An Exploration on Auto-Encoding Variational Bayes

Jingyiran Li

1 Introduction

Efficient approximate inference and learning with continuous latent variables that have intractable posterior distributions have become some of the most important aspects of variational inference. There exist numerous algorithms for approximate inference and the mean-field approximation is a common one. However, this particular approach requires analytical solutions to the evaluation of expectation with respect to the approximated posterior distribution which might be intractable. Consequently, Kingma and Welling proposed an algorithm, Auto-Encoding Variational Bayes (AEVB), that utilizes the Stochastic Gradient Variational Bayes (SGVB) estimator to overcome the aforementioned obstacle by performing efficient approximate posterior inference with continuous latent variables [2]. Kingma and Welling's paper demonstrated the AEVB algorithm on the Variational Auto-Encoder example where a Multi-Layer Perceptron (MLP) is used for the probabilistic encoder to jointly optimize the variational and true parameters. Unfortunately, the authors omitted model uncertainty quantification in the demonstration. As an extension, this report would attempt to address the aleatoric and epistemic uncertainty of the VAE in addition to an in-depth exploration of the AEVB algorithm.

2 Methodology

The SGVB is obtained from reparameterizing the variational lower bound. Prior to discussing the AEVB algorithm, the theoretical background behind the algorithm shall be explored. Let $X = \{x^{(i)}\}_{i=1}^{N}$ be N i.i.d. samples of either continuous or discrete x. Define z to be the latent variable generated by some random process where $z^{(i)} \sim p_{\theta^*}(z)$ and $x^{(i)} \sim p_{\theta^*}(x|z)$. The priors $p_{\theta^*}(z)$ and $p_{\theta^*}(x|z)$ originate from parametric families of $p_\theta(z)$ and $p_\theta(x|z)$, respectively. Furthermore, $p_\theta(z)$ and $p_\theta(x|z)$ are differentiable almost everywhere with respect to θ and z. The AEVB works even when $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$ and $p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$ are both intractable. In short, the AEVB efficiently approximate the following:

1. The maximum likelihood or maximum a-posteriori estimation for θ.
2. $p_\theta(z|x)$
3. $p_\theta(x)$

Let the recognition model $q_\phi(z|x)$ be the approximation to the true posterior $p_\theta(z|x)$. The marginal likelihood of the i^{th} data point can be expressed as the following:

$$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x)||p_\phi(z|x)) + \mathcal{L}(\theta, \phi; x^{(i)}) \tag{1}$$

Since D_{KL} is non-negative, $\mathcal{L}(.)$ is called the *variational lower bound* on the marginal likelihood of datapoint i:

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(\theta, \phi; x^{(i)}) = \mathbb{E}_{q_\phi(z|x)}(-\log q_\phi(z|x) + \log p_\phi(x|z))) \tag{2}$$

which could be rewritten as:

$$\mathcal{L}(\theta, \phi; x^{(i)}) = -D_{KL}(q_\phi(z|x^{(i)})||p_\phi(z|x)) + \mathbb{E}_{q_\phi(z|x^{(i)})}(\log p_\phi(x^{(i)}|z))) \tag{3}$$

The most straightforward approach is to evaluate the gradient of $\mathcal{L}(.)$ with respect to both ϕ and θ then perform stochastic gradient descent to optimize the objective function. Unfortunately, the gradient of $\mathcal{L}(.)$ with respect to ϕ yields an estimator with very high variance rendering the inference ineffective. In order to obtain an more effective gradient estimator of $\mathcal{L}(.)$, Kingma and Welling reparameterized it using the following setup:

Let $\tilde{z} \sim q_\phi(z|x)$ with a differentiabel transformation $g_\phi(\epsilon, x)$:

$$\tilde{z} = g_\phi(\epsilon, x) \text{ with } \epsilon \sim p(\epsilon) \tag{4}$$

where ϵ is a noise variable sampled from some appropriate distribution p. Using the Monte Carlo estimates of the expectations of some function $f(z)$ with respect to the variational distribution:

$$\mathbb{E}_{q_\phi(z|x^{(i)})} f(g_\phi(\epsilon, x^{(i)})) \approx \frac{1}{L} \sum_{l=1}^{L} f(g_\phi(\epsilon^{(l)}, x^{(i)})) \tag{5}$$

The variational lower bound can be reexpressed using the SGVB $\tilde{\mathcal{L}}^M(\theta, \phi; x^M)$ with a minibatch of size M:

$$\mathcal{L}(\theta, \phi; x^{(i)}) \approx \tilde{\mathcal{L}}^M(\theta, \phi; x^M) = \frac{N}{M} \sum_{i=1}^{M} \tilde{L}(\theta, \phi; x^{(i)}) \tag{6}$$

$$\tilde{L}(\theta, \phi; x^{(i)}) = \underbrace{\frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)})}_{\text{reconstruction loss}} - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) \tag{7}$$

where $z^{(z,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$, $X^M = \{x^{(i)}\}_{i=1}^M$ is a random sample of M datapoints from the full dataset X with N datapoints [2].

Subsequently, the SGVB is channeled into the AEVB algorithm to obtain the parameter estimates.

Algorithm 1: Auto-Encoding VB Algorithm

θ, $\phi \leftarrow$ Initialize parameters

while *Not converge* **do**

$X^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset

$\epsilon \leftarrow$ Random samples from noise distribution p(ϵ)

$g \leftarrow \nabla_{\theta,\phi}\tilde{\mathcal{L}}^M(\theta, \phi; X^M, \epsilon)$

$\theta, \phi \leftarrow$ Update parameters using gradients g

end

return *updated* θ, ϕ

2.1 Simulation

A neural network is used for $q_\phi(z|x)$ where ϕ and θ are jointly optimized using the AEVB algorithm. Let $p_\theta(z) = \mathcal{N}(z; 0, 1)$, $p_\theta(x|z)$ be a Gaussian distribution where the distribution parameters are computed from z with a multilayer perceptron (fully connected NN with 1 hidden layer). Assume the true posterior is approximately Gaussian with diagonal covariance so that $\log q_\phi(z|x) = \log \mathcal{N}(z; \mu, \sigma^2)$ where μ and σ are outputs of the encoding MLP. More specifically, $z^{(l)} \sim q_\phi(z|x)$ where $z^{(l)} = g_\phi(x, \epsilon^{(l)}) = \mu + \sigma \odot \epsilon^{(l)}$, $\epsilon^{(l)} \sim \mathcal{N}(0, 1)$ after reparameterization. The function g_ϕ is a neural network that could predict the variational parameters μ and σ. Now the SGVB estimator becomes:

$$\tilde{\mathcal{L}}(\theta, \phi; x) = \frac{1}{2}\sum_{j=1}^{n}(1 + \log(\sigma_j)^2 - (\mu_j)^2 - (\sigma_j)^2) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x|z^{(l)})$$

where $z^{(l)} = \mu + \sigma \odot \epsilon^{(l)}$ and $\epsilon^{(l)} \sim \mathcal{N}(0, 1)$.

z and x are simulated as follows:

Algorithm 2: Data Generation

$i \in n = \{1, \cdots, 150\}$

$w0 = 0.1$

$b0 = 4.5$

$x_i \in [-25, 55]$

$x_i \sim N(0, 1) * 80 - 20$

$s(x_i) = -3 * (0.25 + (\frac{x+20}{80})^2)$

$\epsilon_i \sim N(0, 1) * s(x)$

$z_i = w0 * (-x_i) * (1 + \cos(x_i) + b0) + \epsilon$

Standardize z_i

Sort data points in ascending order

return z, x

Both μ and σ are modeled with a fully connected neural network with 2 hidden layers with size 25. The size is chosen

to be 25 after a series of hyperparameter tuning that yielded the highest evidence lower bound value. Each layer is activated by a LeakyReLU activation function [3]:

$$\text{LeakyReLU}(x) = \max(0, x) + \text{negative slope} * \min(0, x) \tag{8}$$

The reason that a LeakyReLU is used as opposed to a ReLU is that it resolves the dying ReLU problem by including a small slope for negative values instead of setting them all to zero like ReLU. A 25% dropout is applied between the output and final hidden layer to avoid overfitting. After model fitting, samples were drawn from the approximate posterior distribution and aleatoric uncertainty is visualized using a 90% credible interval. Furthermore, the epistemic uncertainty of a Bayesian neural network can be addressed by modeling weights w of the neural networks as $w \sim N(0, 1)$ and $z \sim p(z|x, w)$ [1]. Now, $p(w|z, x)$ is intractable, but it can be estimated by conducting variational inference with a variational distribution $q_\theta(w)$. Firstly, a linear variational layer is constructed by intializing μ_w and σ_w, and weight is sampled as $w \sim N(\mu_w, diag(log(1 + e^{p_w})))$ with bias $b \sim N(\mu_b, diag(log(1 + e^{p_b})))$. A forward pass of layer $z = xw + b$ is conducted after sampling the neural network parameters. Similar to modeling the aleatoric uncertainty, a neural network is built with 2 hidden layers of size 25 and LeakyReLU activation in-between the layers. However, the KL divergence of each variational layer is accumulated. The model is evaluated using the mean, 5% and 95% quantiles of the posterior distribution approximated by taking 1000 samples per data point. Each sample can be thought of as a new neural network with different parameters.

3 Results

Approximating μ and σ involves a dependency on the data x which inevitably contains aleatoric uncertainty that could be visualized in figure 1. The blue line in figure 1 is the estimated μ obtained using the AEVB algorithm. The orange region indicates the 90% credible interval of the posterior distribution of z where σ used in the interval construction is estimated using the AEVB algorithm. The uncertainty quantification is coherent with theoretical results because the credible interval is narrower for more clustered regions and wider for less clustered regions.

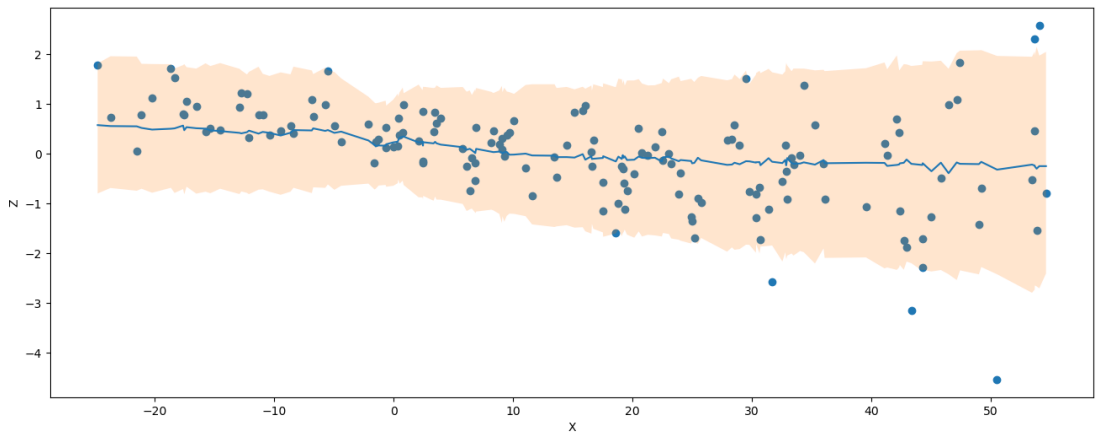

Figure 1: 90% Credible Interval of $p(z|x)$

The aforementioned aleatoric uncertainty is heteroscedastic because the behavior of σ is illustrated in figure 1 for this particular set of inputs, but σ changes for a different set of inputs after its estimation using AEVB.

Furthermore, the epistemic uncertainty of the Bayesian neural network is shown in figure 2. The blue line indicates $E(Z|X)$ and the orange region is the uncertainty of this true mean in the form of a 90% credible interval. It could be discerned that the AEVB on a neural network yielded fairly certain parameter estimations for this set of simulation based on the width of the interval.

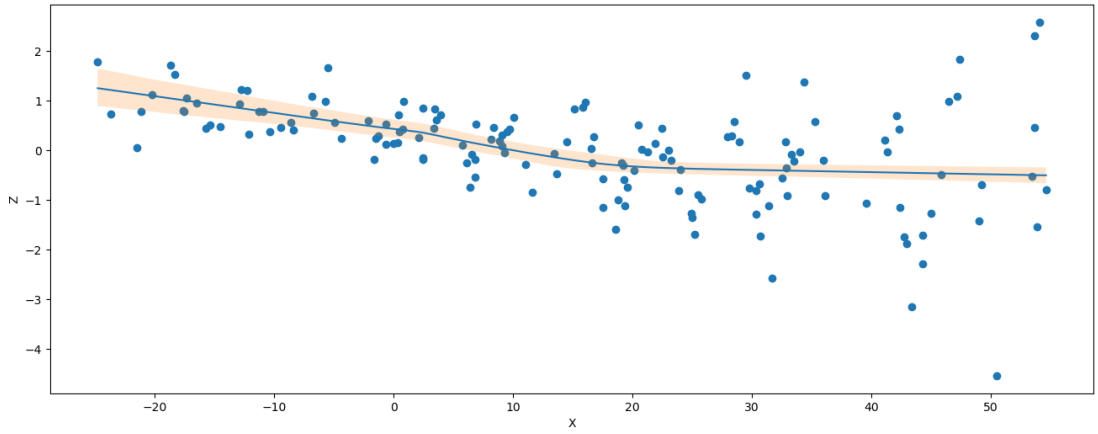

Figure 2: Uncertainty of $E(Z|X)$ (true mean)

4 Conclusion and Future Directions

The AEVB algorithm effectively solves the cases with intractable expectation and inefficient computation when conducting variational inference. Via introducing a reparameterized variational lower bound, the SGVB estimator could optimize over a recognition model such as a neural network to obtain approximate posterior inference. In the example of a variational auto-encoder where a fully connect neural network is used for the probabilistic encoder $q_\phi(z|x) \sim \mathcal{N}(z; \mu, \sigma^2)$, ϕ and θ was jointly optimized using a neural network with 2 hidden layers each with a size of 25, a LeakyReLU activation function between each layer, and a 25% dropout between the last hidden layer and the output layer. Using the simulated dataset, the AEVB algorithm was tested on the variation auto-encoder example. μ and σ were estimated and plotted. Furthermore, the aleatoric uncertainty was captured and visualized using the 90 % credible interval. Lastly, the epistemic uncertainty of the Bayesian neural network was evaluated using a similar neural network setup with an exception of using linear variational layers in order to provide insights into uncertainty in the model parameters. A limitation of the exploration is that it approximates the posterior $q_\phi(x|z)$ over data points that map to z with a fixed variance Gaussian. However, in real world application settings, common distributions such as natural images almost never have a mixture of Gaussian structure. Additionally, unless $q_\phi(z|x)$ is lossless, it will map multiple x to the same encoding z, resulting in a highly non-Gaussian posterior $q_\phi(x|z)$ [4]. Lastly, this exploration does not consider the implications of choosing different classes of perturbation (noise distribution). In

the future, it is worthwhile to address the aforementioned limitations by testing optimizing the algorithm on non-Gaussian distributions with varying perturbations. Furthermore, one could investigate ways to reduce the epistemic uncertainty in Bayesian neural network using variational inference.

5 References

[1] Alex Kendall and Yarin Gal. *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?* 2017. arXiv: 1703.04977 [cs.CV].

[2] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2014. arXiv: 1312.6114 [stat.ML].

[3] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Citeseer. 2013, p. 3.

[4] Shengjia Zhao, Jiaming Song, and Stefano Ermon. *Towards Deeper Understanding of Variational Autoencoding Models*. 2017. arXiv: 1702.08658 [cs.LG].