An Automated Prior Robustness Analysis in Bayesian Model Comparison

Benjamin Maurel
Gabriel Watkinson

Contents

1 Introduction

In this report, we present a methodology developed in the paper **An automated prior robustness analysis in Bayesian model comparison** by Chan, Jacobi, and Zhu [1] for assessing the sensitivity of marginal likelihoods to prior hyperparameters in Bayesian statistics. The marginal likelihood is a key concept in Bayesian statistics, playing a central role in various tasks such as model comparison and averaging, but is typically estimated using computationally intensive simulation-based methods. One well-known drawback of the marginal likelihood is its sensitivity to the choice of prior, as small changes in the prior can greatly affect the marginal likelihood. For example, in a model comparison scenario, a small change in the prior of one model can cause it to have a higher marginal likelihood than a competing model, leading to a false conclusion about which model is more favoured by the data. As a result, sensitivity analysis for marginal likelihoods is important but is not routinely done in empirical work due to computational complexity. To address this issue, the authors propose a computationally feasible and systematic approach that uses Automatic Differentiation (AD) to compute partial derivatives of the marginal likelihood with respect to various hyperparameters.

These partial derivatives can be used in a number of ways: first, they can be used to identify key hyperparameters that have the greatest impact on the marginal likelihood values, allowing for a more focused analysis of those influential hyperparameters, for example, eliciting those prior hyperparameters more carefully or choosing their values optimally. Secondly, the partial derivatives can be used to analyse how small changes in any hyperparameter would impact the ranking of a set of competing models, giving the user a convenient way to assess prior sensitivities without re-estimating all the models and re-computing the corresponding marginal likelihoods. Finally, it is common in many applications to adopt an empirical Bayes approach, which selects hyperparameters values by maximizing the marginal likelihood. For most models in which the marginal likelihood needs to be estimated using Monte Carlo methods, this brute-force approach is typically computationally infeasible. In those cases, the partial derivatives of the marginal likelihood become necessary in order to speed up the marginal likelihood maximization problem.

To illustrate the feasibility of the proposed methodology, we realize an empirical application on simulated data, that compares two Gaussian models. We use AD to compute the partial derivatives of Chib's marginal likelihood estimator with respect to a variety of hyperparameters and assess various aspects of the marginal likelihood sensitivity. To do so, we reimplemented the algorithms using Python and JAX for the Automatic differentiation.

The report will be organized as follows: first, we will present the marginal likelihood estimation for both the Chib's estimator and the cross-entropy estimator. Next, we will explain how we use the Automatic Differentiation method to compute the partial derivatives for both estimators. Afterwards, we will present our experiments and results on the Gaussian models. We will discuss the results and draw conclusions about the feasibility of the proposed methodology and the importance of performing a prior sensitivity analysis in Bayesian model comparison.

2 Marginal Likelihood Estimation

The main goal of the paper is to develop a methodology to automatically analyse the robustness of the prior in Bayesian modelling with regard to the marginal likelihood. For a model M, we define the marginal likelihood as :

$$p(\mathbf{y} \mid M) = \int p(\mathbf{y} \mid \boldsymbol{\psi}; M) p(\boldsymbol{\psi} \mid M) \mathrm{d}\boldsymbol{\psi} \tag{1}$$

where $p(\mathbf{y} \mid \boldsymbol{\psi}; M)$ is the likelihood function of the model, and $p(\boldsymbol{\psi} \mid M)$ is a prior on the model parameters $\boldsymbol{\psi}$.

The marginal likelihood is a measure of how well a particular model fits the observed data. It can be thought of as the joint density forecast from that model evaluated at the observed data, $\mathbf{y}$. If the observed data are likely under the model, the corresponding marginal likelihood would be "large", meaning the model is a good fit for the data. On the other hand, if the observed data are unlikely under the model, the marginal likelihood would be "small", indicating that the model is not a good fit for the data.

The marginal likelihood is used as a tool for model selection and comparison. A model with a larger marginal likelihood is considered to be more probable than one with a smaller marginal likelihood. This can be calculated through the Bayes factor. For two models M_1 and M_2 , the Bayes factor is:

$$\mathrm{BF}_{1,2} = \frac{p(\mathbf{y} \mid M_1)}{p(\mathbf{y} \mid M_2)}$$

If, $\mathrm{BF}_{1,2}$ is greater than 1, that means that the model M_1 is more likely given the data and the prior chosen. Indeed, the choice of the prior can have an impact on the marginal likelihood and therefore on the prior. Hence, the importance to conduct a sensitivity analysis of the marginal likelihood with respect to the priors hyperparameters.

To estimate the marginal likelihood, the authors focus on two methods, the Chib's method and the cross-entropy estimator. For the rest of the report, we will drop the model in the notations, and add $\boldsymbol{\theta}_0$ which denotes the hyperparameters of interest of the prior. For example, we write the prior density as $p(\boldsymbol{\psi}; \boldsymbol{\theta}_0)$ and the marginal likelihood as $p(\mathbf{y}; \boldsymbol{\theta}_0)$.

2.1 Chib's Method

Chib's method [2, 3] is based on the observation that the marginal likelihood is the normalizing constant of the posterior distribution :

$$p(\mathbf{y}; \boldsymbol{\theta}_0) = \frac{p(\mathbf{y} \mid \boldsymbol{\psi}) p(\boldsymbol{\psi}; \boldsymbol{\theta}_0)}{p(\boldsymbol{\psi} \mid \mathbf{y}; \boldsymbol{\theta}_0)}$$

From this equation, we can define Chib's estimator (in log scale) as :

$$\log \widehat{p(\mathbf{y}; \boldsymbol{\theta}_0)}_{\mathrm{Chib}} = \log p(\mathbf{y} \mid \boldsymbol{\psi}^*) + \log p(\boldsymbol{\psi}^*; \boldsymbol{\theta}_0) - \log \widehat{p(\boldsymbol{\psi}^* \mid \mathbf{y}; \boldsymbol{\theta}_0)} \tag{2}$$

where $\boldsymbol{\psi}^*$ is a point in the posterior support. It can be arbitrary, but it is usually chosen where the density of the posterior is high (such as the mean or mode).

In practice, the log-likelihood and the log-prior can be evaluated. However, the log-posterior has to be estimated using a series of Gibbs samplers, to estimate $\widehat{p(\boldsymbol{\psi}^* \mid \mathbf{y}; \boldsymbol{\theta}_0)}$.

2.2 Cross-Entropy Estimator

To estimate the marginal likelihood in (1), we would like to sample points from the posterior density $p(\boldsymbol{\psi} \mid \mathbf{y})$, as it is the theoretical zero-variance importance sampling density. However, we only know it up to a constant (the marginal likelihood).

The idea of the cross-entropy method is to approach the posterior density, noted f^*, with a parameterized density that is close to it. We consider a parametric family of densities, $\mathcal{F}\{f(\boldsymbol{\psi};\boldsymbol{\nu})\}$ where $\boldsymbol{\nu} \in \mathbb{R}^{\dim_\nu}$ is the vector of parameters. The objective is to find $\boldsymbol{\nu}^*$ that minimizes the cross-entropy, equivalent to :

$$\boldsymbol{\nu}^*_{\text{ce}} \in \arg\max_{\nu} \int p(\mathbf{y} \mid \boldsymbol{\psi}) p(\boldsymbol{\psi}) \log(f(\boldsymbol{\psi};\boldsymbol{\nu}))\mathrm{d}\boldsymbol{\psi}$$

It can be estimated with :

$$\widehat{\boldsymbol{\nu}^*_{\text{ce}}} = \arg\max_{\nu} \frac{1}{R} \sum_{r=1}^{R} \log(f(\boldsymbol{\psi}^r;\boldsymbol{\nu})) \tag{3}$$

where $\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \ldots, \boldsymbol{\psi}^R$ are posterior draws.

We then use this parametrized approximation of the posterior to do importance sampling, giving the following marginal likelihood estimator :

$$\widehat{p(\mathbf{y};\boldsymbol{\theta}_0)}_{\text{ce}} = \frac{1}{N} \sum_{j=1}^{N} \frac{p(\mathbf{y} \mid \boldsymbol{\psi}^j) p(\boldsymbol{\psi}^j;\boldsymbol{\theta}_0)}{f(\boldsymbol{\psi}^j;\widehat{\boldsymbol{\nu}^*_{\text{ce}}})} \tag{4}$$

where $\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \ldots, \boldsymbol{\psi}^N$ are drawn from the approximation importance sampling density $f(\boldsymbol{\psi};\widehat{\boldsymbol{\nu}^*_{\text{ce}}})$.

3 Automatic Differentiation for Marginal Likelihood

In the following section, we will describe how to use automatic differentiation to calculate the partial derivatives of the marginal likelihood with respect to the prior hyperparameters for both the Chib's estimator and the cross-entropy estimator. This will allow us to analyse the sensitivity of the marginal likelihood to the choice of prior, and to identify which hyperparameters have the greatest impact on the marginal likelihood values.

3.1 Gradient of Chib's Estimator

In this section, we will explain how we calculate the gradient $\frac{\partial}{\partial \boldsymbol{\theta}_0} \log \widehat{p(\mathbf{y};\boldsymbol{\theta}_0)}_{\text{Chib}}$. Chib's estimator is composed of three components (Eqn. (2)), evaluated at some posterior ordinate $\boldsymbol{\psi}^*$. Furthermore, we dispose of $\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \ldots, \boldsymbol{\psi}^R$ draws of the posterior, obtained using MCMC.

The partial derivatives of the first two components, $\frac{\partial \log p(\mathbf{y}|\boldsymbol{\psi}^*)}{\partial \boldsymbol{\theta}_0}$ and $\frac{\partial \log p(\boldsymbol{\psi}^*;\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}$, can be obtained easily, assuming that the likelihood and prior distributions are continuously differentiable in $\boldsymbol{\psi}$. However, for the log-posterior, we need an additional Monte Carlo simulation, with a multiple block Gibbs sampler for instance. More details are given in the appendix (see Proposition 1).

3.2 Gradient of the Cross-Entropy Estimator

For the cross-entropy method, we first need to calculate the gradient $\frac{\partial \boldsymbol{\nu}^*_{\text{ce}}}{\partial \boldsymbol{\theta}_0}$. The paper gives a formula if certain conditions of differentiability on the likelihood and prior are let (see Proposition 2 in appendix).

The empirical counter-part is :

$$\frac{\partial \widehat{\boldsymbol{\nu}^*_{\text{ce}}}}{\partial \boldsymbol{\theta}_0} = -\left[\sum_{r=1}^{R} \frac{\partial^2 \log f(\boldsymbol{\psi}^r; \widehat{\boldsymbol{\nu}^*_{\text{ce}}})}{\partial \boldsymbol{\nu}^2}\right]^{-1} \left[\sum_{r=1}^{R} \frac{\partial^2 \log f(\boldsymbol{\psi}^r; \boldsymbol{\nu}^*_{\text{ce}})}{\partial \boldsymbol{\nu} \partial \boldsymbol{\psi}} \frac{\partial \boldsymbol{\psi}^r}{\partial \boldsymbol{\theta}_0}\right] \tag{5}$$

where $\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \ldots, \boldsymbol{\psi}^R$ are draws of the posterior density.

Finally, we obtain the marginal partial derivatives as :

$$\begin{aligned}\frac{\partial \widehat{p(\mathbf{y}; \boldsymbol{\theta}_0)_{\text{ce}}}}{\partial \boldsymbol{\theta}_0} = \frac{1}{N} \sum_{j=1}^{N} \Bigg(\frac{\partial}{\partial \boldsymbol{\psi}} \left(\frac{p(\mathbf{y} \mid \boldsymbol{\psi}^j) p(\boldsymbol{\psi}^j; \boldsymbol{\theta}_0)}{f(\boldsymbol{\psi}^j; \widehat{\boldsymbol{\nu}^*_{\text{ce}}})} \right) \frac{\partial \boldsymbol{\psi}^j}{\partial \boldsymbol{\nu}^*_{\text{ce}}} - \frac{p(\mathbf{y} \mid \boldsymbol{\psi}^j) p(\boldsymbol{\psi}^j; \boldsymbol{\theta}_0)}{f(\boldsymbol{\psi}^j; \widehat{\boldsymbol{\nu}^*_{\text{ce}}})^2} \frac{\partial f(\boldsymbol{\psi}^j; \widehat{\boldsymbol{\nu}^*_{\text{ce}}})}{\partial \boldsymbol{\nu}} \Bigg) \frac{\partial \widehat{\boldsymbol{\nu}^*_{\text{ce}}}}{\partial \boldsymbol{\theta}_0} \\ + \frac{p(\mathbf{y} \mid \boldsymbol{\psi}^j)}{f(\boldsymbol{\psi}^j; \widehat{\boldsymbol{\nu}^*_{\text{ce}}})} \frac{\partial p(\boldsymbol{\psi}^j; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}\end{aligned}$$

this time, $\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \ldots, \boldsymbol{\psi}^N$ are drawn from the importance sampling density $f(\boldsymbol{\psi}; \widehat{\boldsymbol{\nu}^*_{\text{ce}}})$.

3.3 Model Comparison

The marginal likelihood is usually used for model comparison through the Bayes factor. The partial derivatives of the Bayes factor are easily available thanks to automatic differentiation. Hence, we can look at the trend of the Bayes factor in regard to the hyperparameters, and thus examine how the relative ranking of models changes as the hyperparameters are varied.

4 Experiments and Results

In this section, to illustrate the proposed automatic prior sensitivity analysis, we will present our experiments and results on two simple Gaussian models. We focused on Chib's estimator and the associated partial derivatives.

You can find the code in Appendix B.

4.1 The Models

In both models, we suppose that we have M observations $y \in \mathbb{R}$. We possess endogenous variables $X_1, X_2, X_3 \in \mathbb{R}$ from which we want to predict y.

Model 1 uses X_1 and X_2 :

$$y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \sigma^2\right) \tag{Model 1}$$

Whereas model 2 uses X_1 and X_3 :

$$y = \alpha + \beta_1 X_1 + \beta_3 X_3 + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \sigma^2\right) \tag{Model 2}$$

In both models, the variance of the error σ^2 is supposed to be known. We took it equal to 1.

We consider the following priors :

$$\begin{aligned}\pi(\alpha) &\sim \mathcal{N}(\mu, 1)\\ \pi(\beta_1) &\sim \mathcal{N}\left(0, \sigma_1^2\right)\\ \pi(\beta_2) &\sim \mathcal{N}\left(0, \sigma_2^2\right)\\ \pi(\beta_3) &\sim \mathcal{N}\left(0, \sigma_3^2\right)\end{aligned}$$

Therefore, we have 4 hyperparameters for our priors, 3 of which are common to both models. We will use AD to estimate the partial derivatives of the likelihood with regard to those parameters. Then we will look at the Bayes factor and conduct a sensitivity initiative.

We show in the appendix (Prop. 3 and 4) that the posteriors have the following form. This is needed in order to use a Gibbs sampler, to estimate the posterior.

$$\begin{aligned}\pi(\alpha \mid y, \beta_1, \beta_2; M_1) &\sim \mathcal{N}\left(\frac{1}{N+\sigma^2}\left(\sum_{i=1}^{N}(y_i - \beta_1 X_{1,i} - \beta_2 X_{2,i}) + \sigma^2\mu\right), \frac{\sigma^2}{N+\sigma^2}\right)\\ \pi(\beta_1 \mid y, \alpha, \beta_2; M_1) &\sim \mathcal{N}\left(\frac{1}{\sigma^2 + \overline{X_1^2}\sigma_1^2}\sigma_1^2\sum_{i=1}^{N}\left[X_{1,i}(y_i - \alpha - \beta_2 X_{2,i})\right], \frac{\sigma^2\sigma_1^2}{\sigma^2 + \overline{X_1^2}\sigma_1^2}\right)\\ \pi(\beta_2 \mid y, \alpha, \beta_1; M_1) &\sim \mathcal{N}\left(\frac{1}{\sigma^2 + \overline{X_2^2}\sigma_2^2}\sigma_2^2\sum_{i=1}^{N}\left[X_{2,i}(y_i - \alpha - \beta_1 X_{1,i})\right], \frac{\sigma^2\sigma_2^2}{\sigma^2 + \overline{X_2^2}\sigma_2^2}\right)\end{aligned}$$

where $\overline{X_1^2} = \sum_{i=1}^{N} X_{1,i}^2$ is the sum, not the mean.

For the second model, the results are the same, we just need to swap β_2 and X_2 for β_3 and X_3.

4.2 Generating Data

We used generated data to conducts the experiments. We supposed that the first model was the right one. Given the parameters, α, β_1 and β_2 and the variables X_1 and X_2, we can generate the observations.

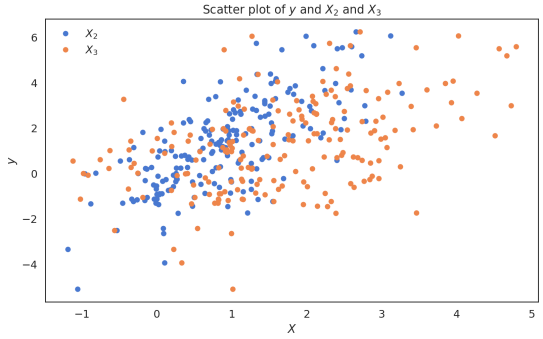
In practice, the real parameters are $\alpha = 0.25$, $\beta_1 = -1$ and $\beta_2 = 1.5$. The observation error ε has a variance of $\sigma^2 = 1$. Lastly, to generate the variables X_1, X_2 and X_3, we supposed that X_3 is correlated to X_2. We drew all three variable simultaneously from a multivariate Gaussian distribution :

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \sim \mathcal{N}(\Lambda, \Sigma)$$

where the mean is $\Lambda = \begin{pmatrix} 0.5 \\ 0.75 \\ 1.5 \end{pmatrix}$, and the covariance matrix $\Sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.7 & 0.7 \\ 0 & 0.7 & 1.44 \end{pmatrix}$

Figure 1 shows the generated data.

(a) Scatter plot of y vs X_1



(b) Scatter plot of y vs X_2 and X_3

Figure 1: Scatter plots of the simulated data

4.3 The Results

We implemented the models and generated all the plots with Python. We used JAX for the Automatic Differentiation part. We reimplemented the algorithms, which took quite some time.

As a reminder, the aim of Chib's method is to estimate the marginal likelihood with

$$\widehat{\log p(\mathbf{y};\boldsymbol{\theta}_0)}_{\text{Chib}} = \log p(\mathbf{y} \mid \boldsymbol{\psi}^*) + \log p(\boldsymbol{\psi}^*;\boldsymbol{\theta}_0) - \widehat{\log p(\boldsymbol{\psi}^* \mid \mathbf{y};\boldsymbol{\theta}_0)}$$

The first two terms have analytic expressions. We use a Gibbs sampler for the last term. The results of the Gibbs sampler are in Fig. 2. They are quite close to the real values of the coefficients.

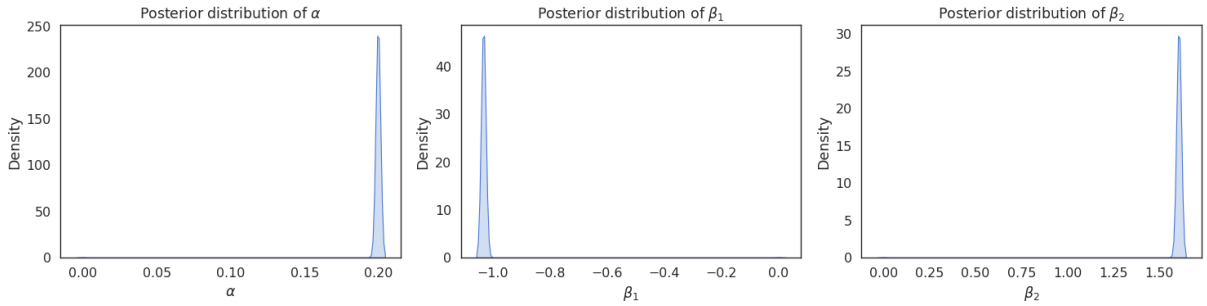


Figure 2: Estimation of the posterior distribution for Model 1

We can then estimate the Bayes factor, we found a value of

$$\log(\mathrm{B}_{1,2}) = 974.08$$

For the partial derivatives of the marginal likelihood, we obtain the results of Table 1 :

Model 1			Model 2		
μ	σ_1	σ_2	μ	σ_1	σ_3
-0.17	2207	-5418	0.22	25800	27497

Table 1: Partial derivatives of the marginal likelihood with regard to the hyperparameters

The results obtained for our problem have values way different from expected. This is likely due to difficulties encountered during the implementation of Automatic Differentiation, as it was challenging to

implement in Python. A missing scaling factor or numeric instability in the gradient may have caused this discrepancy.

Conclusion

In conclusion, Kwok et al. [4] present a methodology, inspired from other domain such as machine learning and finance, for using Automatic Differentiation to improve the practice of Bayesian statistics by providing a more efficient and systematic approach for sensitivity analysis of marginal likelihoods with respect to prior hyperparameters. The proposed methodology can improve the practice of Bayesian statistics by providing a more efficient and systematic approach for sensitivity analysis, which is important for verifying model selection and model averaging.

The authors used this method to analyse two well-known estimators, Chib's estimator and the cross-entropy estimator. The proposed methodology can be applied in a wide range of applications. It is possible to extend this idea to other estimators and more complex models. However, it should be noted that the proposed methodology relies on the availability of gradient information and may not be applicable to models where gradient computation is intractable.

They illustrated the feasibility of the methodology with two concrete examples, using real data. On our end, we reimplemented Chib's estimator with Automatic Differentiation, in a more simple example, comparing two Gaussian models on simulated data.

Appendix

A Propositions

Proposition 1 *Assuming that we have already obtained the Jacobian of the posterior ordinate* $\frac{\partial \boldsymbol{\psi}^*}{\partial \boldsymbol{\theta}_0}$ *and the one of the draws* $\frac{\partial \boldsymbol{\psi}^r}{\partial \boldsymbol{\theta}_0}$ *for* $r = 1, \ldots, R$.

The Jacobian of the first two elements of Eqn. 2 are :

$$\frac{\partial \log p(\mathbf{y} \mid \boldsymbol{\psi}^*)}{\partial \boldsymbol{\theta}_0} = \frac{1}{p(\mathbf{y} \mid \boldsymbol{\psi}^*)} \frac{\partial p(\mathbf{y} \mid \boldsymbol{\psi})}{\partial \boldsymbol{\psi}}\Big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^*} \frac{\partial \boldsymbol{\psi}^*}{\partial \boldsymbol{\theta}_0} \tag{6}$$

$$\frac{\partial \log p(\boldsymbol{\psi}^*; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0} = \frac{1}{p(\mathbf{y} \mid \boldsymbol{\psi}^*)} \left[\frac{\partial p(\boldsymbol{\psi}; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\psi}}\Big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^*} \frac{\partial \boldsymbol{\psi}^*}{\partial \boldsymbol{\theta}_0} + \frac{\partial p(\boldsymbol{\psi}^*; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0} p(\boldsymbol{\psi}^*; \boldsymbol{\theta}_0) \right] \tag{7}$$

The third term can be estimated with a multiple block Gibbs sampler. For example, for $\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3)$, *we have :*

$$\frac{\partial}{\partial \boldsymbol{\theta}_0} \left(\log \widehat{p(\boldsymbol{\psi}^* \mid \mathbf{y}; \boldsymbol{\theta}_0)} \right) = \frac{\frac{\partial p(\psi_1^* | \mathbf{y}; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}}{p(\psi_1^* \mid \mathbf{y}; \boldsymbol{\theta}_0)} + \frac{\frac{\partial p(\psi_2^* | \mathbf{y}, \psi_1^*; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}}{p(\psi_2^* \mid \mathbf{y}, \psi_1^*; \boldsymbol{\theta}_0)} + \frac{\frac{\partial p(\psi_3^* | \mathbf{y}, \psi_1^*, \psi_2^*; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}}{p(\psi_3^* \mid \mathbf{y}, \psi_1^*, \psi_2^*; \boldsymbol{\theta}_0)} \tag{8}$$

Proposition 2 *Assuming that the importance sampling density* $f(\boldsymbol{\psi}; \boldsymbol{\nu})$ *is twice continuously in* $\boldsymbol{\nu}$ *and* $\boldsymbol{\psi}$ *, with*

$$\mathbb{E}_\pi \left[\left\| \frac{\partial^2 \log f(\boldsymbol{\psi}; \boldsymbol{\nu}_{ce}^*)}{\partial \boldsymbol{\nu}^2} \right\| \right] < +\infty, \quad \mathbb{E}_\pi \left[\left\| \frac{\partial^2 \log f(\boldsymbol{\psi}; \boldsymbol{\nu}_{ce}^*)}{\partial \boldsymbol{\nu} \partial \psi} \right\| \right] < +\infty, \quad \mathbb{E}_\pi \left[\frac{\partial^2 \log f(\boldsymbol{\psi}; \boldsymbol{\nu}_{ce}^*)}{\partial \boldsymbol{\nu}^2} \right] \succ 0$$

Then, we have

$$\frac{\partial \boldsymbol{\nu}_{ce}^*}{\partial \boldsymbol{\theta}_0} = -\mathbb{E}_\pi \left[\frac{\partial^2 \log f(\boldsymbol{\psi}; \boldsymbol{\nu}_{ce}^*)}{\partial \boldsymbol{\nu}^2} \right]^{-1} \left(\mathbb{E}_\pi \left[\frac{\partial \log f(\boldsymbol{\psi}; \boldsymbol{\nu}_{ce}^*)}{\partial \boldsymbol{\nu}} \frac{\partial \log \pi(\boldsymbol{\psi}; \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0'} \right] \right) \tag{9}$$

where the expectation $\mathbb{E}_\pi$ *is taken with respect to the posterior measure.*

Proposition 3

$$\begin{aligned}
\pi(\alpha \mid y, \beta_1, \beta_2; M_1) &\propto \mathcal{L}(y \mid \alpha, \beta_1, \beta_2; M_1) \pi(\alpha \mid \beta_1, \beta_2; M_1) \\
&\propto \prod_{i=1}^{N} f(y_i \mid \alpha, \beta_1, \beta_2; M_1) \pi(\alpha; M_1) \\
&\propto \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}[y_i - \alpha - \beta_1 X_{1,i} - \beta_2 X_{2,i}]^2} e^{-\frac{(\alpha-\mu)^2}{2}} \\
&\propto \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}\left[\alpha^2 - 2\alpha(y_i - \beta_1 X_{1,i} - \beta_2 X_{2,i})\right]} e^{-\frac{\alpha^2\sigma^2 - 2\alpha\sigma^2\mu}{2\sigma^2}} \\
&\propto e^{-\frac{1}{2\sigma^2}\left[N\alpha^2 - 2\alpha \sum_{i=1}^{N}(y_i - \beta_1 X_{1,i} - \beta_2 X_{2,i}) + \alpha^2\sigma^2 - 2\alpha\sigma^2\mu\right]} \\
&\propto e^{-\frac{N+\sigma^2}{2\sigma^2}\left[\alpha^2 - 2\alpha \frac{1}{N+\sigma^2}\left(\sum_{i=1}^{N}(y_i - \beta_1 X_{1,i} - \beta_2 X_{2,i}) + \sigma^2\mu\right)\right]} \\
&\propto e^{-\frac{N+\sigma^2}{2\sigma^2}\left[\alpha - \frac{1}{N+\sigma^2}\left(\sum_{i=1}^{N}(y_i - \beta_1 X_{1,i} - \beta_2 X_{2,i}) + \sigma^2\mu\right)\right]^2} \\
&\sim \mathcal{N}\left(\frac{1}{N+\sigma^2} \left(\sum_{i=1}^{N}(y_i - \beta_1 X_{1,i} - \beta_2 X_{2,i}) + \sigma^2\mu \right), \frac{\sigma^2}{N+\sigma^2} \right)
\end{aligned}$$

Proposition 4

$$
\begin{aligned}
\pi(\beta_1 \mid y, \alpha, \beta_2; M_1) &\propto \mathcal{L}(y \mid \alpha, \beta_1, \beta_2; M_1)\pi(\beta_1 \mid \alpha, \beta_2; M_1) \\
&\propto \prod_{i=1}^{N} f(y_i \mid \alpha, \beta_1, \beta_2; M_1)\pi(\beta_1; M_1) \\
&\propto \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}[y_i-\alpha-\beta_1 X_{1,i}-\beta_2 X_{2,i}]^2} e^{-\frac{\beta_1^2}{2\sigma_1^2}} \\
&\propto \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}\left[\beta_1^2 X_{1,i}^2 - 2\beta_1 X_{1,i}(y_i-\alpha-\beta_2 X_{2,i})\right]} e^{-\frac{\beta_1^2}{2\sigma_1^2}} \\
&\propto e^{-\frac{1}{2\sigma^2}\left\{\beta_1^2 \sum_{i=1}^{N} X_{1,i}^2 - 2\beta_1 \sum_{i=1}^{N}[X_{1,i}(y_i-\alpha-\beta_2 X_{2,i})]\right\}} e^{-\frac{\beta_1^2}{2\sigma_1^2}} \\
&\propto e^{-\frac{1}{2\sigma^2\sigma_1^2}\left\{\sigma^2\beta_1^2 + \beta_1^2\overline{X_1^2}\sigma_1^2 - 2\beta_1\sigma_1^2 \sum_{i=1}^{N}[X_{1,i}(y_i-\alpha-\beta_2 X_{2,i})]\right\}} \\
&\propto e^{-\frac{\sigma^2+\overline{X_1^2}\sigma_1^2}{2\sigma^2\sigma_1^2}\left\{\beta_1^2 - 2\beta_1 \frac{1}{\sigma^2+\overline{X_1^2}\sigma_1^2}\sigma_1^2 \sum_{i=1}^{N}[X_{1,i}(y_i-\alpha-\beta_2 X_{2,i})]\right\}} \\
&\propto e^{-\frac{\sigma^2+\overline{X_1^2}\sigma_1^2}{2\sigma^2\sigma_1^2}\left[\beta_1 - \frac{1}{\sigma^2+\overline{X_1^2}\sigma_1^2}\sigma_1^2 \sum_{i=1}^{N}[X_{1,i}(y_i-\alpha-\beta_2 X_{2,i})]\right]^2} \\
&\sim \mathcal{N}\left(\frac{1}{\sigma^2+\overline{X_1^2}\sigma_1^2}\sigma_1^2 \sum_{i=1}^{N}\left[X_{1,i}(y_i-\alpha-\beta_2 X_{2,i})\right], \frac{\sigma^2\sigma_1^2}{\sigma^2+\overline{X_1^2}\sigma_1^2}\right)
\end{aligned}
$$

B Code

This code is the main module implementing Chib's method and the Autom

```
import jax.numpy as np
import matplotlib.pyplot as plt
import numpy as nps
from jax import grad, jit, random


def generate_data(n_samples):
    RANDOM_SEED = 8927
    rng = nps.random.default_rng(RANDOM_SEED)
    alpha = 0.25
    sigma_e = 1
    betas = [-1, 1.5]
    mean = [0.5, 0.75, 1.5]
    sigmas = [1, 0.7**0.5, 1.2]
    cov = [[sigmas[0] ** 2, 0, 0], [0, sigmas[1] ** 2, 0.7], [0, 0.7, sigmas[2] ** 2]]

    X_samples = nps.random.multivariate_normal(mean=mean, cov=cov, size=n_samples)

    y = (
        alpha
        + betas[0] * X_samples[:, 0]
        + betas[1] * X_samples[:, 1]
        + rng.normal(size=n_samples) * sigma_e
    )

    return y, X_samples


# Mean of the distribution
key = random.PRNGKey(0)
# Covariance matrix of the distribution
alpha = 0.25

Y, X_samples = generate_data(50)
Y = np.array(Y)
X1 = np.array(X_samples[:, 0])
X2 = np.array(X_samples[:, 1])
X3 = np.array(X_samples[:, 2])

choice_X = X2


def likelihood(alpha, beta_1, beta_2, sigma, X1=X1, X2=choice_X, Y=Y):
    return np.sum(
        (
            -0.5
            / sigma**2
            * (np.linalg.norm(Y - alpha - beta_1 * X1 - beta_2 * X2, axis=0)) ** 2
        )
    ) - len(Y) * np.log(sigma * np.sqrt(2 * np.pi))


def prior_alpha(mu):
```

```
    key = random.PRNGKey(0)
    key, subkey = random.split(key)
    return random.normal(subkey) + mu


def prior_beta(sigma):
    key = random.PRNGKey(0)
    key, subkey = random.split(key)
    return random.normal(subkey) * sigma


def generate_samples(mu, sigma, sigma_beta_1, sigma_beta_2, X1=X1, X2=choice_X, Y=Y):
    N = 1000
    n = len(Y)
    beta_1 = np.zeros(N)
    beta_2 = np.zeros(N)
    alphas = np.zeros(N)

    burn_in = 100
    alpha_tmp = prior_alpha(mu)
    beta_tmp1 = prior_beta(sigma_beta_1)
    beta_tmp2 = prior_beta(sigma_beta_2)
    for i in range(N + burn_in):
        key = random.PRNGKey(0)
        key, subkey = random.split(key)
        alpha_tmp = random.normal(subkey) * np.sqrt((sigma**2 / (n + sigma**2))) + (
            np.sum(Y - beta_tmp1 * X1 - beta_tmp2 * X2) + sigma**2 * mu
        ) / (n + sigma**2)
        if i > burn_in:
            alphas = alphas.at[i - burn_in].set(alpha_tmp)

        key, subkey = random.split(key)
        beta_tmp1 = random.normal(subkey) * np.sqrt(
            (sigma * sigma_beta_1) ** 2
            / (sigma**2 + np.mean(X1**2) * sigma_beta_1**2)
        ) + sigma**2 * np.sum(X1 * (Y - alpha_tmp - beta_tmp2 * X2)) / (
            sigma**2 + np.sum(X1**2) * sigma_beta_1**2
        )
        if i > burn_in:
            beta_1 = beta_1.at[i - burn_in].set(beta_tmp1)

        key, subkey = random.split(key)
        beta_tmp2 = random.normal(subkey) * np.sqrt(
            (sigma * sigma_beta_2) ** 2
            / (sigma**2 + np.mean(X2**2) * sigma_beta_2**2)
        ) + sigma**2 * np.sum(X2 * (Y - alpha_tmp - beta_tmp1 * X1)) / (
            sigma**2 + np.sum(X2**2) * sigma_beta_2**2
        )
        if i > burn_in:
            beta_2 = beta_2.at[i - burn_in].set(beta_tmp2)
    return alphas, beta_1, beta_2


def Chibs_method(param_dict, samples=Y, X2=choice_X):
    alpha_samples = np.array(
        generate_samples(
```

```
            mu=param_dict["mu"],
            sigma=param_dict["sigma"],
            sigma_beta_1=param_dict["sigma_beta_1"],
            sigma_beta_2=param_dict["sigma_beta_2"],
            X2=X2,
        )
    )

    psi_etoile = np.mean(alpha_samples, axis=0)
    var_psi = np.var(alpha_samples, axis=0)
    psi_etoile_moins, psi_etoile_plus = psi_etoile - 0.5, psi_etoile + 0.5

    proba_psi = np.sum(
        (alpha_samples > psi_etoile_moins) * (alpha_samples < psi_etoile_plus), axis=0
    ) / len(alpha_samples)

    log_proba = np.sum(np.log(proba_psi))
    log_prior = (
        -0.5 * ((psi_etoile[0] - param_dict["mu"]) / param_dict["sigma"]) ** 2
        - 0.5 * (psi_etoile[1] / param_dict["sigma_beta_1"]) ** 2
        - 0.5 * (psi_etoile[2] / param_dict["sigma_beta_2"]) ** 2
        - np.log(
            param_dict["sigma"]
            * param_dict["sigma_beta_1"]
            * param_dict["sigma_beta_2"]
            * (2 * np.pi) ** (3 / 2)
        )
    )
    log_lik = likelihood(
        psi_etoile[0], psi_etoile[1], psi_etoile[2], param_dict["sigma"]
    )

    log_marg = log_lik + log_prior - log_proba

    return log_marg


compute_grad = grad(Chibs_method)
param_dict = {"mu": 0.2, "sigma": 0.1, "sigma_beta_1": 0.1, "sigma_beta_2": 0.1}
M1_X2 = Chibs_method(param_dict, X2=X2)
M2_X3 = Chibs_method(param_dict, X2=X3)
print("We have: log(B12) = ", M1_X2 - M2_X3)

value_at_param_dict = compute_grad(param_dict)
print("values of grad for mu = .2, sigma = .1, etc.", value_at_param_dict)
```

References

[1] Joshua CC Chan, Liana Jacobi, and Dan Zhu. An automated prior robustness analysis in bayesian model comparison. *Journal of Applied Econometrics*, 37(3):583–602, 2022.

[2] Siddhartha Chib. Marginal likelihood from the gibbs output. *Journal of the american statistical association*, 90(432):1313–1321, 1995.

[3] Siddhartha Chib and Ivan Jeliazkov. Marginal likelihood from the metropolis–hastings output. *Journal of the American statistical association*, 96(453):270–281, 2001.

[4] Chun Fung Kwok, Dan Zhu, and Liana Jacobi. An analysis of vectorised automatic differentiation for statistical applications. *Available at SSRN*, 2022.

[5] Joshua CC Chan, Liana Jacobi, and Dan Zhu. Efficient selection of hyperparameters in large bayesian vars using automatic differentiation. *Journal of Forecasting*, 39(6):934–943, 2020.

[6] Joshua CC Chan, Liana Jacobi, and Dan Zhu. How sensitive are var forecasts to prior hyperparameters? an automated sensitivity analysis. In *Topics in Identification, Limited Dependent Variables, Partial Observability, Experimentation, and Flexible Modeling: Part A*. Emerald Publishing Limited, 2019.