Notes On Variational Inference

Ludwig Winkler

November 26, 2017

Variational Bayesian Inference is a family of techniques for approximating intractable integrals arising in Bayesian inference and machine learning. They are typically used in complex statistical models consisting of observed variables as well as unknown parameters and latent variables, with various sorts of relationships among the three types of random variables.

1 PROBLEM SETTING

Let's assume that we have a set of observations x which we know where somehow produced by a process. We can call our set of observations $\mathbf{x}$ our data set.

But the process that generated the observations $\mathbf{x}$ also has a set of latent, random variables $\mathbf{z}$ which we cannot observe. An intuitive example would be a classification task where we have an observation $\mathbf{x}$ and a label $\mathbf{z}$. A standard classifier would construct a decision boundary from which we could obtain the probablity $p(\mathbf{z}|\mathbf{x})$. An analogy would be a neural network classifier with a final softmax layer which outputs a probability of a class, dependent on the input.

Variational Inference (VI) trains a generative model on the data which constructs a joint probability $p(\mathbf{x},\mathbf{z})$ from which we can conveniently infer much more information than just the class-conditional probability. By building the entire generative process $p(\mathbf{x},\mathbf{z})$ and not just the class-dependent probability $p(\mathbf{z}|\mathbf{x})$ we have the entire generative process at our disposal. Take for example studying for an exam: It is usually way better to actually

understand the entire topic intrinsically than just learning by heart a mapping from some question to some answer.

Furthermore we can assume that the true model $p(\mathbf{x}, \mathbf{z})$ has a set of parameters θ which parameterize the joint probability density function (PDF) and which we don't know. It is important to note that is helpful to assume that joint PDF $p(\mathbf{x}, \mathbf{z})$ consists of an arbitrary combination of parameterized PDFs. This combination of parameterized PDFs could be as simple as two normal distributions with independent parameters or a complicated combination of complex PDFs.

Variational Inference aims to find suitable parameters θ which explain the generation of the observations $\mathbf{x}$ with the latent $\mathbf{z}$. Using the analogy of images and labels, VI tries to find parameters which create great pictures for any label. Since we only receive the images, more generally $\mathbf{x}$, we want to model the generative process that produced them from labels, more generally $\mathbf{z}$, which we can't observe. Once we modeled the 'forward' process from $\mathbf{z}$ to $\mathbf{x}$ we can also go 'backward' and infer $\mathbf{z}$, the label, from $\mathbf{x}$, the image. This is possible because we are working with a generative model.

VI therefore aims to maximize the evidence of the observations $\mathbf{x}$ by finding the right parameter θ_{max}. If we were to find the right parameters θ_{max} we could generate valid observations $\mathbf{x}$ for any $\mathbf{z}$.

$$\theta_{\text{max}} = \text{argmax}_\theta \log p_\theta(\mathbf{x}) \tag{1.1}$$

Another nice thing to know, given some observations $\mathbf{x}$, for example images, would be to know the latent factors $\mathbf{z}$, for example the labels, that generated them. So we would like to know

$$p_{\theta_{\text{max}}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\theta_{\text{max}}}(\mathbf{x}, \mathbf{z})}{\int p_{\theta_{\text{max}}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}} = \frac{p_{\theta_{\text{max}}}(\mathbf{x}, \mathbf{z})}{p_{\theta_{\text{max}}}(\mathbf{x})} \tag{1.2}$$

The problem is that the denominator $\int p_{\theta_{\text{max}}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ is usually an intractable integral. It might work for simpler models but for models which try to infer something on high-dimensional data, calculating the integral becomes quickly very difficult. Due to the usually intractable denominator it is difficult to correctly estimate $p_{\theta_{\text{max}}}(\mathbf{z}|\mathbf{x})$.

Because we condition the distribution $p_{\theta_{\text{max}}}(\mathbf{z}|\mathbf{x})$ on $\mathbf{x}$ we have to calculate the evidence $\int p_{\theta_{\text{max}}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$. In a nutshell, the trick of VI is to try to estimate an 'unconditional' distribution $q_\Psi(\mathbf{z})$ which doesn't require an intractable integral like $\int p_{\theta_{\text{max}}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ does.

2 ELBO

The Kullback-Leibler-Divergence $\mathrm{KL}\big(q(\mathbf{x})||p(\mathbf{x})\big)$ measures the distance between two probability distributions $q(\mathbf{x})$ and $p(\mathbf{x})$. In variational inference it is used as a criterion with which we minimize the difference between $q_\Psi(\mathbf{z})$ and $p_\theta(\mathbf{z}|\mathbf{x})$.

$$\begin{aligned}
\min_\Psi \mathrm{KL}\big(q_\Psi(\mathbf{z}) \,||\, p_\theta(\mathbf{z}|\mathbf{x})\big) &= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] && (2.1)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{z}|\mathbf{x})\right] && (2.2)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{x})}\right] && (2.3)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{x},\mathbf{z})\right] + \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{x})\right] && (2.4)\\
&= \underbrace{\mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{x},\mathbf{z})\right]}_{-\mathrm{ELBO}[q_\Psi(\mathbf{z})]} + \log p_\theta(\mathbf{x}) && (2.5)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})}\right] + \log p_\theta(\mathbf{x}) && (2.6)\\
&= \mathrm{KL}\big(q_\Psi(\mathbf{z}) \,||\, p_\theta(\mathbf{x},\mathbf{z})\big) + \log p_\theta(\mathbf{x}) && (2.7)
\end{aligned}$$

We can intuitively see from (2.5) and (2.7) that we can reformulate the minimization problem as a minimization of the KL-Divergence between the variational distribution $q_\Psi(\mathbf{z})$ and $p_\theta(\mathbf{x},\mathbf{z})$. The term $\log p_\theta(\mathbf{x})$ is a constant with respect to our variational distribution $q_\Psi(\mathbf{z})$.

The name 'Evidence Lower Bound' (ELBO) is derived from a property of the Kullback-Leibler-Divergence, namely that $\mathrm{KL}(q(\mathbf{x}) \,||\, p(\mathbf{x})) \geq 0$ for any $q(\mathbf{x})$ and $p(\mathbf{x})$. For variational inference this implicates the following identity:

$$\begin{aligned}
\mathrm{KL}\big(q_\Psi(\mathbf{z}) \,||\, p_\theta(\mathbf{z}|\mathbf{x})\big) &\geq 0 && (2.8)\\
\mathrm{KL}\big(q_\Psi(\mathbf{z}) \,||\, p_\theta(\mathbf{x},\mathbf{z})\big) + \log p_\theta(\mathbf{x}) &\geq 0 && (2.9)\\
-\mathrm{ELBO} + \log p_\theta(\mathbf{x}) &\geq 0 && (2.10)\\
\log p_\theta(\mathbf{x}) &\geq \mathrm{ELBO} && (2.11)
\end{aligned}$$

In order to minimze our original objective in (2.1) we have to maximize the ELBO in (2.5). With the property of the KL-Divergence, namely $\mathrm{KL}\big(q(\mathbf{x}) \,||\, p(\mathbf{x})\big) \geq 0$, we can see that the ELBO is bounded from above by the log-probability of $p_\theta(\mathbf{x})$. So we can only maximize the ELBO up to the log-probability of $p_\theta(\mathbf{x})$.

In order to minimize our original objective in (2.1) we will try to approximate the joint distribution $p_\theta(\mathbf{x},\mathbf{z})$ with a simpler distribution $q_\Psi(\mathbf{z})$. The KL-Divergence is always ≥ 0 so we will be always left with the constant $\log p_\theta(\mathbf{x})$ which we will not be able to optimize. In a nutshell by approximating the complicated conditional distribution $p_\theta(\mathbf{z}|\mathbf{x})$ with a simpler $q_\Psi(\mathbf{z})$ we don't have to deal with the intractable integral but we also obtain a term in our optimization problem which we cannot reduce.

To gain more intuition about the ELBO we can go two steps back and look at the following:

$$\begin{aligned}
\text{ELBO}\left(q_\Psi(\mathbf{z})\right) &= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{x},\mathbf{z})\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] && (2.12)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})\, p_\theta(\mathbf{z})}{p_\theta(\mathbf{z})}\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] && (2.13)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] + \mathbb{E}_{q_\Psi(\mathbf{z})}\left[p_\theta(\mathbf{z})\right] - \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log q_\Psi(\mathbf{z})\right] && (2.14)\\
&= \mathbb{E}_{q_\Psi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \text{KL}\left(q_\Psi(\mathbf{z}) \,||\, p_\theta(\mathbf{z})\right) && (2.15)
\end{aligned}$$

Keeping in mind that we have to maximize the ELBO term in (2.5) can furthermore see in (2.15) that $q_\Psi(\mathbf{z})$ will be balanced between putting weight to the likelihood as well as the prior. The expected likelihood emphasizes $q_\Psi(\mathbf{z})$ putting its probability on configurations of $\mathbf{z}$ that explain the observed data $\mathbf{x}$. If too much emphasis is paid to the expected likelihood the KL-Divergence between the prior $p_\theta(\mathbf{z})$ and $q_\Psi(\mathbf{z})$ will grow. The ELBO term therefore is encouraged to find a balanced approximation of the prior $p_\theta(\mathbf{z})$ and the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. This is corroborated by the fact that we try to approximate the entire joint probability $p_\theta(\mathbf{x},\mathbf{z})$ with $q_\Psi(\mathbf{z})$.

3 Stochastic Gradient Optimization

Similarly to how neural networks are nowadays trained with stochastic gradient descent we can train variational inference algorithms with stochastic gradient otpimization [1]. Recall from (2.5) that in order to minimize our original objective we need to maximize the Evidence Lower Bound.

$$\nabla_\Psi \left[-\text{ELBO}\left(q_\Psi(\mathbf{z})\right)\right] = \nabla_\Psi \left[\text{KL}\left(q_\Psi(\mathbf{z}) \,||\, p_\theta(\mathbf{x},\mathbf{z})\right)\right] \tag{3.1}$$

$$= \nabla_\Psi \left[\int_{\mathbf{z}} q_\Psi(\mathbf{z}) \log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})}\right] \tag{3.2}$$

$$= \int_{\mathbf{z}} \nabla_\Psi \left[q_\Psi(\mathbf{z})\right] \log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})} + \int_{\mathbf{z}} q_\Psi(\mathbf{z}) \nabla_\Psi \left[\log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})}\right] \tag{3.3}$$

$$= \int_{\mathbf{z}} \nabla_\Psi \left[q_\Psi(\mathbf{z})\right] \log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})} + \int_{\mathbf{z}} q_\Psi(\mathbf{z}) \left(\nabla_\Psi \left[\log q_\Psi(\mathbf{z})\right] - \underbrace{\nabla_\Psi \left[p_\theta(\mathbf{x},\mathbf{z})\right]}_{=0}\right) \tag{3.4}$$

$$= \int_{\mathbf{z}} \nabla_\Psi \left[q_\Psi(\mathbf{z})\right] \log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})} + \underbrace{\int_{\mathbf{z}} q_\Psi(\mathbf{z}) \nabla_\Psi \left[\log q_\Psi(\mathbf{z})\right]}_{=0} \tag{3.5}$$

$$= \int_{\mathbf{z}} q_\Psi(\mathbf{z}) \nabla_\Psi \left[\log q_\Psi(\mathbf{z})\right] \log \frac{q_\Psi(\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{z})} \tag{3.6}$$

$$\approx \frac{1}{N} \sum_{\mathbf{z}_j}^{N} \nabla_\Psi \left[q_\Psi(\mathbf{z}_j)\right] \left(\log \frac{q_\Psi(\mathbf{z}_j)}{p_\theta(\mathbf{x},\mathbf{z}_j)} + K\right) \tag{3.7}$$

This gives us a training algorithm in which we don't have to use the entire set, but with which we can train batch by batch just like stochastic gradient descent in neural networks.

The identity $\nabla \log p(x) = \frac{\nabla p(x)}{p(x)}$ was used for

$$\int_{\mathbf{z}} q_\Psi(\mathbf{z}) \nabla_\Psi \left[\log q_\Psi(\mathbf{z})\right] = \int_{\mathbf{z}} \nabla_\Psi \left[q_\Psi(\mathbf{z})\right] \tag{3.8}$$

$$= \nabla_\Psi \left[\int_{\mathbf{z}} q_\Psi(\mathbf{z})\right] \tag{3.9}$$

$$= \nabla_\Psi [1] \tag{3.10}$$

$$= 0 \tag{3.11}$$

REFERENCES

[1] D. Wingate and T. Weber, "Automated variational inference in probabilistic programming," *arXiv preprint arXiv:1301.1299*, 2013.