Variational Auto-Encoders

Mauro Camara Escudero

School of Mathematics, University of Bristol

Table of contents

1. Background
2. AEVB Objective Function
3. ELBO Optimization
4. Marginal Likelihood Estimation
5. Variational Auto-Encoders
6. Relationship between EM and VAE

Background

Latent Variable Models (LVMs)

Notation: $\mathbf{x}$ observed, $\mathbf{z}$ latent, $\boldsymbol{\theta}$ parameter of interest.

Goals:

- Generative Modelling

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

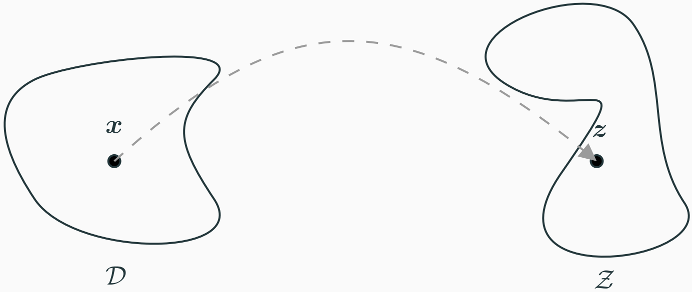
- Posterior Inference

$$p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$$

- Parameter Estimation

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \prod_{\mathbf{x} \in \mathcal{D}} p_{\boldsymbol{\theta}}(\mathbf{x})$$

Space Diagram



EM Algorithm for MLE

Initialize $\boldsymbol{\theta}^{(0)}$ and $t = 0$.

- Compute conditional distribution of latent given observations $\{p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z} \mid \mathbf{x}) \ : \ \mathbf{x} \in \mathcal{D}\}$
- Choose new parameter value $\boldsymbol{\theta}^{(t+1)}$ so that it maximises $\sum_{\mathbf{x} \in \mathcal{D}} \mathbb{E}_{p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right]$

Problem: Breaks if $p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z} \mid \mathbf{x})$ are intractable.

Mean-Field VI for Parameter Posterior

Define factorized **variational** distribution

$$\prod_{i=1}^{|\mathcal{Z}|} q_{\boldsymbol{\phi}_i}(\mathbf{z}_i) \approx \prod_{\mathbf{z} \in \mathcal{Z}} p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$$

Minimize KL-divergence for each data point / variational parameter

$$\min_{\boldsymbol{\phi}} \mathrm{KL}(q_{\boldsymbol{\phi}_i}(\mathbf{z}) \;\|\; p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x}))$$

Problem: Breaks when $\mathbb{E}_{q_{\boldsymbol{\phi}_i}}[\cdot]$ are intractable and doesn't scale to big data.

At a Glance

- **What it is used for**: Inference and Generative Modelling in LVMs.
- **How it works**: Optimization of an unbiased estimator of the ELBO (objective function) using SGD.
- **What's a VAE**: AEVB where probability distributions in LVM are parametrized by Neural Networks.

AEVB Objective Function

Objective Function: ELBO Derivation

Introduce recognition model $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ with one parameter vector ϕ *shared across* data points.

$$
\begin{aligned}
\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})) &= \mathbb{E}_{q_{\boldsymbol{\phi}}} \left[\log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})\right] \\
&= -\mathbb{E}_{q_{\boldsymbol{\phi}}} \left[\log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}\right)\right] + \log p_{\boldsymbol{\theta}}(\mathbf{x})
\end{aligned}
$$

Objective Function: ELBO Derivation

Introduce recognition model $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ with one parameter vector ϕ *shared across* data points.

$$\begin{aligned}
\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})) &= \mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) - \log p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})\right] \\
&= -\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}\right)\right] + \log p_{\boldsymbol{\theta}}(\mathbf{x})
\end{aligned}$$

First term on RHS is Evidence Lower BOund (ELBO) denoted $\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x})$

$$\sum_{\mathbf{x} \in \mathcal{D}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\mathbf{x} \in \mathcal{D}} \left(\mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})) + \mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x})\right) \geq \sum_{\mathbf{x} \in \mathcal{D}} \mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x})$$

Since $\mathrm{KL}(\cdot \mid\mid \cdot) \geq 0$.

Objective Function: Two Birds, One Stone

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) \implies \begin{cases} \max_{\boldsymbol{\theta}} \sum_{\mathbf{x}\in\mathcal{D}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) & \text{as } \log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) \\ \min_{\boldsymbol{\phi}} \sum_{\mathbf{x}\in\mathcal{D}} \mathrm{KL} & \text{as } \log p_{\boldsymbol{\theta}}(\mathbf{x}) - \mathrm{KL} = \mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) \end{cases}$$

Therefore maximizing ELBO leads to:

- $p_{\boldsymbol{\theta}}(\mathbf{x})$ (i.e. generative model) improving.
- $q_{\boldsymbol{\phi}}$ becoming a better approximation.

Objective Function: Alternative ELBO

Rewrite ELBO as expected reconstruction error regularized by penalizing approximate posteriors $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ that are far from the prior $p_{\boldsymbol{\theta}}(\mathbf{z})$.

$$\begin{aligned}
\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) &= \log p_{\boldsymbol{\theta}}(\mathbf{x}) - \mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})) \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}} \left[\log \left(p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z})\right) - \log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})\right]}_{\text{Expected Log-Likelihood}} - \underbrace{\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z}))}_{\text{Regularization Term}}
\end{aligned}$$

Notice: Optimize ELBO using stochastic gradient optimization requires $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x})$

ELBO Optimization

Intractability of Objective Function Gradients

ELBO gradient $\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}}[\cdot]$ difficult to approximate with Monte Carlo as we cannot exchange gradient and expectation $\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}}[\cdot] \neq \mathbb{E}_{q_{\boldsymbol{\phi}}}[\nabla_{\boldsymbol{\phi}}]$

$$\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) = \begin{cases} \mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})\right] \\ \\ \mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})\right] - \mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z})) \end{cases}$$

Can we write expectation with respect to a distribution independent of ϕ?

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\cdot\right] \stackrel{?}{=} \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}}\right]$$

Reparametrization Trick

We know how to write a general MVN in terms of a Standard MVN

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \Longrightarrow \quad \mathbf{z} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^\top)$$

Thus expectations can be written as

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^\top)}\left[f(\mathbf{z})\right] = \mathbb{E}_{\mathcal{N}(\mathbf{0}, \mathbf{I})}\left[f(\boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon})\right]$$

In general write $\mathbf{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$ as a deterministic and differentiable function of $\mathbf{x}$ and $\boldsymbol{\epsilon}$, where $p(\boldsymbol{\epsilon})$ is independent of $\boldsymbol{\phi}$.

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}}\left[f(\mathbf{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}))\right]$$

SGC on Objective Function Unbiased Estimators

Obtain two unbiased estimators for ELBO based on $\epsilon^{(i)} \overset{\text{i.i.d.}}{\sim} p(\epsilon)$.

$$\widetilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \begin{cases} \frac{1}{L}\sum_{i=1}^{L}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i)}, \mathbf{x})) - \log q_{\boldsymbol{\phi}}(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i)}, \mathbf{x}) \mid \mathbf{x})\right] \\ \frac{1}{L}\sum_{i=1}^{L}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i)}, \mathbf{x}))\right] - \underbrace{\mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z}))}_{\text{Often available in closed form}} \end{cases}$$

SGD randomly samples a minibatch of data $M \subseteq \mathcal{D}$ and uses mini-batch gradients

$$\frac{1}{|\mathcal{M}|}\sum_{\mathbf{x}\in\mathcal{M}} \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$$

Marginal Likelihood Estimation

Estimating $\log p_{\boldsymbol{\theta}}(\mathcal{D})$

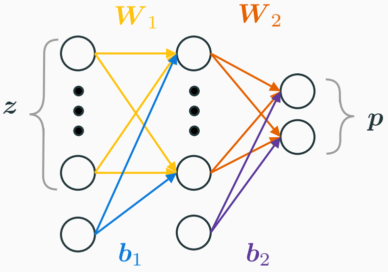
After training we can estimate the log marginal likelihood usingg **importance sampling**.

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \\
&= \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})} \right] \\
&\approx \log \frac{1}{L} \sum_{i=1}^{L} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}^{(i)})}{q_{\boldsymbol{\phi}}(\mathbf{z}^{(i)} \mid \mathbf{x})} \qquad \mathbf{z}^{(i)} \overset{\text{i.i.d.}}{\sim} q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})
\end{aligned}$$

Variational Auto-Encoders

Parametrizing Distributions via Neural Networks

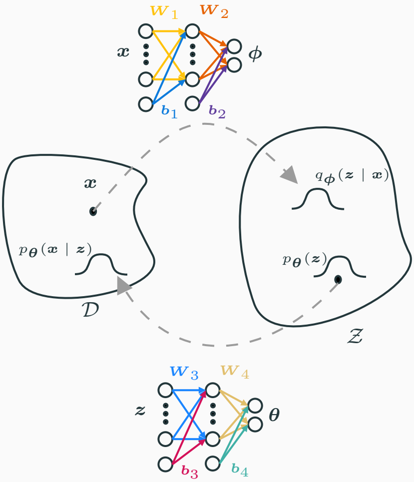
Suppose $\mathbf{x}$ is binary vector of Bernoulli trials. Then $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$ is parametrized by a vector of probabilities $\mathbf{p}$ which can be constructed via an MLP.



The log-likelihood then becomes

$$\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) = \sum_j x_j \log p_j + (1 - x_j) \log(1 - p_j)$$

VAE = AEVB + NN



$\boldsymbol{W}_1$
$\boldsymbol{W}_2$
$\boldsymbol{x}$
$\boldsymbol{\phi}$
$\boldsymbol{b}_1$
$\boldsymbol{b}_2$
$\boldsymbol{x}$
$q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$
$p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$
$p_{\boldsymbol{\theta}}(\boldsymbol{z})$
$\mathcal{D}$
$\mathcal{Z}$
$\boldsymbol{W}_3$
$\boldsymbol{W}_4$
$\boldsymbol{z}$
$\boldsymbol{\theta}$
$\boldsymbol{b}_3$
$\boldsymbol{b}_4$

Relationship between EM and VAE

Variational EM

Recall the EM algorithm:

- Compute approximate posteriors $\{p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z} \mid \mathbf{x}) \ : \ \mathbf{x} \in \mathcal{D}\}$
- Find optimal parameter $\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{x} \in \mathcal{D}} \mathbb{E}_{p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z}|\mathbf{x})} [\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})]$

Consider ELBO as a functional of $q_{\boldsymbol{\phi}}$ and a function of $\boldsymbol{\theta}$.

$$\mathcal{L}_{\mathbf{x}}(\boldsymbol{\theta}, q_{\boldsymbol{\phi}}) = \begin{cases} \log p_{\boldsymbol{\theta}}(\mathbf{x}) - \mathrm{KL}(q_{\boldsymbol{\phi}} \,||\, p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})) & (1) \\ \\ \mathbb{E}_{q_{\boldsymbol{\phi}}} [\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q_{\boldsymbol{\phi}}} [\log q_{\boldsymbol{\phi}}] & (2) \end{cases}$$

E-step: Maximize (1) wrt $q_{\boldsymbol{\phi}}$ (KL is zero and the bound is tight).

$$\left\{ p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z} \mid \mathbf{x}) = \arg\max_{q_{\boldsymbol{\phi}}} \mathcal{L}_{\mathbf{x}}(\boldsymbol{\theta}^{(t)}, q_{\boldsymbol{\phi}}) \ : \ \mathbf{x} \in \mathcal{D} \right\}$$

M-step: Maximize (2) wrt $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{x} \in \mathcal{D}} \mathcal{L}_{\mathbf{x}}(\boldsymbol{\theta}, p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z} \mid \mathbf{x}))$$

VAE = EM

In summary, EM algorithm and VAE optimize the same objective.

- When expectations are in closed-form, one should use EM, which uses coordinate ascent.
- When expectations are intractable, VAE uses stochastic gradient ascent on an unbiased estimator of the objective function.

Thank you

References i