j)} \\ &\quad + \frac{1}{2\sigma^2_{q(\beta_j)}}(\beta_j-\mu_{q(\beta_j)})^2\Bigg] \\ &= \frac{p}{2}\log 2\pi + \frac{1}{2}\sum_{j=1}^{p}\log\sigma^2_{q(\beta_j)} + \frac{p}{2},\end{aligned}$$

the second term is

$$\sum_{j=1}^{p}\mathrm{H}\left\{q(\gamma_j;\mu_{q(\gamma_j)})\right\} = \sum_{j=1}^{p}\mathbb{E}_q\Big[-\log q(\gamma_j;\mu_{q(\gamma_j)})\Big]$$

$$
\begin{aligned}
&= \sum_{j=1}^{p} \mathbb{E}_q \Big[- \Big[\gamma_j \log \mu_{q(\gamma_j)} + (1-\gamma_j) \log(1-\mu_{q(\gamma_j)}) \Big] \Big] \\
&= - \sum_{j=1}^{p} \Big\{ \mu_{q(\gamma_j)} \log \mu_{q(\gamma_j)} + (1-\mu_{q(\gamma_j)}) \log(1-\mu_{q(\gamma_j)}) \Big\},
\end{aligned}
$$

the third term is

$$
\begin{aligned}
\mathrm{H}\{q(\rho)\} &= \mathbb{E}_q \Big[-\log q(\rho) \Big] \\
&= \mathbb{E}_q \Big[- \Big[(\alpha_{q(\rho)} - 1) \log \rho + (\delta_{q(\rho)} - 1) \log(1-\rho) - \log B(\alpha_{q(\rho)}, \delta_{q(\rho)}) \Big] \Big] \\
&= -(\alpha_{q(\rho)} - 1) \mu_{q(\log \rho)} - (\delta_{q(\rho)} - 1) \mu_{q(\log(1-\rho))} + \log B(\alpha_{q(\rho)}, \delta_{q(\rho)}),
\end{aligned}
$$

the fourth term is

$$
\begin{aligned}
\mathbb{E}_q \big[\log p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\gamma}) \big] &= \mathbb{E}_q \Big[\sum_{i=1}^{n} \Big[\mathbf{x}_i^{\intercal} \Gamma \boldsymbol{\beta} y_i - \exp\big\{ \mathbf{x}_i^{\intercal} \Gamma \boldsymbol{\beta} \big\} - \log y_i! \Big] \Big] \\
&= \sum_{i=1}^{n} \Big\{ \sum_{j=1}^{p} x_{ij} \mu_{q(\gamma_j)} \mu_{q(\beta_j)} y_i - \prod_{j=1}^{p} \Big[1 - \mu_{q(\gamma_j)} \\
&\quad + \mu_{q(\gamma_j)} \exp\Big\{ x_{ij} \mu_{q(\beta_j)} + \frac{\sigma^2_{q(\beta_j)} x_{ij}^2}{2} \Big\} \Big] - \log y_i! \Big\},
\end{aligned}
$$

the fifth term is

$$
\begin{aligned}
\sum_{j=1}^{p} \mathbb{E}_q \big[\log p(\beta_j) \big] &= \sum_{j=1}^{p} \mathbb{E}_q \Big[-\frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma^2 - \frac{\beta_j^2}{2\sigma^2} \Big] \\
&= \sum_{j=1}^{p} \Big\{ -\frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \mu_{q(\beta_j^2)} \Big\} \\
&= -\frac{p}{2} \log 2\pi - \frac{p}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{j=1}^{p} \Big\{ \sigma^2_{q(\beta_j)} + \mu^2_{q(\beta_j)} \Big\},
\end{aligned}
$$

the sixth term is

$$
\begin{aligned}
\sum_{j=1}^{p} \mathbb{E}_q \big[\log p(\gamma_j|\rho) \big] &= \sum_{j=1}^{p} \mathbb{E}_q \Big[\gamma_j \log \rho + (1-\gamma_j) \log(1-\rho) \Big] \\
&= \sum_{j=1}^{p} \Big\{ \mu_{q(\gamma_j)} \mu_{q(\log \rho)} + (1-\mu_{q(\gamma_j)}) \mu_{q(\log(1-\rho))} \Big\},
\end{aligned}
$$

the seventh term is

$$
\begin{aligned}
\mathbb{E}_q \Big[\log p(\rho) \Big] &= \mathbb{E}_q \Big[(\alpha - 1) \log \rho + (\delta - 1) \log(1-\rho) - \log B(\alpha, \delta) \Big] \\
&= (\alpha - 1) \mu_{q(\log \rho)} + (\delta - 1) \mu_{q(\log(1-\rho))} - \log B(\alpha, \delta).
\end{aligned}
$$

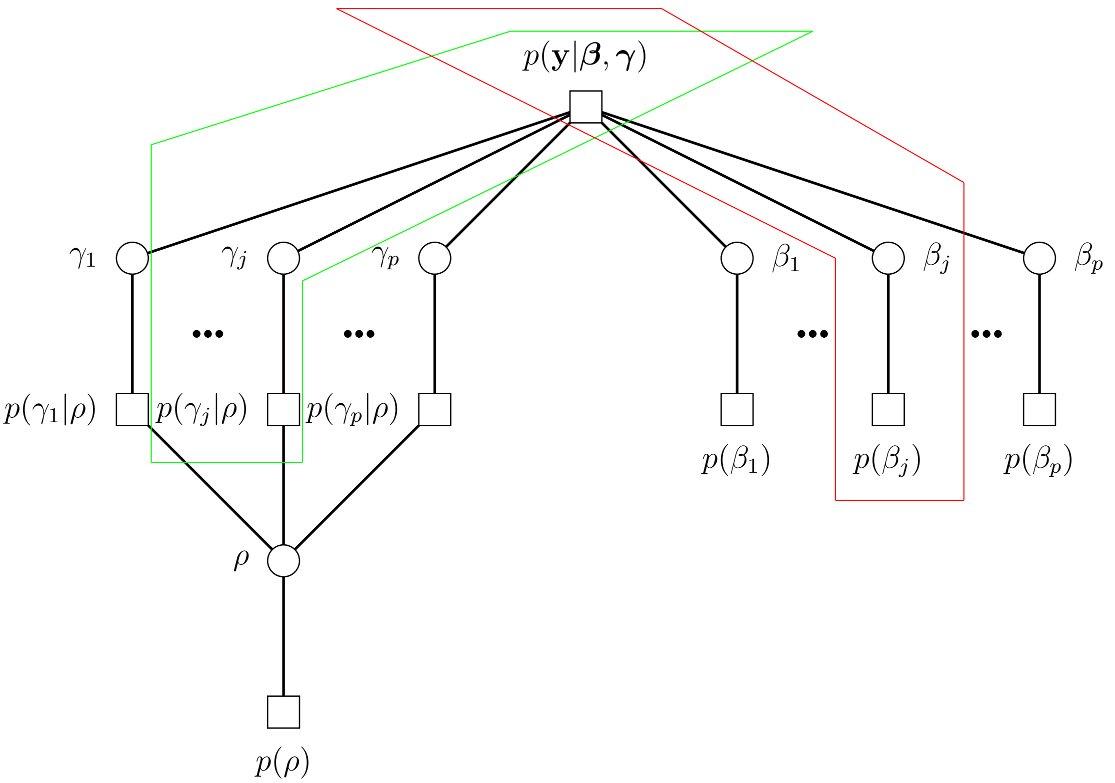


Figure 2.3: Factor graph for the model (2.31) with stochastic nodes corresponding to the mean field restriction (2.41).

At this point, substituting in (2.43) and after the simplification of $\frac{p}{2}\log 2\pi$ in $\sum_{j=1}^{p} \mathrm{H}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\}$ with $-\frac{p}{2}\log 2\pi$ in $\sum_{j=1}^{p} \mathbb{E}_q\left[\log p(\beta_j)\right]$, we obtain the expression in Proposition 2.17. □

At this point, we show how to update both the optimal parameters $(\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})$ of $q^*(\beta_j)$, and $\mu_{q(\gamma_j)}$ of $q(\gamma_j)$ within a coordinate ascent scheme. We maximize the β_j-localized and the γ_j-localized component of lower bound $\log \underline{p}(y;q)$, respectively. In particular, the red line box in Figure 2.3 shows that the neighbours of β_j are $p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma})$ and $p(\beta_j)$, while the green line box highlights the neighbours of γ_j, which are $p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma})$ and $p(\gamma_j|\rho)$. Thus we have the following expressions of $\log \underline{p}(y;q)^{[\beta_j]}$ and $\log \underline{p}(y;q)^{[\gamma_j]}$.

Proposition 2.18. *The β_j-localized component of lower bound $\log \underline{p}(\mathbf{y};q)$ can be ex-*

pressed in a closed form and it is equal to:

$$
\begin{aligned}
\log \underline{p}(y;q)^{[\beta_j]} = \frac{1}{2}\log\sigma^2_{q(\beta_j)} + \frac{1}{2} + \sum_{i=1}^{n}\Bigg\{ \sum_{j=1}^{p} x_{ij}\mu_{q(\gamma_j)}\mu_{q(\beta_j)}y_i - \\
\prod_{j=1}^{p}\left[1-\mu_{q(\gamma_j)}+\mu_{q(\gamma_j)}\exp\left\{x_{ij}\mu_{q(\beta_j)}+\frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\right\}\right] - \log y_i!\Bigg\} \qquad (2.44) \\
-\frac{1}{2}\log\sigma^2 - \frac{\sigma^2_{q(\beta_j)}+\mu^2_{q(\beta_j)}}{2\sigma^2}.
\end{aligned}
$$

Proof. Since the β_j-localized component of $\log \underline{p}(\mathbf{y};q)$ is equal to

$$
\begin{aligned}
\log \underline{p}(y;q)^{[\beta_j]} &= \mathrm{H}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\} + \overline{\mathrm{H}}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\} \\
&= \mathbb{E}_q\big[-\log q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\big] + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma})\big] + \mathbb{E}_q\big[\log p(\beta_j)\big],
\end{aligned} \qquad (2.45)
$$

the first term is

$$
\begin{aligned}
\mathrm{H}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\} &= \mathbb{E}_q\big[-\log q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\big] \\
&= \mathbb{E}_q\left[\frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2_{q(\beta_j)} + \frac{1}{2\sigma^2_{q(\beta_j)}}(\beta_j-\mu_{q(\beta_j)})^2\right] \\
&= \frac{1}{2}\log 2\pi + \frac{1}{2}\log\sigma^2_{q(\beta_j)} + \frac{1}{2},
\end{aligned}
$$

while the second term is

$$
\begin{aligned}
\overline{\mathrm{H}}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\} &= \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma})\big] + \mathbb{E}_q\big[\log p(\beta_j)\big] \\
&= \mathbb{E}_q\Bigg\{\sum_{i=1}^{n}\left[\mathbf{x}_i^{\intercal}\Gamma\boldsymbol{\beta}y_i - \exp\left\{\mathbf{x}_i^{\intercal}\Gamma\boldsymbol{\beta}\right\} - \log y_i!\right]\Bigg\} \\
&\quad + \mathbb{E}_q\Bigg\{-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\beta_j^2\Bigg\} \\
&= \sum_{i=1}^{n}\Bigg\{\sum_{j=1}^{p}x_{ij}\mu_{q(\gamma_j)}\mu_{q(\beta_j)}y_i - \prod_{j=1}^{p}\Bigg[1-\mu_{q(\gamma_j)} \\
&\quad + \mu_{q(\gamma_j)}\exp\left\{x_{ij}\mu_{q(\beta_j)}+\frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\right\}\Bigg] - \log y_i!\Bigg\} \\
&\quad - \frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma^2 - \frac{\sigma^2_{q(\beta_j)}+\mu^2_{q(\beta_j)}}{2\sigma^2}.
\end{aligned}
$$

At this point, substituting in (2.45) and after the simplification of $\frac{1}{2}\log 2\pi$ in $\mathrm{H}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\}$ with $-\frac{1}{2}\log 2\pi$ in $\overline{\mathrm{H}}\left\{q(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})\right\}$, we obtain the expression in Proposition 2.18. □

Proposition 2.19. *The γ_j-localized component of lower bound $\log \underline{p}(\mathbf{y};q)$ can be expressed in a closed form and it is equal to:*

$$\begin{aligned}\log \underline{p}(\mathbf{y};q)^{[\gamma_j]} = &-\Big[\mu_{q(\gamma_j)}\log\mu_{q(\gamma_j)} + (1-\mu_{q(\gamma_j)})\log(1-\mu_{q(\gamma_j)})\Big]\\ &+\sum_{i=1}^{n}\Bigg\{\sum_{j=1}^{p} x_{ij}\mu_{q(\gamma_j)}\mu_{q(\beta_j)}y_i - \prod_{j=1}^{p}\Bigg[1-\mu_{q(\gamma_j)}\\ &+\mu_{q(\gamma_j)}\exp\left\{x_{ij}\mu_{q(\beta_j)} + \frac{\sigma^2_{q(\beta_j)}x_{ij}^2}{2}\right\}\Bigg] - \log y_i!\Bigg\}\\ &+\sum_{j=1}^{p}\Big\{\mu_{q(\gamma_j)}\mu_{q(\log\rho)} + (1-\mu_{q(\gamma_j)})\mu_{q(\log(1-\rho))}\Big\}.\end{aligned} \tag{2.46}$$

Proof. Since the γ_j-localized component of $\log \underline{p}(\mathbf{y};q)$ is equal to

$$\begin{aligned}\log \underline{p}(\mathbf{y};q)^{[\gamma_j]} &= \mathrm{H}\left\{q(\gamma_j;\mu_{q(\gamma_j)})\right\} + \overline{\mathrm{H}}\left\{q(\gamma_j;\mu_{q(\gamma_j)})\right\}\\ &= \mathbb{E}_q\big[-\log q(\gamma_j;\mu_{q(\gamma_j)})\big] + \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma})\big] + \mathbb{E}_q\big[\log p(\gamma_j|\rho)\big],\end{aligned} \tag{2.47}$$

the first term is

$$\begin{aligned}\mathrm{H}\left\{q(\gamma_j;\mu_{q(\gamma_j)})\right\} &= \mathbb{E}_q\big[-\log q(\gamma_j;\mu_{q(\gamma_j)})\big]\\ &= \mathbb{E}_q\Bigg\{-\Big[\gamma_j\log\mu_{q(\gamma_j)} + (1-\gamma_j)\log(1-\mu_{q(\gamma_j)})\Big]\Bigg\}\\ &= -\Big[\mu_{q(\gamma_j)}\log\mu_{q(\gamma_j)} + (1-\mu_{q(\gamma_j)})\log(1-\mu_{q(\gamma_j)})\Big],\end{aligned}$$

while the second term is

$$\begin{aligned}\overline{\mathrm{H}}\left\{q(\gamma_j;\mu_{q(\gamma_j)})\right\} &= \mathbb{E}_q\big[\log p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\gamma})\big] + \mathbb{E}_q\big[\log p(\gamma_j|\rho)\big]\\ &= \mathbb{E}_q\Bigg[\sum_{i=1}^{n}\Big[\mathbf{x}_i^\intercal\Gamma\boldsymbol{\beta}y_i - \exp\left\{\mathbf{x}_i^\intercal\Gamma\boldsymbol{\beta}\right\} - \log y_i!\Big]\Bigg]\\ &\quad+\mathbb{E}_q\Big[\gamma_j\log\rho + (1-\gamma_j)\log(1-\rho)\Big]\\ &= \sum_{i=1}^{n}\Bigg\{\sum_{j=1}^{p} x_{ij}\mu_{q(\gamma_j)}\mu_{q(\beta_j)}y_i - \prod_{j=1}^{p}\Bigg[1-\mu_{q(\gamma_j)}\\ &\quad+\mu_{q(\gamma_j)}\exp\left\{x_{ij}\mu_{q(\beta_j)} + \frac{\sigma^2_{q(\beta_j)}x_{ij}^2}{2}\right\}\Bigg] - \log y_i!\Bigg\}\\ &\quad+\sum_{j=1}^{p}\Big\{\mu_{q(\gamma_j)}\mu_{q(\log\rho)} + (1-\mu_{q(\gamma_j)})\mu_{q(\log(1-\rho))}\Big\}.\end{aligned}$$

Substituting in (2.47), we obtain the expression in Proposition 2.19. □

We use the natural fixed-point iteration for the update of $(\mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)})$ and in the case of the Poisson model with Bernoulli-Gaussian prior it corresponds to the following updating scheme:

$$\left\{\begin{aligned}
&\nu_{q(\beta_j)} \leftarrow -\frac{\mu_{q(\beta_j)}}{\sigma^2} + \sum_{i=1}^{n}\Bigg\{x_{ij}\mu_{q(\gamma_j)}y_i - \prod_{k\neq j}\Bigg\{1-\mu_{q(\gamma_k)}+\mu_{q(\gamma_k)}\exp\Bigg\{x_{ik}\mu_{q(\beta_k)} \\
&\qquad\qquad +\frac{\sigma^2_{q(\beta_k)}x^2_{ik}}{2}\Bigg\}\Bigg\}x_{ij}\mu_{q(\gamma_j)}\exp\Bigg\{x_{ij}\mu_{q(\beta_j)}+\frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\Bigg\}\Bigg\} \\
&\sigma^2_{q(\beta_j)} \leftarrow -\Bigg\{-\frac{1}{\sigma^2}+\sum_{i=1}^{n}\Bigg\{-\prod_{k\neq j}\Bigg\{1-\mu_{q(\gamma_k)}+\mu_{q(\gamma_k)}\exp\Bigg\{x_{ik}\mu_{q(\beta_k)}+\frac{\sigma^2_{q(\beta_k)}x^2_{ik}}{2}\Bigg\}\Bigg\} \\
&\qquad\qquad \times x^2_{ij}\mu_{q(\gamma_j)}\exp\Bigg\{x_{ij}\mu_{q(\beta_j)}+\frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\Bigg\}\Bigg\}\Bigg\}^{-1} \\
&\mu_{q(\beta_j)} \leftarrow \mu_{q(\beta_j)} + \sigma^2_{q(\beta_j)}\nu_{q(\beta_j)}.
\end{aligned}\right. \tag{2.48}$$

On the other hand, for the optimal densities $q^*(\mu_{q(\gamma_j)})$, we maximize $\log \underline{p}(\mathbf{y};q)^{[\gamma_j]}$ using the L-BFGS-B algorithm. This is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) using a limited amount of computer memory and allows box constraints. At this point, it is straightforward to implement the correspondent semiparametric MFVB iterative procedure, as shown in Algorithm 10.

In the Poisson model with Bernoulli-Gaussian prior, similarly to the Poisson model with spike-and-slab prior, we include the j-th regressor if the optimal $\mu_{q(\gamma_j)}$ is greater or equal to a threshold $s \in (0,1)$ and we exclude it otherwise. On the other hand, as concerns the point estimates, in this model the effect of the j-th covariate on the mean of the response variable is not given by β_j but it is equal to $\tilde{\beta}_j = \beta_j\gamma_j$. Thus, the final step is to establish the optimal variational density of $\tilde{\beta}_j$, $q^*(\tilde{\beta}_j)$,which is provided by the next proposition.

Proposition 2.20. *Let $q^*(\beta_j;\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)})$ and $q^*(\gamma_j;\mu_{q(\gamma_j)})$ be the optimal variational densities of β_j and γ_j, respectively. Define $\tilde{\beta}_j = \gamma_j\beta_j$. The optimal variational density of $\tilde{\beta}_j$ is given by a mixture of an univariate Normal distribution and a Dirac in* 0*:*

$$q^*(\tilde{\beta}_j) = \mu_{q(\gamma_j)}\mathsf{N}(\mu_{q(\beta_j)},\sigma^2_{q(\beta_j)}) + (1-\mu_{q(\gamma_j)})\delta_0. \tag{2.49}$$

Moreover, mean and variance can be computed analytically and they are equal to:

$$\begin{aligned}
\mu_{q(\tilde{\beta}_j)} &= \mu_{q(\gamma_j)}\mu_{q(\beta_j)}, \\
\sigma^2_{q(\tilde{\beta}_j)} &= \mu_{q(\gamma_j)}(1-\mu_{q(\gamma_j)})\mu_{q(\beta_j)} + \sigma^2_{q(\beta_j)}\mu_{q(\gamma_j)}.
\end{aligned} \tag{2.50}$$

Proof. Consider the following transformation of random variables $(\gamma_j = \gamma_j, \tilde{\beta}_j =$

Algorithm 10: Semi-parametric MFVB for Poisson regression model with Bernoulli-Gaussian prior.

```
Initialize: $q^*(\beta_1;\mu_{q(\beta_1)},\sigma^2_{q(\beta_1)})$, ...,$q^*(\beta_p;\mu_{q(\beta_p)},\sigma^2_{q(\beta_p)})$, $q^*(\gamma_1;\mu_{q(\gamma_1)})$, ..., $q^*(\gamma_p;\mu_{q(\gamma_p)})$, $q^*(\rho)$, $\varepsilon_{\beta_1}, \ldots, \varepsilon_{\beta_p}, \varepsilon_{global}$
while convergence not reached do
    for $j = 1, \ldots, p$ do
        while convergence not reached do
            $\nu_{q(\beta_j)} \leftarrow -\frac{\mu_{q(\beta_j)}}{\sigma^2} + \sum_{i=1}^{n}\Big\{x_{ij}\mu_{q(\gamma_j)}y_i - \prod_{k\neq j}\Big\{1-\mu_{q(\gamma_k)}+\mu_{q(\gamma_k)} \times \exp\Big\{x_{ik}\mu_{q(\beta_k)}+\frac{\sigma^2_{q(\beta_k)}x^2_{ik}}{2}\Big\}\Big\}x_{ij}\mu_{q(\gamma_j)}\exp\Big\{x_{ij}\mu_{q(\beta_j)} + \frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\Big\}\Big\}$
            $\sigma^2_{q(\beta_j)} \leftarrow -\Big\{-\frac{1}{\sigma^2}+\sum_{i=1}^{n}\Big\{-\prod_{k\neq j}\Big\{1-\mu_{q(\gamma_k)}+\mu_{q(\gamma_k)}\exp\Big\{x_{ik}\mu_{q(\beta_k)} + \frac{\sigma^2_{q(\beta_k)}x^2_{ik}}{2}\Big\}\Big\}x^2_{ij}\mu_{q(\gamma_j)}\exp\Big\{x_{ij}\mu_{q(\beta_j)}+\frac{\sigma^2_{q(\beta_j)}x^2_{ij}}{2}\Big\}\Big\}\Big\}^{-1}$
            $\mu_{q(\beta_j)} \leftarrow \mu_{q(\beta_j)} + \sigma^2_{q(\beta_j)}\nu_{q(\beta_j)}$
            compute $\log \underline{p}(y;q)^{[\beta_j](z)}$;
            evaluate $|\log \underline{p}(y;q)^{[\beta_j](z)} - \log \underline{p}(y;q)^{[\beta_j](z-1)}| < \varepsilon_{\beta_j}$;
        end
    end
    for $j = 1, \ldots, p$ do
        $\mu_{q(\gamma_j)} \leftarrow$ optimization of $\log \underline{p}(\mathbf{y};q)^{[\gamma_j]}$ with L-BFGS-B algorithm and
          initial value the optimal $\mu_{q(\gamma_j)}$ in the previous iteration
    end
    $\alpha_{q(\rho)} \leftarrow \alpha + \sum_{j=1}^{p}\mu_{q(\gamma_j)}$
    $\delta_{q(\rho)} \leftarrow \delta + p - \sum_{j=1}^{p}\mu_{q(\gamma_j)}$
    $\mu_{q(\log\rho)} \leftarrow \psi(\alpha_{q(\rho)}) - \psi(\alpha_{q(\rho)}+\delta_{q(\rho)})$
    $\mu_{q(\log(1-\rho))} \leftarrow \psi(\delta_{q(\rho)}) - \psi(\alpha_{q(\rho)}+\delta_{q(\rho)})$
    compute $\log \underline{p}(\mathbf{y};q)^{(iter)}$;
    evaluate $|\log \underline{p}(\mathbf{y};q)^{(iter)} - \log \underline{p}(\mathbf{y};q)^{(iter-1)}| < \varepsilon_{global}$;
end
```

$\gamma_j\beta_j)$, so that $\beta_j = \gamma_j^{-1}\tilde{\beta}_j$. Hence it follows that:

$$\mathbf{J} = \begin{pmatrix} \dfrac{d}{d\gamma_j}\gamma_j & \dfrac{d}{d\tilde{\beta}_j}\gamma_j \\ \dfrac{d}{d\gamma_j}\gamma_j^{-1}\tilde{\beta}_j & \dfrac{d}{d\tilde{\beta}_j}\gamma_j^{-1}\tilde{\beta}_j \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\dfrac{\tilde{\beta}_j}{\gamma_j^2} & \gamma_j^{-1} \end{pmatrix} \tag{2.51}$$

and so $|\mathbf{J}| = \gamma_j^{-1}$. The joint distribution of $(\tilde{\beta}_j, \gamma_j)$ can be written as:

$$q(\tilde{\beta}_j, \gamma_j) = \gamma_j^{-1} q(\gamma_j^{-1}\tilde{\beta}_j) q(\gamma_j) = f(\tilde{\beta}_j|\gamma_j) q(\gamma_j), \tag{2.52}$$

where q are then replaced by the optimal densities q^*. For the conditional distribution in (2.52), we have that:

$$\begin{aligned} f(\tilde{\beta}_j|\gamma_j) &= \gamma_j^{-1}\phi(\mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)}) \\ &= \frac{1}{\sqrt{2\pi\gamma_j^2\sigma^2_{q(\beta_j)}}} \exp\left\{-\frac{1}{2\sigma^2_{q(\beta_j)}}(\gamma_j^{-1}\tilde{\beta}_j - \mu_{q(\beta_j)})^2\right\} \\ &= \frac{1}{\sqrt{2\pi\gamma_j^2\sigma^2_{q(\beta_j)}}} \exp\left\{-\frac{1}{2\sigma^2_{q(\beta_j)}}\left(\gamma_j^{-2}\tilde{\beta}_j^{\,2} + \mu^2_{q(\beta_j)} - 2\gamma_j^{-1}\tilde{\beta}_j\mu_{q(\beta_j)}\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left[\tilde{\beta}_j^{\,2}\left(\frac{1}{\sigma^2_{q(\beta_j)}\gamma_j^2}\right) - 2\tilde{\beta}_j\left(\frac{\mu_{q(\beta_j)}}{\sigma^2_{q(\beta_j)}\gamma_j}\right)\right]\right\}, \end{aligned}$$

and, exploiting that γ_j is equal to 0 or 1, this is the kernel of an univariate Normal distribution

$$\tilde{\beta}_j|\gamma_j \sim \mathsf{N}(\mu(\gamma_j), \sigma^2(\gamma_j)),$$

with

$$\begin{aligned} \mu(\gamma_j) &= \gamma_j\mu_{q(\beta_j)}, \\ \sigma^2(\gamma_j) &= \gamma_j^2\sigma^2_{q(\beta_j)}. \end{aligned}$$

At this point, the marginal distribution for $\tilde{\beta}_j$ can be found as:

$$\begin{aligned} q(\tilde{\beta}_j) &= f(\tilde{\beta}_j|\gamma_j = 1)q(\gamma_j = 1) + f(\tilde{\beta}_j|\gamma_j = 0)q(\gamma_j = 0) \\ &= \mu_{q(\gamma_j)}\mathsf{N}(\mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)}) + (1 - \mu_{q(\gamma_j)})\mathsf{N}(0, 0) \\ &= \mu_{q(\gamma_j)}\mathsf{N}(\mu_{q(\beta_j)}, \sigma^2_{q(\beta_j)}) + (1 - \mu_{q(\gamma_j)})\delta_0, \end{aligned}$$

and the distributional result concerning $\tilde{\beta}_j$ is therefore proven.

Now, in order to compute the marginal mean and variance recall that $\mathbb{E}_x(x) = \mathbb{E}_y\left[\mathbb{E}_x(x|y)\right]$ and $\text{Var}_x(x) = \text{Var}_y\left[\mathbb{E}_x(x|y)\right] + \mathbb{E}_y\left[\text{Var}_x(x|y)\right]$. Thus we have

$$\begin{aligned}
\mathbb{E}\left(\tilde{\beta}_j\right) &= \mathbb{E}\left[\mathbb{E}(\tilde{\beta}_j|\gamma_j)\right] = \mathbb{E}\left[\gamma_j\mu_{q(\beta_j)}\right] = \mu_{q(\gamma_j)}\mu_{q(\beta_j)}, \\
\text{Var}(\tilde{\beta}_j) &= \text{Var}\left[\mathbb{E}(\tilde{\beta}_j|\gamma_j)\right] + \mathbb{E}\left[\text{Var}(\tilde{\beta}_j|\gamma_j)\right] \\
&= \text{Var}\left[\gamma_j\mu_{q(\beta_j)}\right] + \mathbb{E}\left[\sigma^2_{q(\beta_j)}\gamma_j\right] \\
&= \mu_{q(\gamma_j)}(1-\mu_{q(\gamma_j)})\mu_{q(\beta_j)} + \sigma^2_{q(\beta_j)}\mu_{q(\gamma_j)},
\end{aligned}$$

which concludes the proof. □

2.4 Illustrative example

In order to show how these models work, a simulated dataset with $n = 500$ observations and $p = 6$ covariates from a standard univariate Normal distribution has been considered. Concerning the vector $\boldsymbol{\beta}$, we set $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)^\intercal = (-1, -1, 0, 0, 1, 1)^\intercal$.

Reminding that an advantage of the Poisson regression model with horseshoe prior is that does not require the choice of the hyperparameters by the user, Table 2.1 show hyperparameters' setting for the spike-and-slab prior. In particular, we consider two different values for the variance of the spike component, $\lambda_0 = 0.01$ and $\lambda_0 = 0.001$, in order to evaluate a possible sensitivity of the model to the value of λ_0. Furthermore, since any existing work deals with an Inverse-Gamma distribution for the variance of the slab component, we propose two possible combinations for the values of r and δ. Keeping in mind that $\lambda_{1j} \geq \lambda_0$ for $j = 1, \dots, p$, the first (that we call VSS) is the arbitrary choice of the values $r = 2502$ and $\delta = 125050$ in order to have $\mathbb{E}\left[\lambda_{1j}\right] = 50$ and $\text{Var}\left[\lambda_{1j}\right] = 1$ for each j. In the second (that we call VSS$_{NH}$), we fix r and δ in order to have $\text{Var}\left[\lambda_{1j}\right] = 1$ and $\mathbb{E}\left[\lambda_{1j}\right] = \hat{\sigma}^2 \max\left(\frac{p^{2.1}}{100n^*}, \log n^*\right)$, where $\hat{\sigma}^2$ is the sample variance of n^* pre-observations of Y (Narisetty and He, 2014). In this example, we have $n^* = 500$, $r = 14638$ and $\delta = 1770784$. Thus, the second combination of r and δ is more flexible with respect to the former because it depends on the variance and the number of pre-observations and on the number of variables.

Finally, we set $a = b = 1$ in order to have Uniform, non-informative, distribution in $[0, 1]$ for θ, $\theta \sim \mathsf{U}[0, 1]$.

Hyper-parameter	Value	Description
λ_0	0.01, 0.001	spike component
r	2502, 14638	shape of $p(\lambda_{1j})$
δ	125050, 1770784	rate of $p(\lambda_{1j})$
a	1	shape 1 of $p(\theta)$
b	1	shape 2 of $p(\theta)$

Table 2.1: Hyper-parameters' settings for Poisson regression model with spike-and-slab prior.

On the other hand, Table 2.2 shows hyperparameter's setting in the case of Bernoulli-Gaussian prior. In particular, we set $\sigma^2 = 100$ and $\alpha = \delta = 1$ in order to have a non-informative prior for $\boldsymbol{\beta}$, $\boldsymbol{\beta} \sim \mathsf{N}_p(\mathbf{0}, 100\mathbf{I}_p)$, and Uniform distribution in $[0, 1]$ for θ, $\theta \sim \mathsf{U}\,[0, 1]$.

Hyper-parameter	Value	Description
σ^2	100	variance of $p(\beta_j)$
α	1	shape 1 of p(θ)
δ	1	shape 2 of p(θ)

Table 2.2: Hyper-parameters' settings for Poisson regression model with Bernoulli-Gaussian prior.

Making use of these settings and using as stopping criterion a change in the lower bound less than 1e-5, the semi-parametric MFVB algorithm converged always after 3 iterations, except for VSS(0.01) and BG where we needed 4 iterations. The convergence path of the lower-bound is depicted in Figure 2.4.

As concerns inference accuracy, Figure 2.5 shows true values of $\boldsymbol{\beta}$, its point estimates and credibility intervals obtained through Poisson regression model with horseshoe, spike-and-slab and Bernoulli-Gaussian prior. We notice that the point estimates are all near to the true $\boldsymbol{\beta}$ values, and the credibility intervals include the true values of the coefficients. Furthermore, the results obtained with VSS and VSS$_{NH}$, for a fixed value of λ_0, are almost equal.

As concerns variable selection, Table 2.3 shows the true values of $\boldsymbol{\gamma}$, its estimate with the application of SAVS algorithm in the case of horseshoe prior (HS-SAVS) and the estimates of $\boldsymbol{\mu}_{q(\gamma)}$ with spike-and-slab and Bernoulli-Gaussian prior. We notice that

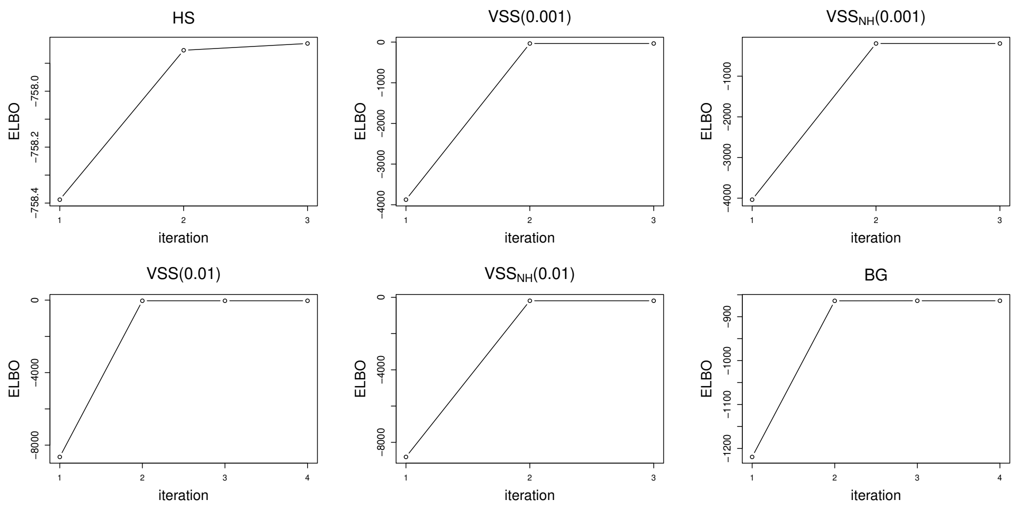


Figure 2.4: Convergence path of the lower bound according to different models.

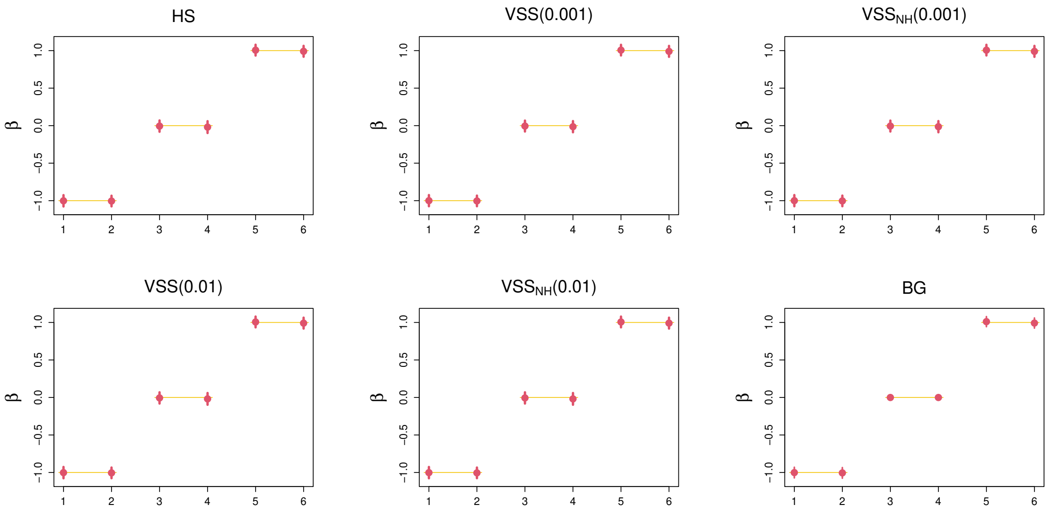


Figure 2.5: Point estimates (red circles) and 95% credible intervals (red lines) for the optimal expected values of the regression coefficients according to different models, compared with true values of $\boldsymbol{\beta}$ (yellow lines).

with the horseshoe prior we have a perfect variable selection: the coefficients equal and different from zero are correctly excluded and included in the model, respectively. On the other hand, with spike-and-slab and Bernoulli-Gaussian prior variable selection depends on the threshold value $s \in (0, 1)$. Figure 2.6 shows the ROC curve for these models, where for semplicity we used the same color for VSS(0.01), VSS$_{NH}$(0.01), VSS(0.001) and VSS$_{NH}$(0.001). We notice that they capture all the signal for any value of the threshold.

Parameter	True value	HS-SAVS	VSS, VSS$_{NH}$ ($\lambda_0 = 0.001$)	VSS, VSS$_{NH}$ ($\lambda_0 = 0.01$)	BG
γ_1	1	1	0.99	0.99	0.99
γ_2	1	1	0.99	0.99	0.99
γ_3	0	0	0.01	0.01	0.01
γ_4	0	0	0.01	0.01	0.01
γ_5	1	1	0.99	0.99	0.99
γ_6	1	1	0.99	0.99	0.99

Table 2.3: Point estimates of $\boldsymbol{\gamma}$ fitting Poisson model with horseshoe prior and of $\boldsymbol{\mu}_{q(\gamma)}$ fitting Poisson model with spike-and-slab and Bernoulli-Gaussian prior via MFVB.

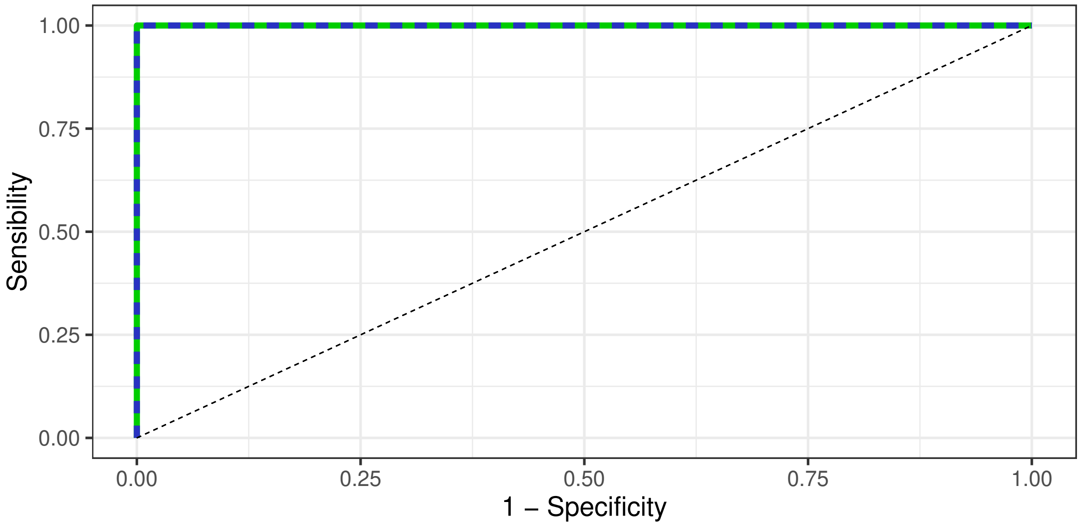


Figure 2.6: ROC curve for variable selection in Poisson regression model with spike-and-slab (blue) and Bernoulli-Gaussian (green) prior via MFVB. For semplicity, VSS(0.001), VSS$_{NH}$(0.001), VSS(0.01) and VSS$_{NH}$(0.01) have the same color.

2.5 Simulation study

In this Section, we provide a simulation study to demonstrate the performances of the proposed models. We compare them to two popular regularization methods: the EM variable selection approach, known as EMVS (Ročková and George, 2014), and the lasso (Tibshirani, 1996). In particular, we consider Poisson lasso (that we call PoiLASSO) on the original data, Y_i, and Gaussian lasso (that we call LASSO) and EMVS on $\log(Y_i + 1)$.

We focused on the case of $n > p$ and orthogonal design matrix $\mathbf{X}$, leaving the evalua-

n	300	600	1000	300	600	1000	300	600	1000
p	20	20	20	40	40	40	80	80	80

Table 2.4: Combinations of n and p considered in the simulation study.

tion of other cases (e.g. $n < p$ and/or non-orthogonal design matrix) to possible future works. We considered three different number of observations, $n = (300, 600, 1000)$, and of variables, $p = (20, 40, 80)$, with a fixed number of coefficients different from zero, $p_0 = 10$. Thus we considered the combinations in Table 2.4. Moreover, for each combination we replicated the estimate $R = 50$ times.

As concerns the evaluation of the models, we used the mean squared error (MSE) for the inference accuracy of $\boldsymbol{\beta}$, while F1-score and classification accuracy with threshold equal to 0.5 for the quality of variable selection. In addition, we made also a comparison in terms of computational cost between all models and in terms of area under the curve (AUC) between EMVS and the models with spike-and-slab and Bernoulli-Gaussian prior.

The results obtained are shown in Figures 2.7, 2.8, 2.9, 2.10, 2.11. As concerns variable selection, Figures 2.7 and 2.8 highlight a very good performance of all the models proposed in terms of F1-score and classification accuracy. Indeed, for each of the combination of n and p considered and for each model, the median of these two metrics are always between 90% and 97%. In particular, several considerations can be made. Firstly, we notice a little better performance of HS, VSS(0.001) and VSS$_{NH}$(0.001) with respect to the other models (PoiLASSO, LASSO and EMVS included) and lower variability of HS with respect to VSS(0.001) and VSS$_{NH}$(0.001). Secondly, for a fixed number of independent variables, the increase of the observations leads to an improvement of F1 and classification accuracy and a decrease in terms of variability. On the other hand, similar results and variability are obtained for a fixed number of observations. Furthermore, in Poisson model with spike-and-slab prior, the results achieved with VSS and VSS$_{NH}$ for a fixed value of λ_0 are almost equal. Finally, we notice a good performance of LASSO and EMVS, even if they show the worst measures for any combination considered.

Figure 2.9 depicts the inference accuracy of the point estimates of $\boldsymbol{\beta}$ in terms of MSE. We notice that all the models, except for LASSO and EMVS, show a similar and good behavior. Indeed, they have the median of the MSE between 0.01 and 0.04 for any combination of n and p. In addition, they have always small variability. On

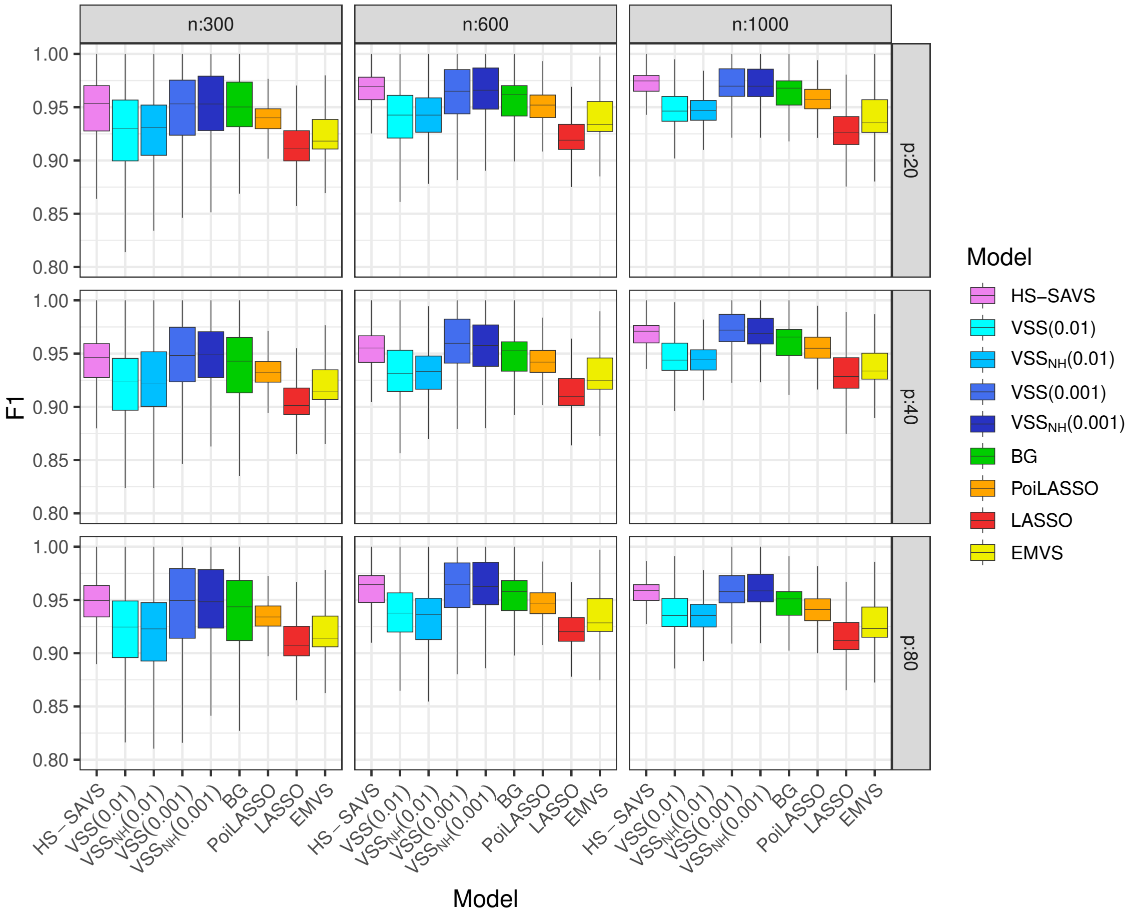


Figure 2.7: F1-score according to different models.

the other hand, the MSE of EMVS is always the greatest, followed by LASSO.

In particular, for any value of p, the MSE of EMVS and LASSO are respectively around 0.13 and 0.11 with $n = 300$, they are between 0.07 and 0.11 and between 0.06 and 0.09 when $n = 600$, and they are around 0.05 when $n = 1000$. Thus, for a fixed number of variables, the MSE of EMVS and LASSO decrease with the increase of observations. Furthermore, it is important to remember that our proposed models have a different behaviour with respect to PoiLASSO, LASSO and EMVS: while the former provide also posterior densities for $\boldsymbol{\beta}$, PoiLASSO, LASSO and EMVS only return us the point estimates.

Figure 2.10 shows the results of the comparison in terms of computational cost. We notice a greater computational effort and variability of BG model with respect to the other proposed models when $n = 600$ or $n = 1000$, for any value of p. The reason is that in this model we have $2p$ numeric optimizations (of β_j and γ_j, $j = 1, \dots, p$), while in HS, VSS and VSS$_{NH}$ there is only one (of $\boldsymbol{\beta}$). Furthermore, the median time of HS, VSS and VSS$_{NH}$ is always similar and between 1 and 4 seconds, a little bit

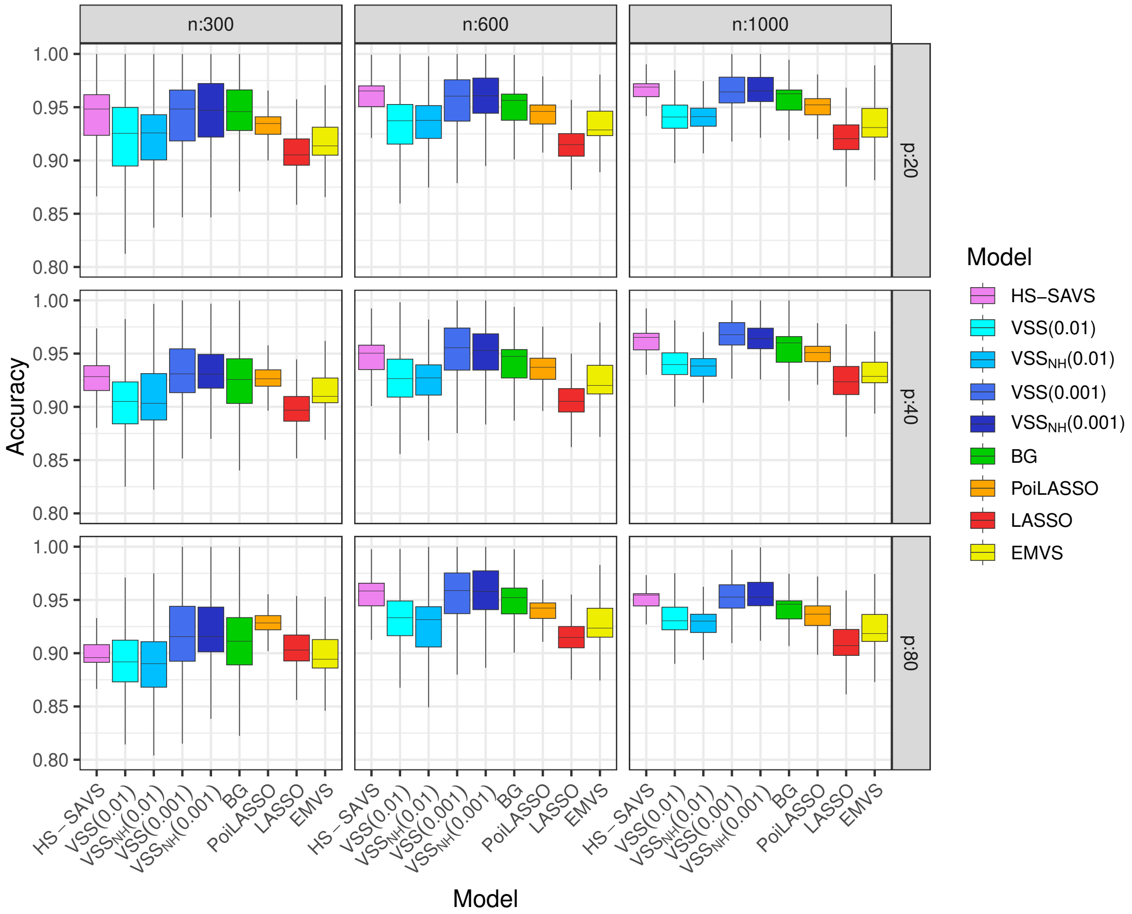


Figure 2.8: Classification accuracy according to different models.

greater than PoiLASSO, LASSO and EMVS.

Finally, in order to evaluate the performance of Poisson model with spike-and-slab and Bernoulli-Gaussian prior and of EMVS overall and not for a fixed value of the threshold, Figure 2.11 depicts the results obtained in terms of AUC. The considerations that could be made are similar to those made for the F1-score and for the classification accuracy in Figures 2.7 and 2.8.

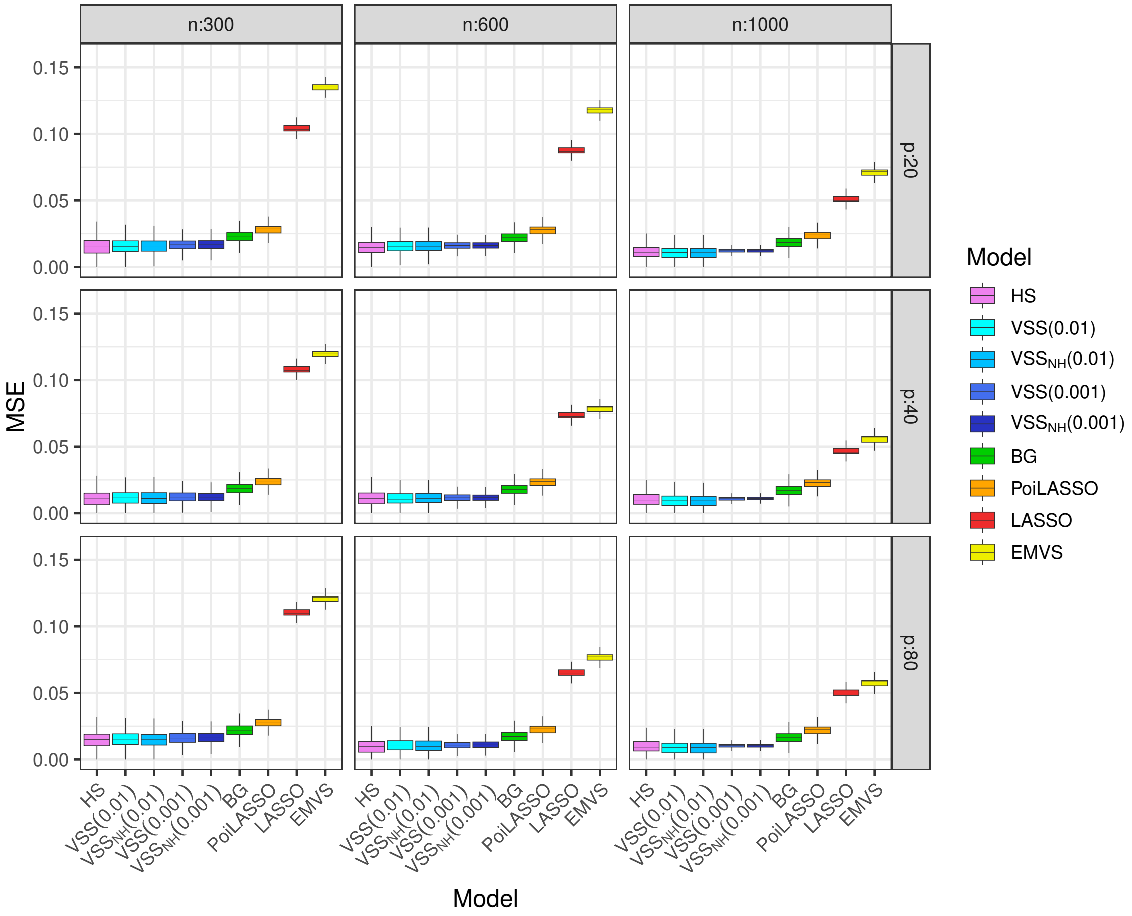


Figure 2.9: MSE according to different models.

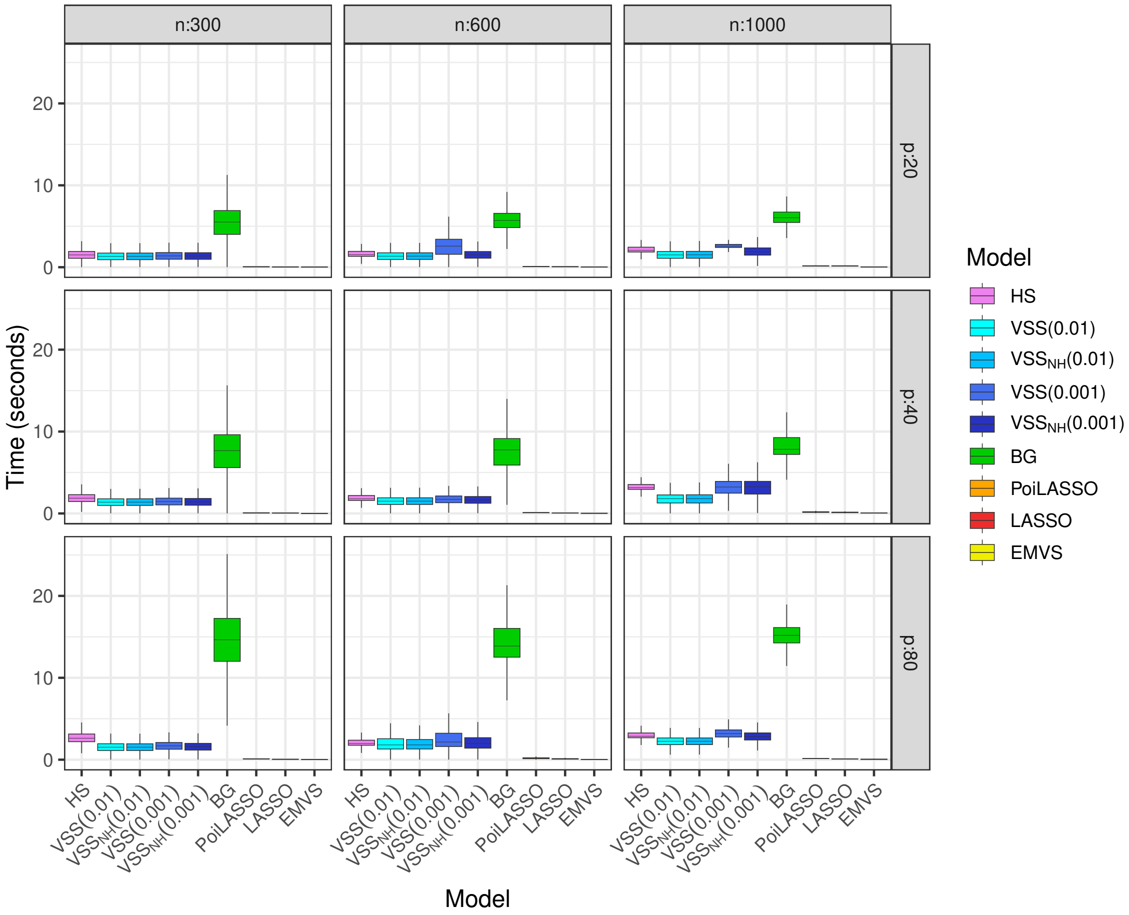


Figure 2.10: Time (in seconds) according to different models.

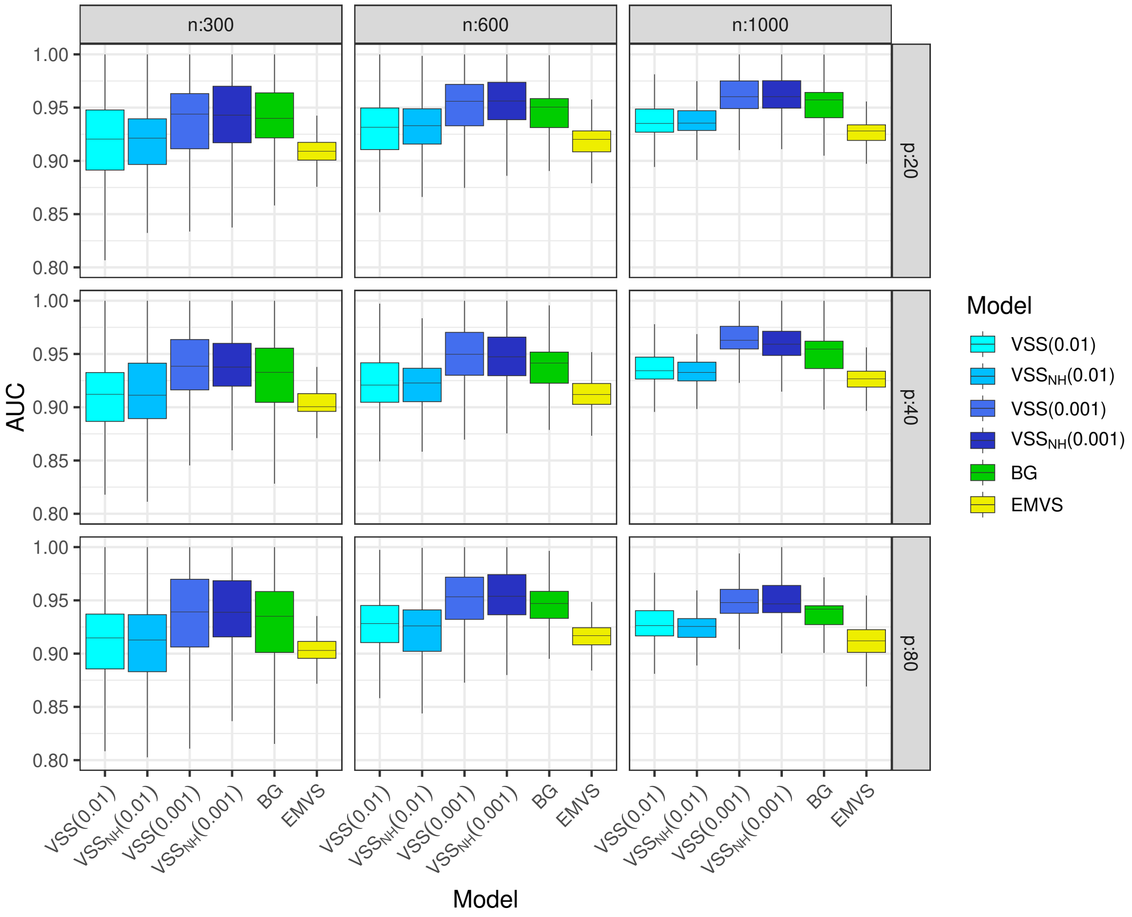


Figure 2.11: AUC according to Poisson model on Y with spike-and-slab and Bernoulli-Gaussian prior via MFVB and according to EMVS on $\log(Y + 1)$.

Chapter 3

Application to real data

In this Chapter, we applied our methods to the analysis of a real dataset, called *Football 2022-2023* dataset, available on Kaggle. The dataset contains 2689 observations and 27 variables concerning 2022-2023 football player performances in the major european leagues, namely Premier League, Ligue 1, Bundesliga, Serie A and Liga. The variables are explained in Table 3.1. We focus our attention on the strikers, meaning that we consider only football player with level of the variable *Pos* equal to “FW”. Moreover, we group the 105 levels of the categorical variable *Nation* in 7 levels (“BRA”, “ENG”, “ESP”, “FRA”, “GER”, “ITA”, “Other”). The final dataset for the analysis is composed by 683 observations and 25 variables (*Player* and *Pos* are excluded), and we consider the variable *Goals* as the response of interest.

The statistical analysis has two independent goals. The first is the interpretation of the estimated regression coefficients (in-sample analysis), while the second focuses on the performance of the models in predict the expected number of goals scored given the characteristics of the striker (out-of-sample analysis).

In-sample estimates. We fit the models proposed in the previous chapters, comparing them with PoiLASSO and the generalized linear model (GLM) Poisson (that we call PoiGLM) in terms of inference. As concerns the values of r and δ in VSS$_{NH}$(0.01) and VSS$_{NH}$(0.001), we exploit the informations included in *Football 2021-2022* dataset. In particular, we calculate the variance of the goals scored by the 631 strikers in 2021-2022 in the same five championships, that is equal to $\hat{\sigma}^2 = 4.96$. This value leads to $\mathbb{E}\left[\lambda_{1j}\right] = 32.87$ and, as a consequence, to $r = 1082$ and $\delta = 35531$.

The results obtained are depicted in Figure 3.1. We only show VSS(0.001), BG and HS because the estimates of VSS(0.001), VSS$_{NH}$(0.001), VSS(0.01) and VSS$_{NH}$(0.01)

Variable	Description
Player	Player's name
Pos	Position
Goals	Goals scored
BlkPass	Number of times blocking a pass by standing in its path
Fls	Fouls committed
PasTotDist	Total distance, in yards, that completed passes have traveled in any direction
CK	Corner kicks
CarMis	Number of times a player failed when attempting to gain control of a ball
ShoDist	Average distance, in yards, from goal of all shots taken
TklDriPast	Number of times dribbled past by an opposing player
2CrdY	Second yellow card
Starts	Number of match started
Car3rd	Carries that enter the 1/3 of the pitch closest to the goal
CrdY	Yellow cards
CarDis	Number of times a player loses control of the ball after being tackled by an opposing player
TB	Completed pass sent between back defenders into open space
Fld	Fouls drawn
PKcon	Penalty kicks conceded
CrdR	Red cards
PasTotCmp	Passes completed
MP	Number of match played
PKwon	Penalty kicks won
Age	Player's age
SoT	Shots on target
Assists	Assists
Comp	League name
Nation	Player's nationality

Table 3.1: Description of the variables contained in the *Football 2022-2023* dataset.

are similar to each other. We notice that the point estimates of the HS's regression coefficients are always different from 0, while VSS and BG have some point estimates equal to 0: this aspect highlights the difference between variable selection, performed in VSS and BG, and shrinkage, performed in HS.

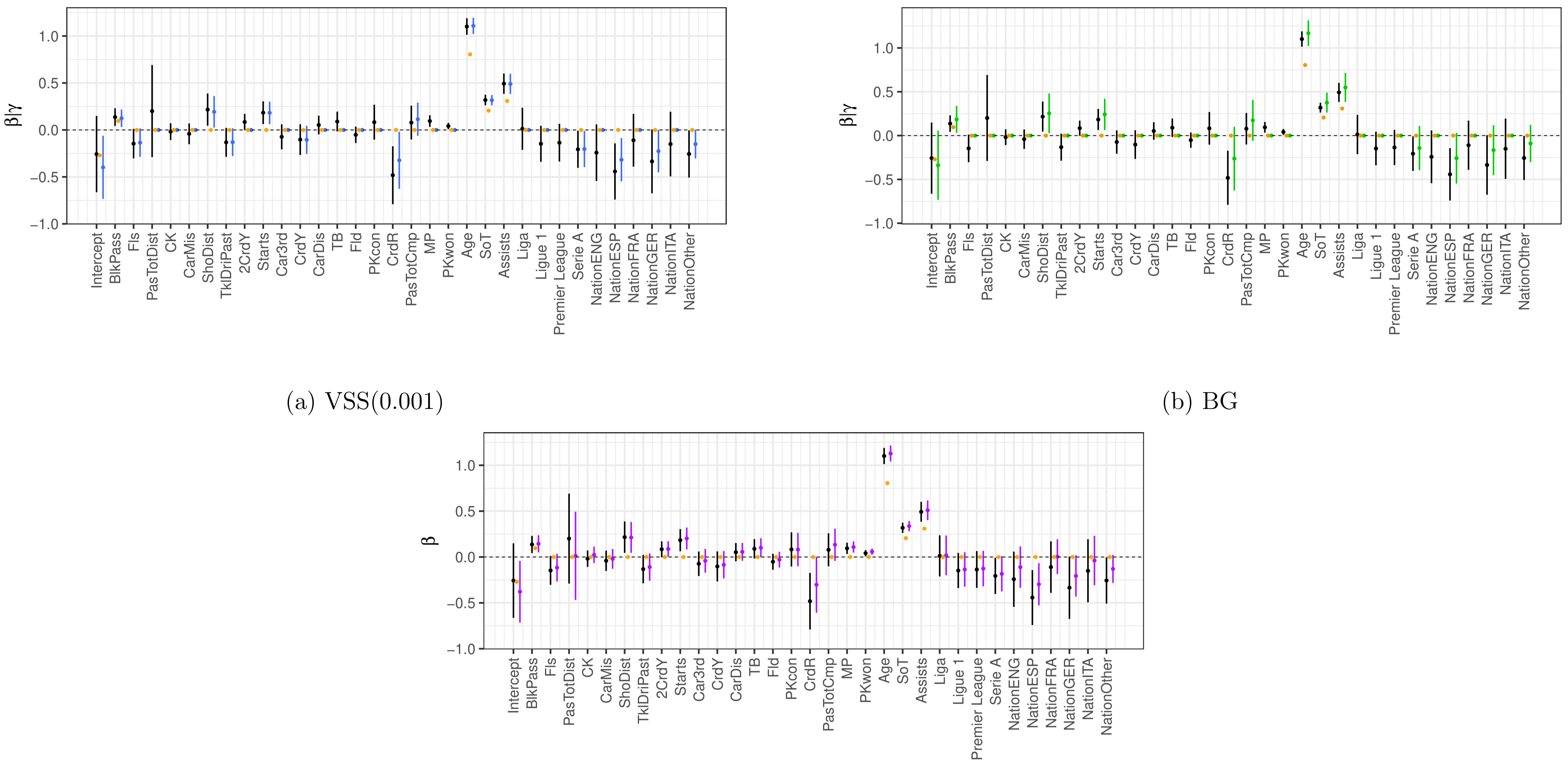
As concerns variable selection, we notice a similar behaviour between VSS and BG. Indeed, with VSS 17 coefficients are set to 0, while with BG we set to 0 the same 17 coefficients plus three others ("Fls", "TklDriPast" and "CrdY"). In addition, when the confidence interval for a regression coefficient obtained with PoiGLM includes the value 0, the regression coefficient is not selected as different from 0 with VSS and BG.

As concerns the variability around the point estimates, 95% credible intervals of the proposed models are similar to those obtained with PoiGLM. Interestingly, the credible intervals of BG are wider compared to those of the other models.

In addition, as we made in the simulation study, we fit also LASSO and EMVS on $\log(Goals + 1)$. The point estimates of $\boldsymbol{\beta}$ obtained with all the models are showed in Table 3.3. We notice that the variable selection of PoiLASSO, LASSO and EMVS is the same. On the other hand, LASSO and EMVS lead to quite different point estimates of $\boldsymbol{\beta}$ compared to those of the other models.

Finally, focusing our attention on the point estimates of VSS and BG different from zero, it is worth noting that the more a striker provides assists, the higher his expected number of goals becomes. It is also of interest that the expected number of goals increases with the increasing number of match started from the first minute. In addition, as concerns the championship, it is notable that the expected number of goals scored by the Serie A's strikers is lower compared to that of the strikers in the other four leagues.

Out-of-sample forecasting accuracy. In order to inspect the ability in predict the number of goals based on player's characteristics, we build a simple forecasting scenario. In particular, we split randomly the 683 observations in training and test set, in order to have 75% of the strikers in the training set and 25% in the test set. Subsequently, we fit all the models on the 513 observations in the training set. As concerns $\text{VSS}_{NH}(0.001)$ and $\text{VSS}_{NH}(0.01)$, the values of r and δ are 1082 and 35531, respectively. At this point, we evaluate the point forecasts for the 170 strikers in the test set based on the out-of-sample predictive MSE and the mean absolute error (MAE). Table 3.2 shows the results obtained. The best performance is given by PoiLASSO both in terms of MSE and MAE, but the results of all our



(a) VSS(0.001)

(b) BG

(c) HS

Figure 3.1: Point estimates (circles) and 95% credible intervals (lines) of $\boldsymbol{\beta}$ according to VSS(0.001) (blue), BG (green) and HS (violet), compared to PoiGLM (black) and PoiLASSO (orange). The results of VSS(0.001), VSS$_{NH}$(0.001), VSS(0.01) and VSS$_{NH}$(0.01) are similar to each other.

models are comparable. More importantly, our models outperform LASSO, EMVS and PoiGLM. The latter probably incurs in overfitting issues since no regularization or variable selection is performed. The bad performances of LASSO and EMVS suggest that transform the orginal count variable to be approximated by a normal is a sub-optimal solution with respect to ad-hoc models.

Model	MSE	MAE
HS	4.89	1.35
VSS(0.001)	4.01	1.25
VSS$_{NH}$(0.001)	4.03	1.26
VSS(0.01)	4.07	1.27
VSS$_{NH}$(0.01)	4.14	1.30
BG	3.87	1.22
PoiGLM	5.14	1.41
PoiLASSO	3.77	1.19
LASSO	6.53	1.57
EMVS	8.66	1.87

Table 3.2: MSE and MAE out-of-sample according to different models.

Variable	HS	VSS(0.001)	BG	PoiGLM	PoiLASSO	LASSO	EMVS
Intercept	-0.38 (0.17)	-0.40 (0.16)	-0.38 (0.20)	-0.26 (0.21)	-0.27	0.77	0.85
BlkPass	0.14 (0.05)	0.12 (0.04)	0.14 (0.08)	0.14 (0.05)	0.10	0.05	0.09
Fls	-0.12 (0.08)	-0.14 (0.06)	0.00	-0.15 (0.08)	0.00	0.00	0.00
PasTotDist	0.01 (0.25)	0.00	0.00	0.20 (0.25)	0.00	0.00	0.00
CK	0.02 (0.05)	0.00	0.00	-0.02 (0.05)	0.00	0.00	0.00
CarMis	-0.02 (0.05)	0.00	0.00	-0.04 (0.06)	0.00	0.00	0.00
ShoDist	0.21 (0.09)	0.19 (0.07)	0.20 (0.12)	0.22(0.09)	0.00	0.00	0.00
TklDriPast	-0.11 (0.08)	-0.13 (0.07)	0.00	-0.13 (0.08)	0.00	0.00	0.00
2CrdY	0.09 (0.04)	0.00	0.00	0.08 (0.04)	0.00	0.00	0.00
Starts	0.20 (0.06)	0.18 (0.06)	0.20 (0.09)	0.18 (0.06)	0.00	0.00	0.00
Car3rd	-0.04 (0.07)	0.00	0.00	-0.07 (0.07)	0.00	0.00	0.00
CrdY	-0.08 (0.08)	-0.10 (0.07)	0.00	-0.10 (0.08)	0.00	0.00	0.00
CarDis	0.06 (0.05)	0.00	0.00	0.05 (0.05)	0.00	0.00	0.00
TB	0.10 (0.05)	0.00	0.00	0.09 (0.05)	0.00	0.00	0.00
Fld	-0.03 (0.04)	0.00	0.00	-0.05 (0.04)	0.00	0.00	0.00
PKcon	0.08 (0.09)	0.00	0.00	0.08 (0.09)	0.00	0.00	0.00
CrdR	-0.30 (0.16)	-0.32 (0.14)	-0.32 (0.19)	-0.48 (0.16)	0.00	0.00	0.00
PasTotCmp	0.13 (0.09)	0.11 (0.08)	0.13 (0.12)	0.08 (0.09)	0.00	0.00	0.00
MP	0.11 (0.03)	0.00	0.00	0.09 (0.03)	0.00	0.00	0.00
PKwon	0.06 (0.02)	0.00	0.00	0.04 (0.02)	0.00	0.00	0.00
Age	1.13 (0.04)	1.11 (0.03)	1.13 (0.07)	1.10 (0.04)	0.81	0.39	0.46
SoT	0.34 (0.03)	0.32 (0.02)	0.34 (0.06)	0.32 (0.03)	0.21	0.04	0.00
Assists	0.51 (0.05)	0.49 (0.04)	0.50 (0.08)	0.49 (0.06)	0.31	0.24	0.28
Liga	0.02 (0.11)	0.00	0.00	0.01 (0.11)	0.00	0.00	0.00
Ligue 1	-0.14 (0.10)	0.00	0.00	-0.15 (0.10)	0.00	0.00	0.00
Premier League	-0.13 (0.10)	0.00	0.00	-0.14 (0.10)	0.00	0.00	0.00
Serie A	-0.18 (0.10)	-0.20 (0.10)	-0.18 (0.13)	-0.21 (0.10)	0.00	0.00	0.00
NationENG	-0.11 (0.12)	0.00	0.00	-0.24 (0.15)	0.00	0.00	0.00
NationESP	-0.30 (0.12)	-0.32 (0.11)	-0.30 (0.15)	-0.44 (0.15)	0.00	0.00	0.00
NationFRA	0.00 (0.10)	0.00	0.00	-0.11 (0.14)	0.00	0.00	0.00
NationGER	-0.21 (0.12)	-0.23 (0.11)	-0.21 (0.15)	-0.33 (0.17)	0.00	0.00	0.00
NationITA	-0.04 (0.14)	0.00	0.00	-0.15 (0.18)	0.00	0.00	0.00
NationOther	-0.13 (0.08)	-0.15 (0.07)	-0.13 (0.11)	-0.26 (0.13)	0.00	0.00	0.00

Table 3.3: Point estimates of $\boldsymbol{\beta}$ according to different models fitted on *Football 2022-2023* dataset.

Conclusions

Variational methods are feasible and computationally efficient deterministic approximations to perform inference within the Bayesian framework and they represent a valid alternative to the usual simulation-based stochastic approximation methods, such as MCMC. In this thesis we focused on the application of these inferential techniques in order to make variable selection in the Bayesian Poisson regression model. Variable selection is rising in importance because the number of variables to work with is growing more and more in lots of modern applications. Furthermore, the problem of working with many covariates is even more important in the Poisson regression model with canonic link because the explosion of the predictor could lead to computational issues.

After the comparison between MFVB and MCMC with two simple examples, we developed and we implemented, via semi-parametric MFVB, Bayesian Poisson regression model with three different options: horseshoe prior (Carvalho et al., 2010), spike-and-slab prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993, 1997) and Bernoulli-Gaussian prior (Ormerod et al., 2017; Bernardi et al., 2023). In order to compare their perfomances, a simulation study has been performed, including also the most popular model selection approaches such as Poisson lasso, lasso (Tibshirani, 1996) and EMVS (Ročková and George, 2014). All the proposed models provided promising results in terms of variable selection accuracy and point estimates of the regression coefficients.

A good performance has been achieved also in the application to *Football 2022-2023* dataset. All our algorithms have been compared to Poisson lasso, generalized linear model (GLM) Poisson, lasso and EMVS in terms of in-sample estimates of the regression coefficients and out-of-sample forecasting accuracy.

In summary, the main contribution of this work is the development and the implementation of three Bayesian Poisson regression model through MFVB approach that allow to make both variable selection and full posterior inference.

Appendix A

R Code

A.1 Multivariate Gaussian distribution

MCMC algorithm

```
multivariate.gaussian.MCMC <- function(R, y, hyp, start) {

  start.time <- Sys.time()

  n <- nrow(y)
  d <- ncol(y)
  y.mean <- apply(y, 2, mean)

  #============= Matrix of results =============
  out.mu <- matrix(NA, nrow = R, ncol = d)
  out.prec <- array(NA, dim = c(d,d,R))

  #============= Hyperparameters' setting =============
  mu0 <- hyp[[1]]
  prec0 <- hyp[[2]]
  nu <- hyp[[3]]
  v <- hyp[[4]]

  #constant hyperparameter in the posterior
  post.nu <- nu + n
  uno.v <- solve(v)
  prec0.nu0 <- prec0 %*% mu0

  #============= Defining parameters =============
  xstar.mu <- start[[1]]
  xstar.prec <- start[[2]]

  #============= MCMC algorithm =============

  require(mvtnorm)
  set.seed(3)
  for(i in 1:R) {

    #drawn precision
    #y.mu.mean <- t(apply(y, 1, function(x) x - xstar.mu))
    y.mu.mean <- t(y) - xstar.mu
    #y.mean.second.sum <- t(y.mu.mean) %*% y.mu.mean
    y.mean.second.sum <- y.mu.mean %*% t(y.mu.mean)
    post.v <- solve(uno.v + y.mean.second.sum)
    xstar.prec <- rWishart(n = 1, df = post.nu, Sigma = post.v)[,,1]

    #drawn mu
    post.var.mu <- solve(n* xstar.prec + prec0)
    post.mean.mu <- as.vector(post.var.mu %*% (n* xstar.prec
                                            %*% y.mean + prec0.nu0))
    xstar.mu <- drop(mvtnorm::rmvnorm(n = 1, mean = post.mean.mu,
                                      sigma = post.var.mu))

    #returning updated values for mu and precision
    out.mu[i,] <- xstar.mu
    out.prec[,,i] <- xstar.prec
  }

  end.time <- Sys.time()

  #============= Computing computational effort =============

  total.time <- end.time - start.time

  return(list(mu.values = out.mu, prec.values = out.prec,
              tempo = total.time[[1]]))
}
```

MFVB algorithm

```
#Updating equations of q.mu
```

```r
update.MFVB <- function(y, y.sum, prec0, mu0, n, d, mu.q.prec,
                        nu.q.prec) {

  sigma.q.mu <- solve(n*mu.q.prec + prec0)
  mu.q.mu <- sigma.q.mu %*% (mu.q.prec %*% y.sum + prec0 %*% mu0)
  y.mean <- t(apply(y, 1, function(x) x - mu.q.mu))
  y.mean.second <- apply(y.mean, 1, function(x) x %*% t(x))
  y.mean.second.sum <- matrix(rowSums(y.mean.second), nrow = d,
                              ncol = d)
  v.q.prec <- solve(y.mean.second.sum + n*sigma.q.mu + solve(v))
  mu.q.prec <- nu.q.prec * v.q.prec
  return(list(mu.q.mu = mu.q.mu, sigma.q.mu = sigma.q.mu,
              v.q.prec = v.q.prec, mu.q.prec = mu.q.prec))
}

#MFVB algorithm
multivariate.gaussian.MFVB <- function(y, hyp, start, maxIter = 100,
                                       eps_ELBO = 1e-4, Trace = 0) {

  start.time <- Sys.time()

  n <- nrow(y)
  d <- ncol(y)
  y.sum <- apply(y, 2, sum)

  #============= Defining history variables =============
  elbo.out <- numeric()
  mu.q.mu.out <- matrix(NA, nrow = maxIter, ncol = d)
  sigma.q.mu.out <- array(NA, dim = c(d,d,maxIter))
  v.q.prec.out <- array(NA, dim = c(d,d,maxIter))

  #============= Hyperparameters' setting =============
  #hyperparameters (hyp = (mu0, prec0, nu, v))
  mu0 <- hyp[[1]]
  prec0 <- hyp[[2]]
  nu <- hyp[[3]]
  v <- hyp[[4]]

  #============= Initializing optimal parameters =============
  mu.q.mu <- start[[1]]
  sigma.q.mu <- solve(start[[2]])
  v.q.prec <- start[[4]]
```

```
mu.q.prec <- start[[3]]*start[[4]]

#constant optimal hyperparameter in the update
nu.q.prec <- nu + n

#============= MFVB algorithm =============
require(CholWishart)
i <- 1
cond <- FALSE
while(cond != TRUE & i <= maxIter) {

  #updating q.mu
  new.val <- update.MFVB(y, y.sum, prec0, mu0, n, d, mu.q.prec,
                         nu.q.prec)
  mu.q.mu <-  new.val[[1]]
  sigma.q.mu <- new.val[[2]]

  #updating q.prec
  v.q.prec <- new.val[[3]]
  mu.q.prec <- new.val[[4]]

  #computing lower bound
  ELBO <- -n*d/2*log(2*pi) - 0.5* as.vector(t(mu.q.mu - mu0) %*%
    prec0 %*% (mu.q.mu - mu0)) + 0.5*log(det(prec0)) +
    0.5 * log(det(sigma.q.mu)) - 0.5 * sum(diag(sigma.q.mu *
    prec0)) + d/2 - nu*d/2*log(2) - nu/2*log(det(v)) -
    lmvgamma(nu/2, d) + nu.q.prec*d/2*log(2) + nu.q.prec/2*
    log(det(v.q.prec)) + lmvgamma(nu.q.prec/2, 2)

  #computing lower bound
  mu.q.mu.out[i,] <- mu.q.mu
  sigma.q.mu.out[,,i] <- sigma.q.mu
  v.q.prec.out[,,i] <- v.q.prec
  elbo.out <- c(elbo.out, ELBO)



  #stopping criterion
  if (i > 1) {
    #delta1 <- max(abs((parNew-parOld)/parOld))
    delta2 <- (abs((elbo.out[i] - elbo.out[i-1])/elbo.out[i-1]))

    #if ((delta1 < eps_Par)&(delta2 < eps_ELBO)) cond <- 1
```

```
      if(delta2 < eps_ELBO) cond <- 1
      if (Trace == 1) {
        #print(paste0("iteration: ",i," - parameter variation: ",
        #delta1," - lower bound increase: ",delta2))
        print(paste0("iteration:", i, "- lower bound increase: ",
                     delta2))
      }
    }
    if (i > maxIter) cond <- 1
    i <- i + 1
  }

  end.time <- Sys.time()

  #============= Computing computational effort =============
  total.time <- end.time - start.time

  return(list(elbo = elbo.out, mu.mu = mu.q.mu.out[1:(i-1),],
              sigma.mu =  sigma.q.mu.out[,,1:(i-1)],
              nu.prec = nu.q.prec, v.prec = v.q.prec.out[,,1:(i-1)],
              iter = i-1, tempo = total.time[[1]]))
}
```

A.2 Poisson regression model

MCMC algorithm

```
lposterior.si.sigma <- function(param, dati, hyp) {
  #hyp = (alpha, delta)
  #param = (beta, sigma2)
  x <- dati[,-1]
  y <- dati[,1]
  p <- ncol(dati) - 1
  alpha <- hyp[1]
  delta <- hyp[2]
  beta <- param[1:p]
  sigma2 <- param[p+1]

  x.beta.y.sum <- as.vector(t(y) %*% (x %*% beta))
  exp.x.beta <- sum(as.vector(exp(x %*% beta)))
```

```r
  if(sigma2 <= 0) {
    return(- Inf)
  }
  else{
    return(-(alpha + p/2 + 1)*log(sigma2) - delta/sigma2 -
             0.5/sigma2 * as.vector(t(beta) %*% beta) +
             x.beta.y.sum - exp.x.beta)
  }
}

#MCMC algorithm
MCMC.si.sigma <- function(dati, hyp, R, start, eps) {

  start.time <- Sys.time()

  n <- nrow(dati)
  p <- ncol(dati) - 1
  accepted <- 0

  #============= Matrix of results =============
  out.beta <- matrix(NA, nrow = R, ncol = p)
  out.sigma <- rep(NA, R)

  #============= Hyperparameters' setting =============
  alpha <- hyp[1]
  delta <- hyp[2]

  #============= Defining parameters =============
  x.beta <- start[1:p]
  x.sigma <- start[p+1]

  #============= MCMC algorithm =============

  set.seed(1)
  require(mvtnorm)
  for(i in 1:R) {

    #drawn sigma2
    x.sigma <- 1/rgamma(1, alpha + p/2, delta + 0.5*
               sum(x.beta*x.beta))

    #drawn beta
    xstar.beta <- drop(rmvnorm(1, mean = x.beta,
                  sigma = eps*solve(post.Jhat.si.sigma[1:p, 1:p])))
    bound <- exp(lposterior.si.sigma(c(xstar.beta, x.sigma), dati,
             hyp) - lposterior.si.sigma(c(x.beta, x.sigma), dati,
             hyp))
    if(runif(1) < bound) {
      x.beta <- xstar.beta
      accepted <- accepted + 1
    }

    #returning updated values for beta and sigma2
    out.sigma[i] <- x.sigma
    out.beta[i,] <- x.beta
  }

  end.time <- Sys.time()

  #============= Computing computational effort =============

  total.time <- end.time - start.time

  return(list(beta.values = out.beta, sigma.values = out.sigma,
              tasso = accepted/R,
              tempo = total.time[[1]]))
}
```

MFVB algorithm

```
library(msos)

#Natural fixed-point iteration for the update of optimal
#variational density q.beta
update.local.PMFVB.si.sigma <- function(x, y, x.y.sum, mu.q.beta,
                                        sigma2.q.beta,
                                        alpha.q.sigma,
                                        delta.q.sigma, mu.q.1.sigma,
                                        n, p, eps_local,
                                        maxIter_local,
                                        Trace_local) {
  #============= Defining history variables =============
  elbo.out.local <- numeric()
  mu.q.beta.out <- matrix(NA, nrow = maxIter_local, ncol = p)
```

```
sigma2.q.beta.out <- array(NA, dim = c(p,p,maxIter_local))

#============= Natural fixed-point iteration =============
j <- 1
cond2 <- FALSE

while(cond2 != TRUE & j <= maxIter_local) {

  #updating equations of q.mu
  log.exp.mu.q.beta <- as.vector(x %*% as.matrix(mu.q.beta)
                                 +0.5 * diag(x %*%
                                  as.matrix(sigma2.q.beta) %*%
                                  t(x)))
  exp.mu.q.beta <- as.vector(exp(log.exp.mu.q.beta))
  x.exp.mu.q.beta.d1 <- t(x) %*% as.matrix(exp.mu.q.beta)
  nu.q.beta <- -mu.q.beta*(alpha.q.sigma/delta.q.sigma) +
    x.y.sum - x.exp.mu.q.beta.d1
  sigma2.q.beta <- solve( diag(1,p)*mu.q.1.sigma + t(x) %*%
                            diag(as.vector(exp.mu.q.beta)) %*% x)
  mu.q.beta <- mu.q.beta + sigma2.q.beta %*% nu.q.beta


  #computing beta-localized lower bound
  entropy.q.beta <- 0.5*p* log(2*pi) + 0.5 * logdet(sigma2.q.beta)
                    + 0.5 * p
  lprior.beta <- -p/2 * (log(2*pi*delta.q.sigma) -
                  digamma(alpha.q.sigma))- 0.5*alpha.q.sigma/
                  delta.q.sigma * (sum(diag(sigma2.q.beta)) +
                  t(mu.q.beta) %*% mu.q.beta)
  logL <- t(y) %*% (x %*% mu.q.beta) - sum(lfactorial(y)) -
    (t(rep(1,n)) %*% as.matrix(exp.mu.q.beta))
  non.entropy.q.beta <- lprior.beta + logL
  ELBO.local <- entropy.q.beta + non.entropy.q.beta

  #updating histories
  mu.q.beta.out[j,] <- mu.q.beta
  sigma2.q.beta.out[,,j] <- sigma2.q.beta
  elbo.out.local <- c(elbo.out.local, ELBO.local)


  #stopping criterion
  if(j > 1) {
      delta2 <- (abs((elbo.out.local[j] - elbo.out.local[j-1])/
                       elbo.out.local[j-1]))

      if(delta2 < eps_local) cond2 <- 1
      if (Trace_local == 1) {
        print(paste0("iteration:", j, "- local lower bound
                     increase: ", delta2))
      }
    }
    if (j > maxIter_local) cond2 <- 1
    j <- j + 1

  }
  return(list(mu.q.beta = mu.q.beta.out[j-1,],
              sigma2.q.beta = sigma2.q.beta.out[,,j-1],
              entropy.q.beta = entropy.q.beta, logL = logL))
}


#MFVB algorithm
PMFVB.si.sigma <- function(y, x,  hyp, start, eps_local, eps_global,
                           maxIter_local, maxIter_global,
                           Trace_local = 0, Trace_global = 0) {

  start.time <- Sys.time()

  n <- nrow(x)
  p <- ncol(x)
  x.y.sum <- colSums(y * x)

  #============= Hyperparameters' setting =============
  mu.beta <- hyp[[1]]
  alpha.sigma <- hyp[[2]]
  delta.sigma <- hyp[[3]]

  #============= Defining history variables =============
  elbo.out.global <- numeric()
  entropy.q.beta.out <- numeric()
  mu.q.beta.out <- matrix(NA, nrow = maxIter_global, ncol = p)
  sigma2.q.beta.out <- array(NA, dim = c(p,p,maxIter_global))
  delta.q.sigma.out <- numeric()
```

```r
#============= Initializing optimal parameters =============
mu.q.beta <- start[[1]]
sigma2.q.beta <- start[[2]]
alpha.q.sigma <- start[[3]]
delta.q.sigma <- start[[4]]
mu.q.1.sigma <- alpha.q.sigma/delta.q.sigma

#============= MFVB algorithm =============

i <- 1
cond1 <- FALSE
while(cond1 != TRUE & i <= maxIter_global) {

  #updating q.beta
  q.mu.update <- update.local.PMFVB.si.sigma(x, y, x.y.sum,
                                             mu.q.beta,
                                             sigma2.q.beta,
                                             alpha.q.sigma,
                                             delta.q.sigma,
                                             mu.q.1.sigma,
                                             n, p, eps_local,
                                             maxIter_local,
                                             Trace_local)
  mu.q.beta <- q.mu.update[[1]]
  sigma2.q.beta <- q.mu.update[[2]]

  #updating di q.sigma
  alpha.q.sigma <- as.vector(alpha.sigma + p/2)
  delta.q.sigma <- as.vector(delta.sigma + 0.5*
                  (sum(diag(sigma2.q.beta)) + t(mu.q.beta) %*%
                  mu.q.beta))
  mu.q.1.sigma <- alpha.q.sigma/delta.q.sigma

  #computing lower bound
  entropy.q.sigma <- alpha.q.sigma + log(delta.q.sigma *
                     gamma(alpha.q.sigma)) - (alpha.q.sigma + 1)*
                     digamma(alpha.q.sigma)
  lprior.sigma <-  alpha.sigma*log(delta.sigma) -
                   lgamma(alpha.sigma) -(alpha.sigma +1)*
                   (log(delta.q.sigma) + (alpha.sigma + 1)*
                   digamma(alpha.q.sigma)) - delta.sigma*
                   mu.q.1.sigma
```

```
    entropy.q.beta <- q.mu.update[[3]]
    logL <- q.mu.update[[4]]
    lprior.beta <- -p/2 * (log(2*pi*delta.q.sigma) -
                  digamma(alpha.q.sigma))
                  - 0.5*mu.q.1.sigma *
                  (sum(diag(sigma2.q.beta)) + t(mu.q.beta) %*%
                  mu.q.beta)
    ELBO.global <- entropy.q.beta + entropy.q.sigma + logL +
                  lprior.beta  + lprior.sigma

    #updating history variables
    mu.q.beta.out[i,] <- mu.q.beta
    sigma2.q.beta.out[,,i] <- sigma2.q.beta
    delta.q.sigma.out <- c(delta.q.sigma.out, delta.q.sigma)
    elbo.out.global <- c(elbo.out.global, ELBO.global)

    #stopping criterion
    if(i > 1) {
      delta1 <- (abs((elbo.out.global[i] - elbo.out.global[i-1])/
                    elbo.out.global[i-1]))
      if(delta1 < eps_global) cond1 <- 1
      if (Trace_global == 1) {
        print(paste0("iteration:", i, "- global lower bound
                    increase: ", delta1))
      }
    }
    if (i > maxIter_global) cond1 <- 1
    i <- i + 1
  }

  end.time <- Sys.time()

  #============= Computing computational effort =============

  total.time <- end.time - start.time

  return(list(elbo = elbo.out.global,
              mu.q.beta = mu.q.beta.out[1:(i-1),],
              sigma2.q.beta =  sigma2.q.beta.out[,,1:(i-1)],
              alpha.q.sigma = alpha.q.sigma,
              delta.q.sigma = delta.q.sigma.out, iter = i-1,
              tempo = total.time[[1]]))
```

```
}
```

A.3 Poisson regression model with horseshoe prior

MFVB algorithm

```
library(msos)

#Natural fixed-point iteration for the update of optimal
#variational density q.beta
update.q.beta.horseshoe <- function(y, x, x.y.sum, mu.q.beta,
                                    sigma2.q.beta, mu.q.1.lambda2,
                                    mu.q.1.tau2, mu.q.log.tau2,
                                    mu.q.log.lambda2, n, p,
                                    eps_beta, maxIter_beta,
                                    Trace_beta) {

  #============= Defining history variables =============
  elbo.out.beta <- numeric()
  mu.q.beta.out <- matrix(NA, nrow = maxIter_beta, ncol = p)
  sigma2.q.beta.out <- array(NA, dim = c(p,p, maxIter_beta))

  #============= Natural fixed-point iteration =============
  z <- 1
  cond.beta <- FALSE
  while(cond.beta != TRUE & z <= maxIter_beta) {

    #updating equations of q.mu
    exp.q.beta <- exp(x %*% mu.q.beta + 0.5 * diag(x %*%
        sigma2.q.beta %*% t(x)))
    nu.q.beta <- x.y.sum - t(x) %*% exp.q.beta -
        mu.q.beta*mu.q.1.lambda2*mu.q.1.tau2
    sigma2.q.beta <- - solve(- t(x) %*%
        diag(as.vector(exp.q.beta)) %*% x - mu.q.1.lambda2*
        mu.q.1.tau2)
    mu.q.beta <- mu.q.beta + sigma2.q.beta %*% nu.q.beta

    #computing beta-localized lower bound
    entropy.q.beta <- p/2* log(2*pi) + 0.5 *
      logdet(sigma2.q.beta) + 0.5 * p
    lprior.beta <- -p/2*log(2*pi) -p/2*mu.q.log.tau2 +
      0.5*sum(-mu.q.log.lambda2 - (mu.q.beta^2 +
      diag(sigma2.q.beta))*mu.q.1.lambda2*mu.q.1.tau2)
    logL <- t(y) %*% (x %*% mu.q.beta) - sum(lfactorial(y)) -
      (t(rep(1,n)) %*% exp(x %*% mu.q.beta + 0.5 *
      diag(x %*% sigma2.q.beta %*% t(x))))
    non.entropy.q.beta <- lprior.beta + logL
    ELBO.beta <- entropy.q.beta + non.entropy.q.beta

    #updating histories
    mu.q.beta.out[z,] <- mu.q.beta
    sigma2.q.beta.out[,,z] <- sigma2.q.beta
    elbo.out.beta <- c(elbo.out.beta, ELBO.beta)

    #stopping criterion
    if(z > 1) {
      delta.beta <- (abs((elbo.out.beta[z] - elbo.out.beta[z-1])
                         /elbo.out.beta[z-1]))

      if(delta.beta < eps_beta) cond.beta <- 1
      if (Trace_beta == 1) {
        print(paste0("iteration:", z,
              "- beta lower bound increase: ", delta.beta))
      }
    }
    if (z > maxIter_beta) cond.beta <- 1
    z <- z + 1
  }

  return(list(mu.q.beta = mu.q.beta.out[z-1,],
              sigma2.q.beta = sigma2.q.beta.out[,,z-1],
              entropy.q.beta = entropy.q.beta,
              lprior.beta = lprior.beta, logL = logL))
}

#MFVB algorithm
MFVB.horseshoe <- function(y, x, hyp, start, eps_beta = 1e-4,
                           eps_ELBO = 1e-4, maxIter_global = 100,
                           maxIter_beta = 10, Trace_global = 0,
                           Trace_beta = 0) {

  start.time <- Sys.time()
```

```r
n <- nrow(x)
p <- ncol(x)
x.y.sum <- colSums(y * x)

#============= Hyperparameters' setting =============
mu.beta <- hyp[[1]]
a.eta <- hyp[[2]]
b.eta <- hyp[[3]]
a.tau2 <- hyp[[4]]
a.nu <- hyp[[5]]
b.nu <- hyp[[6]]
a.lambda2 <- hyp[[7]]


#============= Defining history variables =============
elbo.out.global <- numeric()
mu.q.beta.global.out <- matrix(NA, nrow = maxIter_global,
                               ncol = p)
sigma2.q.beta.global.out <- array(NA, dim = c(p,p,maxIter_global))
a.q.eta.out <- 1
b.q.eta.out <- rep(NA, maxIter_global)
a.q.nu.out <- rep(1, p)
b.q.nu.out <- matrix(NA, nrow = maxIter_global, ncol = p)
a.q.tau2.out <- (p+1)/2
b.q.tau2.out <- matrix(NA, nrow = maxIter_global, ncol = p)
a.q.lambda2.out <- rep(1, p)
b.q.lambda2.out <- matrix(NA, nrow = maxIter_global, ncol = p)

#============= Initializing optimal parameters =============
mu.q.beta <- start[[1]]
sigma2.q.beta <- start[[2]]
omega.q.beta <- solve(sigma2.q.beta)
a.q.eta <- a.q.nu <-  a.q.lambda2 <- 1
a.q.tau2 <- (p+1)/2
b.q.eta <- start[[3]]
b.q.nu <- start[[4]]
b.q.tau2 <- start[[5]]
b.q.lambda2 <- start[[6]]
mu.q.1.tau2 <- a.q.tau2/b.q.tau2
mu.q.1.lambda2 <- a.q.lambda2/b.q.lambda2
mu.q.1.nu <- a.q.nu/b.q.nu
mu.q.log.eta <- log(b.q.eta) - digamma(a.q.eta)
mu.q.log.nu <- log(b.q.nu) - digamma(a.q.nu)
mu.q.log.tau2 <- log(b.q.tau2) - digamma(a.q.tau2)
mu.q.log.lambda2 <- log(b.q.lambda2) - digamma(a.q.lambda2)



#============= MFVB algorithm =============

i <- 1
cond.global <- FALSE
while(cond.global != TRUE & i <= maxIter_global) {



  #updating q.beta
  q.mu.update <- update.q.beta.horseshoe(y, x, x.y.sum, mu.q.beta,
                                         sigma2.q.beta,
                                         mu.q.1.lambda2,
                                         mu.q.1.tau2,
                                         mu.q.log.tau2,
                                         mu.q.log.lambda2,
                                         n, p, eps_beta,
                                         maxIter_beta, Trace_beta)
  mu.q.beta <- q.mu.update[[1]]
  sigma2.q.beta <- q.mu.update[[2]]

  #updating q.eta
  b.q.eta <- 1 + a.q.tau2/b.q.tau2
  mu.q.1.eta <- a.q.eta/b.q.eta
  mu.q.log.eta <- log(b.q.eta) - digamma(a.q.eta)

  #updating q.nu
  b.q.nu <- 1 + a.q.lambda2/b.q.lambda2
  mu.q.1.nu <- a.q.nu/b.q.nu
  mu.q.log.nu <- log(b.q.nu) - digamma(a.q.nu)

  #updating q.tau2
  b.q.tau2 <- a.q.eta/b.q.eta + 0.5*(sum((diag(sigma2.q.beta) +
              mu.q.beta^2)*a.q.lambda2/b.q.lambda2))
  mu.q.1.tau2 <- a.q.tau2/b.q.tau2
  mu.q.log.tau2 <- log(b.q.tau2) - digamma(a.q.tau2)

  #updating q.lambda2
  if(p == 1)
```

```
      b.q.lambda2 <- a.q.nu/b.q.nu + 0.5*(sigma2.q.beta +
                     mu.q.beta^2)*a.q.tau2/b.q.tau2
    else
      b.q.lambda2 <- a.q.nu/b.q.nu + 0.5*(diag(sigma2.q.beta) +
                     mu.q.beta^2)*a.q.tau2/b.q.tau2
    mu.q.1.lambda2 <- a.q.lambda2/b.q.lambda2
    mu.q.log.lambda2 <- log(b.q.lambda2) - digamma(a.q.lambda2)

    #computing lower bound
    entropy.q.beta <- q.mu.update[[3]]

    entropy.q.lambda2 <- sum(a.q.lambda2 + log(b.q.lambda2) +
                         lgamma(a.q.lambda2) - (a.q.lambda2 +1)*
                         digamma(a.q.lambda2))

    entropy.q.tau2 <- a.q.tau2 + log(b.q.tau2) +  lgamma(a.q.tau2) -
                      (a.q.tau2 +1)*digamma(a.q.tau2)

    entropy.q.eta <- a.q.eta + log(b.q.eta) + lgamma(a.q.eta) -
                     (a.q.eta +1)*digamma(a.q.eta)

    entropy.q.nu <- sum(a.q.nu + log(b.q.nu) +  lgamma(a.q.nu) -
                    (a.q.nu +1)*digamma(a.q.nu))

    lprior.beta <- -p/2*log(2*pi) -p/2*mu.q.log.tau2 -
                   0.5*sum(mu.q.log.lambda2)-
                   0.5*sum((diag(sigma2.q.beta)
                   + mu.q.beta^2)*mu.q.1.lambda2*mu.q.1.tau2)

    logL <- q.mu.update[[5]]

    lprior.lambda2 <- -p*lgamma(0.5) - 0.5*sum(0.5*mu.q.log.nu +
                      3/2*mu.q.log.lambda2 +
                      mu.q.1.nu*mu.q.1.lambda2)

    lprior.tau2 <- -0.5*mu.q.log.eta - lgamma(0.5)
                   - 3/2*mu.q.log.tau2 - mu.q.1.eta*mu.q.1.tau2

    lprior.eta <- -lgamma(0.5) - 3/2*mu.q.log.eta - mu.q.1.eta

    lprior.nu <- -p*lgamma(0.5) - sum(3/2*mu.q.log.nu + mu.q.1.nu)
```

```
  ELBO.global <- entropy.q.beta +  entropy.q.lambda2 +
                 entropy.q.tau2 + entropy.q.eta + entropy.q.nu +
                 logL + lprior.beta + lprior.lambda2  +
                 lprior.tau2 + lprior.eta + lprior.nu

  #updating history variables
  elbo.out.global <- c(elbo.out.global, ELBO.global)
  mu.q.beta.global.out[i,] <- mu.q.beta
  sigma2.q.beta.global.out[,,i] <- sigma2.q.beta
  b.q.eta.out[i] <- b.q.eta
  b.q.nu.out[i,] <- b.q.nu
  b.q.tau2.out[i] <- b.q.tau2
  b.q.lambda2.out[i,] <- b.q.lambda2

  #stopping criterion
  if(i > 1) {
    Delta.global <- (abs((elbo.out.global[i] -
              elbo.out.global[i-1])/elbo.out.global[i-1]))
    if(Delta.global < eps_global) cond.global <- 1
    if (Trace_global == 1) {
      print(paste0("iteration:", i,
            "- global lower bound increase: ",Delta.global))
    }
  }
  if (i > maxIter_global) cond.global <- 1
  i <- i + 1
}

end.time <- Sys.time()

#============= Computing computational effort =============
total.time <- end.time - start.time

return(list(elbo = elbo.out.global,
            mu.q.beta =as.matrix(mu.q.beta.global.out[1:(i-1),]),
            sigma2.q.beta=sigma2.q.beta.global.out[,,1:(i-1)],
            a.q.eta = a.q.eta.out,
            b.q.eta = b.q.eta.out[1:(i-1)], a.q.nu = a.q.nu.out,
            b.q.nu = as.matrix(b.q.nu.out[1:(i-1),]),
            a.q.tau2 = a.q.tau2.out, b.q.tau2 = b.q.tau2[1:(i-1)],
            a.q.lambda2 = a.q.lambda2.out,
            b.q.lambda2 = as.matrix(b.q.lambda2.out[1:(i-1),]),
              iter = i-1, tempo = total.time[[1]]))
}
```

SAVS algorithm

```
savs.algorithm <- function(mu.q.beta, X){
  p <- ncol(X)
  mu.q.beta.star <- numeric(p)
  for(j in 1:p){
    mu.j <- 1/mu.q.beta[j]^2
    sum.x.j.2 <- sum(X[,j]^2)
    if(abs(mu.q.beta[j])*sum.x.j.2<= mu.j)
      mu.q.beta.star[j] <- 0
    else
      mu.q.beta.star[j] <- sign(mu.q.beta[j])/sum.x.j.2*
      (abs(mu.q.beta[j])*sum.x.j.2 - mu.j)
  }
  return(mu.q.beta.star)
}
```

A.4 Poisson regression model with spike-and-slab prior

MFVB algorithm

```
library(msos)

#Natural fixed-point iteration for the update of optimal
#variational density q.beta
update.q.beta.spike <- function(y, x, x.y.sum, mu.q.beta,
                                sigma2.q.beta, mu.q.gamma,
                                mu.q.1.lambda1, mu.q.log.lambda1,
                                lambda0,   n, p, eps_beta,
                                maxIter_beta, Trace_beta) {
  #============= Defining history variables =============
  elbo.out.beta <- numeric()
  mu.q.beta.out <- matrix(NA, nrow = maxIter_beta, ncol = p)
  sigma2.q.beta.out <- array(NA, dim = c(p,p, maxIter_beta))

  #============= Natural fixed-point iteration =============
  z <- 1
  cond.beta <- FALSE
```

```
while(cond.beta != TRUE & z <= maxIter_beta) {

  #updating equations of q.mu
  exp.q.beta <- exp(x %*% mu.q.beta + 0.5 * diag(x %*%
                sigma2.q.beta %*% t(x)))
  nu.q.beta <- x.y.sum - t(x) %*% exp.q.beta -
               (1-mu.q.gamma)*mu.q.beta/lambda0 -
               mu.q.gamma*mu.q.beta*mu.q.1.lambda1
  sigma2.q.beta <-  solve((1-mu.q.gamma)/lambda0 +
                               mu.q.gamma*mu.q.1.lambda1+
                               t(x) %*% diag(as.vector(exp.q.beta))
                           %*% x)
  mu.q.beta <- mu.q.beta + sigma2.q.beta %*% nu.q.beta

  #computing beta-localized lower bound
  entropy.q.beta <- 0.5*p* log(2*pi) + 0.5 *logdet(sigma2.q.beta)+
                    0.5 * p
  lprior.beta <- -p/2*log(2*pi) + sum(-(1-mu.q.gamma)*
                 (log(lambda0)/2 + (mu.q.beta^2 +
                 diag(sigma2.q.beta))/(2*lambda0)))+
                 sum(-mu.q.gamma* 0.5*(mu.q.log.lambda1 +
                 (mu.q.beta^2 + diag(sigma2.q.beta))/(2)*
                 mu.q.1.lambda1))
  logL <- t(y) %*% (x %*% mu.q.beta) - sum(lfactorial(y)) -
    (t(rep(1,n)) %*% exp(x %*% mu.q.beta + 0.5 * diag(x %*%
    sigma2.q.beta %*% t(x))))
  non.entropy.q.beta <- lprior.beta + logL
  ELBO.beta <- entropy.q.beta + non.entropy.q.beta

  #updating histories
  mu.q.beta.out[z,] <- mu.q.beta
  sigma2.q.beta.out[,,z] <- sigma2.q.beta
  elbo.out.beta <- c(elbo.out.beta, ELBO.beta)

  #stopping criterion
  if(z > 1) {
    delta.beta <- (abs((elbo.out.beta[z] - elbo.out.beta[z-1])/
                  elbo.out.beta[z-1]))
    if(delta.beta < eps_beta) cond.beta <- 1
    if (Trace_beta == 1) {
      print(paste0("iteration:", z,
            "- beta lower bound increase: ",delta.beta))
      }
    }
    if (z > maxIter_beta) cond.beta <- 1
    z <- z + 1
  }

  return(list(mu.q.beta = mu.q.beta.out[z-1,],
              sigma2.q.beta = sigma2.q.beta.out[,,z-1],
              entropy.q.beta = entropy.q.beta,
              lprior.beta = lprior.beta, logL = logL))
}

#MFVB algorithm
MFVB.spike.slab <- function(y, x, hyp, start, eps_beta, eps_global,
                            maxIter_beta, maxIter_gamma,
                            maxIter_global, Trace_beta = 0,
                            Trace_global = 0) {

  n <- nrow(x)
  p <- ncol(x)
  x.y.sum <- colSums(y * x)

  #============= Hyperparameters' setting =============
  mu.beta <- hyp[[1]][[1]]
  r.lambda1 <- hyp[[2]][[1]]
  delta.lambda1 <- hyp[[2]][[2]]
  a.theta <- hyp[[3]][[1]]
  b.theta <- hyp[[3]][[2]]
  lambda0 <- hyp[[4]]

  #============= Defining history variables =============
  elbo.out.global <- numeric()
  mu.q.beta.global.out <- matrix(NA, nrow = maxIter_global,
                          ncol = p)
  sigma2.q.beta.global.out <- array(NA, dim =c(p,p,maxIter_global))
  mu.q.gamma.global.out <- matrix(NA, nrow = maxIter_global,
                           ncol = p)
  r.q.lambda1.out <- matrix(NA, nrow = maxIter_global, ncol = p)
  delta.q.lambda1.out <- matrix(NA, nrow = maxIter_global, ncol = p)
  a.q.theta.out <- numeric()
  b.q.theta.out <- numeric()
```

```
#============= Initializing optimal parameters =============
mu.q.beta <- start[[1]]
sigma2.q.beta <- start[[2]]
omega.q.beta <- solve(sigma2.q.beta)
mu.q.gamma <- start[[3]]
w <- log(mu.q.gamma / (1-mu.q.gamma))
r.q.lambda1 <- start[[4]]
delta.q.lambda1 <- start[[5]]
a.q.theta <- start[[6]]
b.q.theta <- start[[7]]
mu.q.log.theta <- digamma(a.q.theta) - digamma(a.q.theta +
                  b.q.theta)
mu.q.log.1.theta <- digamma(b.q.theta) - digamma(a.q.theta +
                    b.q.theta)
mu.q.1.lambda1 <- r.q.lambda1/delta.q.lambda1
mu.q.log.lambda1 <- log(delta.q.lambda1) - digamma(r.q.lambda1)


#============= MFVB algorithm =============
start.time <- Sys.time()

i <- 1
cond.global <- FALSE
while(cond.global != TRUE & i <= maxIter_global) {

  #updating q.beta
  q.mu.update <- update.q.beta.spike(y, x, x.y.sum, mu.q.beta,
                                     sigma2.q.beta, mu.q.gamma,
                                     mu.q.1.lambda1,
                                     mu.q.log.lambda1,lambda0,
                                     n, p,eps_beta,
                                     maxIter_beta, Trace_beta)
  mu.q.beta <- q.mu.update[[1]]
  sigma2.q.beta <- q.mu.update[[2]]

  #updating q.gamma
  for(j in 1:p){
    w[j] <- mu.q.log.theta  - mu.q.log.1.theta -
      (mu.q.beta[j]^2 + diag(sigma2.q.beta)[j])*
      (mu.q.1.lambda1[j]-1/lambda0)/2-
      0.5*(mu.q.log.lambda1[j] - log(lambda0))
    mu.q.gamma[j] <- 1/(1 + exp(-w[j]))
```

```
}
mu.q.gamma[which(mu.q.gamma == 1.)] <- 0.99
mu.q.gamma[which(mu.q.gamma == 0.)] <- 0.01

#updating q.lambda
r.q.lambda1 <- r.lambda1 + mu.q.gamma/2
if(p == 1)
  delta.q.lambda1 <- delta.lambda1 +
  mu.q.gamma*(mu.q.beta^2 + sigma2.q.beta)/(2)
else
  delta.q.lambda1 <- delta.lambda1 +
  mu.q.gamma*(mu.q.beta^2 + diag(sigma2.q.beta))/(2)
mu.q.1.lambda1 <- r.q.lambda1/delta.q.lambda1
mu.q.log.lambda1 <- log(delta.q.lambda1) - digamma(r.q.lambda1)

#updating q.theta
a.q.theta <- a.theta + sum(mu.q.gamma)
b.q.theta <- b.theta + p - sum(mu.q.gamma)
mu.q.log.theta <- digamma(a.q.theta) - digamma(a.q.theta +
                  b.q.theta)
mu.q.log.1.theta <- digamma(b.q.theta) - digamma(a.q.theta +
                    b.q.theta)

#computing lower bound
entropy.q.beta <- q.mu.update[[3]]

entropy.q.gamma <- - sum(mu.q.gamma * log(mu.q.gamma) +
                   (1-mu.q.gamma) * log(1-mu.q.gamma))

entropy.q.lambda1 <- sum(-r.q.lambda1*log(delta.q.lambda1) +
                     lgamma(r.q.lambda1)+ delta.q.lambda1*
                     mu.q.1.lambda1+ (r.q.lambda1+1)*
                     mu.q.log.lambda1)

entropy.q.theta <- -(a.q.theta - 1)* mu.q.log.theta -
                   (b.q.theta - 1)*mu.q.log.1.theta +
                   lbeta(a.q.theta, b.q.theta)

lprior.beta <- -p/2*log(2*pi)+ sum(-(1-mu.q.gamma)*
               (log(lambda0)/2 + (mu.q.beta^2 +
               diag(sigma2.q.beta))/(2*lambda0)))+
               sum(-(mu.q.gamma)* 0.5*mu.q.log.lambda1 *
              (mu.q.beta^2 + diag(sigma2.q.beta))/(2)*
              mu.q.1.lambda1)

lprior.gamma <- sum((1-mu.q.gamma)*mu.q.log.1.theta+
                mu.q.gamma*mu.q.log.theta)

lprior.theta <- (a.theta - 1)*mu.q.log.theta + (b.theta - 1)*
                mu.q.log.1.theta - lbeta(a.theta, b.theta)

lprior.lambda1 <- p*r.lambda1[1]*log(delta.lambda1[1]) -
                  p*lgamma(r.lambda1[1]) - delta.lambda1[1]*
                  sum(mu.q.1.lambda1)- (r.lambda1[1] + 1)*
                  sum(mu.q.log.lambda1)

logL <- q.mu.update[[5]]

ELBO.global <- entropy.q.beta + entropy.q.gamma +
               entropy.q.lambda1 + entropy.q.theta +
               lprior.beta + lprior.gamma + logL +
               lprior.lambda1 + lprior.theta

#updating history variables
elbo.out.global <- c(elbo.out.global, ELBO.global)
mu.q.beta.global.out[i,] <- mu.q.beta
sigma2.q.beta.global.out[,,i] <- sigma2.q.beta
mu.q.gamma.global.out[i,] <- mu.q.gamma
r.q.lambda1.out[i,] <- r.q.lambda1
delta.q.lambda1.out[i,] <- delta.q.lambda1
a.q.theta.out <- c(a.q.theta.out, a.q.theta)
b.q.theta.out <- c(b.q.theta.out, b.q.theta)

#stopping criterion
if(i > 1) {
  Delta.global <- (abs((elbo.out.global[i] -
                  elbo.out.global[i-1])/
                  elbo.out.global[i-1]))
  if(Delta.global < eps_global) cond.global <- 1
  if (Trace_global == 1) {
    print(paste0("iteration:", i,
          "- global lower bound increase: ", Delta.global))
  }
}
if (i > maxIter_global) cond.global <- 1
i <- i + 1
}

end.time <- Sys.time()

#============= Computing computational effort =============
total.time <- end.time - start.time

return(list(elbo = elbo.out.global,
          mu.q.beta=as.matrix(mu.q.beta.global.out[1:(i-1),]),
          mu.q.gamma=as.matrix(mu.q.gamma.global.out[1:(i-1),]),
          sigma2.q.beta=sigma2.q.beta.global.out[,,1:(i-1)],
          a.q.theta = a.q.theta.out,
          b.q.theta = b.q.theta.out,
          r.q.lambda1=as.matrix(r.q.lambda1.out[1:(i-1),]),
          delta.q.lambda1=as.matrix(delta.q.lambda1.out[1:(i-1),]),
          iter = i-1, tempo = total.time[[1]]))
}
```

A.5 Poisson regression model with Bernoulli-Gaussian prior

MFVB algorithm

```
#Function for the creation of matrix (n, p) with each row composed
#by the elements in the production for the update of optimal
#variational densities  q.beta.j and q.gamma.j
prod.elementi.start <- function(x, mu.q.gamma, mu.q.beta,
                                sigma2.q.beta, n, p) {
  matrice.prod <- matrix(NA, nrow = n, ncol = p)
  if(p == 1)
    matrice.prod <- as.vector(1 - mu.q.gamma + mu.q.gamma*exp(x *
                    mu.q.beta + sigma2.q.beta * x^2/2))
  else{
    for(j in 1:p)
      matrice.prod[,j] <- 1 - mu.q.gamma[j] + mu.q.gamma[j]*
        exp(x[,j] * mu.q.beta[j] + (sigma2.q.beta[j] * x[,j]^2)/
        2)
  }
  return(as.matrix(matrice.prod))
}

#Function for the change of the column j in the matrix of elements
#in the production for the update of q.beta.j and q.gamma.j
change.prod.elementi.j <- function(x, mu.q.gamma, mu.q.beta,
                                   sigma2.q.beta, j, p) {
  if(p == 1)
    return(1-mu.q.gamma + mu.q.gamma*exp(x * mu.q.beta +
           sigma2.q.beta * x^2/2))
  else
    return(1-mu.q.gamma[j] + mu.q.gamma[j]*exp(x[,j] *
           mu.q.beta[j] + (sigma2.q.beta[j] * x[,j]^2)/2))
}


#Function for the computation of the production in the update of
#optimal variational densities q.beta.j and q.gamma.j
produttoria.q.beta.q.gamma <- function(elementi.prod, j ){
  elementi.prod.partial <- as.matrix(elementi.prod[,-j])
  return(apply(elementi.prod.partial, 1, function(x) prod(x)))
}


#Natural fixed-point iteration for the update of optimal
#variational density q.beta
update.q.beta.j <- function(y, x, sigma2.beta, mu.q.beta,
                            sigma2.q.beta, mu.q.gamma,
                          elementi.prod, maxIter_beta, eps_beta,
                          Trace_beta, j, nu.q.beta) {

  n <- nrow(x)
  p <- ncol(x)


  #============= Defining history variables =============
  elbo.out.beta <- c()
  mu.q.beta.out <- rep(NA, maxIter_beta)
  sigma2.q.beta.out <- rep(NA, maxIter_beta)
  nu.q.beta.out <- rep(NA,maxIter_beta)

  #============= Natural fixed-point iteration =============
    z <- 1
    cond.beta <- FALSE
    if(p == 1)
      produttoria.no.j <- rep(1,n)
    else
      produttoria.no.j<-produttoria.q.beta.q.gamma(elementi.prod,
                        j)
    while(cond.beta != TRUE & z <= maxIter_beta) {

      #updating equations of q.mu
      exp.q.beta.j <- mu.q.gamma[j] * exp(x[,j]*mu.q.beta[j] +
                      sigma2.q.beta[j]*(x[,j]^2)/2)
      nu.q.beta[j] <- -mu.q.beta[j]/sigma2.beta  + t(x[,j] *
                     mu.q.gamma[j]) %*% y - t(produttoria.no.j) %*%
                    (x[,j] * exp.q.beta.j)
      sigma2.q.beta[j] <-  as.vector(solve( 1/sigma2.beta +
                          t(produttoria.no.j) %*%
                          (x[,j]^2 * exp.q.beta.j)))
      mu.q.beta[j] <- mu.q.beta[j] + sigma2.q.beta[j]*nu.q.beta[j]
      elementi.prod[,j] <- change.prod.elementi.j(x, mu.q.gamma,
                           mu.q.beta, sigma2.q.beta, j, p)

      #computing beta-localized lower bound

      produttoria.total <- apply(elementi.prod, 1, function(x)
                           prod(x))

      entropy.q.beta.j <- 0.5*log(sigma2.q.beta[j]) + 1/2 +
                          0.5*log(2*pi)
      lprior.beta.j <- - 0.5*log(2*pi) - 1/2*log(sigma2.beta) -
                       1/(2*sigma2.beta)*(sigma2.q.beta[j] +
                       mu.q.beta[j]^2)
      if(p == 1){
        logL <- as.vector( t(x * mu.q.gamma * mu.q.beta) %*% y -
                sum(lfactorial(y)) - sum(elementi.prod))
      } else{
        logL <- as.vector(t(x %*% diag(mu.q.gamma) %*% mu.q.beta)
                %*% y) - sum(lfactorial(y)) -
                sum(produttoria.total)
      }
      non.entropy.q.beta.j <- lprior.beta.j + logL
```

```
      ELBO.beta.j <- entropy.q.beta.j + lprior.beta.j + logL

      #updating histories
      mu.q.beta.out[z] <- mu.q.beta[j]
      sigma2.q.beta.out[z] <- sigma2.q.beta[j]
      nu.q.beta.out[z] <- nu.q.beta[j]
      elbo.out.beta <- c(elbo.out.beta, ELBO.beta.j)

      #stopping criterion
      if(z > 1) {
        Delta.beta <- (abs((elbo.out.beta[z] -
                      elbo.out.beta[z-1])/
                      elbo.out.beta[z-1]))
        if(Delta.beta < eps_beta) cond.beta <- 1
        if(Trace_beta == 1) {
          print(paste0("iteration:", z,
              "- beta lower bound increase: ", Delta.beta))
        }
      }
      if (z > maxIter_beta) cond.beta <- 1
      z <- z + 1
      }

  return(list(mu.q.beta.j = mu.q.beta.out[z-1],
              sigma2.q.beta.j = sigma2.q.beta.out[z-1],
              nu.q.beta.j = nu.q.beta.out[z-1]))
}

#gamma.j-localized component of lower bound
update.gamma.j <- function(mu.q.gamma.j, y, x, alpha.rho,
                           delta.rho, mu.q.beta, sigma2.q.beta,
                           elementi.prod, mu.q.log.rho,
                           mu.q.log.1.rho, mu.q.gamma, j) {
  p <- ncol(x)
  n <- nrow (x)


  entropy.q.gamma <- - (mu.q.gamma.j * log(mu.q.gamma.j) +
                     (1-mu.q.gamma.j)*log(1-mu.q.gamma.j))

  if(p == 1) {
    logL <- as.vector(t(x * mu.q.gamma.j * mu.q.beta) %*% y -
            sum(lfactorial(y)) -sum(1 - mu.q.gamma.j +
            mu.q.gamma.j*exp(x*mu.q.beta + sigma2.q.beta*x^2/2)))
  }
  else if(p == 2){
    produttoria.no.j <- produttoria.q.beta.q.gamma(elementi.prod,
                        j)
    logL <- (t(x[,-j] * mu.q.gamma[-j] * mu.q.beta[-j]) %*% y) +
            (t(x[,j] * mu.q.gamma.j * mu.q.beta[j]) %*% y) -
            sum(lfactorial(y)) - sum(produttoria.no.j *
            (1 - mu.q.gamma.j + mu.q.gamma.j*exp(x[,j]*
            mu.q.beta[j] + sigma2.q.beta[j]*(x[,j]^2)/2)))
  }
  else{
    produttoria.no.j <- produttoria.q.beta.q.gamma(elementi.prod,
                        j)
    logL <- (t(x[,-j] %*% diag(mu.q.gamma[-j]) %*% mu.q.beta[-j])
             %*% y) + (t(x[,j] * mu.q.gamma.j * mu.q.beta[j])%*%
             y) - sum(lfactorial(y)) - sum(produttoria.no.j *
            (1 - mu.q.gamma.j + mu.q.gamma.j*exp(x[,j]*
            mu.q.beta[j] + sigma2.q.beta[j]*(x[,j]^2)/2)))
  }

  lprior.gamma <- mu.q.gamma.j*mu.q.log.rho +(1-mu.q.gamma.j)*
                  mu.q.log.1.rho

  ELBO.gamma <- entropy.q.gamma + lprior.gamma + logL
  return(ELBO.gamma)
}


#first derivative of gamma.j-localized component of lower bound
der.prime.gamma.j <- function(mu.q.gamma.j, y, x, mu.q.beta,
                              sigma2.q.beta, mu.q.gamma,
                              elementi.prod, mu.q.log.rho,
                              mu.q.log.1.rho, j) {
  p <- ncol(x)

  if(p == 1)
    produttoria.no.j <- rep(1, n)
  else
    produttoria.no.j <- produttoria.q.beta.q.gamma(elementi.prod,
                        j)
  output <- mu.q.log.rho - mu.q.log.1.rho - log(mu.q.gamma.j) -
            1 + 1/(1-mu.q.gamma.j) + log(1 - mu.q.gamma.j)-
            mu.q.gamma.j/(1 - mu.q.gamma.j) + t(x[,j] *
            mu.q.beta[j]) %*% y - sum(produttoria.no.j *
            (exp(x[,j]*mu.q.beta[j] + sigma2.q.beta[j]*(x[,j]^2)
            /2) -1))
  return(output)
}

#MFVB algorithm
MFVB.norm.bern.beta <- function(y, x, hyp, start, eps_beta,
                                eps_gamma, eps_global,
                                maxIter_beta,
                                maxIter_gamma, maxIter_global,
                                Trace_beta = 0,Trace_global = 0){
  start.time <- Sys.time()

  #============= Hyperparameters' setting =============
  mu.beta <- hyp[[1]][[1]]
  sigma2.beta <- hyp[[1]][[2]]
  alpha.rho <- hyp[[2]][[1]]
  delta.rho <- hyp[[2]][[2]]
  n <- nrow(x)
  p <- ncol(x)

  #============= Defining history variables =============
  elbo.out.global <- numeric()
  mu.q.beta.global.out <- matrix(NA, nrow = maxIter_global,
                          ncol = p)
  sigma2.q.beta.global.out <- matrix(NA, ncol = p,
                              nrow =  maxIter_global)
  mu.q.gamma.out <- matrix(NA, nrow = maxIter_global, ncol = p)
  alpha.q.rho.out <- numeric()
  delta.q.rho.out <- numeric()
  nu.q.beta.global.out <- matrix(NA, nrow = maxIter_global,
                                 ncol = p)

  #============= Initializing optimal parameters =============
  mu.q.beta <- start[[1]]
  sigma2.q.beta <- start[[2]]
  omega.q.beta <- 1/sigma2.q.beta
```

```
mu.q.gamma <- start[[3]]
alpha.q.rho <- start[[4]]
delta.q.rho <- start[[5]]
mu.q.log.rho <- digamma(alpha.q.rho) + digamma(alpha.q.rho +
                delta.q.rho)
mu.q.log.1.rho <- digamma(delta.q.rho) + digamma(alpha.q.rho +
                  delta.q.rho)

nu.q.beta <- mu.q.beta

elementi.prod <- prod.elementi.start(x, mu.q.gamma, mu.q.beta,
                                     sigma2.q.beta, n, p)

#============= MFVB algorithm =============
i <- 1
cond.global <- FALSE

while(cond.global != TRUE & i <= maxIter_global) {

  #updating q.beta
  for(j in 1:p){
    q.beta.j.update <- update.q.beta.j(y, x, sigma2.beta,
                       mu.q.beta, sigma2.q.beta, mu.q.gamma,
                       elementi.prod, maxIter_beta[j],
                       eps_beta[j], Trace_beta[j], j,
                       nu.q.beta)
    mu.q.beta[j] <- q.beta.j.update[[1]]
    sigma2.q.beta[j] <- q.beta.j.update[[2]]
    nu.q.beta[j] <- q.beta.j.update[[3]]
    }

  #updating q.gamma
  for(j in 1:p){
    mu.q.gamma[j] <- optim(mu.q.gamma[j],
                           function(z) -update.gamma.j(z, y, x,
                                        alpha.rho, delta.rho,
                                        mu.q.beta,
                                        sigma2.q.beta,
                                        elementi.prod,
                                        mu.q.log.rho,
                                        mu.q.log.1.rho,
                                        mu.q.gamma, j),
                         function(z) -der.prime.gamma.j(z, y,
                                      x, mu.q.beta,
                                      sigma2.q.beta,
                                      mu.q.gamma,
                                      elementi.prod,
                                      mu.q.log.rho,
                                      mu.q.log.1.rho, j),
                         method = "L-BFGS-B",
                         lower = 0.01, upper = 0.99)$par
  elementi.prod[,j] <- change.prod.elementi.j(x, mu.q.gamma,
                       mu.q.beta, sigma2.q.beta, j, p)
  }

#updating q.rho
alpha.q.rho <- alpha.rho + sum(mu.q.gamma)
delta.q.rho <- delta.rho + p - sum(mu.q.gamma)
mu.q.log.rho <- digamma(alpha.q.rho) + digamma(alpha.q.rho +
                delta.q.rho)
mu.q.log.1.rho <- digamma(delta.q.rho) +digamma(alpha.q.rho+
                  delta.q.rho)

#computing lower bound

entropy.q.beta <- p/2*log(2*pi)+0.5*sum(log(sigma2.q.beta)) +
                  p/2

entropy.q.gamma <- - sum(mu.q.gamma * log(mu.q.gamma) +
                  (1-mu.q.gamma) * log(1-mu.q.gamma))

entropy.q.rho <- -(alpha.q.rho - 1)* mu.q.log.rho -
                 (delta.q.rho - 1)*mu.q.log.1.rho +
                 lbeta(alpha.q.rho, delta.q.rho)

lprior.beta <- -p/2*log(2*pi) - p/2*log(sigma2.beta) -
               1/(2*sigma2.beta)*sum(sigma2.q.beta+
               mu.q.beta^2)

lprior.gamma <- sum(mu.q.gamma*mu.q.log.rho +(1-mu.q.gamma)*
                mu.q.log.1.rho)

if(p == 1){
  logL <- as.vector(t(x * mu.q.gamma * mu.q.beta) %*% y -
            sum(lfactorial(y)) - sum(elementi.prod))
  }
  else{
    produttoria.total <- apply(elementi.prod, 1, function(x)
                         prod(x))
    logL <- as.vector(t(x %*%diag(mu.q.gamma)%*%mu.q.beta) %*%
            y) - sum(lfactorial(y)) - sum(produttoria.total)
  }

  lprior.rho <- (alpha.rho - 1)*mu.q.log.rho + (delta.rho - 1)*
                mu.q.log.1.rho - lbeta(alpha.rho, delta.rho)

  ELBO.global <- entropy.q.beta + entropy.q.gamma +
                 entropy.q.rho + lprior.beta +
                 lprior.gamma + logL + lprior.rho

  #updating history variables
  mu.q.beta.global.out[i,] <- mu.q.beta
  sigma2.q.beta.global.out[i,] <- sigma2.q.beta
  mu.q.gamma.out[i,] <- mu.q.gamma
  alpha.q.rho.out <- c(alpha.q.rho.out, alpha.q.rho)
  delta.q.rho.out <- c(delta.q.rho.out, delta.q.rho)
  elbo.out.global <- c(elbo.out.global, ELBO.global)

  #stopping criterion
  if(i > 1) {
    Delta.global <- (abs((elbo.out.global[i] -
                    elbo.out.global[i-1])/
                    elbo.out.global[i-1]))
    if(Delta.global < eps_global) cond.global <- 1
    if (Trace_global == 1) {
      print(paste0("iteration:", i,
          "- global lower bound increase: ", Delta.global))
    }
  }
  if (i > maxIter_global) cond.global <- 1
  i <- i + 1
}

end.time <- Sys.time()

#============= Computing computational effort =============
  total.time <- end.time - start.time

  return(list(elbo = elbo.out.global,
              mu.q.beta = mu.q.beta.global.out[1:(i-1),],
              mu.q.gamma = mu.q.gamma.out[1:(i-1),],
              sigma2.q.beta=sigma2.q.beta.global.out[1:(i-1),],
              alpha.q.rho = alpha.q.rho.out,
              delta.q.rho = delta.q.rho.out, iter = i-1,
              tempo = total.time[[1]]))
}
```

Bibliography

Bernardi, M., Bianchi, D. and Bianco, N. (2023). Dynamic variable selection in high-dimensional predictive regressions, *arXiv preprint arXiv:2304.07096* .

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017). Variational inference: A review for statisticians, *Journal of the American Statistical Association* **112**: 859–877.

Brown, P., Vannucci, M. and Fearn, T. (2002). Multivariate bayesian variable selection and prediction, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **60**: 627–641.

Carvalho, C., Polson, N. and Scott, J. (2010). The horseshoe estimator for sparse signals, *Access* **0**: 465–480.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J. and Blei, D. M. (2017). Variational inference via χ upper bound minimization, *Advances in Neural Information Processing Systems* **30**: 2733–2742.

Friedman, J., Hastie, T., Höfling, H. and Tibshirani, R. (2007). Pathwise coordinate optimization, *The Annals of Applied Statistics* **1**.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities, *Journal of the American Statistical Association* **85**: 398–409.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences, *Statistical Science* **7**: 457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. ieee trans. pattern anal. mach. intell. pami-6(6), 721-741, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**: 721–741.

George, E. and McCulloch, R. (1993). Variable selection via gibbs sampling, *Journal of The American Statistical Association* **88**: 881–889.

George, E. and McCulloch, R. (1997). Approaches for bayesian variable selection, *Statistica Sinica* **7**: 339–373.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications, *Biometrika* **57**: 97–109.

Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights, *Annual Conference Computational Learning Theory*.

Jordan, M., Ghahramani, Z., Jaakkola, T. and Saul, L. (1999). An introduction to variational methods for graphical models, *Machine Learning* **37**: 183–233.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency, *The Annals of Mathematical Statistics* **22**: 79–86.

Minka, T. (2005). Divergence measures and message passing.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression, *Journal of the American Statistical Association* **83**: 1023–1032.

Narisetty, N. N. and He, X. (2014). Bayesian variable selection with shrinking and diffusing priors, *The Annals of Statistics* **42**: 789–817.

Ormerod, J. and Wand, M. (2010). Explaining variational approximations, *Centre for Statistical & Survey Methodology Working Paper Series* **64**: 140–153.

Ormerod, J., You, C. and Muller, S. (2017). A variational bayes approach to variable selection, *Electronic Journal of Statistics* **11**: 3549–3594.

Panagiotelis, A. and Smith, M. (2008). Bayesian identification, selection and estimation of semiparametric functions in high-dimensional additive models, *Journal of Econometrics* **143**: 291–316.

Parisi, G. (1988). *Statistical field theory*, Frontiers in Physics, Addison-wesley.

Park, T. and Casella, G. (2008). The bayesian lasso, *Journal of the American Statistical Association* **103**: 681–686.

Ray, P. and Bhattacharya, A. (2018). Signal adaptive variable selector for the horseshoe prior, *arXiv: Methodology* .

Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*, Vol. 2, Springer.

Robert, C. and Casella, G. (2011). A short history of markov chain monte carlo: Subjective recollections from incomplete data, *Statistical Science* **26**.

Rohde, D. and Wand, M. (2016). Semiparametric mean field variational bayes: General principles and numerical issues, *Journal of Machine Learning Research,* **17**: 5975–6021.

Ročková, V. and George, E. I. (2014). EMVS: The EM Approach to Bayesian Variable Selection, *Journal of the American Statistical Association* **109**: 828–846.

Rustagi, J. (1976). *Variational Methods in Statistics*, Mathematics in science and engineering : a series of monographs and textbooks, Academic Press.

Sakurai, J. J. (1994). *Modern quantum mechanics; rev. ed.*, Addison-Wesley, Reading, MA.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* **58**: 267–288.

Wand, M., Ormerod, J., Padoan, S. and Fuhrwirth, R. (2011). Mean field variational bayes for elaborate distributions, *Bayesian Analysis* **6**.

Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (2018). Yes, but did it work?: Evaluating variational inference, pp. 5581–5590.