Beyond MCMC: A Variational Inference Workflow

Candidate no. 1060665

Word count: 17,265

A thesis submitted for the degree of

Master of Science in Advanced Computer Science

Trinity 2022

Abstract

Practitioners in various disciples rely more and more on models for their decision making. From such models, they not only demand precise estimates, but also precise uncertainty estimates [Gelman et al., 2020, 2013]. Luckily, Markov Chain Monte Carlo (MCMC) sampling provides a generic tool that allows practitioners to obtain both predictions and uncertainty estimates for their models. However, practitioners demand that these tools are highly efficient and scale to large data sets. Unfortunately, MCMC is often criticised to not provide sufficient scalability [Johndrow et al., 2020, Song et al., 2020, Homan and Gelman, 2014] and therefore practitioners fall back on alternatives.

This research extensively studies an alternative class of algorithms that promises to address some of the limitations of MCMC: *Variational inference (VI)*. In addition, we extend publicly available implementations of Variational Inference by implementing missing distributions that are commonly used in practice. Through simulation studies, we show that caution is required by practitioners as VI may provide a false sense of accuracy and lead to poor estimates even on simple data sets. To mitigate this problem, we propose a *Variational Inference Workflow* building on our findings, recent research [Welandawe et al., 2022, Dhaka et al., 2020] and existing best-practices for MCMC [Vehtari et al., 2021a, Gelman et al., 2013, Brooks and Gelman, 1998]. This workflow aims to help practitioners to obtain confidence in their model estimates when using Variational Inference.

Contents

Appendices

List of Figures

1
Introduction

Contents

1.1 Motivation

For many machine learning applications, the objective is to train a function $f_\theta(\boldsymbol{x})$ that approximates a sample of training data $(\boldsymbol{x}, y)$ for a given loss function L, i.e., find

$$\theta^* = \underset{\theta \in \Theta}{\arg\min}\, L(f_\theta(\boldsymbol{x}), y). \tag{1.1}$$

Unfortunately, such a point estimate θ^* neither contains information about the distribution of the unknown parameter θ, nor the distribution of new predictions $\hat{y} = f_{\theta^*}(\hat{\boldsymbol{x}})$ at a new point $\hat{\boldsymbol{x}}$.

Bayesian inference provides a general framework that addresses this limitation and offers extensive information about the distribution of the parameter θ and the predictions $\hat{y}$.

To achieve that, Bayesian inference extends Equation (1.1) such that it optimizes for the full posterior distribution of θ given the training data $(\boldsymbol{x}, y)$, i.e., $p(\theta \mid$

$y, \boldsymbol{x})$. If we know the posterior distribution $p(\theta \mid y, \boldsymbol{x})$, then the distribution of predictions $\hat{y}$ can be inferred by

$$p(\hat{y} \mid y, \boldsymbol{x}) = \int\int \underbrace{p(\hat{y} \mid \theta)}_{\text{likelihood of } \hat{y}} \underbrace{p(\theta \mid y, \boldsymbol{x})}_{\text{posterior}} d\theta.$$

Fortunately, using Bayes rule, we can reformulate the posterior distribution $p(\theta \mid \boldsymbol{x})$ (omitting y for ease of notation) as the product of the prior over the latent variables $p(\theta)$ as well as a likelihood $p(\boldsymbol{x} \mid \theta)$, i.e.,

$$p(\theta \mid \boldsymbol{x}) = \frac{1}{\int p(\boldsymbol{x} \mid \theta) d\theta} p(\boldsymbol{x} \mid \theta) p(\theta). \tag{1.2}$$

Unfortunately, the integral $\int p(\boldsymbol{x} \mid \theta) d\theta$ is typically not analytically tractable and therefore approximation algorithms for $p(\theta \mid \boldsymbol{x})$ are required in general.

Remark 1.1.1 (MAP estimator)**.** Note that Equation (1.2) is often used to as a loss function in Equation (1.1), i.e.,

$$\theta^* = \arg\max_{\theta \in \Theta} p(\theta \mid \boldsymbol{x}) = \arg\max_{\theta \in \Theta} p(\boldsymbol{x} \mid \theta) p(\theta),$$

where the second equality follows by the fact that $\int p(\boldsymbol{x} \mid \theta) d\theta$ does not depend on θ.

The complexity of this problem is significantly lower than estimating the full posterior since both the likelihood and prior are specified by the practitioners and the problem therefore reduces to a (non-convex) optimization problem. θ^* in this case is commonly known as the *maximum a posterior estimate* or short MAP.

Example 1.1.1. Figure 1.1 shows $n = 500$ data points drawn from some unknown probability distribution $p(\mathbf{x} \mid \theta)$ with unknown parameter θ. Visual inspection of Figure 1.1 suggest that $p(\mathbf{x} \mid \theta)$ originates from a multivariate Gaussian with $\theta = (\mu, \Sigma) \in \mathbb{R}^2 \times R^{2\times 2}$. Note that Σ is a covariance matrix, i.e., it can be decomposed into the variance σ_i^2 of x_i as well as the correlation ρ of x_1 and x_2, i.e, there exists $\sigma_1, \sigma_2, \rho \in \mathbb{R}_{>0}$ such that

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \cdot \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

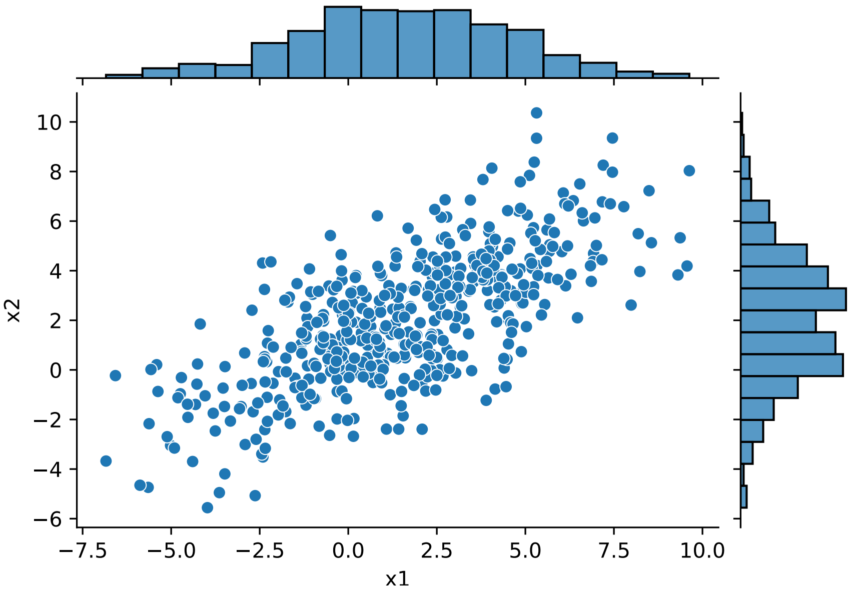


Figure 1.1: Observed data $\mathbf{x}$ from an unknown distribution $p(\boldsymbol{x} \mid \theta)$. Bayesian inference is required to calculate the posterior distribution of θ given $\boldsymbol{x}$.

With this parameterization, the unknown parameter θ has 5 degrees of freedom, i.e., $\theta = \{\mu_1, \mu_2\} \times \{\sigma_1, \sigma_2, \rho\} \in \mathbb{R}^2 \times \mathbb{R}^3_{>0}$. Given the likelihood $p(\mathbf{x} \mid \theta)$ (i.e., multivariate Gaussian) as well as the prior distribution $p(\theta)$, the aim of Bayesian inference is to compute $p(\theta \mid \mathbf{x})$. Note both the likelihood and the prior are provided by the practitioner through knowledge of the underlying problem that is being solved.

Figure 1.2 shows estimates for the full posterior distribution of each parameter instead of only point estimates. These estimates were obtain using MCMC sampling (see Section 2.2). We see that the distribution of ρ is very narrow, i.e., it can be estimates with high certainty. The distributions of the other parameters are wider and therefore less certain. Practitioners can use these output to calculate detailed uncertainty estimates.

Example 1.1.1 shows the variety of challenges associated with Bayesian inference. From the perspective of a practitioner, the following challenges arise when trying to apply Bayesian inference to a data set $\mathbf{x}$:

Challenge 1: How to choose the likelihood $p(\mathbf{x} \mid \theta)$?

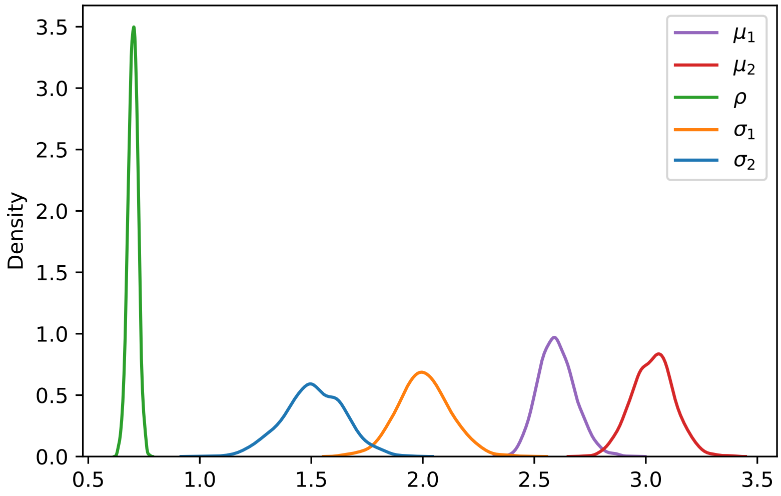


Figure 1.2: Estimate of the posterior distribution $p(\theta \mid \boldsymbol{x})$ obtained from MCMC sampling.

Challenge 2: How to choose the prior distribution $p(\theta)$?

Challenge 3: How to calculate or approximate the posterior $p(\theta \mid \mathbf{x})$?

Challenge 4: How to assess the quality of an approximation of $p(\theta \mid \mathbf{x})$?

1.2 Contributions

This work focuses on Challenges 3 and 4 as outlined in Section 1.1. More specifically, we focus on Variational Inference as a class of methods to approximate the posterior (Challenge 3) and investigate metrics to asses the quality of the approximation (Challenge 4). Challenges (1) and (2) are typically very specific to the underlying problem that is being solved. They require domain knowledge and are considered out of scope for this work.

With respect to quality metrics, we investigate three different approaches in Section 4.2:

1. Apply best-practices from MCMC such as Gelman-Rubin criteria $\hat{R}$ to VI [Vehtari et al., 2021a, Gelman et al., 2013, Brooks and Gelman, 1998].

2. Study quality metrics from recent research on VI [Welandawe et al., 2022, Dhaka et al., 2020, Vehtari et al., 2021b, Embrechts et al., 1997].
3. Derive an entirely new metric that provides an upper bound on the Wasserstein-2 distance between the true posterior and its variational approximation.

Through simulation studies, we apply the above metrics and show that caution is required when using off-the-shelve available VI algorithms (e.g., ADVI). In addition, we evaluate the newly proposed RAABBVI algorithm [Welandawe et al., 2022] and extend their publicly available implementation such that it is able to cover the multivariate Gaussian distribution with full rank covariance matrix (instead of mean-field only) as variational distribution family. This is required such that RAABBVI is able to cover the range of data sets that we investigate in our simulation studies. The code associated with the thesis is publicly available on GitHub[1]

Our results show that Variational Inference may provide a false sense of accuracy and lead to poor estimates even on simple models and data sets. As a solution to this problem, this work proposes a novel *Variational Inference Workflow* that can be used by practitioners interested in applying VI. Therefore, we extend the *Bayesian Workflow* [Gelman et al., 2020, Section 3.3] by providing additional guidance specifically focusing on Variational Inference.

---

[1] https://github.com/patrickzoechbauer/beyondmcmc

2

Background

Contents

2.1 Introduction

In general, $p(\theta \mid \boldsymbol{x})$ cannot be calculated analytically and hence needs to be approximated. A widely used and well understood method for that task is Markov Chain Monte Carlo (MCMC) sampling. MCMC is a powerful class of algorithm that provides strong guarantees to the practitioner in terms of convergence. Unfortunately, this often comes at the cost of long runtimes even for moderately complex examples [Yao et al., 2018].

To address some of these challenges, Variational Inference (VI) provides an interesting alternative. In this Chapter, we briefly review the motivation and general objective of MCMC and VI. In Section 2.3.1 and Section 2.3.2, we introduce two specific VI algorithms that form the basis for our simulation studies in Chapter 3.

2.2 Markov Chain Monte Carlo (MCMC)

In general, the objective of Bayesian inference is to sample from an unknown posterior distribution. Luckily based on Equation (1.2), this unknown posterior is known up to a normalizing constant $\int p(\boldsymbol{x} \mid \theta) d\theta$. MCMC describes a class of algorithms that are capable of sampling from unnormalized probability distributions, i.e., the normalizing constant does not need to be known.

Formally, the idea is to generate a Markov process that converges to a stationary distribution π which equals the unknown posterior distribution.

A Markov process with transition probabilities $k(\theta' \mid \theta)$ converges to a stationary distribution π if the detailed balance conditions holds, i.e., for each two states θ and θ', it is equally likely to transition from θ to θ' as it is to transition from θ' to θ, that is

$$\pi(\theta) k(\theta' \mid \theta) = \pi(\theta') k(\theta \mid \theta'). \tag{2.1}$$

The Metropolis-Hastings algorithm is a simple example of such an MCMC algorithm. It generates a Markov chain step by step, where in each step a proposal distribution q samples a new candidate state θ^*. This new state in step $i+1$ is either accepted or rejected depending on an *acceptance probability* $A(\theta_i, \theta^*)$ for which the unknown posterior distribution only needs to be know up to a constant. For the Metropolis-Hastings algorithm this acceptiance probability is given by

$$A(\theta_i, \theta^*) = \min\left\{1, \frac{p(\theta^* \mid \boldsymbol{x}) q(\theta_i \mid \theta^*)}{p(\theta_i \mid \boldsymbol{x}) q(\theta^* \mid \theta_i)}\right\}. \tag{2.2}$$

Note the ratio $p(\theta^* \mid \boldsymbol{x}) / p(\theta_i \mid \boldsymbol{x})$ in Equation (2.2) can be calculated even if the normalizing constant is unknown as it cancels out. Algorithm 1 shows the full Metropolis-Hastings algorithm. The algorithm defines a Markov chain with transition kernel

$$k(\theta' \mid \theta) = q(\theta' \mid \theta) A(\theta, \theta') + \delta_\theta(\theta') R(\theta),$$

with

$$R(\theta) = P(\text{reject } \theta') = 1 - \int A(\theta, u) q(u \mid \theta) du.$$

Based on this, we can directly show that the detailed balance conditions Equation (2.1) holds for $\pi(\theta) = p(\theta \mid \boldsymbol{x})$, i.e.,

$$\begin{aligned} p(\theta \mid \boldsymbol{x})k(\theta' \mid \theta =) &= p(\theta \mid \boldsymbol{x})q(\theta' \mid \theta)A(\theta, \theta') \\ &= \min\left\{1, \frac{p(\theta' \mid \boldsymbol{x})q(\theta \mid \theta')}{p(\theta \mid \boldsymbol{x})q(\theta' \mid \theta)}\right\} p(\theta \mid \boldsymbol{x})q(\theta' \mid \theta) \\ &= \min\left\{p(\theta \mid \boldsymbol{x})q(\theta' \mid \theta), p(\theta' \mid \boldsymbol{x})q(\theta \mid \theta')\right\} \\ &= p(\theta' \mid \boldsymbol{x})k(\theta \mid \theta'). \end{aligned}$$

This shows that Algorithm 1 generates a Markov Chains with stationary distribution equal to the unknown posterior distribution. That means after a 'warm-up' period that is required for the Markov chain to converge, we can us Algorithm 1 to sample from $p(\theta \mid \boldsymbol{x})$.

Algorithm 1 Metropolis-Hastings algorithm

```
Require: Initial state $\theta_0$
for $i = 0$ to $N - 1$ do
    Sample $u \sim \mathcal{U}(0, 1)$
    Sample a proposal state $\theta^* \sim q(\theta^* \mid \theta_i)$
    Calculate the acceptance ratio
        $A(\theta_i, \theta^*) = \min\left\{1, \frac{p(\theta^* \mid \boldsymbol{x})q(\theta_i \mid \theta^*)}{p(\theta_i \mid \boldsymbol{x})q(\theta^* \mid \theta_i)}\right\}$
    If $u \leq A(\theta_i, \theta^*)$ then $\theta_{i+} = \theta^*$ (accept), else $\theta_{i+1} = \theta_i$ (reject)
end for
```

However, despite the above theoretical guarantees also MCMC algorithms come with a set of limitations:

1. Exploring the full support: The Markov chain may get stuck in certain areas of the support of the true posterior if that area is separated from other areas by another area of low density. For example if the posterior is a multi-modal Gaussian mixture, then new proposals θ^* may never be able to cross that regions of low probability step-by-step between two modes.

2. Speed of convergence: Unfortunately, the theoretical guarantee does not tell us how many 'warm-up' steps are required until stationarity has been reached.

3. Scaling towards high dimensions: MCMC faces various challenges when scaling to very large data sets. This is due to the high computational cost per step and growth of the variance as a function of dimension [Johndrow et al., 2020].

4. Scaling towards large data sets: Calculating $A(\theta_i, \theta^*)$ requires to load the full data set $\boldsymbol{x}$ into memory [Song et al., 2020]. For very large problems (e.g., image classification) this is not possible due to hardware limitations and therefore alternatives (e.g., mini-batches) are needed.

5. Sensitivity towards correlations: MCMC algorithms such as Metropolis Hastings are sensitive towards correlated parameters [Homan and Gelman, 2014]. In particular in higher dimension this becomes a problem in practice.

There exists a wide body of research that aims to improve MCMC algorithms such as as Algorithm 1. Homan and Gelman [2014] proposed the No-U-Turn Samplers (NUTS) which is a specialized MCMC sampler that addresses some of the above problems (e.g., sensitivity towards correlation) and is currently considered the state-of-the-art of general purpose MCMC samplers that work for any probabilistic model.

2.3 Variational inference

The core idea of Variational Inference is to formulate the posterior approximation as an optimization problem similar to Equation (1.1). For that a family of variational distributions q_λ is defined and an optimisation problem is solved to find the member of this family that minimises the error L between to the true posterior distribution and the variational distribution family. More formally, for a variational family $q_\lambda(\theta)$ parameterised by a vector $\lambda \in \mathbb{R}^m$, VI minimises

$$\lambda_* = \arg\min_{\lambda \in \mathbb{R}^m} \mathrm{L}(q_\lambda(\theta), p(\theta|\boldsymbol{x})). \tag{2.3}$$

For Equation (2.3), a measure of distance L between two probability distributions is required. Variational inference commonly uses the Kullback-Leibler (KL) divergence for that purpose.

Definition 2.3.1 (Kullback-Leibler divergence)**.** Given two probability measure p and q, the KL divergence is given by

$$\mathrm{KL}(q||p) = \int_\Theta q(\theta) \log \left(\frac{q(\theta)}{p(\theta)} \right) d\theta.$$

Based on Definition 2.3.1, it holds that $\mathrm{KL}(q||p) = 0$ if and only if $q = p$. This property makes it an ideal candidate to be used for Equation (2.3), which leads to the objective

$$\lambda_* = \arg \min_{\lambda \in \mathbb{R}^m} \mathrm{KL}(q_\lambda(\theta)||p(\theta|\boldsymbol{x})). \tag{2.4}$$

Unfortunately, the KL contains the intractable posterior distribution $p(\theta|\boldsymbol{x})$. Therefore, it cannot be minimized directly. However, it holds that

$$\begin{aligned}
\mathrm{KL}(q_\lambda(\theta)||p(\theta|\boldsymbol{x})) &= \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{p(\theta|\boldsymbol{x})} d\theta \\
&= \int q_\lambda(\theta) \log q_\lambda(\theta) - q_\lambda(\theta) \log p(\theta|\boldsymbol{x}) d\theta \\
&= E_{q_\lambda}(\log q_\lambda(\theta)) - E_{q_\lambda}(\log p(\theta|\boldsymbol{x})) \\
&= E_{q_\lambda}(\log q_\lambda(\theta)) - E_{q_\lambda}\left(\log \frac{p(\theta, \boldsymbol{x})}{p(\boldsymbol{x})}\right) \\
&= E_{q_\lambda}(\log q_\lambda(\theta)) - E_{q_\lambda}(\log p(\boldsymbol{x}, \theta) - \log p(\boldsymbol{x})) \\
&= E_{q_\lambda}(\log q_\lambda(\theta) - \log p(\boldsymbol{x}, \theta)) + E_{q_\lambda}(\log p(\boldsymbol{x})) \\
&= E_{q_\lambda}(\log q_\lambda(\theta) - \log p(\boldsymbol{x}, \theta)) + \underbrace{\log p(\boldsymbol{x})}_{\leq 0} \\
&\leq -(E_{q_\lambda}(\log p(\boldsymbol{x}, \theta)) - E_{q_\lambda}(\log q_\lambda(\theta)).
\end{aligned}$$

Hence, minimizing the KL divergence is equivalent to maximizing the Evidence Lower Bound (ELBO) given by

$$\mathrm{ELBO}(\lambda) = \mathbb{E}_{q_\lambda(\theta)}[\log p(\boldsymbol{x}, \theta)] - \mathbb{E}_{q_\lambda(\theta)}[\log q_\lambda(\theta)]$$

Note the ELBO is tractable as it only depends on the joint distribution of $\boldsymbol{x}$ and θ as well as the variational distribution family q_λ.

2.3.1 Automatic Differentiation Variational Inference

One constraint of VI is that $\mathrm{supp}(q_\lambda(\theta)) \subseteq \mathrm{supp}(p(\theta|\boldsymbol{x}))$ since the KL divergence between the variational distribution and the posterior would otherwise always be infinite. Without loss of generality, it can be assumed that $\mathrm{supp}(p(\theta)) = \mathrm{supp}(p(\theta|\boldsymbol{x}))$. Thus, if one wants to use variational inference for a given probabilistic model, the variational family will have to be hand picked to have the same support as the prior distribution of the model.

The aim of ADVI is to offer a way of applying variational inference to a given probabilistic model without having to think about this support matching constraint. ADVI achieves this by bijecitvely mapping the original latent variable space to the real coordinate space and then performing variational inference in the transformed space.

For a one-to-one differentiable function $T : \mathrm{supp}(p(\theta)) \to \mathbb{R}^K$, the transformed joint density $p(\boldsymbol{x}, \zeta)$ has the representation $p(\boldsymbol{x}, \zeta) = p(\boldsymbol{x}, T^{-1}(\zeta))|\mathrm{det} J_{T^{-1}}(\zeta)|$. Intuitively, the Jacobian here describes how the transformation warps the unit volumes and thus ensures that the integral of the transformed distribution equals 1. The ELBO in real coordinate space then takes the form:

$$\mathrm{ELBO}(\lambda) = \mathbb{E}_{q_\lambda(\zeta)}[\log p(x, T^{-1}(\zeta))) + \log|\mathrm{det} J_{T^{-1}}(\zeta)|] - \mathbb{E}_{q_\lambda(\zeta)}[\log q_\lambda(\zeta)] \quad (2.5)$$

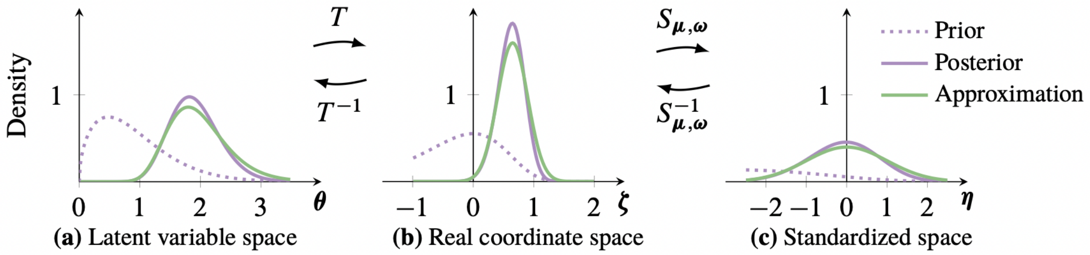
Commonly, a Gaussian distribution is assumed for $q_\lambda(\zeta)$ with mean $\boldsymbol{\mu}$ and variance $\mathrm{diag}(\boldsymbol{\sigma})$. However, the ELBO in Equation (2.5) involves an expectation over $q_\lambda(\zeta)$ which depends on $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Unfortunately, this leads to an intractable integral and hence automatic differentiation cannot be applied to this term directly. Instead, an additional transformation is applied to transform the variational distribution into a standard Gaussian. This second transformation is given by $S : \mathbb{R}^K \to \mathbb{R}^K$ is given by

$$S(\boldsymbol{\zeta}) = \mathrm{diag}(\boldsymbol{\sigma})^{-1}(\boldsymbol{\zeta} - \boldsymbol{\mu}) .$$

By doing so the expectation in the ELBO is computed with respect to a standard Gaussian rather than the original variational distribution. Finaly, we parameterize the standard deviation, $\boldsymbol{\sigma} = \exp(\boldsymbol{w})$, to ensure it remains positive. Based on this, the ELBO can be written as

$$\mathrm{ELBO}(\lambda) = \mathbb{E}_{\mathcal{N}(\boldsymbol{\eta};0,1)}[\log p(x, T^{-1}(S^{-1}_{\boldsymbol{\mu},\boldsymbol{w}}(\boldsymbol{\eta}))) + \log|\det J_{T^{-1}}(S^{-1}_{\boldsymbol{\mu},\boldsymbol{w}}(\boldsymbol{\eta}))|] + \sum_{k=1}^{K} w_k. \tag{2.6}$$

This means that automatic differentiation can be applied to the expression inside of the expectation to calculate the gradient of Equation (2.6). The expectation is then approximated via Monte Carlo sampling from the standard Gaussian.



(a) Latent variable space (b) Real coordinate space (c) Standardized space

Figure 2.1: Illustration of the mappings T and S used in ADVI to ensure that using the Gaussian variational distribution family is well-defined for a general probabilistic model [Kucukelbir et al., 2015].

Optimizing the ELBO in Equation (2.6)

Based on this, we can use any gradient decent algorithms to maximize the ELBO in Equation (2.6). As a stopping criteria, the author of the original on paper ADVI [Kucukelbir et al., 2015] suggest to stop training if the relative improve in the ELBO falls below a threshold δ. The implementation of ADVI in *Stan* [Stan Development Team, 2018] uses $\delta = 0.01$ as a default value. *Numpyro* [Bingham et al., 2019] does not provide a default to the user.

Note that the threshold δ is primarily a stopping criteria rather than a measure to assess the quality of the approximation q of $p(\theta \mid \boldsymbol{x})$. There is no direct connection that likes the absolute size or the relative improvements in the ELBO to the quality of q. In Chapter 3, we will show that even small changes in ELBO can result in large changes in q.

Benefits and limitations of AVDI for practitioners

By design, ADVI can be applied to an arbitrary probabilistic model as long as suitable transformations T are specified. Luckily, existing implementations available in *Stan* [Stan Development Team, 2018] or *numpyro* [Bingham et al., 2019] has the most important transformations already integrated. Therefore, ADVI, similar to MCMC, is a general purpose algorithm that can be used by practitioners for an arbitrary problem without the need to worry if the inference is well-defined.

The downside of ADVI, in contrast to MCMC, is that it does not provide theoretical guarantees that it converges to the true posterior distribution in general. For practitioners, there are two main concerns to be aware of:

1. Robustness: To what extent is a small change in ELBO a good enough criteria to asses if the gradient descent algorithm has converged? See detailed discussion in Section 2.3.1.

2. Accuracy: To what extent does the Gaussian variational distribution family provide a good approximation to the true posterior distribution in the transformed parameter space?

Based on our findings in Chapter 3, this research derives a *Variational Inference Workflow* in Section 4.4 that addresses the above concerns by providing a set of standard diagnostics methods to practitioners.

2.3.2 Robust, Automated, and Accurate Black-box Variational Inference

In addition to ADVI, this research also evaluate a more recent method named *Robust, Automated, and Accurate Black-box Variational Inference (RAABBVI)*. RAABBVI is an extension of ADVI proposed by Welandawe et al. [2022] that aims to address some of the limitations of ADVI stated in Section 2.3.1.

The objective of RAABBVI is to provide a VI algorithm that offers integrated diagnostics of the posterior approximation to automatically inform the practitioner if the obtained model estimates are reliable. To achieve this objective, RAABBVI addresses four dimensions:

1. Robustness: An algorithm that ensures robustness of the parameter estimates λ of the variational distribution q_λ such that it does not heavily depend on the choice of an individual tuning parameters (e.g., δ in ADVI).
2. Automated: An algorithm that requires minimal input from the practitioner beyond the data and the likelihood function.
3. Accuracy: An algorithm that stops model training only if the accuracy of q provides a sufficiently good approximation to the true posterior. If no suitable fit has been found after reaching the maximum iteration, the user will be automatically informed by the algorithm. For example, the authors replaces the stopping criteria based on the relative improvement in ELBO with more advanced quality measures.
4. Black-box: An algorithm that is of general purpose, i.e., that can be used with any probabilistic model and variational distribution family for which the KL-divergence can be calculated.

Algorithm 2 provides a high-level overview of the key steps of the RAABBVI algorithm. Section 2.3.2 and Section 2.3.2 detail the Steps 1, 2 and 3 in Algorithm 2 and show how they address the objectives of RAABBVI.

Step 1 and 4 in Algorithm 2: Averaging of $\lambda^{(k)}$

First, we look into why RAABBVI uses $\bar{\lambda}_\gamma$ instead of $\lambda_\gamma^{(k)}$ as the final parameter of the variational distribution q.

For stochastic optimizations of the form

$$\lambda^{(k+1)} = \lambda^{(k)} - \gamma g^{(k)},$$

where γ is the learning rate and $g^{(k)}$ the gradient at the k-th iteration, it holds that $\lambda^{(1)}, \lambda^{(2)}, ...$ form a homogeneous Markov Chain that under suitable conditions

Algorithm 2 RAABBVI algorithm

```
Require: Select an initial learning rate $\gamma$, let $\lambda_\gamma^{(k)}$ be the parameter of the $k$-th
    iteration of the variational distribution.
1: Run fixed-learning rate gradient descent for Equation (2.3) until the following
   stopping criteria are met:
     • $\hat{R}$ below 1.1, see [Dhaka et al., 2020]
     • Number of effective samples (ESS) of $\lambda_\gamma^{(k)}$ below threshold
     • Monte Carlo Standard Error (MCSE) of $\bar{\lambda}_\gamma = E_{\mu_\gamma}(\lambda)$ below threshold (see
       Section 2.3.2 for a formal definition of $\bar{\lambda}_\gamma$)
2: Check if a smaller learning rate $\gamma_*$ might improve the fit using an 'inefficiency
   index' $\mathcal{I}$ (see Equation (2.7)). The measure $\mathcal{I}$ balances the improvement in
   accuracy versus the additional computation time required.
3: If $\hat{\mathcal{I}}$ below threshold, then return to step 1 otherwise continue with step 4.
4: Calculate the average $\bar{\lambda}_\gamma$ and return $q_{\bar{\lambda}_\gamma}$
```

converges to a stationary distribution μ_γ. Furthermore, Dieuleveut et al. [2017] show that under regularity conditions on the loss function and unbiased gradient estimates, it holds that $\bar{\lambda}_\gamma = E_{\mu_\gamma}(\lambda)$ is a good estimate for the optimal variational distribution parameter λ_* defined by Equation (2.3). Formally, they show that there exist $A, B \in \mathbb{R}^m$ such that

$$\bar{\lambda}_\gamma - \lambda_* = A\gamma + B\gamma^2 + o(\gamma^2),$$

and $A' \in \mathbb{R}^{m\times m}$ such that

$$\int (\lambda - \lambda_*)(\lambda - \lambda_*)^T \mu_\gamma(\lambda) d\lambda = A'\gamma + \mathcal{O}(\gamma^2).$$

Therefore, at stationarity it holds that $\bar{\lambda}_\gamma - \lambda_* = \mathcal{O}(\gamma)$, which is much better than $\lambda^{(k)} - \lambda_* = \mathcal{O}(\gamma^{1/2})$ for $\lambda < 1$. Hence, Step 4 in Algorithm 2 aims to improve accuracy.

In addition, evaluating the Monte Carlo Standard Error of $\bar{\lambda}_\gamma$ as a convergence criteria in Step 1 of Algorithm 2 helps to improve robustness of RAABBVI as the algorithm continues to update $\lambda^{(k)}$ until it has reached a stable region with low variance.

Finally, we note that $\bar{\lambda}_\gamma = E_{\mu_\gamma}(\lambda)$ needs to be estimated in practice as no closed form expression for μ_γ is available. Welandawe et al. [2022] propose to use the sample mean of the last W samples of $\lambda_\gamma^{(1)}, ..., \lambda_\gamma^{(W)}$ for that.

Step 2 in Algorithm 2: The inefficiency index $\mathcal{I}$

Let q_{γ^*} be the optimal variational approximation when using the fixed learning rate γ, then the inefficiency index $\mathcal{I}$ in Step 2 of Algorithm 2 balances improvements of reducing the learning rate γ by a factor ρ with the additional computing time required to again reach convergence with the new learning rate $\rho\gamma$ in Step 1. To formalize this, consider the following definition of the inefficiency index $\mathcal{I}$.

Definition 2.3.2 (Inefficiency index)**.** Let q_* be the optimal variational approximation and q_{γ^*} be the optimal variational approximation when using learning rate γ, then the inefficiency index $\mathcal{I}$ is given by

$$\mathcal{I} = \underbrace{\left(\frac{SKL(q_*, q_{\rho\gamma^*})^{1/2} + \xi}{SKL(q_*, q_{\gamma^*})^{1/2}}\right)}_{\text{Relative SKL improvement (RSKL)}} \cdot \underbrace{\left(\frac{K_{\rho\gamma^*}}{K_{\gamma^*} + K_0}\right)}_{\text{Relative iteration increase (RI)}}, \tag{2.7}$$

where SKL denotes the symmetrized KL divergence, ξ denotes the target accuracy and K_0 denotes the number of iterations a user considers small.

The two factors in Definition 2.3.2 can be interpreted as follows:

- RSKL: The relative SKL improvement is small if $q_{\rho\gamma^*}$ provides a much better fit than q_{γ^*}, i.e., $SKL(q_*, q_{\rho\gamma^*}) \ll SKL(q_*, q_{\gamma^*})$
- RI: The relative iteration increase is small if the number of steps to reach convergence using learning rate $\rho\gamma$ is smaller than the steps required for γ plus some additional small amount of steps, i.e., $K_{\rho\gamma^*} < K_{\gamma^*} + K_0$.

Therefore, if the product of these factors is small, it is worth the additional cost of exploring a smaller learning rate. Unfortunately, we need to know our target q_* to calculate $\mathcal{I}$. Therefore, Welandawe et al. [2022] propose an estimator for $\mathcal{I}$.

Proposition 2.3.1. *Let* $q_\lambda \sim \mathcal{N}(\tau, diag(\exp(\psi)))$ *and with* $\lambda = (\tau, \psi) \in \mathbb{R}^{2d}$*, i.e., the mean-field Gaussian, and assume there exists constant vectors* $A, B \in \mathbb{R}^{2d}$ *such that*

$$\bar{\lambda}_\gamma - \lambda_* = A\gamma + B\gamma^2 + o(\lambda^2), \tag{2.8}$$

and $\gamma' \in \mathcal{O}(\gamma)$*. Then there exists a constant* $C > 0$ *such that*

$$SKL(q_{\gamma^*}, q_{\gamma'*}) = C(\gamma - \gamma') + o(\gamma^2).$$

See Welandawe et al. [2022] for a proof of Proposition 2.3.1. If we assume γ_0 is the initial learning rate, then at the t-th iteration of Step 1 in Algorithm 2 the learning rate is $\gamma_t = \gamma_0 \rho^t$. Let $\delta_t := SKL(q_{\gamma_t^*}, q_{\gamma_{t-1}^*})$, then we can use Proposition 2.3.1 to estimate C by

$$\delta_t = C\left(\gamma_t - \frac{\gamma_t}{\rho}\right) + o(\gamma_t^2). \tag{2.9}$$

Note, at the end of the t-th iteration of step 1 of Algorithm 2, we can sample from $q_{\gamma t}^*$. Hence, we can calculate a Monte Carlo estimate for δ_t and C is the only unknown in Equation (2.9). Therefore for each iteration $t = 1, ..., T$, we store the history δ_t and use a regression model to calculate an estimate $\hat{C}$ given by

$$\log \delta_t = \log(C) + 2\log\left(\frac{1}{\rho_t} - 1\right) + 2\log(\gamma_t) + \eta_t, \tag{2.10}$$

where $\eta_t \sim \mathcal{N}(0, \sigma^2)$. Then it follows that

$$\widehat{RSKL}_{t+1} = \rho + \frac{\xi}{\hat{C}^{1/2}\gamma_{t+1}}.$$

Note that Welandawe et al. [2022] provide additional variations as well as optimizations of the above approach to account for various effects including the relaxation of the mean-field assumption on Proposition 2.3.1 as well as a weighted regression approach for Equation (2.10) to account for the fact that early estimate of SKL may be inaccurate.

Finally, we need to estimate the relative iteration increase RI. For that Welandawe et al. [2022] assume that the number of iterations grows exponentially as the learning rate decreases and hence they estimate K using

$$\log(K_{\gamma_t}) = \alpha \log(\gamma_t) + \beta + \nu_t, \quad \nu_t \sim \mathcal{N}(0, \sigma_t^2) \quad \text{for } t = 1, ..., T, \tag{2.11}$$

and since K_{γ_t} is know at time $t+1$, we obtain the estimate

$$\widehat{RI}_{t+1} = \frac{\widehat{K}_{\gamma_{t+1}}}{K_{\gamma_t} + K_0} = \frac{\gamma_{t+1}^{\hat{\alpha}}}{K_{\gamma_t} + K_0} \exp(\hat{\beta}).$$

Benefits and limitations of RAABBVI for practitioners

The main benefit of RAABBVI is that it provides integrated diagnostics to practitioner that help them to understand if the obtained model predictions provide a good approximation to the true posterior. Therefore, RAABBVI address both robustness and accuracy concerns of ADVI as described in Section 2.3.1.

However, practitioners should also be aware of the following three main concerns with respect to RAABBVI that we identified as part of this research:

1. The algorithm relies on some complex assumption with respect to regularity as well as the variational distribution family (see Proposition 2.3.1). Verifying these assumptions is usually not possible in practice and introduces additional model risk.

2. The existing implementation of RAABBVI on GitHub[1] is very experimental and therefore not easily accessible to practitioners. This includes the following limitations:

    - It only supports mean-field Gaussian as a variational distribution family. For our experiments in Chapter 3, we extend the existing code to support multivariate Gaussian with full-rank covariance matrix.
    - It depends on an old deprecated version of pystan that is used to estimate the coefficients in the regression models defined by Equation (2.10) and Equation (2.11). Currently, the user has to manually install pystan version 2 or older for RAABBVI to work.

3. The existing implementation of RAABBVI does not leverage state-of-the-art libraries for machine learning (e.g., pytorch or TensorFlow). Therefore, implementing GPU support or approaches to increase computational efficiency cannot no easily be integrated.

---

[1] https://github.com/Manushi22/viabel

Finally, Table 2.1 provides s summary of the theoretical aspects of ADVI and RAABBVI as described in Section 2.3.1 and Section 2.3.2. Note, in Table 2.1, ΔELBO-ADVI refers to ADVI using the ΔELBO criteria introduced in Section 2.3.1 as a stopping criteria. In Chapter 3, we show that some limitation of ADVI can be addressed by introducing a more advanced stopping criteria.

Based on our findings, we are able to propose a novel Variational Inference workflow that is able to address the limitation of ΔELBO-ADVI. More specifically, we are able to show that the robustness limitations can be largely associated with the ΔELBO rule, see Section 3.3.2 and Section 3.3.3. The accuracy limitations can be addressed by using appropritate quality measure as proposed in Section 4.2.

Table 2.1: Comparison between ΔELBO-ADVI and RAABBVI on the core objectives for VI formulated by Welandawe et al. [2022].

		Variational inference algorithm	
Objective	Description	ΔELBO-ADVI	RAABBVI
Robustness	Robustness assess if small changes in tuning parameters does not lead to large changes in the model predictions.	× (Section 3.3.2, Section 3.3.3)	✓
Automated	Automated asses if minimal input is required from the practitioner beyond the data and likelihood function.	✓	✓
Accuracy	Accuracy assess if the model predictions provide a good estimate for the true posterior.	× (Section 4.2)	✓
Black-box	Black-box asses if the algorithm can be applied to any probabilistic model	✓	✓

Legend: ✓ = Integrated in algorithm, × = Not integrated in algorithm.

3
Simulation studies

Contents

3.1 Introduction

In this chapter, we use simulated data to compare the performance of MCMC, ADVI and RAABBVI with respective to accuracy as well as computation time. The simulated data sets vary in complexity in order to test the proposed algorithms under various conditions. Based on these findings, we aim to develop a workflow that provides guidance to practitioners on how to use variational inference methods.

3.2 Data generating processes for $p(\theta)$ and $p(\mathbf{x} \mid \theta)$

We test the algorithms on four different data sets: (1) Linear regression, (2) Poisson regression, (3) Multivariate Gaussian, and (4) Hierarchical regression.

The benefit of using simulated data is that we know the underlying data generating process. This allows us to investigate various interesting quantities, e.g., average quality of each model across multiple simulations. In practice, one does not know $p(\boldsymbol{x} \mid \theta)$ and $p(\theta)$, but it is subject to the expertise of the practitioner to choose a suitable model (see Challenges (1) and (2) in Section 1.1).

As our focus lies on analyzing Challenges (3) and (4) as outline in Section 1.1, we use the same likelihood $p(\mathbf{x} \mid \theta)$ and prior $p(\theta)$ for both data generation and model estimation. Hence, there is no additional error due to model misspecification in the results of our simulation studies. Therefore, any resulting predictions errors can be associated to the inference algorithm, e.g., converge or the limitations of the variational distribution family.

3.2.1 Multivariate linear data with Gaussian noise

The first data generating process is given by

$$\begin{aligned}
\beta_d &\sim \mathcal{N}(0,1) \quad \text{for } d = 0, ..., D \\
\varepsilon_i &\sim \mathcal{N}(0,\sigma) \quad \text{for } i = 1, ..., n \\
x_{ij} &\sim \mathcal{N}(0,1) \quad \text{for } i = 1, ..., n, \text{ and } j = 1, ..., D \\
y_i &= \mathbf{x_i}^T \boldsymbol{\beta} + \varepsilon_i \quad \text{with } \mathbf{x_i} = (1, x_{i1}, ..., x_{iD}) \text{ and } \boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_D)
\end{aligned}$$

Figure 3.1 shows an example of the data generating process for $D = 2$ and $n = 100$. This model is mostly known as *Linear regression* and widely used by practitioners [Gelman et al., 2013].

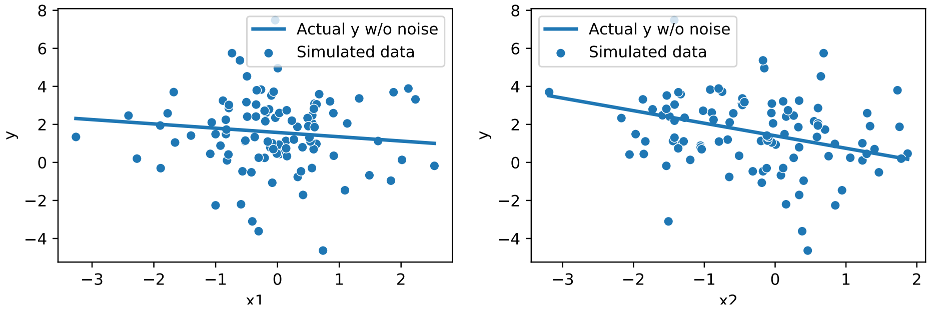


Figure 3.1: Example output of data generating process using $D = 2$ and $n = 100$ and $\sigma = 2$.

Remark 3.2.1. For this data generating process, it is straight forward exercise to show that the posterior $p(\boldsymbol{\beta}|y_1, ..., y_n)$ is mean-field Gaussian. Since for both ADVI and RAABBVI, mean-field Gaussian is also the default assumptions, it holds that the true posterior distribution is included in the variational distribution family assumed by ADVI and RAABBVI. Consequently, if enough data is available, we may expect from both inference algorithms a very high quality in recovering the true posterior.

3.2.2 Poisson regression with log-linear intensity

Second, we simulate data from a Poisson distribution with a log-linear intensity function, i.e., for $i = 1, ..., n$

$$\begin{aligned}
\beta_d &\sim \mathcal{N}(0,1) \quad \text{for } d = 1, ..., D \\
x_{ij} &\sim \mathcal{N}(0,1) \quad \text{for } i = 1, ..., n, \text{ and } j = 1, ..., D \\
\beta_0 &= -\log\left(\frac{1}{n}\sum_{i=}^{n}\exp\left(\sum_{j=1}^{D} x_{ij}\beta_j\right)\right) + \log(10) \\
\lambda(\mathbf{x}_i) &= \exp(\mathbf{x_i}^T\boldsymbol{\beta}), \qquad \text{with } \mathbf{x_i} = (1, x_{i1}, ..., x_{iD}) \text{ and } \boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_D) \\
y_i &\sim \text{Poisson}(\lambda(\mathbf{x}_i))
\end{aligned}$$

The choice of β_0 is such that on $\mathbb{E}(y_i) = 10$. In particular the expectation of y_i neither depends on D nor n which is important to avoid overflow when increase the size and dimensionality of the data set. Figure 3.2 shows an example of the data generating process for $D = 2$ and $n = 100$.

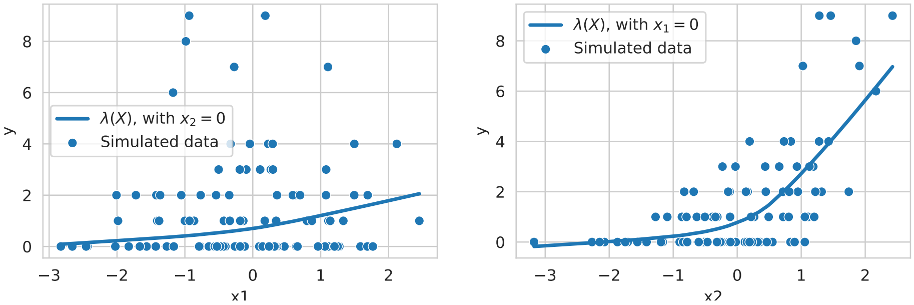


Figure 3.2: Example output of data generating process using $D = 2$ and $n = 100$.

Remark 3.2.2. Note that the conjugate prior of the Poisson distribution is the Gamma distribution resulting in a Gamma-distributed posterior. In our case, the intensity function is the product of random variables with log-normal prior which is not conjugate to the Poisson model. Therefore, no closed-form expression for the posterior exists. In particular, this implies that the posterior distribution for a Poisson regression with log-linear intensity and normal prior is not Gaussian. Even though the model is still relatively simple, it will already impose larger challenges on the inference algorithms and accuracy metrics.

3.2.3 Multivariate Gaussian with full rank covariance matrix

In the previous data generating processes, the likelihood factorized, i.e.,

$$p(\mathbf{x} \mid \theta) = \prod_{j=1}^{D} p(x_j \mid \theta).$$

This is commonly referred to the mean-field assumption and it reduces the complexity of the posterior distribution significantly as there are no correlation between individual dimensions (given θ) that need to be learned.

Next, we simulate a data set that does not fulfill this assumption and the respective algorithms have to learn the entire joint distribution. This increased complexity results in longer runtimes by the inference algorithm, including MCMC, as the dimensionality of the parameter space increases exponentially.

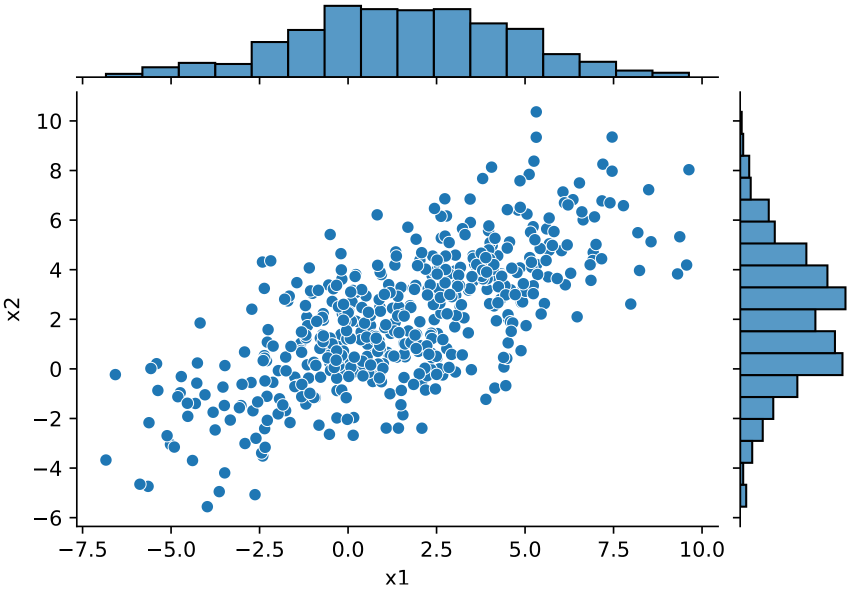


Figure 3.3: Observed data $\mathbf{x}$ from an unknown distribution $p(\boldsymbol{x} \mid \theta)$. Bayesian inference is required to calculate the posterior distribution of θ given $\boldsymbol{x}$.

For that we simulate data from a multivariate normal with full rank covariance matrix, i.e.,

$$\begin{aligned} a &\sim \mathcal{N}(1,1) \\ b &\sim \mathcal{N}(2,1) \\ \mu &= (a,b) \in \mathbb{R}^2 \\ \log(\sigma_a) &\sim \mathcal{N}(\log(3),1) \\ \log(\sigma_b) &\sim \mathcal{N}(\log(2.5),1) \\ \rho &\sim \mathcal{N}(0.7,1) \\ \Sigma &= \begin{pmatrix} \sigma_a^2 & \rho\sigma_a\sigma_b \\ \rho\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix} \\ \mathbf{x}_i &\sim \mathcal{N}(\mu,\Sigma) \quad \text{for } i = 1,...,n. \end{aligned}$$

Figure 3.3 shows $n = 500$ data points from the above data generating process.

Remark 3.2.3. For this data set, we had to extend the RAABBVI package as available on GitHub[1] to support the multivariate Gaussian distribution as a variational distribution family. The available implementation is limited to the mean-field approximation.

---

[1] https://github.com/Manushi22/viabel

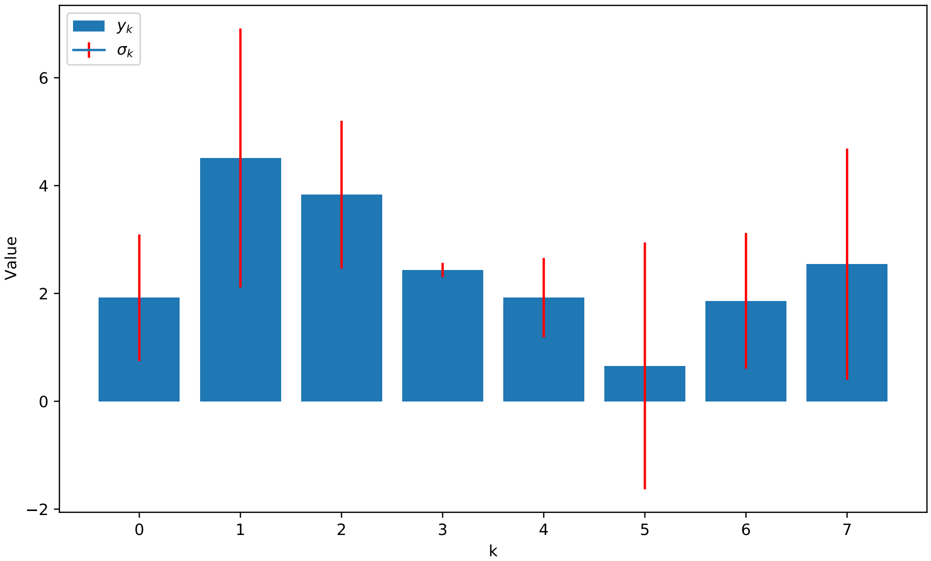


Figure 3.4: Example output of data generating process in Section 3.2.4 using $D = 8$.

3.2.4 Hierarchical regression

Finally, we investigate data for a hierarchical model that was previously used by Yao et al. [2018] to study the quality of variational inference algorithms, i.e., we sample from

$$\begin{aligned} \mu &\sim \mathcal{N}(2,1) \\ \log(\tau) &\sim \mathcal{N}(0,1) \\ \theta_k &\sim \mathcal{N}(\mu,\tau) \quad \text{for } k = 1,\ldots,D \\ \log(\sigma_k) &\sim \mathcal{N}(0,1) \\ y_k &\sim \mathcal{N}(\theta_k,\sigma_k) \quad \text{for } k = 1,\ldots,D. \end{aligned}$$

Figure 3.4 shows a sample of the data set for $D = 8$. This model is referred to as hierarchical because the unknown parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$ depends on the shared parameters μ and τ. Note $\boldsymbol{y}$ does not depend on μ and τ given $\boldsymbol{\theta}$ and therefore the the posterior is given by

$$\begin{aligned} p(\mu,\tau,\boldsymbol{\theta} \mid \boldsymbol{y}) &= Cp(\boldsymbol{y} \mid \mu,\tau,\boldsymbol{\theta})p(\mu,\tau,\boldsymbol{\theta}) \\ &= Cp(\boldsymbol{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta} \mid \mu,\tau)p(\mu,\tau), \end{aligned}$$

for some unknown normalizing constant $C \in \mathbb{R}$.

Remark 3.2.4. This data generating process / model is commonly used for the 8schools data set [Rubin, 1981]. In this data set, each of 8 schools reported a treatment effect y_i and standard deviation σ_i. Since there is no prior believe that the treatments were more or less effective in any of the schools, it is commonly modeled as a hierarchical model with shared parameter μ and τ.

3.3 Experiments

Using the data generating processes / models described in Section 3.2, we conduct multiple experiments to answer the following research questions:

1. *How well do ΔELBO-ADVI and RAABBVI recover the true posterior as given by MCMC?*
2. *Is RAABBVI always better than ADVI or does it only depend on the stopping criteria (i.e., ΔELBO)?*

3.3.1 Accuracy metrics

To answer each of the above research questions, it is important to define a set of metrics to determine the accuracy of the posterior approximation. For this research, we evaluate four different metrics: (1) Error in mean (2) Error in standard deviation (3) Kolmogrov-Smirnov Test [Massey, 1951], and (4) Wasserstein distance [Ramdas et al., 2015].

Given a probability distribution $p(\theta)$, we use the notation $\mu(\theta) := \mathbb{E}_p(\theta)$ and $\sigma^2(\theta) = Var_p(\theta)$. Furthermore, $\hat{\mu}$ and $\hat{\sigma}$ denote the sample mean, resp. sample standard deviation based on the samples $\theta_1, ..., \theta_n$ from $p(\theta)$. Using this notation, Table 3.1 provides a formal definition of the metrics previously introduced. Note that for most models, θ is D-dimensional. To account for this, each metrics defined in Table 3.1 aggregates the univariate metrics across the D dimension either by averaging or taking the minimum. This has the benefit that it allows us to quantify the quality of an approximation with a single number.

Table 3.1: Accuracy metrics to compare algorithms.

Metrics	Description	Formula
μ-MAE(θ)	Mean absolute error of the mean μ of $p(\theta \mid \boldsymbol{x})$	$\frac{1}{D}\left(\sum_{j=1}^{D} \lvert\mu(\theta_j) - \hat{\mu}(\theta_j)\rvert\right)$
σ-MAE(θ)	Mean absolute error of the standard deviation of $p(\theta \mid \boldsymbol{x})$	$\frac{1}{D}\left(\sum_{j=1}^{D} \lvert\sigma(\theta_j) - \hat{\sigma}(\theta_j)\rvert\right)$
KS(θ)	Minimum p-value of each univariate Kolmogorov–Smirnov Test for θ_j	Let $\tilde{\theta}^j \sim p^1(\theta)$ and $\hat{\theta}^j \sim p^2(\theta)$, calculate $\min_{j=1,\ldots,D} \{\text{KS p-value of} \{\tilde{\theta}^j\} \text{ and } \{\hat{\theta}^j\}\}$
W$_1$(θ)	Average univariate Wasserstein-1 distance across each dimension of θ for two distributions p_1 and p_2	see Ramdas et al. [2015]

3.3.2 Experiment 1: How well do ΔELBO-ADVI and RAABBVI recover the true posterior?

In this section, we evaluate the performance of ΔELBO-ADVI and RAABBVI by comparing the estimated posterior distribution against MCMC.

Figure 3.5 shows the estimated posterior distributions for MCMC, ΔELBO-ADVI with $\delta = 0.01$ and RAABBVI trained on the data sets described Section 3.2. Visual inspection of Figure 3.5 shows that ADVI using the ΔELBO stopping criteria does not recover the true posterior distribution successfully (comparison against MCMC). In fact, ΔELBO-ADVI not only struggles to recover the variance of the posterior, but it is also struggling to provide accurate estimates for the mean. This problem is most prominent for the Multivariate Gaussian data sets, where the ΔELBO-ADVI estimates are far off from the true posterior.

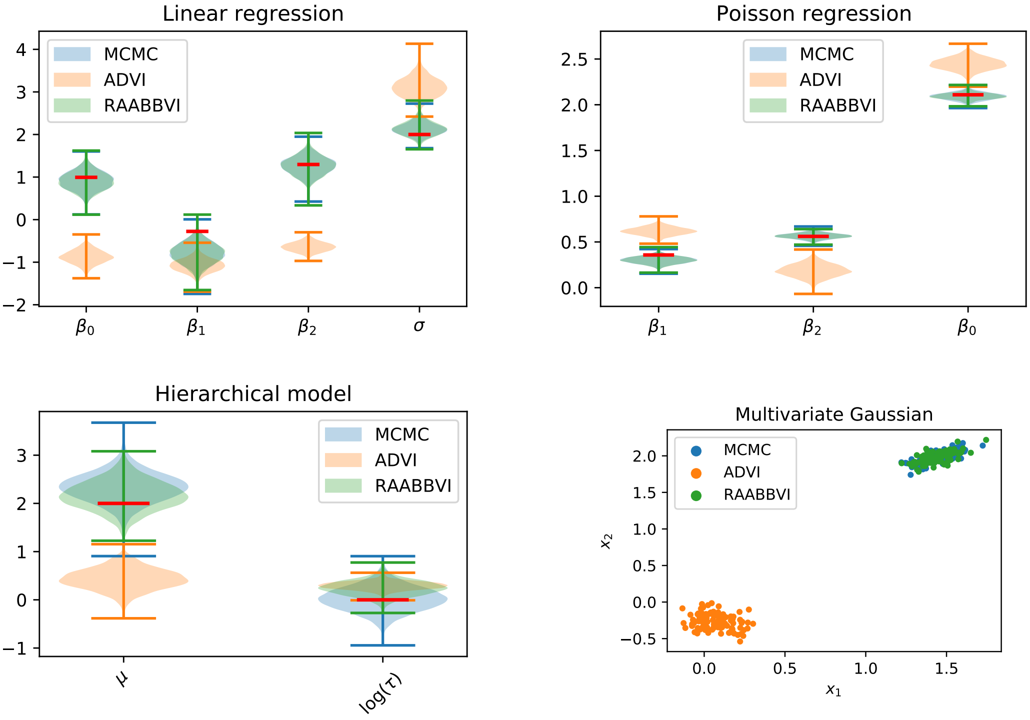


Figure 3.5: Comparison of estimated posterior distribution of each model parameter using $\delta = 0.01$ for ΔELBO-ADVI.

In order to formalize the above observation, we run a simulation study for each data set. For each data set, we simulate $n_{sim} = 10$ copies. For each copy of the data set, we train a ΔELBO-ADVI model and a RAABBVI model and evaluate the accuracy metrics shown in Table 3.1. For the simulation, we increased the complexity of the simulated data sets in comparision to Figure 3.5. Table 3.2 shows the chosen parameters for each data set in the simulation studies.

Table 3.2: Parameters for each data set in the simulation studies.

Data set	Parameter(s)
Linear regression	$N = 1000,\ D = 100,\ \sigma = 2$
Poisson regression	$N = 250, D = 5$
Hierarchical regression	$D = 20$
Multivariate Gaussian	$N = 20$

Note that the metrics in Table 3.1 can only be calculated if the true posterior is known. As the true posterior is not known in general (see Remark 3.2.2), we also fit an MCMC model and use its theoretical guarantees (see Section 2.2) to assume that the MCMC samples are sufficiently close to the true posterior.

For the MCMC models, we use the NUTS-MCMC sampler [Homan and Gelman, 2014] with 10'000 warm up samples and 10'000 regular samples as well as 4 independent chains. Each MCMC model is automatically verified for convergence using the $\hat{R}$ diagnostics [Gelman et al., 2013, Dhaka et al., 2020, Vehtari et al., 2021a].

The results for the linear regression data set are shown in Table 3.3. To get more detailed insights, we split $\theta = \{\boldsymbol{\beta}, \sigma\}$, where $\boldsymbol{\beta}$ captures the coefficients of the mean function and σ the noise variance. The splitting is also important due to the fact that the scales of the β_i and σ are different. Averaging across both may skew the resulting metrics. Therefore, Table 3.3 depicts the accuracy metrics for the posterior distribution for $\boldsymbol{\beta}$ and σ, individually. We observe that RAABBVI outperforms ΔELBO-AVDI for both $\boldsymbol{\beta}$ and σ across all metrics defined in Table 3.1.

The results for the other data sets defined in Section 3.2 are consistent with the insights direct from the linear regression data set and are shown in Appendix A.

Table 3.3: Performance for linear regression $n = 1000$ and $D = 100$

Model	μ-MAE($\boldsymbol{\beta}$)	μ-MAE(σ)	σ-MAE($\boldsymbol{\beta}$)	σ-MAE(σ)	$W(\boldsymbol{\beta})$	$W(\sigma)$	KS($\boldsymbol{\beta}$)	KS(σ)
ΔELBO-ADVI	0.77	0.67	0.04	0.06	0.72	0.74	<1e-5	0.05
RAABBVI	**0.22**	**0.14**	**0.02**	**0.02**	**0.02**	**0.02**	**0.29**	**0.40**

Best performing model in bold

Figure 3.5 and Table 3.3 confirm that ΔELBO-ADVI may result in poor estimates even if there is no model error or for relatively simple data for which the exact posterior distribution lies in the variational distribution family assumed for ADVI (e.g., linear regression data set).

This support our initial hypothesis that one cannot rely on the relative improvement in ELBO only when using ADVI. Hence, it is important to provide additional guidance to practitioners when using VI algorithms for posterior approximation. In contrast, the RAABBVI algorithm successfully managed to provide significantly better approximations as promised by the authors.

Next, we proceed to better understand the drivers for the poor performance of ΔELBO-ADVI.

3.3.3 Experiment 2: Is RAABBVI always better than ΔELBO-ADVI or does it only depend on the stopping criteria?

In this section, we investigate if ADVI achieves better performance when running the gradient descent algorithm as described in Section 2.3.1 for a longer period, i.e., not using ΔELBO as a stopping criteria. We aim to verify if the poor performance of ΔELBO-ADVI in Section 3.3.2 can be partially attributed to early stopping.

As in Section 3.3.2, we use the performance metrics μ-MAE and σ-MAE to asses the quality of different models. As previously, the true posterior is unknow, however we assume that MCMC samples can be used to calculate the performance metrics.

Instead of using a stopping criteria, we continuously train the models and evaluate their performance at pre-specified run time intervals. Figures 3.6 to 3.12 show μ-MAE and σ-MAE for run times $t = 2^0, ..., 2^{12}$.

Note that for RAABBVI, we do not evaluate its performance at different time intervals since determining the optimal runtime is a crucial part of the algorithm (see the Inefficiency Index in Section 2.3.2). Instead, we show a constant red line that starts at the run time when RAABBVI has converged. After that time the algorithm stops training and therefore its performance is not further improved, i.e., remains constant.

Similar to Section 3.3.2, we run multiple simulations for each data set and use the same parameter configuration as described in Table 3.2. The blue lines in Figures 3.6 to 3.12 denote the average performance and the light blue shaded areas the 95% confidence intervals.

We observe that ADVI is able to match the performance of RAABBVI if the algorithms runs for sufficiently many iterations. This suggests that the ΔELBO rule leads to early stopping before ADVI has found a good fit. Also, on our simulated data sets, we observe that for most data sets (i.e., Poisson, Hierarchical and Multivariate Gaussian) ADVI has outperformed RAABBVI with respect to recovering the mean

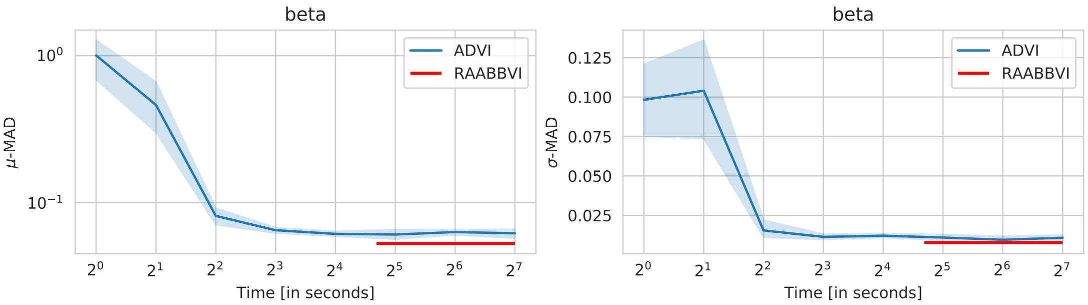


Figure 3.6: Linear regression (1/2): Quality of posterior approximation of $\boldsymbol{\beta}$ as a function of the number of iterations k.

(μ-MAE). However the difference in performance is very marginal. Further, it is worth noting that at the run time when RAABBVI converged (start of red lines), also ADVI has converged to a high quality fit.

Based on this experiment, we draw the following conclusions:

1. ADVI is capable of approximating the posterior distribution with similar precision as RAABBVI if there was a better stopping criteria than the ΔELBO rule.
2. The stopping criteria of RAABBVI based on the Inefficiency Index (see Definition 2.3.2) provides a good stopping rule to ensure that the RAABBVI model achieves high accuracy.
3. For some data sets, ADVI already achieves the optimal performance much earlier than RAABBVI. This raises the question if exist better criteria for ADVI to achieve good performance without being as computationally intensive as RAABBVI.

Based on these concludes, we continue in the next chapter to investigate an alternative workflow that allows to detect convergence for ADVI by practitioners.

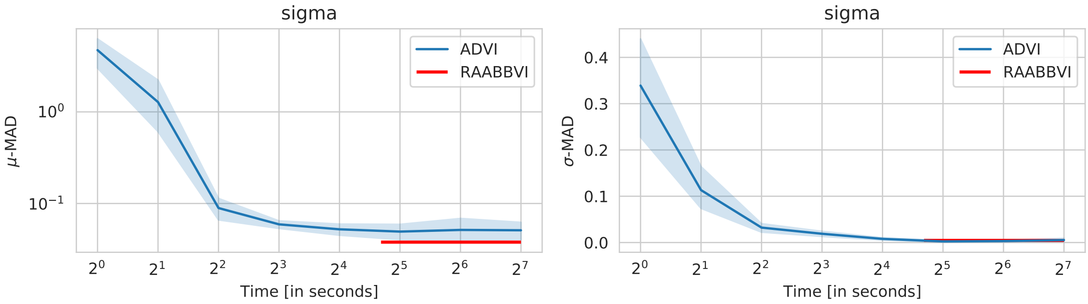


Figure 3.7: Linear regression (2/2): Quality of posterior approximation of σ_0 as a function of the number of iterations k.

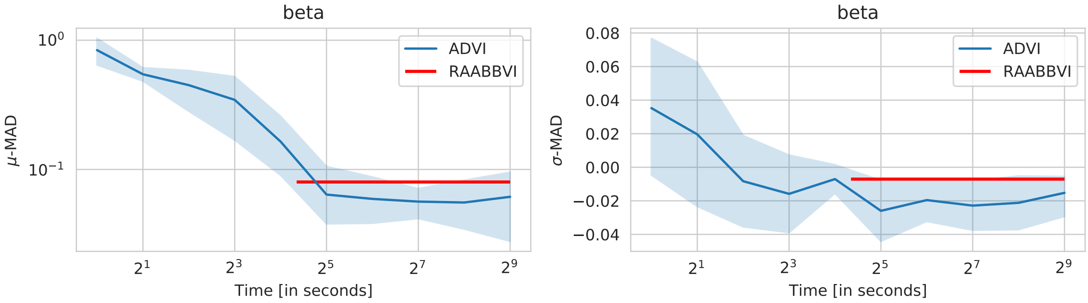


Figure 3.8: Poisson regression: Quality of posterior approximation of $\boldsymbol{\beta}$ as a function of the number of iterations k.

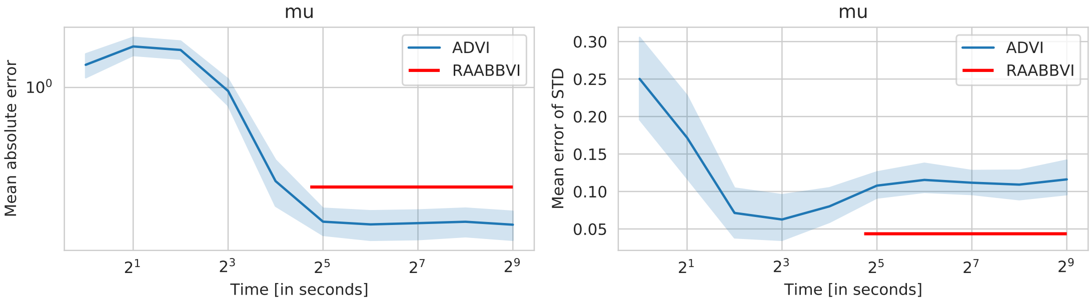


Figure 3.9: Hierarchical regression (1/2): Quality of posterior approximation of μ as a function of the number of iterations k.

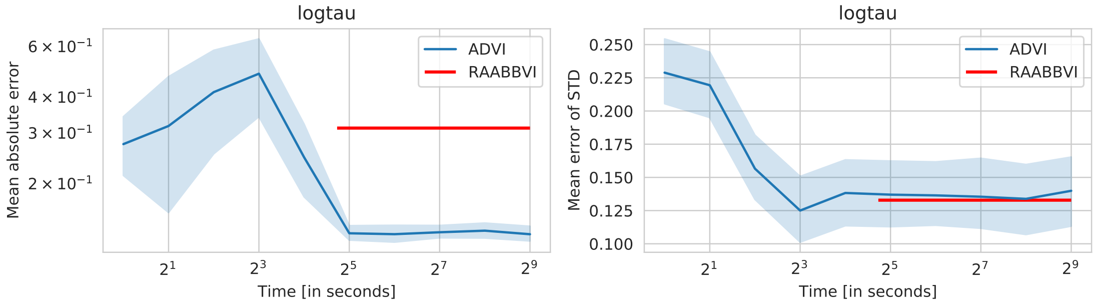


Figure 3.10: Hierarchical regression (2/2): Quality of posterior approximation of $\log(\tau)$ as a function of the number of iterations k.

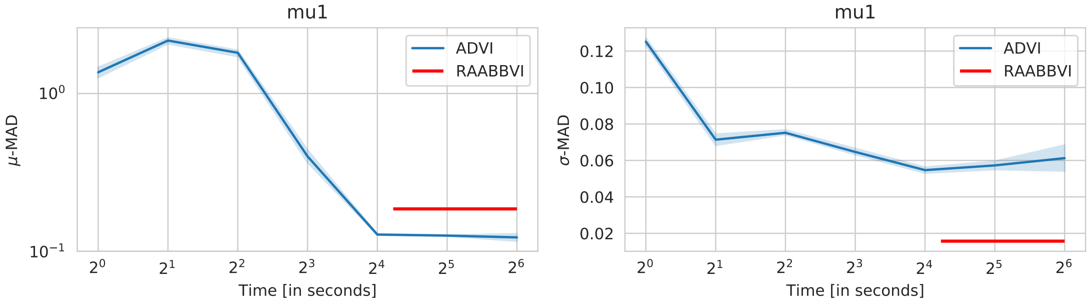


Figure 3.11: Multivariate Gaussian (1/2): Quality of posterior approximation of μ_1 as a function of the number of iterations k.

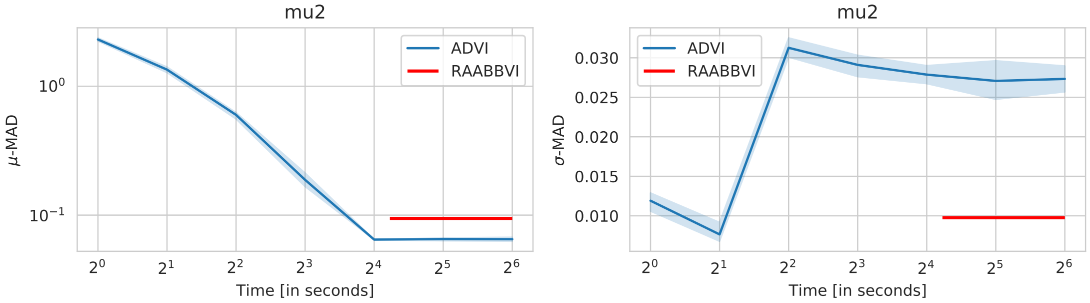


Figure 3.12: Multivariate Gaussian (2/2): Quality of posterior approximation of μ_2 as a function of the number of iterations k.

4

Variational Inference Workflow

Contents

4.1 Introduction

In Section 3.3.3, we showed that ADVI is able to approximate the posterior distribution equally well as RAABBVI if sufficiently many iterations are used for training. However, Section 3.3.2 shows that the widely used ΔELBO criteria does not provide a good stopping criteria for the number of iterations.

In this chatper, we investigate multiple alternative quality metrics that can be used to asses the convergence of the ADVI algorithm. We ground the choice of these rules in existing literature on variational inference, MCMC sampling, and probability theory. First, we define and introduce the theory of the alternative stopping criteria in Sections 4.2.1 to 4.2.3. In Section 4.3, we present the results of a variety of experiments using the data sets defined in Section 3.2 across all quality metrics.

Based on the findings in Section 4.3, we derive a novel Variational Inference workflow in Section 4.4 that helps practitioners to assess the quality of their model when using Variational Inference.

4.2 Quality Metrics for Variational Inference

4.2.1 Pareto Smoothed Importance Sampling (PSIS)

The PSIS coefficient k was proposed by Yao et al. [2018] as a measure to diagnose if a variational approximation provides a good fit to the true posterior. Formally, the coefficient k is given by

$$k := \inf \left\{ k' > 0 : E_q \left(\frac{p(\theta|\boldsymbol{x})}{q(\theta)} \right)^{\frac{1}{k'}} < \infty \right\}, \tag{4.1}$$

i.e., if the variational approximation $q(\theta)$ recovers the posterior $p(\theta \mid \boldsymbol{x})$, then the expectation should be finite even if k' converges to 0. Consequently, a small value of k means that high order moments exists and therefore indicates a good approximation.

Empirical studies by Vehtari et al. [2021b] suggest that for $k < 0.7$, one can conclude that the variational approximation q is close enough to the true density. This can be formulated as a stopping criteria for which we estimate k in each iteration an stop if $k < 0.7$.

Hence it remains to derive an estimator for k. For that, note that k is invariant under multiplications of q or p by a constant. Therefore, we can replace $p(\theta \mid \boldsymbol{x})$ with the known joint distribution $p(\theta, \boldsymbol{x})$ in Equation (4.1).

Furthermore, let F^n be the distribution of the maximum of iid distributed random variables, i.e.,

$$F^n(x) = P(\max\{X_1, ..., X_n\} < x) = P(X_1 < x, ..., X_n < x) = P(X_1 < x)^n.$$

Based on this, we introduce the maximum domain of attraction (MDA) of a distribution F.

Definition 4.2.1 (Maximum Domain of Attraction (MDA))**.** A probability distribution F lays in the MDA of a non-degenerate distribution G, if there exists sequences a_n and b_n such that

$$\lim_{n\to\infty} F^n(a_n x + b_n) = G(x)$$

for all x.

Furthermore, for a random variable X, let F_u be the excess distribution over a threshold u, i.e.,

$$F_u(x) = P(X - u < x \mid X > u).$$

and we denote by $GPD_{k,\beta}$ the generalized Pareto distribution (GPD) given by

$$GPD_{k,\beta} = \begin{cases} 1 - (1 + kx/\beta)^{-1/k}, & k \neq 0 \\ 1 - e^{-x/\beta}, & k = 0. \end{cases}$$

then the following proposition holds:

Proposition 4.2.1. *There exists a function* $\beta(u)$ *such that*

$$\lim_{u\to x_F} \sup_{0\leq x<x_F-u} |F_u(x) - GPD_{k,\beta(u)}(x)| = 0$$

if and only if $F \in MDA(H_k)$*, where* MDA *denotes the Maximum Domain of Attraction of a Generalized Extreme Value distribution* H_k*.*

Remark 4.2.1. Note that by the Fisher-Tippet-Gnedenko theorem [Embrechts et al., 1997], it holds that if a distribution function F belongs to the maximum domain of attraction (MDA) of any non-degenerate probability distribution, then it belongs to the MDA of a generalized extreme value distribution. Hence, the condition $F \in MDA(H_k)$ is applicable to many common distribution, see Example 4.2.1 and Example 4.2.2.

Example 4.2.1 (MDA of Gaussian distribution)**.** The Gaussian distribution lays in the MDA of the standard Pareto distribution with normalizing sequences

$$b_n = \sqrt{2\log n - \log\log n - \log(2\pi)} \quad \text{and} \quad a_n = \frac{1}{b_n}.$$

See Example 1.1.7 in de Haan and Ferreira [2010] for a proof of the above statement.

Example 4.2.2 (Poisson distribution and the MDA)**.** In contrast to the Gaussian distribution, for the Poisson distribution, there are no sequences a_n and b_n such that Definition 4.2.1 holds and therefore Proposition 4.2.1 is not applicable [Leadbetter et al., 1983].

Next, Lemma 4.2.1 details how Proposition 4.2.1 can be used to estimate Equation (4.1).

Lemma 4.2.1 (Moments of GPD)**.** *A Generalized Pareto distribution $GPD_{k,\beta}$ has $1/k$ finite moments.*

The proof for Lemma 4.2.1 can be found in Vehtari et al. [2021b]. We can use Proposition 4.2.1 and Lemma 4.2.1 to derive an estimator for Equation (4.1) by approximating

$$\frac{p(\theta, \boldsymbol{x})}{q(\theta)} \bigg| \frac{p(\theta, \boldsymbol{x})}{q(\theta)} > M \sim GPD_{k,\beta},$$

for a sufficiently large $M > 0$. Algorithm 3 summarizes the computational steps required to estimate k when training a variational inference model.

Algorithm 3 Pareto Smoothed Importance Sampling

```
Require: A Bayesian model $p(\theta, \boldsymbol{x}) = p(\theta)p(\boldsymbol{x} \mid \theta)$ as well as variational
    approximation $q_{\lambda_k}(\theta)$, and number of tail samples $M$.
1: At iteration $t$, sample $S$ observations from $\theta_1, ..., \theta_S \sim q_{\lambda_t}(\theta)$.
2: Calculate the probability ratios $r_s = p(\theta_s, \boldsymbol{x})/q_{\lambda_t}(\theta)$ for $s = 1, ..., S$.
3: Fit a generalized Pareto distribution to the largest $M$ samples of $r_s$.
4: Return the estimated shape parameter $\hat{k}$ of the GPD
```

In terms of computational complexity, Algorithm 3 has multiple components:

1. Sampling from $q_{\lambda_t}(\theta)$: This depends on the variational distribution family. For example, for an D-dimensional multivariate Gaussian the cost is driven by the inversion of the covariance matrix Σ which has a one-off complexity of $\mathcal{O}(D^3)$.

2. Fitting of the GDP distribution can be achieved using the maximum likelihood estimator (MLE). Unfortunately, there does not exist a closed-form expression for the shape parameter k and hence gradient optimization methods have be used.

Based on these observations, it may be very costly to estimate k for each gradient step when minimizing the ELBO. This can be mitigated by only evaluating k every m-th iterations.

4.2.2 Gelman-Rubin $\hat{R}$

One limitation of criteria such as ELBO or PSIS is that they do not considered if the optimization has converged in each dimension of $\theta \in \mathbb{R}^m$. A similar problem exists for MCMC sampling, where practitioners needs to diagnose if the sampled chain have reached stationarity.

The MCMC literature commonly uses the Gelman-Rubin criteria $\hat{R}$ as a general purpose tool to assess the convergence properties of the Markov chain samples. The main idea of $\hat{R}$ is to compare the between-chain variance and the within-chain variance with each other.

Definition 4.2.2 (Gelman-Rubin $\hat{R}$)**.** Given L Markov chains with samples $\theta_1^l, ..., \theta_N^l$ after the warm-up period from an unknown random variable θ, let $\hat{\theta}_l$ and $\hat{\theta}$ be the posterior mean across all chains and within each chain l, i.e.,

$$\hat{\theta}_l = \frac{1}{N}\sum_{i=1}^{N}\theta_i^l \quad \text{and} \quad \hat{\theta} = \frac{1}{L}\sum_{i=1}^{L}\hat{\theta}_l.$$

Similarly, let σ_l^2, B and W be the l-th chain variance, the **B**etween-chain variance, and the **W**ithin-chain variance, i.e.,

$$\sigma_l^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\theta_i^l - \hat{\theta}_l)^2, \quad B = \frac{1}{L-1}\sum_{l=1}^{L}(\hat{\theta}_l - \hat{\theta})^2 \quad \text{and} \quad W = \frac{1}{L}\sum_{m=1}^{L}\sigma_l^2$$

Then

$$\hat{\sigma}^2 = \frac{N-1}{N}W + \frac{L+1}{LN}B$$

is and unbiased estimator for variance of θ [Brooks and Gelman, 1998]. Therefore, the coefficient

$$R = \sqrt{\frac{\hat{\sigma}^2}{\sigma^2}},$$

should be close to 1, by definition, in case of convergence. As σ^2 is unknown, we use W as an (under)estimate for σ^2 and define the (over)estimate for R by

$$\hat{R} = \sqrt{\frac{\hat{\sigma}^2}{W}},$$

$\hat{R}$ can be interpreted as follows: If it is large, further simulations will increase W since the current simulated sequence has not yet explored the full distribution of θ. If it is close to 1, we can conclude that the L chains of length N are close the true target.

Next, we will apply this idea to our stochatic optimization of the variational distribution defined in Equation (2.3). As discussed in Section 2.3.2, the parameter estimates λ_k for each iteration $k = 1, ..., K$ for the variational distribution q_{λ_k} form a homogeneous Markov Chain converging to a stationary distribution μ under suitable convergence criteria.

In contrast to Definition 4.2.2, our setting does not provide multiple chains of λ_k^l. To circumvent this problem, for each iteration i and given windows size W we define the two chains

$$\text{Chain } 1 = \{\lambda_{i-2W}, ..., \lambda_{i-W-1}\}$$
$$\text{Chain } 2 = \{\lambda_{i-W}, ..., \lambda_{i}\}$$

and calculate $\hat{R}$ for Chains 1 and 2.

Note that in general $\lambda_k \in \mathbb{R}^m$, but the estimates in Definition 4.2.2 only focus on the univariate case. We propose to extend the univariate case to the multivariate case by calculating $\hat{R}_i$ for each dimension i and then average across all dimensions, i.e.,

$$\hat{R}^{MV} = \frac{1}{m}\sum_{i=1}^{m} \hat{R}_i.$$

Remark 4.2.2 (Alternative multivariate extensions)**.** Note that Brooks and Gelman [1998] also propose alterantive extensions of $\hat{R}$ to the multivariate settings that estimate the full covariance structure of λ. Additional research may extend the above proposed approach which might lead to further improvements.

Remark 4.2.3 (Optimal threshold)**.** In order to use $\hat{R}^{MV}$ as a stopping criteria, one needs to define a threshold that suggests that a good fit has been found. By Definition 4.2.2 this should should be close to 1. Following the MCMC literature on $\hat{R}$ [Dhaka et al., 2020, Vehtari et al., 2021a], we have chosen 1.1 for our simulation experiments.

Algorithm 4 provides a summarised description of the previously described approach.

Algorithm 4 Gelman-Rubin $\hat{R}$ as a stopping criteria

```
Require: Windows size $W$ and parameters $\lambda_k$ of the variational distribution $q_{\lambda_k}$
    for $k = i - 2W, ...., i$
At iteration $i$ define the two chains
    $\text{Chain } 1 = \{\lambda_{i-2W}, ..., \lambda_{i-W-1}\}$
    $\text{Chain } 2 = \{\lambda_{i-W}, ..., \lambda_{i}\}$
Calculate $\hat{R}$ for each dimension of $\lambda_i \in \mathbb{R}^m$, i.e., the ratio of the between chain
    variance to the in-chain variance
Stop if the average of $\hat{R}^{MV}$ across all $m$ dimensions is below 1.1.
```

4.2.3 Wasserstein-2 bound

Finally, we investigate a third criteria. This criteria is based on the Wasserstein distance [Ramdas et al., 2015] that measures the distance between two probability distributions. Similar to the KL-divergence, it holds that the Wasserstein distance is zero if and only if the two distribution are equal. It, therefore, provides an intuitive choice to assess the quality of a variational approximation to the true posterior distribution. Furthermore, the Wasserstein-2 distance provides useful bounds for the difference in mean and standard deviations as summarized in Proposition 4.2.2.

Proposition 4.2.2. *For any probability distribution π, let μ_π its mean, Σ_π its covariance matrix, and $\sigma_{\pi,i} = \Sigma_{\pi,ii}^{1/2}$ the i-th component marginal standard deviation. Then for two probability distribution π and $\hat{\pi}$ with $\mathcal{W}_2(\hat{\pi}, \pi) \leq \varepsilon$, it holds that*

$$\|\mu_{\hat{\pi}} - \mu_\pi\| \leq \varepsilon, \quad \max_i |\sigma_{\hat{\pi},i} - \sigma_{\pi,i}| \leq \varepsilon,$$
$$\textit{and} \quad \|\Sigma_{\hat{\pi}} - \Sigma_\pi\| < 2\varepsilon(\varepsilon + \sqrt{\min\{\|\Sigma_{\hat{\pi}}\|, \|\Sigma_\pi\|\}}.$$

In other words, we can use Proposition 4.2.2 to bound the maximum error for important statistics of a variational approximation of the posterior distribution if we are able to bound the Wasserstein-2 distance. A proof of Proposition 4.2.2 can be found in Huggins et al. [2020].

Unfortunately, the exact Wasserstein-2 distance between a variational approximation q and the true posterior $p(\theta \mid \boldsymbol{x})$ cannot be calculated without knowing the true posterior. However, Huggins et al. [2020] provide bounds for the Wasserstein distance that we can build on to derive an upper bound.

Proposition 4.2.3. *If π is absolutely continuous with respect to $\hat{\pi}$, and if $\hat{\pi}$ is p-exponentially integrable, then*

$$\mathcal{W}_p(\hat{\pi}, \pi) \leq C_p(\hat{\pi}) \left(KL(\pi|\hat{\pi})^{\frac{1}{p}} + [KL(\pi|\hat{\pi})/2]^{\frac{1}{2p}} \right) \tag{4.2}$$

with

$$C_p(\hat{\pi}) = 2 \inf_{\theta_0, \varepsilon} \left(\frac{1}{\varepsilon} \left[\frac{3}{2} + \log \int \exp\left(\varepsilon \|\theta - \theta_0\|_2^p\right) d\hat{\pi}(\theta) \right] \right)^{\frac{1}{p}} < \infty$$

Proposition 4.2.4. *Let π be absolute continuous with respect to η, then*

$$KL(\pi \mid \hat{\pi}) \leq H(\hat{\pi}) := 2\left(CUBO_2(\hat{\pi}) - ELBO(\eta)\right)$$

This allow us to bound the intractable KL divergence terms in Proposition 4.2.3 using a known distribution η. Further more combining Proposition 4.2.5 and Proposition 4.2.4 provides a bound for $W_p(\hat{\pi}, \pi)$ that can be estimates without access to π. This makes it accessible for our use case, i.e., $\hat{\pi}$ may denote the variational approximation and π the unknown true posterior $p(\theta \mid \boldsymbol{x})$.

Unfortunately, the bound in Equation (4.2) consists of the multiplication of two expressions that both need to be estimated from samples of $\hat{\pi}$ and η. This can lead to vary noise estimate. In addition, the constant $C_p(\hat{\pi})$ itself is prone to very high variance as it contains the expectation of the exponential of $\hat{\pi}$.

Fortunately, we can extend the results from Huggins et al. [2020] by assuming that $\hat{\pi}$ is Multivariate Gaussian. This allows us to explicitly calculate a bound for the constant $C_p(\hat{\pi})$ for $p = 2$. Furthermore, the Multivariate Gaussian assumption holds for the ADVI algorithm.

Proposition 4.2.5. *If $\theta \sim \hat{\pi} = \mathcal{N}(\mu, \Sigma)$, with eigenvalues $\lambda_1, ..., \lambda_p$ for Σ, $\theta_0 = \mu$ and $\varepsilon < \min(1/(2\lambda_i))$, then*

$$\log \int \exp\left(\varepsilon \|\theta - \theta_0\|_2^2\right) d\hat{\pi} = \sum_i \log\left((1 - 2\lambda_i \varepsilon)^{-0.5}\right)$$

and hence

$$C_2(\hat{\pi}) \leq 2\left(\frac{1}{\varepsilon}\left[\frac{3}{2} + \sum_i \log\left((1 - 2\lambda_i\varepsilon)^{-0.5}\right)\right]\right)^{\frac{1}{2}}.$$

Proof. Let $\hat{\pi}(\theta) \sim \mathcal{N}(\theta|\mu, \Sigma)$, choose $\theta_0 = \mu$, then $X = \theta - \theta_0 \sim \mathcal{N}(0, \Sigma)$ and

$$\log \int \exp\left(\varepsilon \|\theta - \theta_0\|_2^2\right) d\hat{\pi} = \log E(\varepsilon \exp(\|X\|_2^2)).$$

Because Σ is positive-semidefinitve, there exists an orthogonal matrix P such that $\Sigma = PDP^T$ with $D = \mathrm{diag}(\lambda_1, ..., \lambda_p)$ and therefore it holds that

$$\begin{aligned}
\|X\|_2^2 &= \|P\|\|Y\|_2^2\|P\|^T \quad \text{with } Y \sim \mathcal{N}(0, D) \\
&= \|Y\|_2^2 \quad \text{(due to orthogonality of P)} \\
&= \sum_{k=1}^{p} Y_i^2 \quad \text{with independent } Y_i \sim \mathcal{N}(0, \lambda_i) \\
&= \sum_{k=1}^{p} \lambda_i Z_i^2, \quad \text{with i.i.d. } Z_i \sim \mathcal{N}(0, 1) \\
&= \sum_{k=1}^{p} \lambda_i K_i \quad \text{with i.i.d. } K_i \sim \chi^2(1).
\end{aligned}$$

Since $E(\exp(\varepsilon\lambda_i K_i) = (1 - 2\varepsilon\lambda_i)^{-1/2}$, it follows that and hence,

$$E(\exp(\varepsilon\|X\|_1^2)) = \prod_{k=1}^{p} E(\exp(\varepsilon\lambda_i K_i)) = \prod_{k=1}^{p} (1 - 2\varepsilon\lambda_i)^{-1/2}.$$

□

Proposition 4.2.5 allow us to calculate the constant $C_2(\hat{\pi})$ without the need for sampling from $\hat{\pi}$ and relying on noisy Monte Carlo estimates. This has both computational and accuracy benefits.

Finally, combining Proposition 4.2.3, Proposition 4.2.4, and Proposition 4.2.5 provides a bound for the Wasserstein-2 distance between the variational distribution q and the unknown posterior $p(\theta \mid \boldsymbol{x})$.

Similarly to the ΔELBO criteria, we can continuously calculate the upper bound given in Proposition 4.2.3 during model training. We stop the training algorithm if the relative improvements of the bound is smaller than a threshold δ. For our simulation results, we choose $\delta = 0.01$ as it is also the default for the ELBO criteria.

Remark 4.2.4 (Absolute continuity of $p(\theta \mid \boldsymbol{x})$ with respect to q)**.** Proposition 4.2.3 assumes that π is absolutely continuous with respect to $\hat{\pi}$. By definition, for two probability measures $\hat{\pi}$ and π with sample space Ω and σ-algebra $\mathcal{F}$, it holds that π is absolutely continous with respect to $\hat{\pi}$ if for any $A \in \mathcal{F}$ with $\pi(A) = 0$, then $\hat{\pi}(A) = 0$.

In case of the ADVI algorithm, this holds by design since $\operatorname{supp}(q_\lambda(\theta)) \subseteq \operatorname{supp}(p(\theta \mid \boldsymbol{x}))$ (see Section 2.3.1). Therefore, we can conclude that $p(\theta \mid \boldsymbol{x})$ is absolutely continuous with respect to q.

Remark 4.2.5 (Choice of η)**.** For Proposition 4.2.4 we have to choose η such that $p(\theta \mid \boldsymbol{x})$ is absolutely continuous with respect to η. Due to Remark 4.2.4, we can choose $\hat{\pi} = \eta$.

4.3 Performance comparison across all criteria

In this section, we compare the performance of the alternative quality criteria introduced in Sections 4.2.1 to 4.2.3 on the data sets introduced in Section 3.2. Similarly to the ΔELBO rule, we use these criteria as stopping rules to automatically detect convergence of the ADVI algorithm. For comparison, we also report the performance of the ΔELBO criteria as well as RAABBVI as an alternative algorithm.

In terms of accuracy metrics, we report the μ-MAE as well as σ-MAE as defined in Table 3.1. Similar to previous estimates, we assume that the true posterior is sufficiently well approximated by an MCMC model using the NUTS sampler. To improve robustness of the results, we simulate $n_{sim} = 10$ copies of each data sets and train the models newly on each simulation. The accuracy metrics are then averaged across the simulations.

Table 4.1 shows the performance of different models in approximating the average of the posterior distribution of each parameter, i.e., μ-MAE. In addition, Table 4.2 shows the performance of approximating the variance of the posterior, i.e., σ-MAE.

Table 4.1: μ-MAE for each data set.

		ADVI				
Data set	Parameters	ΔELBO	PSIS	$\hat{R}^{MV}$	W_2	RAABBVI
Linear regression	$\boldsymbol{\beta}$	1.49	0.36	**0.06**	0.94	0.17
	σ	6.99	1.44	**0.09**	8.39	0.12
Poisson regression	$\boldsymbol{\beta}$	0.62	1.01	0.03	0.46	**0.01**
Hierarchical regression	μ	1.25	**0.15**	0.18	0.98	0.25
	$\log(\tau)$	0.59	0.36	**0.23**	0.51	0.51
Multivariate Gaussian	μ_1	1.0	0.23	0.22	**0.11**	0.24
	μ_2	1.9	0.2	**0.19**	2.95	0.20

From Table 4.1, we note that for the majority of data sets and parameter $\hat{R}^{MV}$-ADVI performed best in recovering the average of the posterior. The method consistently outperforms ΔELBO-ADVI. Furthermore, $\hat{R}^{MV}$-ADVI even outperforms RAABBVI on all data sets except the Poisson regression. Having said that the accuracy of $\hat{R}^{MV}$-ADVI for the Poisson regression data set is only marginally behind RAABBVI.

PSIS-ADVI offers an improvement against ΔELBO except for the Poisson regression. This observation is not surprising when taking Example 4.2.2 into account. Example 4.2.2 shows that the Poisson distribution does not fulfill the assumptions that are required for Proposition 4.2.1 which builds the basis for the estimator for the PSIS coefficient k (see Equation (4.1)).

Finally, we note that W_2-ADVI offers some improvements in comparison to ΔELBO but it is not competitive against $\hat{R}^{MV}$. Our simulations have shown that the our upper bound of the Wasserstein-2 distance is not always very tight. Therefore, it might not always be a good measure to assess convergence.

Table 4.2 shows the performance of the various models in terms of approximating the variance of the posterior distribution. Similar to the mean, we observe that $\hat{R}^{MV}$-ADVI performs best. In the linear regression data set is even significantly outperforms RAABBVI in approximating the variance of the regression coefficients with an error of 0.01 vs 0.15.

Table 4.2: σ-MAE for each data set.

		ADVI				
Data set	Parameters	ΔELBO	PSIS	$\hat{R}^{MV}$	W_2	RAABBVI
Linear regression	$\boldsymbol{\beta}$	0.05	0.06	**0.01**	0.16	0.15
	σ_0	1.02	0.23	**0.06**	1.32	0.23
Poisson regression	$\boldsymbol{\beta}$	0.04	0.05	**0.01**	0.03	**0.01**
Hierarchical regression	μ	0.11	0.12	0.09	0.06	**0.02**
	$\log(\tau)$	0.31	0.27	0.26	0.26	**0.25**
Multivariate Gaussian	μ_1	0.05	**0.01**	**0.01**	0.16	0.02
	μ_2	0.06	**0.01**	**0.01**	0.12	**0.01**

4.4 Variational Inference Workflow

Our results in Chapter 3 show the importance of having a robust workflow when applying variational inference in practice. Even when applying VI to simple data sets without model specification error, commonly used methods may lead to poor estimates. Especially ΔELBO-ADVI which is the default for Stan and therefore widely used has to be be treated with caution.

However, we can confirm that the newly proposed RAABBVI algorithm [Welandawe et al., 2022] does offer significant help for practitioners as it is more robust due to the integrated convergence checks. Having said that we did not observe that RAABBVI offers significantly better fits than ADVI alternatives. In fact using $\hat{R}^{MV}$-ADVI even outperformed RAABBVI on many data sets.

At the same time, using RAABBVI comes with a risk for practitioners. First, it builds on variety of assumptions that cannot be easily verified or understood by practitioners that are purely interested in VI as a tool rather than its theory. Second, RAABBVI is currently not supported in common probabilistic programming frameworks such as pyro or Stan. Both risks limit the ability of a practitioner to debug RAABBVI if something goes wrong or extend it if the existing implementation does not support a certain set up (see Section 2.3.2).

Instead of using an end-to-end back-box model with complex integrated convergence checking (such as RAABBVI), we advocate the usage of a robust workflow when applying Variational Inference. For VI, such a workflow includes the evaluation of quality and convergence metrics after model inference. This can also be seen as an extension of the Bayesian workflow proposed by Gelman et al. [2020] for VI. This approach will also be familiar to practitioners that have used MCMC is the past. For MCMC, most available implementations offer standardized metrics for diagnostics after sampling. It is an overdue extension to start offering similar diagnostics also for VI.

In terms of diagnostics criteria, Section 4.2 provides a good resource of criteria to be used. From our findings in Section 4.3, we saw that there does not exist one single best metric and therefore we encourage to evaluate a broad selection when using VI. The python code assosciated with the thesis provides off-the-shelve implementations of all criteria discussed in Section 4.2 in the probabilistic programming framework numpyro.

Also, we note that the quality metrics in Section 4.2 are not limited to ADVI. They can be adapted to other VI algorithms if needed. Furthermore, all metrics are computationally cheap and may even be implemented as online metrics that are updated during model training.

5

Discussion

The focus of this thesis lays on helping practitioners that develop Bayesian models and consider VI an alternative to MCMC for Bayesian inference. Building on their needs, we have chosen the data sets in Section 3.2 and conducted simulations in Chapter 3.

However, more often Bayesian inference is also being applied to problems at even larger scale, e.g., Bayesian deep learning [Vasconcelos et al., 2022, Notin et al.]. In those examples, the dimensions m of the parameter λ of the variational distribution q_λ may be significantly larger than in our experiments. Additional research is required to understand the behaviour of the criteria introduced in Section 4.2 in this setting.

Furthermore, we have focused our evaluation on simulated data. This choice has been made deliberately as it allows us to evaluate the performance of various methods in more details. However, additional research is required to study the setting of model mis-specifications. This is linked *Challenges (1) and (2)* as mentioned in Section 1.1. In practice, it is typically not possible to perfectly specify the likelihood $p(\boldsymbol{x} \mid \theta)$ and therefore model mis-specification may be involved.

Appendices

A
Appendix

Table A.1: Performance for Poisson regression.

Model	μ-MAE($\boldsymbol{\beta}$)	σ-MAE($\boldsymbol{\beta}$)	$W(\hat{\pi}, \pi)$	KS-Test
ΔELBO	0.79	0.01	0.71	<1e-5
RAABBVI	0.11	0.02	0.04	0.005

Table A.2: Performance for Hierarchical Regression.

Model	μ-MAE(μ)	μ-MAE($\log(\sigma)$)	σ-MAE(μ)	σ-MAE($\log(\sigma)$)	$W(\mu)$	$W(\log(\sigma))$	KS(μ)	KS($\log(\sigma)$)
ΔELBO	0.48	0.67	0.25	0.08	0.49	1.54	0.3	0.0
RAABBVI	0.43	0.14	0.21	0.05	0.42	0.17	0.4	0.9

Bibliography

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20: 28:1–28:6, 2019. URL `http://jmlr.org/papers/v20/18-403.html`.

Stephen Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *J. Comput. Graphi. Stat.*, 7:434–455, 12 1998. doi: 10.1080/10618600.1998.10474787.

Laurens de Haan and Ana Ferreira. *Extreme Value Theory: An Introduction (Springer Series in Operations Research and Financial Engineering).* Springer, 1st edition. edition, 2010. ISBN 144192020X.

Akash Kumar Dhaka, Alejandro Catalina, Michael R Andersen, Må ns Magnusson, Jonathan Huggins, and Aki Vehtari. Robust, accurate stochastic optimization for variational inference. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10961–10973. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/7cac11e2f46ed46c339ec3d569853759-Paper.pdf`.

Aymeric Dieuleveut, Alain Durmus, and Francis Bach. Bridging the gap between constant step size stochastic gradient descent and markov chains, 2017. URL `https://arxiv.org/abs/1707.06386`.

Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling extremal events*, volume 33 of *Applications of Mathematics (New York).* Springer-Verlag, Berlin, 1997. ISBN 3-540-60931-8. For insurance and finance.

A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian Data Analysis, Third Edition.* Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013. ISBN 9781439840955. URL `https://books.google.ch/books?id=ZXL6AQAAQBAJ`.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow, 2020. URL `https://arxiv.org/abs/2011.01808`.

Matthew D. Homan and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, jan 2014. ISSN 1532-4435.

Jonathan Huggins, Mikolaj Kasprzak, Trevor Campbell, and Tamara Broderick. Validated variational inference via practical posterior error bounds. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1792–1802. PMLR, 26–28 Aug 2020. URL `https://proceedings.mlr.press/v108/huggins20a.html`.

James Johndrow, Paulo Orenstein, and Anirban Bhattacharya. Scalable approximate mcmc algorithms for the horseshoe prior. *Journal of Machine Learning Research*, 21 (73):1–61, 2020. URL `http://jmlr.org/papers/v21/19-536.html`.

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic variational inference in stan. 2015. doi: 10.48550/ARXIV.1506.03431. URL `https://arxiv.org/abs/1506.03431`.

M. R. Leadbetter, G. Lindgren, and H. Rootzen. *Extremes and Related Properties of Random Sequences and Processes.* Springer Series in Statistics. Springer Verlag, 1983. ISBN 0387907319.

F. J. Massey. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

Pascal Notin, José Miguel Hernández-Lobato, and Yarin Gal. Principled uncertainty estimation for high dimensional data.

Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of nonparametric tests, 2015. URL `https://arxiv.org/abs/1509.02237`.

Donald B. Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4):377–401, 1981. ISSN 03629791. URL `http://www.jstor.org/stable/1164617`.

Qifan Song, Yan Sun, Mao Ye, and Faming Liang. Extended stochastic gradient mcmc for large-scale bayesian variable selection, 2020. URL `https://arxiv.org/abs/2002.02919`.

Stan Development Team. Stan modeling language, version 2.18.0, 2018. URL `http://mc-stan.org/`.

Francisca Vasconcelos, Bobby He, Nalini Singh, and Yee Whye Teh. Uncertainr: Uncertainty quantification of end-to-end implicit neural representations for computed tomography, 2022. URL `https://arxiv.org/abs/2202.10847`.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved r^ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2), jun 2021a. doi: 10.1214/20-ba1221. URL `https://doi.org/10.1214%2F20-ba1221`.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling, 2021b.

Manushi Welandawe, Michael Riis Andersen, Aki Vehtari, and Jonathan H. Huggins. Robust, automated, and accurate black-box variational inference, 2022. URL `https://arxiv.org/abs/2203.15945`.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5581–5590. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/yao18a.html`.