Adaptive Optimizer Design for Constrained Variational Inference

Alp Sarı
TU Eindhoven
Eindhoven, Netherlands
a.sari@student.tue.nl

Semih Akbayrak
TU Eindhoven
Eindhoven, Netherlands
s.akbayrak@tue.nl

İsmail Şenöz
TU Eindhoven
Eindhoven, Netherlands
i.senoz@tue.nl

Bert de Vries
TU Eindhoven
Eindhoven, Netherlands
bert.de.vries@tue.nl

***Abstract*—This paper addresses the problem of implementing robust and hyperparameter-free natural gradient variational inference. Natural gradient methods are often employed in variational inference strategies, which maximize a variational lower bound on the model evidence. Generally, gradient-based optimization algorithms require the user to pre-specify values for hyperparameters such as step size and number of iterations. Optimal values for these hyperparameters are problem-specific and may significantly affect the algorithm's performance. We propose a model-aware optimizer that adaptively adjusts its step size parameter. The proposed optimizer determines the necessary number of iterations and evaluates the accuracy of the variational approximation, compared to the actual posterior distribution, using convergence diagnostics. We verify in this paper that the proposed adaptive optimizer alleviates the fine tuning problem with no manually initialized step size and a number of iterations. The performance of the optimization results is reported using the convergence diagnostics implemented within the proposed optimizer.**

***Index Terms*—Bayesian Diagnostics, Constrained Bayesian Inference, Exponential-family Distributions, Hyper-parameter Free Optimization, Natural Gradient Variational Inference, Stochastic Gradients**

I. Introduction

We address the problem of implementing robust and hyper-parameter free natural gradient variational inference. When exact Bayesian inference to find a posterior distribution of a parameter is not tractable, an approximate distribution for the posterior estimate is searched. Finding this approximate posterior using optimization and through variational calculus is known as variational inference [1]. Conjugate-computation variational inference (CVI) is a variational inference algorithm that uses stochastic gradients on the non-conjugate term whereas using efficient conjugate computations on the conjugate term [2]. CVI is highly dependent on the choice of hyper parameters, such as the step size and the number of iterations. The optimal choice of these hyper-parameters differs for each problem, so it requires fine tuning, which is time consuming. This paper focuses on developing an automatized optimizer by making modifications to the CVI algorithm with already existing methods on the variational inference literature and adding more heuristics to the approaches when necessary. In this paper we will show:

- Limitations of vanilla implementations of the CVI algorithm in Sec. II-C and how to address them in Sec. III.
- Methods to automate the inference process by automatically determining proper step size and number of iterations to prevent manual fine tuning in Sec. III-B and Sec. III-C, respectively.
- How to evaluate the accuracy of our variational approximation in Sec. III-D.

II. Conjugate Computation Variational Inference

A. Variational Objective

In Bayesian inference, a model is specified as joint distribution:

$$p(y, z) = p(y|z)p(z) \tag{1}$$

where y stands for observations and z stands for latent variables of the model. Having observed y, we can use the Bayes rule to calculate the posterior distribution of latent variables z as:

$$p(z|y) = \frac{p(y|z)p(z)}{\int p(y|z)p(z)\,\mathrm{d}z} \tag{2}$$

The problem is, the marginal likelihood term $\left(\int p(y|z)p(z)\,\mathrm{d}z\right)$ might be intractable. This usually happens when the prior term is not a *conjugate prior* to the likelihood.. $p(z)$ is called a *conjugate prior* for the likelihood $p(y|z)$ if the posterior $p(z|y)$ is in the same probability distribution family as the prior distribution $p(z)$ [3, Ch. 2].

A workaround would be to introduce another distribution $q(z)$ that will approximate the exact posterior $p(z|y)$ [1]. Then the marginal likelihood term can be rewritten as:

$$p(y) = \int p(y|z)p(z)\,\mathrm{d}z = \int \frac{p(y|z)p(z)}{q(z)} q(z)\,\mathrm{d}z \tag{3}$$

Using Jensen's inequality [4], we obtain a lower bound on the log-likelihood function, also known as the evidence lower bound ELBO, given by the expression:

$$\mathcal{L}[q] \triangleq \int \log\left(\frac{p(y,z)}{q(z)}\right) q(z)\mathrm{d}z = \mathbb{E}_q\left[\log\left(\frac{p(y,z)}{q(z)}\right)\right]. \tag{4}$$

Our objective is to maximize $\mathcal{L}[q]$ with respect to our approximate variational distribution $q(z)$.

Note that ELBO $\mathcal{L}[q]$ is a *functional*, a function of functions, in this setting, without further assumptions on $q(z)$. In variational inference, we further assume a fixed-form variational approximation $q_\lambda(z)$, parametrized by λ. Then, functional

maximization problem reduces to maximization of a function $\mathcal{L}(\lambda)$ with respect to its parameters. Thus, we are trying to find optimal values for the parameters λ by the following optimization problem:

$$\begin{aligned}\max_{\lambda\in\Omega}\mathcal{L}(\lambda) &= \mathbb{E}_q\left[\log\left(\frac{p(y|z)p(z)}{q_\lambda(z)}\right)\right] \\ &= \mathbb{E}_q\left[\log\left(p(y|z)\right) - \log\left(\frac{q_\lambda(z)}{p(z)}\right)\right]\end{aligned} \tag{5}$$

where Ω is the space of valid parameters [1].

B. CVI Algorithm

CVI method utilizes conjugate computations on the conjugate part of the model, whereas it computes the natural gradients on the non-conjugate part of the model [2]. For CVI, the variational approximation is chosen to be in the minimal exponential family of distributions. A distribution $q_\lambda(z)$ in the exponential family with natural parameters λ has the probability density function of the following form:

$$q_\lambda(z) = h(z)\ \exp\left(\phi(z)^T\lambda - A(\lambda)\right) \tag{6}$$

where $h(z)$ is the base measure, $\phi(z)$ is the sufficient statistics vector and the $A(\lambda)$ is the log-partition function. An exponential family representation is called minimal if the components of the sufficient statistics vector are linearly independent [5, Ch. 3] . We would assume that our prior distribution $p_{\lambda_p}(z)$ is in the same minimal exponential family as the variational approximation, with natural parameters λ_p. Thus, the conjugate part is the prior term $p_{\lambda_p}(z)$, and the non-conjugate term is the log-likelihood term $p(y|z)$. CVI algorithm updates the parameters using a natural gradient descent approach:

$$\lambda \leftarrow \lambda + \beta\hat{g} \tag{7}$$

The natural gradient of the ELBO $\hat{g} \triangleq \nabla_m\mathcal{L}$ is the Euclidean gradient with respect to the expectation parameters $m \triangleq \mathbb{E}_q\left[\phi(z)\right]$ and can also be computed as $m = \nabla_\lambda A(\lambda)$ for minimal exponential family distributions. Calculating natural gradients also give rise to local exponential-family approximations of the non-conjugate terms. Then, combining (5) and (7), the natural gradient update for the CVI algorithm becomes:

$$\lambda \leftarrow \lambda + \beta\left[\nabla_m\mathbb{E}_q\left[\log\left(p(y|z)\right)\right] + \lambda_p - \lambda\right] \tag{8}$$

using the property that:

$$\nabla_m\left(\mathbb{E}_q\left[\log\left(\frac{p_{\lambda_p}(z)}{q_\lambda(z)}\right)\right]\right) = \lambda_p - \lambda. \tag{9}$$

For more details about the derivation of (8), we refer the interested reader to [2].

C. Considerations Using CVI

In CVI, the parameters of $q_\lambda(z)$ are updated using natural gradients to optimize ELBO. Using such an approach comes with practical considerations, such as:

- The update scheme does not take into account the constraints of the parameters by default. For example, the precision parameter of a Gaussian distribution must be positive-definite. Since there are no constraints on the values of step size or the natural gradient vector can take, this constraint may be violated in (8). The constraints are addressed in Sec. III-A.
- Convergence of gradient-based methods are dependent on the hyper-parameters, which are step size and the number of iterations, and the optimal choice of these parameters differ for different model specifications. CVI algorithm does not offer any specification for these parameters. Finding appropriate parameters for the step size and the number of iterations are addressed in Sec. III-B and Sec. III-C, respectively.
- After a given number of iterations, CVI algorithm does not give information about the convergence of the parameters. A metric to evaluate the posterior approximation is given in Sec. III-D.

III. Adaptive Optimizer Design for Constrained CVI

Our proposed optimizer addresses the problems mentioned in Sec. II-C. We propose a modification to the CVI update using existing methods in the variational inference literature and adding more heuristics to the approaches when necessary. The proposed optimizer is capable of initializing and updating the hyper-parameters of the inference process, adapting to the given model.

A. Handling Positive Definite Constraints of the Parameters

A modified version of the CVI update (8) is proposed in [6], which is called the improved Bayesian Learning Rule (iBLR). This update scheme handles the positive-definite constraints of the valid parameter space when the approximation $q_\lambda(z)$ attains a certain parameterization, which the authors call *block-coordinate natural parameterization*(BCN). This modification allows us to freely choose the step size parameter β_t. For the sake of completeness, we briefly summarize iBLR approach below.

Let BCN parameters are denoted with $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}$ contains blocks of parameters as $\boldsymbol{\lambda} = \{\boldsymbol{\lambda^{[1]}}, ..., \boldsymbol{\lambda^{[n]}}\}$. Let λ^{a_i} denote the parameter at $a-$th entry of the $i-$th block parameter $\boldsymbol{\lambda^{[i]}}$, $\hat{g}^{c_i}$ denote the c-th entry of natural gradient $\boldsymbol{\hat{g}^{i}}$ with respect to $\boldsymbol{\lambda^{[i]}}$. Then, modified gradient ascent update takes the form:

$$\lambda^{c_i} \leftarrow \lambda^{c_i} + \beta_t\hat{g}^{c_i} - \frac{\beta_t^2}{2}\sum_{a_i}\sum_{b_i}\Gamma^{c_i}_{a_ib_i}\hat{g}^{a_i}\hat{g}^{b_i} \tag{10}$$

where each summation is to sum over all entries of the $i-$th block, $\Gamma^{c_i}_{a_ib_i} := \frac{1}{2}\partial_{m_{c_i}}\partial_{\lambda^{a_i}}\partial_{\lambda^{b_i}}A(\lambda)$ and m_{c_i} is the $c-$th entry of the expectation parameter $\boldsymbol{m_{[i]}} := \nabla_{\lambda^{[i]}}A(\lambda)$.

Note that (10) only differs from CVI update by the last term $-\frac{\beta_t^2}{2}\sum_{a_i}\sum_{b_i}\Gamma^{c_i}_{a_ib_i}\hat{g}^{a_i}\hat{g}^{b_i}$, which is to take the curvature information in a Riemannian manifold into account. For some of the BCN parameterizations, such as for the Gaussian distribution, (10) can be efficiently applied. For the list of BCN parameterizations in the exponential family of distributions, their simplified update rules and the detailed derivation of their update rules, see [6].

B. Determining Step Size

In this section, a fast heuristic approach and an adaptive method are presented to find an appropriate step size parameter β_t.

1) Inexact Line Search to Determine Step Size: To determine the step size, our optimizer uses an inexact line search method, which is a heuristic approach. Note that inexact line search inequality conditions used in Euclidean spaces cannot be utilized directly since our parameter space induces a Riemannian manifold. An adaptation of line search methods to manifolds can be utilized, but the study of it is left for future work.

Our heuristic approach searches for an appropriate step size only for the first iteration to be computationally efficient. Found step size is kept fixed throughout the optimization.

2) Adaptive Step Size: Adaptive step size approach is based on [7], which is developed for stochastic variational inference. Stochastic variational inference is used to scale variational inference to models with large data sets by instead of computing a batch gradient, a sample data from the data set is used to calculate the gradient for its computational efficiency. The proposed method determines the step size β_t such that it minimizes the expected distance between updated parameters λ_{t+1}, where the update from λ_t to λ_{t+1} is the CVI update given in (7), and the updated parameters using the whole batch λ_*, by minimizing the expectation of the following cost function:

$$J(\beta_t) = (\lambda_{t+1}(\beta_t) - \lambda_*)^T(\lambda_{t+1}(\beta_t) - \lambda_*) \tag{11}$$

The cost function J is a function of step size β_t through λ_{t+1} term and is a random variable since update from λ_t to λ_{t+1} includes the natural gradient term $\hat{g}_t$ in (7). Thus, the minimization is done for its expectation value $\mathbb{E}\left[J|\lambda_t\right]$, given the current iterate λ_t. Minimizing the expectation yields the optimal step size as:

$$\beta_t^* = \frac{\mathbb{E}[\hat{g}_t]^T\mathbb{E}[\hat{g}_t]}{\mathbb{E}[\hat{g}_t{}^T\hat{g}_t]} \tag{12}$$

and the expectations can be calculated using moving average windows and they can be plugged in (12), to calculate β_t at each time step t. For the detailed derivation of the result, see [7].

C. Determining the Number of Iterations

Convergence of CVI optimization scheme is highly dependent on the number of iterations. Doing too many iterations might result in unnecessary increase in the computation time, whereas small number of iterations might result in premature termination of the optimization process before convergence. Unfortunately, there does not exist a specified number of iterations, which is optimal for any optimization problem. But, if the convergence of the parameters can be checked using some diagnostics, we can come up with a stopping criteria which will be used to terminate the optimization process. In our proposed optimizer, a method based on tracking the relative change of the variational objective $\mathcal{L}(\lambda)$, to determine the stopping criterion is implemented.

This stopping criteria is similar to the Automatic Variational Inference in Stan algorithm [8]. For a (optional) specified number $k \in \mathbb{Z}^+$, the variational objective and the relative change of variational objective is calculated. If one iteration index at which the calculation occurred is T, then the relative change for that step is calculated as:

$$\Delta\mathcal{L}_T = 100 \cdot \left|\frac{\mathcal{L}(\lambda_T) - \mathcal{L}(\lambda_{T-k})}{\mathcal{L}(\lambda_{T-k})}\right| \tag{13}$$

$$\mathcal{L}_T = \mathbb{E}_q\left[\log\left(p(y|z)\right) - \log\left(\frac{q_{\lambda_T}(z)}{p(z)}\right)\right], \tag{14}$$

and stored in a vector $\Delta\mathcal{L}_{vect} = [..., \Delta\mathcal{L}_{T-k}, \Delta\mathcal{L}_T]$. Then, the running mean and median of this vector is calculated and compared to a threshold and the algorithm is terminated when either of the criteria are satisfied.

Downside of this algorithm is that it can prematurely end the optimization algorithm. This is shown with a simulated example displayed in Fig. 1. The variational objective is Free Energy, which is calculated as the negative of ELBO. As we are using the relative change in the free energy, this algorithm will be referred as ΔFE. Since there are no guarantees to reduce the free energy in each step with a given step size, simulated scenario involves an increase in the free energy at first then it converges to a lower value after considerable amount of iterations. The convergence algorithm ΔFE calculates the relative change in the free energy and compares the running mean and median to the threshold, which is set as 3%. Then, the algorithm would terminate prematurely, where the termination point is shown with red dashed line in Fig. 1.

To solve this problem, we have defined a burn-in period where the algorithm would not look for convergence until a specified number of iterations are carried out or the initial free energy decreases until a certain amount. With the latter condition, the algorithm will not search for convergence before the free energy decreases compared to the initial value, thus skipping the points where the vanilla implementation would prematurely terminate the optimization process.

D. Evaluating Variational Inference Using Generalized Pareto Distribution

A diagnostic which can be used to assess the goodness of the fit of variational distribution $q_\lambda(z)$ is fitting a generalized pareto distribution to the largest importance ratios and looking at the shape parameter k of the fitted distribution [9]. The motivation to use such a diagnostic comes from importance sampling literature [10, Ch. 9].

We treat our variational approximation $q_\lambda(z)$ as if it were a proposal distribution in importance sampling. When the proposal distribution $q_\lambda(z)$ is a poor approximation to the target distribution $p(z|y)$, the distribution of importance ratios can have a heavy right tail [11]. Thus, checking if distribution of importance ratios having a heavy tailed would indicate the accuracy of our variational approximation.

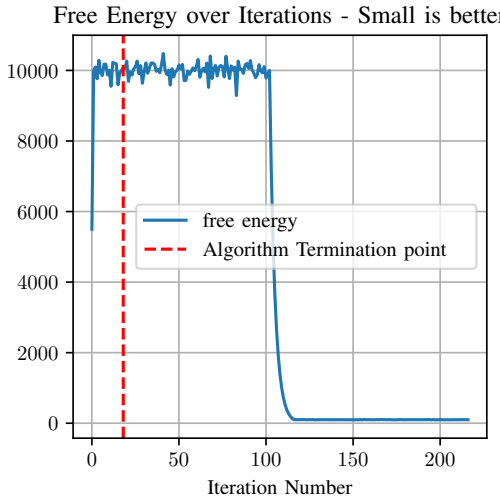

Fig. 1: A simulated example where ΔFE algorithm would result in premature termination. The threshold is set as 3% and the dashed line shows the termination point. Premature termination prevents the algorithm to converge to a lower free energy than the initial free energy.

A distribution used to model tails of another distribution is generalized Pareto distribution. A generalized Pareto distribution with a shape parameter k has finite moments up to the order $1/k$. If the fitted importance ratios have more than 2 finite moments, the convergence rate of the estimator improves [11], then if we have $k < 0.5$, it can be concluded that the variational distribution is close enough to the true posterior. Empirical studies show the number of samples you would need to have reliable estimators increases drastically after $k > 0.7$ [11]. Thus, for the fitted values of $0.5 < k < 0.7$, the variational distribution can still be practically useful.

This diagnostic is used as follows in our optimizer: After updating $q_\lambda(z)$ after a certain number of iterations, which can be fixed or determined by a stopping criterion, S samples from variational distribution are obtained and importance ratios are calculated as:

$$r_s(z) = \frac{p(y|z_s)p(z_s)}{q_\lambda(z_s)} \tag{15}$$

where $z_s \sim q_\lambda(z_s), i = 1, ..., S$. Then, a generalized Pareto distribution is fitted to M largest importance ratios, where M is a function of S and fitted shape parameter k is reported. If $k > 0.7$, the user is warned that the variational inference may not have converged. For the negative values of k, it is predicted that the importance ratios are bounded from above. For detailed explanation of how to fit the generalized Pareto distribution, we refer the interested reader to [12], and for more details about the convergence properties of the generalized Pareto distribution, we refer the interested reader to [11] and [9].

E. Overall Algorithm

We have addressed the practical considerations of the vanilla implementation of the CVI algorithm mentioned in Sec. II-C, with the methods given in Sec. III. Using these methods, we propose our adaptive optimizer in Algorithm 1 which finds the appropriate step size using an adaptive step size algorithm and tracks the relative change of the variational objective to terminate the algorithm. Finally, the accuracy of the approximation is diagnosed by fitting a generalized Pareto distribution to the largest importance ratios.

Algorithm 1 Adaptive Optimizer for Constrained CVI using Relative Change of Variational Objective

Define: Number of iterations of burn-in period: τ
Define: Mean threshold ϵ_1
Define: Median threshold ϵ_2
Define: Window size to evaluate the variational objective W
Require: τ,ϵ_1,ϵ_2,W
$check = false$
$\Delta\mathcal{L}_{vect} = []$
for t=1,...,T_{max} **do**
  if t = 1 **then**
    Compute $F\_thr = \mathcal{L}(\lambda_0)$
  end if
  Compute $\hat{g} = \nabla_m \mathcal{L}$
  Compute β via (12)
  Compute λ_t via (10)
  $\lambda \leftarrow \lambda_t$
  if t $\leq \tau$ **then**
    $continue$ ▷ First additional heuristic
  else if t mod W = 0 **then**
    Compute $\mathcal{L}(\lambda_t)$
    if $check = false$ and $\mathcal{L}(\lambda_t) \geq F\_thr$ **then**
      $check = true$ ▷ Second additional heuristic
    end if
    if $check = true$ **then**
      Compute $\Delta\mathcal{L}_t$ via Eq 13
      Append $\Delta\mathcal{L}_{vect} = [\Delta\mathcal{L}_{vect}, \Delta\mathcal{L}_t]$
      Compute mean m_1 and median m_2 of $\Delta\mathcal{L}_{vect}$
    end if
    if $m1 \leq \epsilon_1$ or $m2 \leq \epsilon_2$ **then**
      $break$
    end if
  end if
end for
Sample $z_s, s = 1, ..., S$ from $q_\lambda(z)$
Compute importance ratios $r_s, s = 1, ..., S$ via Eq 15
Fit generalized Pareto distribution to M largest importance ratios and return shape parameter estimate $\hat{k}$
if $\hat{k} > 0.7$ **then**
  Warn user that variational inference may not have converged.
end if
return λ

IV. Experiments

In this section, the experiment results on simulated examples which investigate how the choice of step size and the number of iterations affect the convergence of the variational distribution are shown. Motivation of using simulated examples is to observe the behavior of the current algorithms with arbitrary nonlinearities/functions to test their robustness.

A. First Experiment:Nonlinear Measurement Model

In the first experiment, the effect of the step size parameter and the number of iterations is studied. The prior distribution of the latent variable z is the Gaussian distribution. Observations y are also Gaussian distributed with known precision γ and the mean parameter is a non-linear function $g(\cdot)$ of latent variables z. The model is given as:

$$p(y \mid z) = \mathcal{N}\left(y \mid g(z), \gamma^{-1}\right) \tag{16a}$$

$$p(z) = \mathcal{N}\left(z \mid \mu_p, S_p^{-1}\right) \tag{16b}$$

The Gaussian prior and the measurement precision are set as:

$$\mu_p = 0, S_p^{-1} = 0.01, \gamma^{-1} = 0.01\,,$$

respectively.

The nonlinear expression $g(\cdot)$ is given as:

$$g(z) = -z^3 \cdot \exp\left(-0.005 \cdot |z|\right) \tag{17}$$

A measurement $\hat{y} = g(120) + \epsilon, \epsilon \sim \mathcal{N}(0,1)$ is observed and a variational approximation $q_\lambda(z)$, which is a Gaussian distribution, of the true posterior $p(z|\hat{y})$ is calculated. The selected variational objective to find $q_\lambda(z)$ is the free energy, defined as the negative of the ELBO. Non-linearity $g(z)$ is selected such that $g(120)$ evaluates to a large number, which also tests numerical stability of the algorithms. For the given nonlinearity $g(\cdot)$, the likelihood $p(\hat{y}|z)$ has two local optima, around $z \approx 120$ and $z \approx 1716$. Having a weak prior $p(z)$, the local variational approximation should converge to a Gaussian distribution with mean around either of the local optima. The posterior with mean value of 120 is the global optimum, since it is closer to the prior, which has lower free energy than the posterior with mean value of 1716, but the noisy estimate of the expectation of the log-likelihood term makes it impossible to distinguish the global optimum.

The parameters of variational approximation $q_\lambda(z)$ are optimized using both the CVI update rule with gradient descent and Adam optimizers and the iBLR update rule. All optimization schemes are tested with various number of iterations and step size hyper-parameters. Step size parameters are set as $ss_i = 10^{-i}, i = 0, 1, ..., 10$, number of iterations are set as $itr_j = 10^j, j = 1, 2, ..., 6$ to cover a variety of hyperparameter combinations. Then, for each point (ss_i, itr_j) in the hyper-parameter space for all three update rules, we perform the experiment with the same hyper-parameter configuration 10 times, and report the median value of 10 experiments.

Fig. 2 and Fig. 3 show the estimated posterior mean parameter of the variational distribution $q_\lambda(z)$ using the iBLR algorithm and the CVI algorithm with Descent optimizer, respectively. Only the results that are in the vicinity of the true posterior mean are shown in the figures, as failed cases have arbitrarily large values and cannot be plotted on the same graph. The results using CVI algorithm with Adam optimizer are not shown since none of the 66 different configurations of hyper-parameters yield a close approximation to the true posterior mean.

As seen in Fig. 2 and Fig. 3, both iBLR and CVI algorithm with Descent converge only for the cases where the step size parameter is less than a certain threshold, which is 10^{-6} for the iBLR case and 10^{-7} for the CVI case. Moreover, if the user selects smaller step sizes, then the optimal number of iterations varies.

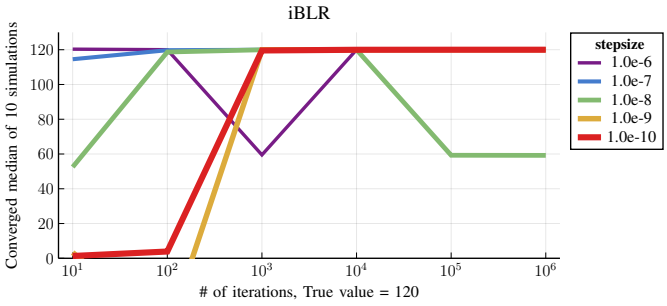

Fig. 2: Hyper-parameter sweep results using iBLR algorithm. It is observed that using stepsizes larger than 10^{-6} yield in algorithm to fail, whereas a certain number of iterations are necessary if one uses very small step sizes, such as 10^{-10}.

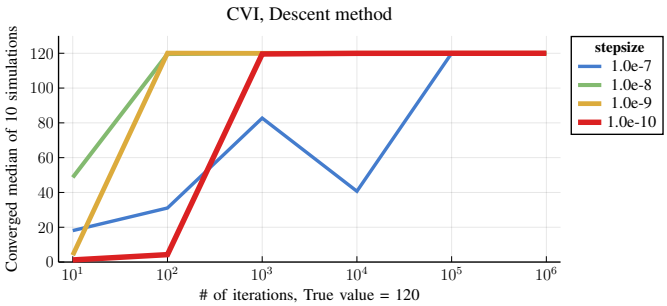

Fig. 3: Hyper-parameter sweep results using CVI algorithm with Descent optimizer. It is observed that using step sizes larger than 10^{-7} yield in algorithm to fail, whereas a certain number of iterations are necessary if one uses very small step sizes, such as 10^{-10}.

Fig. 4 shows the comparison of 3 optimization schemes with a fixed step size of 10^{-7}. It is observed that ADAM needs too many samples for this problem to converge, which is more than a million samples, whereas you would need much less samples for the iBLR case and at least 100000 samples for the CVI case to converge.

Fig. 2, Fig. 3 and Fig. 4 show how hyper-parameters affect the inference performance. The optimal parameters varies with every design choice, and finding suitable parameters requires

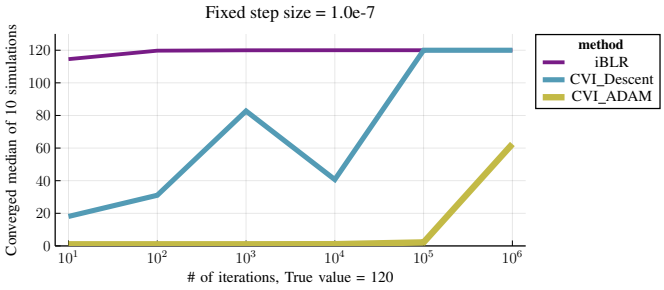

Fig. 4: Comparison of three optimization schemes. It is observed that the optimal number of iterations also changes greatly depending on the optimization algorithm chosen.

tedious work. Our proposed optimization scheme relieves the user of choosing step sizes and the number of iterations. We used our proposed optimizer given in Algorithm 1 for the given problem without any specifications on the initial step size or how many iterations to perform. The step size is selected to be determined by the heuristic based on the inexact line search method in Sec III-B1 to see if the optimizer can propose an appropriate step size for the problem. Initial step size is determined as $7.62 \cdot 10^{-6}$, which is an appropriate step size according to the results of Fig 2, and the algorithm terminated itself after 90000 iterations, converging to the value 120, which is the mean parameter value minimizing the current variational objective.

B. Second Experiment: Variational Message Passing

In the second example, variational message passing(VMP) example introduced in [13] is studied. The model in [13] can be formed using the same factorization in Eq. 16 with two differences. First, the non-linear function $g(\cdot)$ is changed to identity mapping, i.e., $g(z) = z$ and we put a Gamma prior on the measurement precision γ with the shape parameter a and the rate parameter β. VMP example in [13] approximates the posterior $p(x, \gamma \mid y)$ with a variational approximation $q(x, \gamma)$. If we also assume mean-field factorization $q(x, \gamma) = q_x(x)q_\gamma(\gamma)$, the model is conditionally conjugate and VMP algorithm updates the posterior parameters analytically. 5 observations are generated as:

$$y_n = 15 + \epsilon, \epsilon \sim \mathcal{N}(0, 1), n = 1, \ldots, 5 \tag{18}$$

Using VMP on the given problem setting resulted in the posterior mean of z as $\mu_z = 14.938$. We will take this result as the ground truth and test our gradient-based algorithm's performance.

In the first experiment in Sec. IV-A, we have used our heuristic line-search-based approach to find an appropriate step size. In this example, we let our optimizer decide the step size using adaptive step size algorithm mentioned in Section III-B2.

500 Monte Carlo simulations were performed and we calculated the mean and variance of the estimate as $\hat{\mu}_z = 14.90$ and $\sigma^2_{\hat{\mu}_z} = 0.40$, respectively. As expected, the mean is in the vicinity of the ground truth with a small variance value.

We also ran the algorithm using a fixed step size of 0.5 and 10^{-6} and we observe that even though we only needed 20 iterations using a step size of 0.5, we needed at least 10^6 iterations for the step size of 10^{-6}. Even if the VMP model is simple, it still shows how important the hyper-parameters are for performance of the algorithm.

V. Conclusion

We illustrated the practical considerations of implementing a natural gradient based variational inference optimization and what can be done in order to automatize the hyper-parameter tuning process, with their limitations. We proposed an automated optimizer for conjugate computation variational inference, which determines the initial step size, adaptation of step size over the iterations and when to terminate the inference procedure, along with diagnosing the accuracy of the posterior approximation to the true posterior. Its working principle can be improved by careful design of heuristics and implementing more robust solutions from the literature. This paper paves the way for a novel approach of gradient based variational inference algorithms which has its own robust convergence diagnostics and adaptive to the different types of non-conjugate terms in the generative model.

References

[1] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, vol. 112, pp. 859–877, Apr. 2017. arXiv: 1601.00670.

[2] M. E. Khan and W. Lin, "Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models," *arXiv:1703.04265 [cs]*, Mar. 2017.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

[4] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning*, vol. 37, pp. 183–233, Jan. 1999.

[5] M. Wainwright and M. Jordan, "Graphical Models, Exponential Families, and Variational Inference," *Foundations and Trends in Machine Learning*, vol. 1, pp. 1–305, Jan. 2008.

[6] W. Lin, M. Schmidt, and M. E. Khan, "Handling the Positive-Definite Constraint in the Bayesian Learning Rule," *arXiv:2002.10060 [cs, stat]*, Oct. 2020.

[7] R. Ranganath, C. Wang, B. David, and E. Xing, "An Adaptive Learning Rate for Stochastic Variational Inference," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 298–306, PMLR, May 2013. ISSN: 1938-7228.

[8] A. Kucukelbir, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic Variational Inference in Stan," *arXiv:1506.03431 [stat]*, June 2015.

[9] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman, "Yes, but Did It Work?: Evaluating Variational Inference," *arXiv:1802.02538 [stat]*, July 2018.

[10] O. Art B., *Monte Carlo theory, methods and examples*. 2013.

[11] A. Vehtari, D. Simpson, A. Gelman, Y. Yao, and J. Gabry, "Pareto Smoothed Importance Sampling," *arXiv:1507.02646 [stat]*, Feb. 2021.

[12] J. Zhang and M. A. Stephens, "A New and Efficient Estimation Method for the Generalized Pareto Distribution," *Technometrics*, vol. 51, no. 3, pp. 316–325, 2009. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality].

[13] J. Winn and C. M. Bishop, "Variational Message Passing," *Journal of Machine Learning Research*, vol. 6, no. 23, pp. 661–694, 2005.