Notes on Variational Autoencoders

Christoph Heindl
christoph.heindl@gmail.com

February 3, 2021

Contents

1 Introduction

Clear definition of distrubtion vs density needed

Better motivation needed

We seek to perform density estimation between a true (but unknown) probability distribution, and a set of proposal distributions. Formulated as minimization between distributions, this leads to the maximum (marginal) likelihood principle (see Section 2). To extend the expressiveness of our proposal distributions, we introduce latent variable models (see Section 3). The hidden variables raise issues in evaluating the marginal likelihood. We replace the marginal likelihood by a surrogate objective developed by Variational Inference (see Section 4). Combining these ideas gives us a powerful framework to estimate marginal, inference and joint distributions simultaneously. The Variational Autoencoder (see Section 5) is a specific application of these framework, that parametrizes the distributions to be optimized by neural networks.

The Appendix of this document contains additional material to Monte Carlo approximations, optimization of expectations and details of the Kullback-Leibler divergence. This document is accompanied by several Python notebooks, demonstrating derived results numerically.

Remark This document started out as a concise summary of Variational Autoencoders (VAE) [1, 2]. Over time, more and more topics were added, that shed light on the derivation from a broader perspective. Eventually, its content will be turned into a chapter of my thesis. This work is inspired by the contributions and books I found during my research [3, 4, 5, 6].

2 Density Estimation

Consider the task of estimating a density function from observed data. The true underlying density $p^\star(x)$ is unknown. Instead, we are given a set of *iid* observations $\{x_i\}_{i\leq N} \sim p^\star(x)$ having empirical density $p'(x)$ [7]. Let $\mathcal{P}_x$ denote a set of proposal densities $p(x)$. Density estimation attempts to find the element $p(x) \in \mathcal{P}_x$ which minimizes a cost function $C(p'(x), p(x))$. Consider the Kullback-Leibler divergence (see Appendix B) as objective. Then,

$$\begin{aligned}
\hat{p}(x) &= \arg\min_{p(x)\in\mathcal{P}_x} \mathrm{D}_{\mathrm{KL}}(p'(x) \,||\, p(x)) && (1)\\
&= \arg\min_{p(x)\in\mathcal{P}_x} \mathbb{E}_{x\sim p'(x)}\left[\log \frac{p'(x)}{p(x)}\right] && (2)\\
&= \arg\min_{p(x)\in\mathcal{P}_x} \left[\mathbb{E}_{x\sim p'(x)} \log p'(x) - \mathbb{E}_{x\sim p'(x)} \log p(x)\right] && (3)\\
&= \arg\min_{p(x)\in\mathcal{P}_x} -\mathbb{E}_{x\sim p'(x)} \log p(x) && (4)\\
&= \arg\max_{p(x)\in\mathcal{P}_x} \mathbb{E}_{x\sim p'(x)} \log p(x). && (5)
\end{aligned}$$

From the last equation it can be seen that the objective of density estimation using KL-divergence corresponds to maximum (marginal) likelihood estimation under the empirical distribution—a common objective in unsupervised machine learning.

Remark This sheds light on the inner workings of maximum likelihood. As shown in Appendix B, the specific order of arguments in the KL-divergence in the equations above leads a solution that spreads out over regions where $p'(x) > 0$ (i.e at samples) and is arbitrarily defined in between. Considering a proposal family $\mathcal{P}_x$ that contains infinitely complex densities and a small sample set, the resulting $\hat{p}(x)$ therefore tends to overfit. Thus, we need complexity regularization for $\mathcal{P}_x$ when doing maximum likelihood. One obious way: limit $\mathcal{P}_x$ to densities of simpler shape.

In the following, we assume $\mathcal{P}_x$ consists of models containing unobserved random variables. It turns out that Equation 5 cannot be applied in its current form.

3 Latent Variable Models

The latent variable model (LVM) introduces a set of unobserved random variables to explain the observations. The assumption being, observations are generated by transformations of simpler, but hidden causes. These hidden variables are denoted by z in contrast to x, the observables. The left Figure 1a depicts the general latent variable model. The main purpose of include latent variables is (a) to create more expressive models, (b) add causality between (simpler) latent variables and (complex) observations.

From Figure 1a, the joint density of LVM factors as

$$p(x, z) = p(x|z)p(z). \tag{6}$$

To account for the fact of latent variables, we define a new family of joint densities $p(x, z) \in \mathcal{P}_{x,z}$. Using the law of total probability allows us to rewrite Equation 5 as a maximization over elements $p(x, z)$

$$\arg\max_{p(x,z)\in\mathcal{P}_{x,z}} \mathbb{E}_{x\sim p'(x)} \left[\log p(x)\right] = \arg\max_{p(x,z)\in\mathcal{P}_{x,z}} \mathbb{E}_{x\sim p'(x)} \left[\log \int_z p(x, z)\, dz\right]. \tag{7}$$

The objective remains (usually) intractable because it requires marginalization over the latent variables, which poses an non-analytical integral. In the following section, Variational Inference (VI) is introduced. VI algebraically rearranges terms of the Equation 7 to find a surrogate objective that becomes computationally tractable.

(a) Latent variable model (b) Variational Inference

Figure 1: (a) Latent variable model. Observables x are generated by hidden causes z. (b) Variational Inference. Computing $p(z|x)$ in latent variable models is often intractable. Instead, Variational Inference seeks to find the best approximation to $p(z|x)$ from a set $q(z|x) \in \mathcal{Q}_{z|x}$ of proposal distributions.

4 Variational Inference

In the latent variable setting, Variational Inference (VI) replaces the intractable objective $\log p(x)$ of Equation 7 with a surrogate objective—the so called evidence lower bound $\text{ELBO}(q, x)$. Reconsider the marginal log-likelihood

$$\log p(x) = \log \int_z p(x, z)\, dz. \tag{8}$$

VI introduces a new family of distributions $\mathcal{Q}_{z|x}$, whose members are probability distributions of the form $q(z|x)$.

$$\log p(x) = \log \int_z \frac{q(z|x)}{q(z|x)} p(x, z)\, dz \tag{9}$$

$$= \log \int_z q(z|x) \frac{p(x, z)}{q(z|x)}\, dz \tag{10}$$

$$= \log \mathbb{E}_{z\sim q(z|x)} \left[\frac{p(x, z)}{q(z|x)} \right]. \tag{11}$$

Using Jensen's inequality,

$$\log p(x) \geq \mathbb{E}_{z\sim q(z|x)} \log \left[\frac{p(x, z)}{q(z|x)} \right] \tag{12}$$

$$\equiv \text{ELBO}(q, x). \tag{13}$$

The $\text{ELBO}(q, x)$ is a lower bound on the evidence $p(x)$, hence the name. It depends on the choice of q and x. The Appendix C contains an additional derivation using the variational principle. Independent of how you arrive at $\text{ELBO}(q, x)$, the consequences are as follows: Instead of maximizing $\log p(x)$, now we maximize the $\text{ELBO}(q, x)$ instead. Combining the maximum likelihood principle from Equation 5 and the $\text{ELBO}(q, x)$ leads to versatile optimization framework:

$$(\hat{p}(x, z), \hat{q}(z|x)) = \underset{\substack{p(x,z)\in\mathcal{P}_{x,z},\\ q(z|x)\in\mathcal{Q}_{z|x}}}{\arg\max} \mathbb{E}_{x\sim p'(x)} \mathbb{E}_{z\sim q(z|x)} \log \left[\frac{p(x, z)}{q(z|x)} \right]. \tag{14}$$

Optimizing the objective of Equation 14 yields the following quantities

- $q(z|x)$ A model to approximate the inference problem.
- $p(x, z)$ A model of the joint distribution.
- An approximation to the marginal log-likelihood via $\text{ELBO}(q, x)$.

Remark In VI we need to be able to evaluate $p(x, z)$. In practice this is often the case. Consider Bayesian Networks, where the joint distribution often factors into few terms due to independence assumptions. In addition, in VI $p(x, z)$ is often assumed to be known in advance. That is, no optimization over $\mathcal{P}_{x,z}$ takes place.

The Variational AutoEncoder (VAE) presented in the next section, jointly optimizes over $p(x, z)$ and $q(z|x)$ using densities parametrized by neural networks.

5 Variational Autoencoder

Variational Autoencoders (VAE) [1, 2] are a specific instance of models that optimize Equation 14. Starting from the definition of the $\mathrm{ELBO}(q, x)$ in Equation 13 and applying the latent variable model assumption

$$p(x, z) = p(x|z)p(z)$$

leads to

$$\begin{aligned}
\mathrm{ELBO}(q, x) &= \mathbb{E}_{z\sim q(z|x)} \log \left[\frac{p(x, z)}{q(z|x)}\right] && (15)\\
&= \int_z q(z|x) \log p(x|z)\, dz + \int_z q(z|x) \log \frac{p(z)}{q(z|x)}\, dz && (16)\\
&= \underbrace{\mathbb{E}_{z\sim q(z|x)} \left[\log p(x|z)\right]}_{\text{Reconstruction likelihood}} - \underbrace{\mathrm{D}_{\mathrm{KL}}(q(z|x) \,||\, p(z))}_{\text{Divergence from prior}}, && (17)
\end{aligned}$$

where we used the fact that

$$-\mathrm{D}_{\mathrm{KL}}(q(z) \,||\, p(z)) = \int_z q(z) \log \frac{p(z)}{q(z)}.$$

The objective in Equation 14 thus becomes

$$(\hat{p}(x|z), \hat{q}(z|x)) = \underset{\substack{p(x|z)\in\mathcal{P}_{x|z},\\ q(z|x)\in\mathcal{Q}_{z|x}}}{\arg\max}\ \mathbb{E}_{x\sim p'(x)} \left[\mathbb{E}_{z\sim q(z|x)} \log p(x|z) - \mathrm{D}_{\mathrm{KL}}(q(z|x) \,||\, p(z))\right]. \quad (18)$$

VAE assumes parametric families of normal densities

$$\begin{aligned}
\mathcal{Q}_{z|x} &= \{\mathcal{N}(\cdot; \theta) \colon \theta \in \Theta\}, \quad \theta = (\mu_q, \Sigma_q) && (19)\\
\mathcal{P}_{x|z} &= \{\mathcal{N}(\cdot; \phi) \colon \phi \in \Phi\}, \quad \phi = (\mu_p, \Sigma_p). && (20)
\end{aligned}$$

The prior is assumed to be $p(z) = \mathcal{N}(\cdot; 0, \mathbf{I})$. The introduction of parameters allows us to rewrite the objective of Equation 18 as search over parameters

$$\begin{aligned}
(\hat{\theta}, \hat{\phi}) &= \arg\max_{\theta,\phi}\ \mathbb{E}_{x\sim p'(x)} \left[\mathrm{ELBO}(x, \theta, \phi)\right] && (21)\\
&= \arg\max_{\theta,\phi}\ \mathbb{E}_{x\sim p'(x)} \left[\mathbb{E}_{z\sim q(z|x;\theta)} \log p(x|z; \phi) - \mathrm{D}_{\mathrm{KL}}(q(z|x; \theta) \,||\, p(z))\right] && (22)\\
&= \arg\min_{\theta,\phi}\ \mathbb{E}_{x\sim p'(x)} \left[-\mathbb{E}_{z\sim q(z|x;\theta)} \log p(x|z; \phi) + \mathrm{D}_{\mathrm{KL}}(q(z|x; \theta) \,||\, p(z))\right] && (23)\\
&= \arg\min_{\theta,\phi}\ \mathcal{L}(\theta, \phi). && (24)
\end{aligned}$$

Optimization uses (stochastic) gradient descent updates

$$\begin{aligned}
\theta^{(t+1)} &= \theta^{(t)} - \alpha \nabla_\theta \mathcal{L}(\theta^{(t)}, \phi^{(t)}) && (25)\\
\phi^{(t+1)} &= \phi^{(t)} - \beta \nabla_\phi \mathcal{L}(\theta^{(t)}, \phi^{(t)}). && (26)
\end{aligned}$$

Instead of directly optimizing over (μ_q, Σ_q), we predict these values by a function approximator $h(x; \theta)$

$$\begin{aligned}
h &\colon \mathbb{R}^D \to \mathbb{R}^d \times \mathbb{R}^d && (27)\\
x &\mapsto (\mu_q, \Sigma_q). && (28)
\end{aligned}$$

Similarly, let $g(z; \phi)$ be a function approximator of the following form

$$\begin{aligned}
g &\colon \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d && (29)\\
z &\mapsto (\mu_p, \Sigma_p). && (30)
\end{aligned}$$

VAE uses two separate neural networks for function approximation. Thus the parameters of the optimization are the weights of the neural networks. Backpropagation with stochastic gradient descent is used for minimization of $\mathcal{L}(\theta, \phi)$. Taking partial derivatives of $\mathcal{L}(\theta, \phi)$ requires derivatives of expectations, see Appendix A.

Remark In training Σ_p is considered a hyper-parameter and is not part of optimization. Usually, Σ_p factors as $\mathrm{diag}(\sigma_p^2)$. The likelihood term in Equation 17 is of means-squared-error form. This loss unsuited for perceptual purposes [8] and leads to blurry reconstructions. VAE uses path-wise derivatives to avoid computing gradients with respect to random number generators.

6 EM-Algorithm

The EM-Algorithm [9] is used to perform maximum likelihood parameter estimation in partially observed variable models.

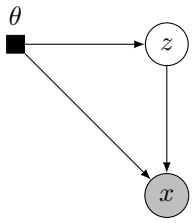

Figure 2: The parametric latent variable model.

Appendices

A Monte Carlo Approximations

Consider the expectation

$$\mathbb{E}_{z\sim p(z)}\left[f(z)\right].$$

Here z is a possibly multi-variate random variable $z \in \mathbb{R}^d$, and $f(z)$ denotes a scalar function $f\colon \mathbb{R}^d \to \mathbb{R}$. We can approximate the expectation using Monte Carlo integration to obtain noisy estimates as follows

$$\{z_i\}_{i\leq N} \sim p(z) \tag{31}$$

$$\mathbb{E}_{z\sim p(z)}\left[f(z)\right] \approx \frac{1}{N}\sum_{i=1}^{N}\left[f(z_i)\right]. \tag{32}$$

That is, we draw N samples according to $p(z)$ and take the mean over $\{f(z_i)\colon i \leq N\}$. This strategy also applies to derivatives of expectations

$$\nabla_\lambda \mathbb{E}_{z\sim p(z)}\left[f(z;\lambda)\right] = \mathbb{E}_{z\sim p(z)}\left[\nabla_\lambda f(z;\lambda)\right] \tag{33}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}\left[\nabla_\lambda f(z_i;\lambda)\right], \tag{34}$$

with parameters λ. Swapping derivatives and integrals is not possible in general, see Section A of Ryan [10].

Missing: unbiased, variance, variance reduction,sampling methods besides simple sampling

The above approximation covers the case when parameters are part of the function that we take the expectation over. In addition, strategies for computing approximate derivatives when the parameters are part of the probability distribution need to be developed. Consider $\nabla_\theta - \mathbb{E}_{z\sim q(z|x;\theta)}\left[\log p(x|z;\phi)\right]$ as an example. Finding efficient methods (in terms of convergence) is part of active research [11]. In the following sub-sections we cover two strategies:

- Score function derivatives
- Pathwise derivatives

A.1 Score Function Gradients

Consider finding $\nabla_\lambda \mathbb{E}_{z\sim p(z;\lambda)}[f(z)]$. The score function gradient is developed as follows

$$\begin{aligned}
\nabla_\lambda \mathbb{E}_{z\sim p(z;\lambda)}[f(z)] &= \nabla_\lambda \int_z f(z)p(z;\lambda)\,dz && (35)\\
&= \int_z f(z)\nabla_\lambda p(z;\lambda)\,dz && (36)\\
&= \int_z f(z)\frac{p(z;\lambda)}{p(z;\lambda)}\nabla_\lambda p(z;\lambda)\,dz && (37)\\
&= \int_z f(z)p(z;\lambda)\frac{\nabla_\lambda p(z;\lambda)}{p(z;\lambda)}\,dz && (38)\\
&= \int_z f(z)p(z;\lambda)\nabla_\lambda \log p(z;\lambda)\,dz && (39)\\
&= \mathbb{E}_{z\sim p(z;\lambda)}\left[\nabla_\lambda \log p(z;\lambda) f(z)\right]. && (40)
\end{aligned}$$

The score function gradient [12] allows us to express the derivate of an expectation as the expectation of a derivative. Thus, we can employ the Monte Carlo approximation to estimate the gradient. The fundamental trick is the introduction of the identity $\frac{p(z;\lambda)}{p(z;\lambda)}$, which allows two things to happen: (1) the denominator together with $\nabla_\lambda p(z;\lambda)$ leads to to $\nabla_\lambda \log p(z;\lambda)$ and (2) the nominator is the missing term for creating the expectation.

The score function gradient does not require $f(z)$ to be differentiable. On the flip-side it exhibits higher variance than path-wise derivatives.

A.2 Path-wise Derivatives

We come back to evaluating $\nabla_\lambda \mathbb{E}_{z\sim p(z;\lambda)}[f(z)]$. Path-wise derivatives use LOTUS backwards. To remove the parameters from $p(z;\lambda)$, we assume that we are able to find a different distribution $\epsilon \sim \pi(\epsilon), \epsilon \in \mathbb{R}^M$, independent of λ, such that

$$z \overset{d}{=} g(\epsilon;\lambda), \epsilon \sim \pi(\epsilon),$$

for some function $g\colon \mathbb{R}^M \to \mathbb{R}^d$ (typically M=d). For example any normal distribution $\mathcal{N}(z;\mu,\sigma^2)$ can be written as a transformation of a standard normal $z = \sigma\epsilon + \mu, \epsilon \sim \mathcal{N}(0,1)$.

This allows us to rewrite as follows

$$\begin{aligned}
\nabla_\lambda \mathbb{E}_{z\sim p(z;\lambda)}[f(z)] &= \nabla_\lambda \mathbb{E}_{\epsilon\sim\pi(\epsilon)}[f(g(\epsilon;\lambda))] && (41)\\
&= \mathbb{E}_{\epsilon\sim\pi(\epsilon)}\left[\nabla_\epsilon f(g(\epsilon;\lambda))\right] && (42)\\
&= \mathbb{E}_{\epsilon\sim\pi(\epsilon)}\left[J_\lambda g(\epsilon;\lambda)\nabla_z f(z)\right], && (43)
\end{aligned}$$

where J_λ denotes the Jacobian of g with respect of λ. Also path-wise derivatives allow us to express the derivative of an expectation as the expectation over derivatives. Compared to score function gradients, path-wise derivatives exhibit less variance, but require $f(z)$ to be differentiable. Path-wise derivatives are used by the VAE framework.

B Kullback-Leibler Divergence

The asymmetric Kullback-Leibler (KL) divergence is measure between two distributions p, q defined on the same probability space

$$\mathrm{D}_{\mathrm{KL}}(p(x) \mid\mid q(x)) = \int_x p(x)\log\left[\frac{p(x)}{q(x)}\right]. \quad (44)$$

The KL divergence can also be written as an expected value

$$\mathrm{D}_{\mathrm{KL}}(p(x) \mid\mid q(x)) = \mathbb{E}_{x\sim p(x)}\log\left[\frac{p(x)}{q(x)}\right]. \quad (45)$$

It is the expected value of the log of ratios between p and q weighted by p. That is, $p(x) \approx q(x)$ implies $\frac{p(x)}{q(x)} \approx 1$, implies $\log \frac{p(x)}{q(x)} \approx 0$. These error terms are then weighted by $p(x)$.

Reconsider density estimation

$$\hat{\theta} = \arg\min_{\theta} \mathrm{D}_{\mathrm{KL}}(p(x) \,||\, q(x;\theta)) \tag{46}$$

leads to $q(x;\hat{\theta})$ that attempts to match $p(x)$ closely wherever $p(x)$ has non-zero density. If $p(x)$ is a multimodal distribution and $q(x;\theta)$ is unimodal, then the resulting effect is that $q(x;\hat{\theta})$ spreads out over all modes of $p(x)$ (see left plot of Figure 3). This order of arguments in the KL-divergence is used in deriving maximum likelihood density estimation (see Section 2).

In contrast, minimizing

$$\hat{\theta} = \arg\min_{\theta} \mathrm{D}_{\mathrm{KL}}(q(x;\theta) \,||\, p(x)) \tag{47}$$

leads to $q(x;\hat{\theta})$ that ignores parts of $p(x)$ where $q(x;\hat{\theta})$ has zero density. If $p(x)$ is a multimodal distribution and $q(x;\theta)$ is unimodal, optimizing leads to mode catching of $p(x)$ (see right plot of Figure 3). This order of arguments in the KL-divergence is used by Variational Inference. (see Sub-section C and Section 4).

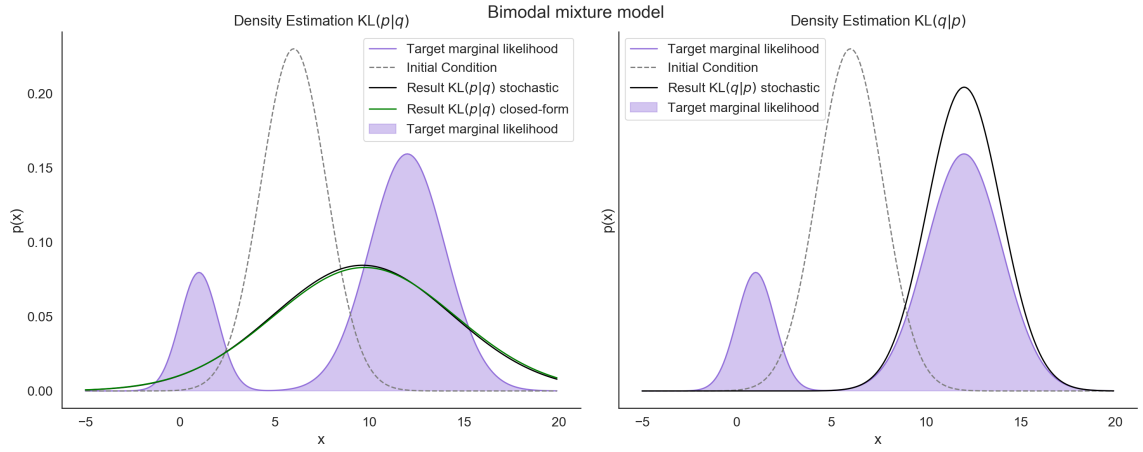

Figure 3: Density estimation by minimizing the KL-divergence between a bi-modal Gaussian mixture model $p(x)$ and an uni-modal normal distribution $q(x)$. Left: Minimizing $\mathrm{D}_{\mathrm{KL}}(p(x) \,||\, q(x))$ tends to put density everywhere $p(x) > 0$. Right: Minimizing $\mathrm{D}_{\mathrm{KL}}(q(x) \,||\, p(x))$ tends to chase a single mode.

B.1 Derivation of Density Estimation Gradients

Both plots of Figure 3 are created by an supplementary notebook `KL Density Estimation.ipynb`. The gradient derivation contains some tricks that are worth paying attention to.

The setup is as follows. The true distribution $p^{\star}(x)$ follows a bimodal Gaussian mixture model, the empirical distribution is denoted $p'(x)$. For brevity, we will refer to both target distributions as $p(x)$. The proposal distribution family $q(x;\theta) \in \mathcal{Q}_x$ consists of unimodal Gaussian distributions with parameters $\theta = (\mu, \sigma^2)$. Note, this example is not concerned with latent variable models. Optimization considers two objectives: (1) minimize $\mathrm{D}_{\mathrm{KL}}(p(x) \,||\, q(x;\theta))$; (2) $\mathrm{D}_{\mathrm{KL}}(q(x;\theta) \,||\, p(x))$. Both optimizations use (stochastic) gradient descent, Monte Carlo sampling to approximate expections (see Sub-section A) and score function gradients where required (see Sub-section A.1).

Both cases require gradients $\nabla_\theta \log q(x;\theta)$. For numerical stability and to avoid negative σ^2 in optimization, we reparametrize as follows

$$r = \log(\sigma^2)$$
$$\sigma^2 = e^r$$

and let $\theta = (\mu, r)$.

$$\nabla_\mu \log q(x;\theta) = \nabla_\mu \log \left[\frac{1}{\sqrt{2\pi e^r}} \exp\left(-\frac{(x-\mu)^2}{2e^r} \right) \right] \tag{48}$$

$$= \nabla_\mu \left[-\log \sqrt{2\pi e^r} - \frac{(x-\mu)^2}{2e^r} \right] \tag{49}$$

$$= \frac{x-\mu}{e^r}. \tag{50}$$

Continuing,

$$\nabla_r \log q(x;\theta) = \nabla_r \log \left[\frac{1}{\sqrt{2\pi e^r}} \exp\left(-\frac{(x-\mu)^2}{2e^r} \right) \right] \tag{51}$$

$$= -\nabla_r \log \sqrt{2\pi e^r} - \nabla_r \frac{(x-\mu)^2}{2e^r} \tag{52}$$

$$= -\frac{\nabla_r \sqrt{2\pi e^r}}{\sqrt{2\pi e^r}} + \frac{(x-\mu)^2}{2e^r} \tag{53}$$

$$= -0.5 \frac{(2\pi e^r)^{-0.5}(2\pi e^r)}{\sqrt{2\pi e^r}} + \frac{(x-\mu)^2}{2e^r} \tag{54}$$

$$= -0.5 \frac{\sqrt{2\pi e^r}}{\sqrt{2\pi e^r}} + \frac{(x-\mu)^2}{2e^r} \tag{55}$$

$$= \frac{(x-\mu)^2}{2e^r} - 0.5 \tag{56}$$

$$= 0.5(x-\mu)^2 e^{-r} - 0.5. \tag{57}$$

Case (1). Minimize

$$\hat{\theta} = \arg\min_\theta \, \mathrm{D}_{\mathrm{KL}}(p(x) \,||\, q(x;\theta)) \tag{58}$$

$$= \arg\min_\theta \, \mathbb{E}_{x\sim p(x)} \log \left[\frac{p(x)}{q(x;\theta)} \right] \tag{59}$$

$$= \arg\min_\theta \, \left[\mathbb{E}_{x\sim p(x)} \log p(x) - \mathbb{E}_{x\sim p(x)} \log q(x;\theta) \right] \tag{60}$$

$$= \arg\min_\theta \, - \mathbb{E}_{x\sim p(x)} \log q(x;\theta). \tag{61}$$

The gradient is found to be

$$-\nabla_\theta \, \mathbb{E}_{x\sim p(x)} \log q(x;\theta) = - \mathbb{E}_{x\sim p(x)} \nabla_\theta \log q(x;\theta). \tag{62}$$

Note that **Case (1)** allows for a closed-form parameter estimation, yielding the maximum likelihood estimators of the normal distribution. Consider a sample $\{x_i\}_{i\leq N} \sim p(x)$ and $p'(x)$ the associated empirical distribution. Then,

$$- \mathbb{E}_{x\sim p'(x)} \nabla_\mu \log q(x;\theta) = 0 \tag{63}$$

$$= -\sum_{i=1}^{N} \frac{1}{N} \frac{x_i - \mu}{\sigma^2} \tag{64}$$

$$\Leftrightarrow \mu = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{65}$$

and

$$- \mathbb{E}_{x\sim p'(x)} \nabla_{\sigma 2} \log q(x;\theta) = 0 \tag{66}$$

$$= -\sum_{i=1}^{N} \frac{1}{N} \left[\frac{(x_i-\mu)^2}{2\sigma^2} - 0.5 \right] \tag{67}$$

$$\Leftrightarrow \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2. \tag{68}$$

Case (2) Find

$$\hat{\theta} = \arg\min_{\theta} \mathrm{D}_{\mathrm{KL}}(q(x;\theta) \,||\, p(x)) \tag{69}$$

$$= \arg\min_{\theta} \mathbb{E}_{x\sim q(x;\theta)} \log\left[\frac{q(x;\theta)}{p(x)}\right] \tag{70}$$

$$= \arg\min_{\theta} \mathbb{E}_{x\sim q(x;\theta)} f(x;\theta), \tag{71}$$

with $f(x;\theta) = \log\left[\frac{q(x;\theta)}{p(x)}\right]$. In this task, the parameters are inside $f(x;\theta)$ and additionally in the distribution the expectation is taken over. Noting that

$$\nabla_\theta f(x;\theta) = \nabla_\theta \log\left[\frac{q(x;\theta)}{p(x)}\right] \tag{72}$$

$$= \frac{p(x)}{q(x;\theta)} \frac{\nabla_\theta q(x;\theta)}{p(x)} \tag{73}$$

$$= \frac{\nabla_\theta q(x;\theta)}{q(x;\theta)} \tag{74}$$

$$= \nabla_\theta \log q(x;\theta), \tag{75}$$

and additionally $f(x;\theta) = \log q(x;\theta) - \log p(x)$. Continuing from Equation 71 yields

$$\nabla_\theta \mathbb{E}_{x\sim q(x;\theta)} f(x;\theta) = \nabla_\theta \int_x q(x;\theta) f(x;\theta)\, dx \tag{76}$$

$$= \int_x \nabla_\theta q(x;\theta) \cdot f(x;\theta) + q(x;\theta) \cdot \nabla_\theta f(x;\theta)\, dx \tag{77}$$

$$= \int_x \frac{q(x;\theta)}{q(x;\theta)} \nabla_\theta q(x;\theta) \cdot f(x;\theta) + q(x;\theta) \cdot \nabla_\theta f(x;\theta)\, dx \tag{78}$$

$$= \int_x q(x;\theta) \nabla_\theta \log q(x;\theta) \cdot f(x;\theta) + q(x;\theta) \cdot \nabla_\theta f(x;\theta)\, dx \tag{79}$$

$$= \int_x q(x;\theta) \nabla_\theta \log q(x;\theta) \cdot f(x;\theta) + q(x;\theta) \cdot \nabla_\theta \log q(x;\theta)\, dx \tag{80}$$

$$= \int_x q(x;\theta) \nabla_\theta \log q(x;\theta) \left(f(x;\theta) + 1\right)\, dx \tag{81}$$

$$= \mathbb{E}_{x\sim q(x;\theta)} \nabla_\theta \log q(x;\theta) \left(\log q(x;\theta) - \log p(x) + 1\right). \tag{82}$$

C Derivation of ELBO from KL-Divergence

Section 4 derived the ELBO from the log-likelihood. This required the trick of multiplying and dividing by $q(z|x)$. This section derives the ELBO from a Kullback-Leibler principle.

Variational Inference (VI) is all concerned with inference of the intractable quantity $p(z|x)$. For this purposes it assumes a set of candidate distributions $q(z|x) \sim \mathcal{Q}_{z|x}$. The natural objective of VI is then to minimize the following divergence.

$$\mathrm{D}_{\mathrm{KL}}(q(z|x) \,||\, p(z|x)) = \mathbb{E}_{q(z|x)} \log\left[\frac{q(z|x)}{p(z|x)}\right] \tag{83}$$

$$= \mathbb{E}_{q(z|x)} \left[\log q(z|x) - \log p(z|x)\right] \tag{84}$$

$$= \mathbb{E}_{q(z|x)} \left[\log q(z|x) - \log p(x,z) + \log p(x)\right] \tag{85}$$

$$= \mathbb{E}_{q(z|x)} \left[\log q(z|x) - \log p(x,z)\right] + \log p(x) \tag{86}$$

$$= \mathbb{E}_{q(z|x)} \log\left[\frac{q(z|x)}{p(x,z)}\right] + \log p(x), \tag{87}$$

where we used $p(z|x) = \frac{p(x,z)}{p(x)}$ in Equation 85. Rearranging terms in the last equation leads to

$$\log p(x) = \mathrm{D}_{\mathrm{KL}}(q(z|x) \,||\, p(z|x)) - \mathbb{E}_{q(z|x)} \log \left[\frac{q(z|x)}{p(x,z)} \right] \tag{88}$$

$$= \underbrace{\mathrm{D}_{\mathrm{KL}}(q(z|x) \,||\, p(z|x))}_{\geq 0} + \mathbb{E}_{q(z|x)} \log \left[\frac{p(x,z)}{q(z|x)} \right] \tag{89}$$

$$\geq \mathbb{E}_{q(z|x)} \log \left[\frac{p(x,z)}{q(z|x)} \right] \tag{90}$$

$$\equiv \mathrm{ELBO}(x). \tag{91}$$

Todos

- [ ] Clear definition of distrubtion vs density needed 1
- [ ] Better motivation needed 1
- [ ] Missing: unbiased, variance, variance reduction,sampling methods besides simple sampling . 5

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[3] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[4] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[5] Rui Shu. Density estimation: Variational autoencoders. http://ruishu.io/2018/03/14/vae/, 2019.

[6] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[7] MS Waterman and DE Whiteman. Estimation of probability densities by empirical density functions. *International Journal of Mathematical Education in Science and Technology*, 9(2):127–137, 1978.

[8] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[9] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[10] Martin Ryan. Likelihood and maximum likelihood estimation. http://www2.stat.duke.edu/~sayan/SAMSI/lec/411notes03.pdf, 2019. Online.

[11] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452, 2018.

[12] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.