Auto-Encoding Variational Bayes
Authors: Diederik P. Kingma & Max Welling

Sai Subramanyam Thoota

SPC Lab, Department of ECE
Indian Institute of Science

August 25, 2018

Table of contents

Problem

- Consider a unlabelled dataset $\mathbf{X} = \{\mathbf{x}\}_{i=1}^{N}$ consisting of N i.i.d. samples of some continuous or discrete random variable $\mathbf{x}$ with some unknown distribution
- Assume data are generated by a random process parameterized by $\boldsymbol{\theta}$
- A common approach in statistical inference is to consider a joint distribution involving an unobserved continuous r.v $\mathbf{z}$ (latent variable in a feature space $\mathcal{Z}$)
- Goal is to learn the posterior distribution of the latent variables given the dataset $\mathbf{X}$
- We impose a simple prior $p_\theta(\mathbf{z})$, and the likelihood function is $p_\theta(\mathbf{x}|\mathbf{z})$
- We propose a general algorithm that learns the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ and/or the marginal distribution $p_\theta(\mathbf{x})$
  - Learning the marginal distribution is computationally intractable, especially when the dataset is very large
- Approach:
  - Introduce a recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ as an approximation to the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$
  - Recognition model is not necessarily factorial and its parameters are not computed from closed form expectations
  - Learn the recognition model parameters ϕ jointly with the generative model parameters $\boldsymbol{\theta}$

- Inference using exact posterior is computationally intractable which necessitates variational approaches
- Variational Bayesian (VB) approach involves the optimization of an approximation to the intractable posterior
- Mean-field approach requires analytical solutions of expectations w.r.t. the approximate posterior, which are intractable in the general case
- A reparameterization of the variational lower bound yields a simple differentiable unbiased estimator of the lower bound
  - Stochastic Gradient Variational Bayes (SGVB) estimator
  - Optimized using standard stochastic gradient ascent techniques
- Auto-encoding VB (AEVB) algorithm is proposed for the case of an i.i.d. dataset and continuous latent variables
  - Inference and learning using the SGVB estimator to optimize a recognition model
  - Neural network used to learn the parameters of the approximate posterior, which leads to the variational auto-encoder (VAE)

The Variational Bound (Evidence Lower Bound)

- The marginal likelihood $\log p_\theta(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)})$ can be rewritten as:

$$\begin{aligned} \log p_\theta\left(\mathbf{x}^{(i)}\right) &= \mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right) + D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) || p_\theta(\mathbf{z}|\mathbf{x}^{(i)})\right) && (1) \\ &\geq \mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right) && (2) \end{aligned}$$

- $\mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right)$ is the variational lower bound on the marginal likelihood of the data point i since KL divergence is non-negative

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[-\log q_\phi\left(\mathbf{z}|\mathbf{x}\right) + \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}, \mathbf{z}\right)\right] && (3) \\ &= -D_{KL}\left(q_\phi\left(\mathbf{z}|\mathbf{x}^{(i)}\right) || p_{\boldsymbol{\theta}}\left(\mathbf{z}\right)\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}|\mathbf{z}\right)\right] && (4) \end{aligned}$$

- We want to differentiate and optimize the lower bound w.r.t. both the variational parameters ϕ and generative parameters $\boldsymbol{\theta}$

The SGVB estimator and AEVB algorithm

- Under mild conditions for a chosen approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$, we can reparameterize the r.v $\tilde{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ using a differentiable transformation $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$ of an auxiliary noise variable $\boldsymbol{\epsilon}$:

$$\tilde{\mathbf{z}} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \tag{5}$$

- Monte Carlo estimates of expectations of some function $f(\mathbf{z})$ can be computed as follows:

$$\begin{aligned}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[f(\mathbf{z})\right] &= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f\left(g_\phi\left(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}\right)\right)\right] \\ &\simeq \frac{1}{L}\sum_{l=1}^{L} f\left(g_\phi\left(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)}\right)\right) \end{aligned} \tag{6}$$

$$\text{where} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

- Applying this technique to the variational lower bound, we get the SGVB estimator $\tilde{\mathcal{L}}^A(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) \simeq \mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right)$ as follows:

$$\tilde{\mathcal{L}}^A\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right) = \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}\right) - \log q_\phi\left(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)}\right) \tag{7}$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_\phi\left(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)}\right) \quad \text{and} \quad \boldsymbol{\epsilon}^{l} \sim p(\boldsymbol{\epsilon})$$

- Second version of the SGVB estimator $\tilde{\mathcal{L}}^B(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) \simeq \mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right)$

$$\tilde{\mathcal{L}}^B(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = -D_{KL}\left(q_\phi\left(\mathbf{z}|\mathbf{x}^{(i)}\right)||p_{\boldsymbol{\theta}}\left(\mathbf{z}\right)\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}\right) \tag{8}$$

- The KL divergence term can be integrated analytically, and hence only the expected reconstruction error requires estimation by sampling (see (8))
- Given a dataset $\mathbf{X}$ with N datapoints, we can construct an estimator of the marginal likelihood lower bound of the full dataset based on minibatches:

$$\mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{X}\right) \simeq \tilde{\mathcal{L}}^M\left(\boldsymbol{\theta}, \phi; \mathbf{X}^M\right) = \frac{N}{M}\sum_{i=1}^{M}\tilde{\mathcal{L}}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right) \tag{9}$$

The Reparameterization Trick

- Goal: To generate samples of a random variable $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$
- Possible to express $\mathbf{z}$ as a deterministic variable $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$
  - ϵ is an auxiliary variable with independent marginal p_ϵ
  - $g_\epsilon(.)$ is some vector-valued function parameterized by ϕ
- Proof: Given the mapping $\mathbf{z} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$, we know that

$$
\begin{aligned}
q_\phi(\mathbf{z}|\mathbf{x}) \prod_i dz_i &= p(\epsilon) \prod_i d\epsilon_i \\
\implies \int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} &= \int p(\epsilon) f\left(g_\phi(\epsilon, \mathbf{x})\right) d\epsilon \\
&\simeq \frac{1}{L} \sum_{l=1}^{L} f\left(g_\phi(\mathbf{x}, \epsilon)^{(l)}\right) \qquad (10)
\end{aligned}
$$

- This trick is used to obtain a differentiable estimator of the variational lower bound
- Example:
  - $z \sim \mathcal{N}(\mu, \sigma^2)$
  - Reparameterization $z = \mu + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$

Variational Auto-Encoder

Encoder Neural Network:

- Probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{x})$
- Simple prior over the latent variables $\mathbf{z}$ chosen as multivariate Gaussian $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$
  - Note that the prior lacks parameters
- Let $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ be a multivariate Gaussian or Bernoulli whose parameters are computed from $\mathbf{z}$ using a fully connected MLP
- We choose the variational approximate posterior to be a multivariate Gaussian:

$$\log q_\phi\left(\mathbf{z}|\mathbf{x}^{(i)}\right) = \log \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I}\right) \tag{11}$$

  where the mean and s.d. of the approximate posterior are the outputs of the encoding MLP, i.e., $\boldsymbol{\mu}^{(i)} = \mathbf{W}_4\mathbf{h}_e + \mathbf{b}_4$, $\log \boldsymbol{\sigma}^{2(i)} = \mathbf{W}_5\mathbf{h}_e + \mathbf{b}_5$, and the hidden layer output $\mathbf{h}_e = \tanh(\mathbf{W}_3\mathbf{x}^{(i)} + \mathbf{b}_3)$
- The parameter $\phi = \{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$ are the weights and biases, which are found by passing $\mathbf{x}^{(i)}$ to the MLP

Decoder Neural Network:

- Probabilistic decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$
- Weights and biases found similar to the encoder NN

- Sample from the posterior $\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^(i))$ using $\mathbf{z}^{(i,l)} = g_\phi(\mathbf{x}^{(i)}, \epsilon^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ where $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- The resulting estimator of the evidence lower bound for this model is given by

$$\mathcal{L}\left(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}\right) \simeq \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\left(\sigma_j^{(i)}\right)^2\right) - \left(\boldsymbol{\mu}_j^{(i)}\right)^2 - \left(\sigma_j^{(i)}\right)^2\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}\right) \tag{12}$$

  where $\mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ and $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- The decoding term $p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}\right)$ is a Bernoulli or Gaussian MLP depending on the type of data we are modelling

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

THANK YOU!