Automatic Differentiation Variational Inference: A Report

Anand S Hegde: 200020007
Harrithha B: 200010018
Shashank P: 200010048

April 9, 2023

This report contains an overview and analysis of the paper titled **Automatic Differentiation Variational Inference** [3]. The link to the paper is given here.

1 Problem Statement

Probabilistic modelling involves fitting a model to data, and optimizing it to get the best possible fit repeatedly. However, in the case of a huge amount of data, fitting complex models would be both analytically and computationally challenging, which presents a bottleneck. To efficiently deal with the above problem, **Automatic Differentiation Variational Inference** was developed, which requires only a dataset and probabilistic model as input. The **ADVI** algorithm automatically derives an efficient variational inference algorithm, optimizing the model in order to get the best possible fit.

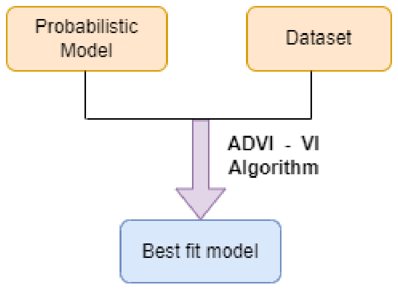


Figure 1: Overview

2 Methodology

The paper proposes scalable variational inference algorithms for a fundamental class of probability models, leveraging recent advancements in variational inference and latent variable space transformations.

The root problem involves defining a joint probabilistic model for observed data x and latent variables θ as given by $p(x, \theta)$ and computing the posterior distribution $p(\theta|x)$ of the latent variables given the observations. This posterior distribution provides insights into underlying data patterns and supports predictions but can be analytically infeasible to compute for complex models.

Variational inference uses a family of parametrized distributions $q(\theta) \in \mathcal{Q}$ and finds a member that minimizes the Kullback-Leibler (KL) [4] divergence to the exact posterior.

Traditionally the following steps had to be followed to obtain the member that minimizes KL Divergence.

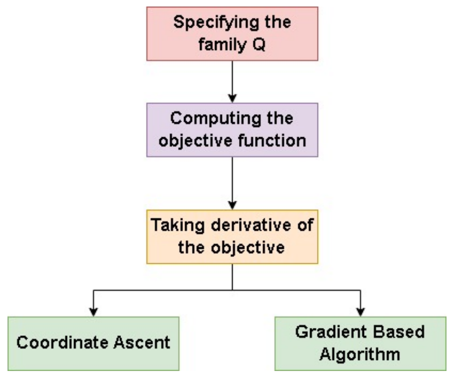

Specifying the family Q
Computing the objective function
Taking derivative of the objective
Coordinate Ascent
Gradient Based Algorithm

Figure 2: Traditional Variational Inference

ADVI solves this problem by taking in the model and generating the corresponding optimization algorithm. It first transforms the objective function into a common space and solves the optimization in this space, which in turn is equivalent to solving the optimization in the original space. The exact steps followed are given below:

1. It transforms the given $p(x, \theta)$ into an unconstrained space say $p(x, \zeta))$. We then define KL divergence $KL(q(\zeta)||p(\zeta|x)$. With this, the latent variables are defined in the same space.

2. **ADVI** recasts the gradient of the variational objective function as an expectation over q. Expressing the gradient as an expectation gives us an opportunity to use Monte Carlo Integration.

3. **ADVI** further applies a transformation to convert the expectation to be over a standard normal distribution. The second transformation allows us to compute Monte Carlo approximations efficiently.

4. It also uses noisy gradients and adaptively tuned step sizes using **RMS-prop** [2]

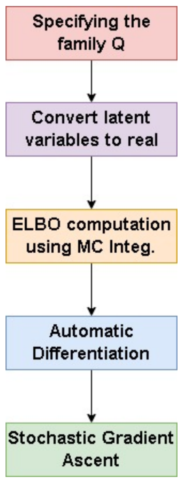

Specifying the family Q
Convert latent variables to real
ELBO computation using MC Integ.
Automatic Differentiation
Stochastic Gradient Ascent

Figure 3: Automatic Differentiation

Consider a dataset $x_{1:n}$ with N observations. Each point is a value produced by a random variable. The goal is to compute an approximation of the posterior density $p(\theta|x)$. **ADVI** approximates the posterior differentiable probability models. The models should have continuous latent variables, and $\nabla_\theta log(p(x, \theta))$ should be continuous. Also, the gradient has to be valid within the support:

$$supp(p(\theta)) = \{\theta|\theta \in \mathbb{R}^K, p(\theta) > 0\} \subseteq \mathbb{R}^K$$

Here K is the dimension of latent variable space. For now, let us take an example of an unknown **Poisson likelihood** $p(x|\theta)$. Assume a **Weibull prior** for θ, $p(\theta)$. The resulting joint $p(x, \theta)$ distribution is non-conjugate, i.e., the posterior is not from the same family as prior. The support of **Weibull** distribution is $supp(p(\theta)) = \mathbb{R}_{>0}$.

The derivation of **ELBO** from variational inference is given in the section 4 The objective function **ELBO** to be maximized wrt to a new family of distributions $q(\theta; \phi)$ parametrized by $\phi \in \Phi$ is shown below.

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\theta)}[log\, p(x,\, \theta)] - \mathbb{E}_{q(\theta)}[log\, q(\theta\,;\, \phi)] \qquad \text{(Eq. 1)}$$

Note that the support of q should lie within p [1]. Solving (Eq. 1) is difficult.

1. Let's begin by transforming the support of the latent variables θ such that $\zeta = T(\theta)$ and they live in the real coordinate space $\mathbb{R}^K$.

$$T : supp(p(\theta)) \rightarrow \mathbb{R}^K$$

[1]If $supp(q) \not\subseteq supp(p)$, we have $KL(q||p) = \mathbb{E}_q[logq] - \mathbb{E}_q[logp] = \infty$

The transformed joint probability looks like:

$$p(x,\zeta) = p(x,\ T^{-1}(\zeta))\ \ |detJ_{T^{-1}}(\zeta)|$$

Taking the previous example of **Poisson** and **Weibull** and taking transformation $\zeta = log(\theta)$, the joint is transformed to:

$$p(x,\zeta) = Poisson(x|e^{\zeta})\ \ Weibull(e^{\zeta}|1.5,1)\ \ e^{\zeta}$$

2. The variational distribution $q(\zeta;\phi)$ can be approximated in two different ways:

   (a) **Mean Field Gaussian**: Here we disregard covariance between ζ_i's, and maximize wrt the vector $\phi = \left(\mu_1,\cdots,\mu_K,\sigma_1^2,\cdots,\sigma_K^2\right)$.

   $$q(\boldsymbol{\zeta};\phi) = \text{Normal}\left(\zeta \mid \mu, \text{diag}\left(\sigma^2\right)\right) = \prod_{k=1}^{K} \text{Normal}\left(\zeta_k \mid \mu_k, \sigma_k^2\right)$$

   Here the constraint can be removed by taking $\omega = log(\sigma)$

   (b) **Full Rank Gaussian**: Here we do consider the covariance between ζ_i's and maximize wrt the vector $\phi = (\mu,\Sigma)$

   $$q(\zeta;\phi) = \text{Normal}(\boldsymbol{\zeta} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

   Here the constraint can be removed by applying (3) on Σ. We can now freely maximize this in real unconstrained space.

3. Now substitute the above transformation in the original objective function **ELBO**, we obtain:

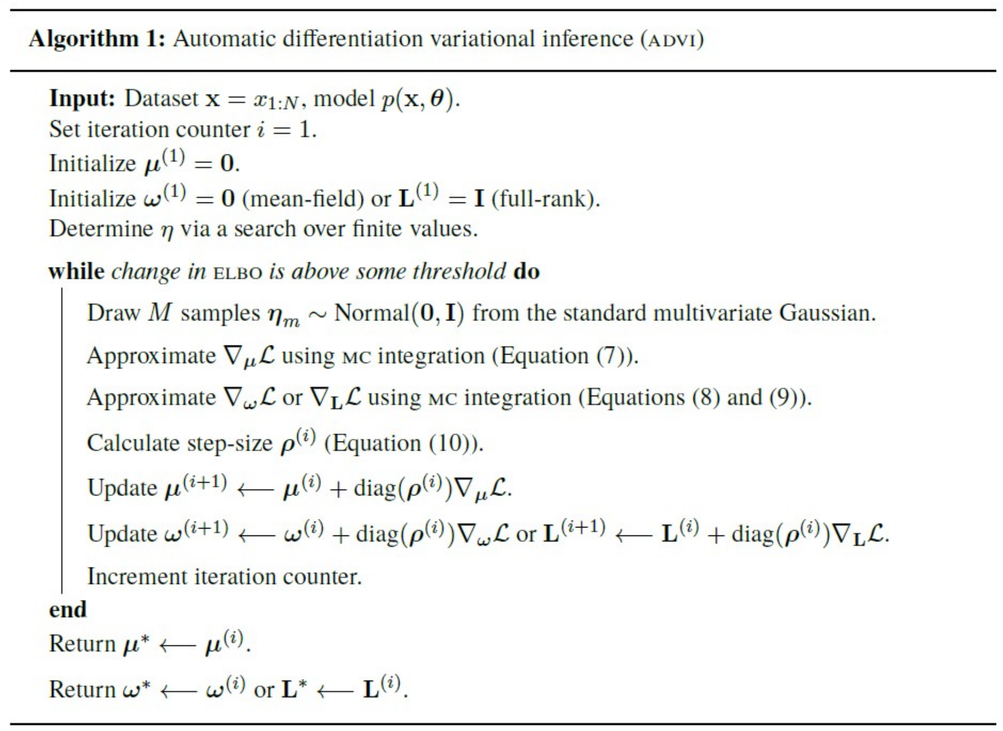
   $$\mathcal{L}(\phi) = \mathbb{E}_{q(\zeta;\phi)}\left[\log p\left(\mathbf{x}, T^{-1}(\boldsymbol{\zeta})\right) + \log\left|\det J_{T^{-1}}(\boldsymbol{\zeta})\right|\right] + \mathbb{H}[q(\boldsymbol{\zeta};\phi)]$$

   The **ELBO** is a function of variational parameter ϕ and the entropy $\mathbb{H}$ of q.

4. To apply gradient descent we need to push the gradient inside the expectation in order to apply Monte Carlo Integration. We employ elliptic standardization to transform q to standard normal. The transformations are shown below:

   $$q(\boldsymbol{\eta}) = \text{Normal}(\boldsymbol{\eta} \mid \mathbf{0},\mathbf{I}) = \prod_{k=1}^{K} \text{Normal}\left(\eta_k \mid 0,1\right)$$

   $$\phi^* = \arg\max_{\phi}\ \mathbb{E}_{\text{N}(\boldsymbol{\eta};\mathbf{0},\mathbf{I})}\left[\log p\left(\mathbf{x}, T^{-1}\left(S_{\phi}^{-1}(\boldsymbol{\eta})\right)\right) + \log\left|\det J_{T^{-1}}\left(S_{\phi}^{-1}(\boldsymbol{\eta})\right)\right|\right] + \mathbb{H}[q(\boldsymbol{\zeta};\boldsymbol{\phi})]$$



```
Algorithm 1: Automatic differentiation variational inference (ADVI)

Input: Dataset $\mathbf{x} = x_{1:N}$, model $p(\mathbf{x}, \boldsymbol{\theta})$.
Set iteration counter $i = 1$.
Initialize $\boldsymbol{\mu}^{(1)} = \mathbf{0}$.
Initialize $\boldsymbol{\omega}^{(1)} = \mathbf{0}$ (mean-field) or $\mathbf{L}^{(1)} = \mathbf{I}$ (full-rank).
Determine $\eta$ via a search over finite values.
while change in ELBO is above some threshold do
    Draw $M$ samples $\boldsymbol{\eta}_m \sim \text{Normal}(\mathbf{0}, \mathbf{I})$ from the standard multivariate Gaussian.
    Approximate $\nabla_{\boldsymbol{\mu}}\mathcal{L}$ using MC integration (Equation (7)).
    Approximate $\nabla_{\boldsymbol{\omega}}\mathcal{L}$ or $\nabla_{\mathbf{L}}\mathcal{L}$ using MC integration (Equations (8) and (9)).
    Calculate step-size $\boldsymbol{\rho}^{(i)}$ (Equation (10)).
    Update $\boldsymbol{\mu}^{(i+1)} \longleftarrow \boldsymbol{\mu}^{(i)} + \text{diag}(\boldsymbol{\rho}^{(i)})\nabla_{\boldsymbol{\mu}}\mathcal{L}$.
    Update $\boldsymbol{\omega}^{(i+1)} \longleftarrow \boldsymbol{\omega}^{(i)} + \text{diag}(\boldsymbol{\rho}^{(i)})\nabla_{\boldsymbol{\omega}}\mathcal{L}$ or $\mathbf{L}^{(i+1)} \longleftarrow \mathbf{L}^{(i)} + \text{diag}(\boldsymbol{\rho}^{(i)})\nabla_{\mathbf{L}}\mathcal{L}$.
    Increment iteration counter.
end
Return $\boldsymbol{\mu}^* \longleftarrow \boldsymbol{\mu}^{(i)}$.
Return $\boldsymbol{\omega}^* \longleftarrow \boldsymbol{\omega}^{(i)}$ or $\mathbf{L}^* \longleftarrow \mathbf{L}^{(i)}$.
```

Figure 4: ADVI Algorithm

5. On computing the gradient for the **Full-Rank** approximation we obtain the following final gradient, here L is from Choleskey decomposition of Σ

$$\nabla_{\mathbf{L}}\mathcal{L} = \mathbb{E}_{\mathrm{N}(\eta)}\left[\left(\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\zeta}} T^{-1}(\boldsymbol{\zeta}) + \nabla_{\boldsymbol{\zeta}} \log |\det J_{T^{-1}}(\boldsymbol{\zeta})|\right) \boldsymbol{\eta}^{\top}\right] + \left(\mathbf{L}^{-1}\right)^{\top} \tag{1}$$

Here the individual gradients inside the expectation are computed using **Automatic Differentiation** and properties of differentiation.

The above algorithm uses **RMS-Prop** to dynamically obtain different step size during Gradient Descent. The time complexity comes out to be $O(N\ M\ K)$ where N is the number of data points, M is number of samples for Monte-Carlo integration (in general $M = 1 - 10$ due to stochastic descent) and K is the dimention of variational parameter ϕ.

3 Experimental Results

In order to study the accuracy, we find the posterior for problems like Bayesian neural network and coin toss problems and compare the results of the analytical solution and the results from ADVI.

3.1 Coin toss

We sample 10 random samples from a Bernoulli distribution with the probability of the head being 0.8. We then used the beta prior and Bernoulli likelihood to compute the posterior. Here we know that the posterior will again be a beta distribution (conjugate prior).

Note that the support of the probability p here is constrained. So we have to transform the space to unconstrained while doing ADVI. But we can see that the model performs fairly well. We can see that the accuracy was not compromised in this case.

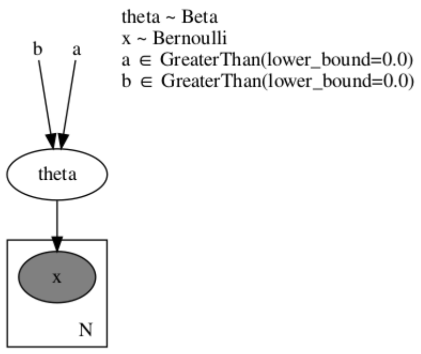

theta ~ Beta
x ~ Bernoulli
a ∈ GreaterThan(lower_bound=0.0)
b ∈ GreaterThan(lower_bound=0.0)
b
a
theta
x
N

Figure 5: Graphical representation of the Bayesian model

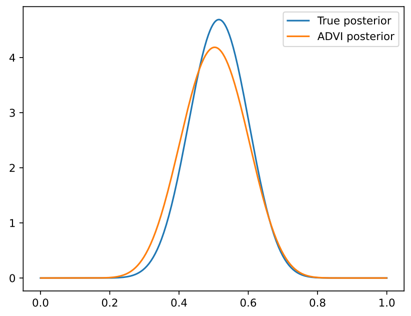


Figure 6: The comparison of the true posterior and posterior obtained by ADVI

3.2 Neural Network

We try to fit it on a toy neural network in the following way. We first generate some toy data using *sin* function and adding some noise. Then we create a neural network with 2 dense layers. We add a *relu* activation block after each of the dense layers.

We randomly initialize the weights of the neural network. We keep our likelihood function as a Student-T distribution. We want the posterior over the mean and the variance of the parameters.

We use a standard normal as prior for the model. Now, we use ADVI to find the posterior. After finding the posterior, we sample the parameters of the neural network and predict the values for the dataset using the sampled parameters.

The following is the plot that we got after plotting the neural network. In the plot, we can see that the variance is lesser where the data is available and higher where no data samples are available.

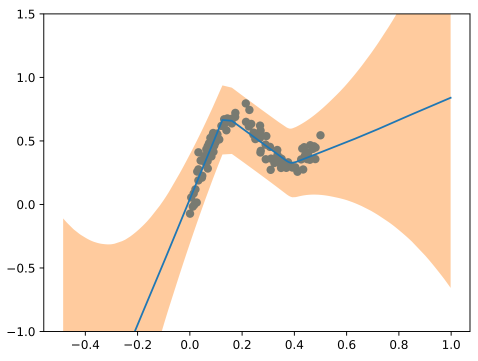


Figure 7: Neural network using the ADVI

4 Mathematics

1. The user inputs a model as a program to the **ADVI** algorithm, which helps in generation of a variational algorithm. The inference problem is first converted to a common space, where the **ADVI** is applied to solve the variational optimization problem.

2. **Variational inference (VI)** converts the approximate posterior inference into an optimization problem.

3. Let us consider a family of approximating densities of the latent variables $q(\theta; \phi)$, parameterized by a vector $\theta \in \Phi$. **VI** finds the parameters that

minimize the **KL-divergence** to the posterior, The original optimization problem is as follows:

$$\phi^* = \arg\min_{\phi \in \Phi} \ \mathbf{KL}(q(\theta;\phi)||p(\theta|x)). \tag{2}$$

4. However, in most cases, we are unable to find the posterior in the above equation $p(\theta|x)$. Thus, we transform it as follows to maximize the **Evidence Lower Bound (ELBO)**.

$$\begin{aligned}\mathbf{KL}\,(q(\theta)\,||\,p(\theta|x)) &= \int_\theta q(\theta)\,log\Big(\frac{q(\theta}{p(\theta|x)}\Big)\,d\theta \\ &= \mathbb{E}_{\theta\sim q}[log\,q(\theta)] - \mathbb{E}_{\theta\sim q}[log\,p(\theta|x)] \\ &= \mathbb{E}[log\,q(\theta)] - \mathbb{E}[log\,p(\theta,x)] + log(p(x)) \\ &= \mathbb{E}[log\,q(\theta)] - \mathbb{E}[log\,p(x|\theta)\,p(\theta)] + log(p(x)) \\ &= -\mathbf{ELBO}(q) + log(p(x))\end{aligned}$$

$$\mathbf{ELBO}(q) = \mathbb{E}_{q(\theta)}[log\,p(x,\,\theta)] - \mathbb{E}_{q(\theta)}[log\,q(\theta\,;\,\phi)]$$

5. The transformed optimization problem is ($\mathbf{ELBO}(q)$ denoted as $\mathcal{L}$):

$$\phi^* = \arg\max_{\phi\in\Phi} \ \mathcal{L}(\phi)$$

4.1 Course topics used here

- Note 1: While doing a full-rank Gaussian variational approximation,

$$q(\zeta;\phi) = \text{Normal}(\mathcal{C}|\mu,\Sigma)$$

To ensure that Σ always remains positive semidefinite, we re-parameterize the covariance matrix using a Cholesky factorization, such that:

$$\Sigma = LL^T \tag{3}$$

- In constrained settings, the gradient descent approach will not work. Thus, here we are converting the support of the posterior from a constrained space to an unconstrained space in order to use the gradient descent method for optimization as mentioned in section 2.

5 Modifications and Related Works

The following areas are open for further research:

- **Accuracy**: **ADVI**'s accuracy can be improved by using a cascade of simple transformations [1], a Gaussian process to learn the optimal transformation or developing rich approximations for non-differentiable latent variable models.

- **Practical Heuristics**: Initialization and step-size scaling can affect **ADVI**'s convergence. Initializing with a standard Gaussian could be improved by adapting to the model and dataset based on moment matching. The scale of the step-size sequence could be adaptively tuned without additional computation.

- **Optimization**: **ADVI** uses first-order automatic differentiation for stochastic gradient ascent. Using higher-order gradients may enable faster convergence but comes at a computational cost. Introducing some bias to reduce variance could also improve convergence speed [5]. Using line search or natural gradient approaches could also improve convergence robustness.

6 Conclusion

In a nutshell, **ADVI** is an initial attempt at an automating variational inference algorithm that performs effectively for a broad range of practical models on contemporary real-world datasets. The various steps in the **ADVI** algorithm include utilizing a compiler to automate the transformation of latent variables, selecting a variational family that yields low-variance gradient estimators for the variational objective, and devising an adaptive stochastic optimization step-size sequence that functions well both in theory and practice. By employing **ADVI**, scientists can construct, investigate, and modify intricate probabilistic models with significant data more conveniently.

References

[1] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. 2015.

[2] Geoffrey Hinton. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012.

[3] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of machine learning research*, 2017.

[4] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[5] Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. The generalized reparameterization gradient. 29, 2016.