6.7800 Spring 2025: Bonus Lecture 05

Variational Techniques in Generative Models

Jongha (Jon) Ryu
jongha@mit.edu

May 9, 2025

Contents

1 Recap: Maximum Likelihood Estimation

Consider observations $y_1, \ldots, y_N$ generated i.i.d. from a data generating distribution $p_{\mathsf{y}}(\cdot)$. In the lecture, we learned that for discrete alphabets, the maximum likelihood estimator (MLE)

$$\hat{x}_{\mathsf{ML}} \in \arg\max_x \frac{1}{N} \sum_{i=1}^{N} \log p(y_i; x) \tag{1}$$

can be equivalently viewed as KL divergence minimization

$$\hat{x}_{\mathsf{ML}} \in \arg\min_x D(\hat{p}_{\mathsf{y}}(\cdot) \,\|\, p_{\mathsf{y}}(\cdot; x)). \tag{2}$$

Recall that $\hat{p}_{\mathsf{y}}(\cdot)$ denotes the empirical distribution induced by the observations $\{y_1, \ldots, y_N\}$. Succinctly, we can thus say that the MLE is the M-projection of the empirical distribution onto the parametric family.

We can also justify the MLE (1) for a continuous alphabet as an M-projection in an approximate sense as follows. In the asymptotic limit, we can define the MLE as

$$x_{\mathsf{ML}} \in \arg\min_x D(p_{\mathsf{y}}(\cdot) \parallel p_{\mathsf{y}}(\cdot; x)). \tag{3}$$

Since $D(p_{\mathsf{y}}(\cdot) \parallel p_{\mathsf{y}}(\cdot; x)) = -\mathbb{E}_{p_{\mathsf{y}}(\cdot)}[\log p_{\mathsf{y}}(\cdot; x)] - h(p_{\mathsf{y}}(\cdot))$, where $h(p_{\mathsf{y}}(\cdot))$ is the differential entropy of y, it is equivalent to writing

$$x_{\mathsf{ML}} \in \arg\max_x \mathbb{E}_{p_{\mathsf{y}}(\cdot)}[\log p_{\mathsf{y}}(y; x)], \tag{4}$$

since the differential entropy term is irrelevant to the optimization. Now, the MLE (1) with finite observations can be understood as a finite-sample approximation of the infinite-sample version (4).

The MLE criterion can be used in both scenarios of (1) parameter estimation, i.e., when we are interested in figuring out x^*, and (2) distribution fitting, i.e., when we are interested in finding a parametric distribution $p_{\mathsf{y}}(\cdot; x)$ closely approximating the data generating distribution $\hat{p}$. Note that the distribution fitting framework becomes extremely popular recently for training generative models.

Maybe not surprisingly, most of the generative models can be understood via the lens of divergence matching, beyond the M-projection. In this note, we will provide a quick overview on the objective functions of a few popular generative models, namely variational autoencoders [4, 5] and generative adversarial networks [3, 7].

Below, we will use a slightly different notation system from the class. We will use θ, φ for model parameters and $\mathbf{x}$ for feature vectors (such as images). We will omit the subscript in the distribution, which was used in the class to denote the random variables that corresponds to a distribution.

2 Variational Autoencoders

Variational autoencoders (VAEs), first proposed by Kingma and Welling [4], are *autoencoders* learned by a *variational* principle, as suggested by its name. In the following, we will explain these two components in order.

2.1 Autoencoders

Suppose that you are given a high-dimensional datum $\mathbf{x} \in \mathbb{R}^D$ such as image and audio, which is assumed to be drawn form an underlying distribution $q_{\mathsf{data}}(\mathbf{x})$. Since it is often hard to deal with such a complex object directly, an important problem in machine learning is how to find a good *low-dimensional embedding*, also known as *dimensionality reduction* or *representation learning* problem. A working assumption behind this procedure is the effective dimension of such high-dimensional modalities is indeed low; this is known as *manifold hypothesis*; see, e.g., [1].

The *autoencoders* is proposed as a neural-network-based approach to find such a low-dimensional representation. Let $L \geq 1$ be a designated dimensionality. An autoencoder consists of two neural networks: an encoder $\mathbf{f}_\varphi \colon \mathbb{R}^D \to \mathbb{R}^L$ which *encodes* a datum $\mathbf{x}$ to a *latent* $\mathbf{z} = \mathbf{f}_\varphi(\mathbf{x}) \in \mathbb{R}^L$, and a

decoder $\mathbf{g}_\theta : \mathbb{R}^L \to \mathbb{R}^D$ which *decodes* a latent $\mathbf{z} \in \mathbb{R}^L$ to a reconstruction $\hat{\mathbf{x}} = \mathbf{g}_\theta(\mathbf{z})$. The encoder and decoder are trained to minimize the expected reconstruction error meaured in the ℓ_2^2-distance:

$$\min_{\theta,\varphi} \mathbb{E}_{p(\mathbf{x})}[\|\mathbf{x} - g_\theta(f_\varphi(\mathbf{x}))\|_2^2].$$

In the autoencoder framework, we believe that a good latent representation, i.e., the encoder $\mathbf{f}_\varphi$, is *automatically* found by trying to minimize the reconstruction error. Conceptually, the decoder only plays a role as a facilitator to train the encoder.

After all, can we *generate* images out of a trained autoencoder? Theoretically, the answer is yes. To see why, note that the data distribution and encoder *induces* a distribution over the latent:

$$q_\varphi(\mathbf{z}) := \int \delta(\mathbf{z} - f_\varphi(\mathbf{x})) q_{\text{data}}(\mathbf{x}) \, \mathrm{d}\mathbf{x}.$$

If we knew how to draw a sample from the distribution $\mathbf{z} \sim q_\varphi(\mathbf{z})$, then its decoding $\hat{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{z})$ would have been distributed close to $q_{\text{data}}(\mathbf{x})$. The problem with this reasoning in practice is that we do not know how to sample from $q_\varphi(\mathbf{z})$! This motivates the following question:

Q. Can we train an autoencoder so that $p_\theta(\mathbf{z})$ is some fixed, easy-to-sample distribution?

2.2 Latent Variable Models and MLE

This leads us to consider a *latent variable model*, which is defined as a joint distribution over the data and latent space:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}).$$

Here, the *prior* $p_\theta(\mathbf{z})$ is often chosen as a fixed distribution such as the standard Gaussian $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}_L, I_L)$. In this case, we drop the dependence on θ in the notation, i.e., $p_\theta(\mathbf{z}) = p(\mathbf{z})$. The *likelihood* $p_\theta(\mathbf{x}|\mathbf{z})$ is parameterized by a spherical Gaussian, whose mean is parameterized by a decoder neural network $\mathbf{g}_\theta : \mathbb{R}^L \to \mathbb{R}^D$, i.e.,

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{g}_\theta(\mathbf{z}), \varepsilon^2 I_D),$$

where $\varepsilon > 0$ is a hyperparameter. Note the Gaussian likelihood model is assumed to let the density $p_\theta(\mathbf{x}, \mathbf{z})$ *explicitly* computable, that is, one can compute the value of the density $p_\theta(\mathbf{x}, \mathbf{z})$ for any $\mathbf{x}$ and $\mathbf{z}$. After training, the decoding process is assumed to be *deterministic*, ignoring the Gaussian noise in $p_\theta(\mathbf{x}|\mathbf{z})$. If we can somehow train this latent variable model so that the induced marginal $p_\theta(\mathbf{x}) := \int p_\theta(\mathbf{x}, \mathbf{z}) \, \mathrm{d}\mathbf{z}$ is close to the data distribution $q_{\text{data}}(\mathbf{x})$, then $\hat{\mathbf{x}} \leftarrow \mathbf{g}_\theta(\mathbf{z})$ for $\mathbf{z} \sim p(\mathbf{z})$ would emulate $\mathbf{x} \sim q_{\text{data}}(\mathbf{x})$. For now, we disregard the encoder from the previous section.

The first idea you can think of to train the latent variable model may be MLE, which is, in the population limit, solving

$$\min_\theta D(q_{\text{data}}(\mathbf{x}) \,\|\, p_\theta(\mathbf{x})), \tag{5}$$

or in the finite-sample regime where we have access to samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ drawn i.i.d. from $q_{\text{data}}(\mathbf{x})$, solving

$$\min_\theta \frac{1}{N} \sum_{n=1}^{N} \log \frac{1}{p_\theta(\mathbf{x}_n)}. \tag{6}$$

This approach is considered infeasible in practice, since the induced marginal density $p_\theta(\mathbf{x})$ is hard to compute by definition, due to the high-dimensional integration over $\mathbf{z}$, similar to the issue with computing partition functions we learned in the very last lecture.

2.3 Variational Relaxation with Variational Encoder

We now illustrate how we can apply the idea of variational relaxation to circumvent with the high-dimensional integration. For the relaxation, we introduce a *variational* encoder, which is a set of conditional distributions $q_\varphi(\mathbf{z}|\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^D$. For each θ (decoder parameter), we wish that each $q_\varphi(\mathbf{z}|\mathbf{x})$ fits to the *model posterior* $p_\theta(\mathbf{z}|\mathbf{x}) := \frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{x})}$. Starting from the population objective of MLE (5), we consider the following relaxation:

$$\begin{aligned}
D(q_{\text{data}}(\mathbf{x}) \parallel p_\theta(\mathbf{x})) &\le D(q_{\text{data}}(\mathbf{x}) \parallel p_\theta(\mathbf{x})) + \mathbb{E}_{q_{\text{data}}(\mathbf{x})}[D(q_\varphi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))] && (7)\\
&= D(q_{\text{data}}(\mathbf{x})q_\varphi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{x})p_\theta(\mathbf{z}|\mathbf{x})) && (8)\\
&= D(q_{\text{data}}(\mathbf{x})q_\varphi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})) && (9)\\
&= \mathbb{E}_{q_{\text{data}}(\mathbf{x})q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\varphi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})}\right] + \mathbb{E}_{q_{\text{data}}(\mathbf{x})}[\log q_{\text{data}}(\mathbf{x})] && (10)\\
&= \mathbb{E}_{q_{\text{data}}(\mathbf{x})q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\varphi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} + \log \frac{1}{p_\theta(\mathbf{x}|\mathbf{z})}\right] + C && (11)\\
&= \mathbb{E}_{q_{\text{data}}(\mathbf{x})}\left[D(q_\varphi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) + \frac{1}{2\varepsilon^2}\mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\|\mathbf{x} - \mathbf{g}_\theta(\mathbf{z})\|_2^2]\right] + C. && (12)
\end{aligned}$$

Here, (7) follows from the nonnegativity of KL divergence, (8) from the chain rule of KL divergence, and (9) from the definition of the joint distribution $p_\theta(\mathbf{x}, \mathbf{z})$ and the model posterior $p_\theta(\mathbf{z}|\mathbf{x})$. In (10), the second term corresponds to the differential entropy of data (assuming that $\mathbf{x}$ is continuous), which is *constant* with respect to both θ, φ, and thus we denote it as $C := -h(q_{\text{data}}(\mathbf{x}))$ in (11). Note that the expression in (11) is computable given that $q_\varphi(\mathbf{z}|\mathbf{x})$ is computable. A more popular expression for the VAE objective is (12), where we write the first term in terms of the KL divergence between $q_\varphi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$, which is often called the *KL regularization loss*, and the second term is from the Gaussian likelihood assumption $p_\theta(\mathbf{x}|\mathbf{z})$, which is called the *reconstruction loss*. The KL regularization term $D(q_\varphi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))$ can be sometimes analytically computed for a given $(\mathbf{x}, \mathbf{z})$, which will be the case of the choice illustrated below.

The most standard parameterization for the variational encoder $q_\varphi(\mathbf{z}|\mathbf{x})$ is a diagonal Gaussian of the form

$$q_\varphi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\varphi(\mathbf{x}), \operatorname{diag}(\boldsymbol{\sigma}_\varphi^2(\mathbf{x}))).$$

Here, $\boldsymbol{\mu}_\varphi \colon \mathbb{R}^D \to \mathbb{R}^L$ and $\boldsymbol{\sigma}_\varphi \colon \mathbb{R}^D \to \mathbb{R}^L$ are the outputs of the encoder. Note that this choice of parameterization is purely based on a practical consideration, without a guarantee. As each latent

coordinate is assumed to be conditionally independent, which may not hold for the model posterior $p_\theta(\mathbf{z}|\mathbf{x})$. This, however, provides a convenient choice widely used, is also known as the *mean field approximation*. Recall that if the variational encoder cannot perfectly fit the model posterior, the inequality (7) becomes loose.

2.4 Reparameterization Trick

So far, we have derived the (population) objective for the VAE, namely, from (11),

$$\mathcal{L}_{\mathsf{VAE}}(\theta, \varphi) = \mathbb{E}_{q_{\mathsf{data}}(\mathbf{x})q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\varphi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} + \log \frac{1}{p_\theta(\mathbf{x}|\mathbf{z})}\right]. \tag{13}$$

Given samples drawn from $\{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^N \sim q_{\mathsf{data}}(\mathbf{x})q_\varphi(\mathbf{z}|\mathbf{x})$, the objective function can be estimated in an unbiased manner by replacing the expectation via a Monte Carlo approximation:

$$\hat{\mathcal{L}}_{\mathsf{VAE}}(\theta, \varphi) = \frac{1}{N}\sum_{i=1}^{N}\left(\log \frac{q_\varphi(\mathbf{z}_i|\mathbf{x}_i)}{p(\mathbf{z}_i)} + \log \frac{1}{p_\theta(\mathbf{x}_i|\mathbf{z}_i)}\right). \tag{14}$$

In practice, however, we optimize the objective function via a gradient-based algorithm such as stochastic gradient descent (SGD) or Adam, in which case it is important to have an *unbiased gradient esitmator* with low variance. A nontrivial issue arises for the VAE objective, however, when we wish to take a gradient with respect to the encoder parameter φ, since the expectation is with respect to a distribution dependent on φ.

To understand the issue more clearly, consider an abstract setting where we wish to compute a gradient of an expectation of the form $\mathbb{E}_{q_\varphi(y)}[h_\varphi(y)]$, where both the distribution $q_\varphi(y)$ and function $h_\varphi(y)$ depend on the parameter φ. In this case, one can check that

$$\nabla_\varphi \mathbb{E}_{q_\varphi(y)}[h_\varphi(y)] \overset{(a)}{=} \mathbb{E}_{q_\varphi(y)}[\nabla_\varphi \log q_\varphi(y) h_\varphi(y) + \nabla_\varphi h_\varphi(y)] \neq \mathbb{E}_{q_\varphi(y)}[\nabla_\varphi h_\varphi(y)].$$

This discrepancy implies that the gradient $\nabla_\varphi \mathbb{E}_{q_\varphi(y)}[h_\varphi(y)]$ cannot be estimated in an unbiased manner by $\mathbb{E}_{q_\varphi(y)}[\nabla_\varphi h_\varphi(y)]$. The unbiased gradient estimator based on the expression in (a) is known as the REINFORCE estimator in the literature, but it is known to suffer high variance.

As an alternative, Kingma and Welling [4] proposed a clever trick called the *reparameterization trick* to come up with a low-variance unbiased gradient estimator. Suppose that drawing a sample $y \sim q_\varphi(y)$ is equivalent to $y = t_\varphi(v)$ with $v \sim q(v)$ for some distribution $q(v)$ and a function t_φ. Then, we can *reparameterize* the expectation $\mathbb{E}_{q_\varphi(y)}[\cdot]$ with $\mathbb{E}_{q(v)}[\cdot]$, so that the gradient can be interchanged with the expectation[1]:

$$\nabla_\varphi \mathbb{E}_{q_\varphi(y)}[h_\varphi(y)] = \nabla_\varphi \mathbb{E}_{p(v)}[h_\varphi(t_\varphi(v))] = \mathbb{E}_{p(v)}[\nabla_\varphi h_\varphi(t_\varphi(v))].$$

The last expression $\mathbb{E}_{p(v)}[\nabla_\varphi h_\varphi(t_\varphi(v))]$ is the final gradient estimator.

Note that the Gaussian encoder $q_\varphi(\mathbf{z}|\mathbf{x})$ is reparameterizable, since $\mathbf{z} \sim q_\varphi(\mathbf{z}|\mathbf{x})$ is equivalent to $\mathbf{z} = \boldsymbol{\mu}_\varphi(\mathbf{x}) + \boldsymbol{\sigma}_\varphi(\mathbf{x}) \odot \mathbf{v}$, where $\mathbf{v} \sim p(\mathbf{v}) = \mathcal{N}(\mathbf{v}; 0, I)$. With this choice of encoder, the gradient can be automatically computed via autograd, based on the unbiased estimate of the objective function (14).

[1]Note that changing expectation and a limit operation needs a certain regularity condition such as uniform integrability; in machine learning practice, it is implicitly assumed to hold by default.

3 Generative Adversarial Networks

While VAE is an important class of generative models that have been used in many different applications, it produces rather blurry images compared to some other alternatives, such as generative adversarial networks (GANs), normalizing flows, and diffusion models. In this note, we introduce the principle of a family of GANs from a simple variational learning perspective, which is different from that of VAE.

Similar to VAEs, in the GAN framework, we aim to train a deterministic decoder $\mathbf{g}_\theta(\mathbf{z})$ with prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, I_L)$. Unlike VAE that assumes a tractable density model $p_\theta(\mathbf{x}, \mathbf{z})$, GAN does not assume such model and is thus classified as an *implicit* density framework. The marginal density induced by $p(\mathbf{z})$ and $g_\theta(\mathbf{z})$ is formally defined as

$$p_\theta(\mathbf{x}) := \int \delta(\mathbf{x} - g_\theta(\mathbf{z}))p(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

With a choice of divergence function $D(p, q)$ defined for distributions p, q, we wish to train the decoder $g_\theta(\mathbf{z})$ by solving

$$\min_\theta D(q_{\mathsf{data}}(\mathbf{x}), p_\theta(\mathbf{x})).$$

Specifically, in this note, we explain how to solve this divergence minimization problem for the f-divergence [2].

3.1 f-Divergence

Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a strictly convex function with $f(1) = 0$. For two distributions with densities $p(\mathbf{x})$ and $q(\mathbf{x})$, the *f-divergence* between p and q is defined as

$$D_f(p(\mathbf{x}) \parallel q(\mathbf{x})) := \mathbb{E}_{q(\mathbf{x})}\left[f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\right],$$

if p is absolutely continuous with respect to q, i.e., $p(\mathbf{x}) = 0$ whenever $q(\mathbf{x}) = 0$. It is easy to check that f-divergence has positive definiteness, i.e., $D_f(p(\mathbf{x}) \parallel q(\mathbf{x})) = 0$ if and only if $p(\cdot) = q(\cdot)$.[2] For example $f(r) = r \log r$ corresponds to $D_f(p \parallel q) = D(p \parallel q)$, and $f(r) = -\log r$ to $D_f(p \parallel q) = D(q \parallel p)$. Another important example is $f(r) = r \log r - (r + 1) \log(r + 1)$, which corresponds to the Jensen–Shannon divergence

$$D_f(p \parallel q) = D_{\mathsf{JS}}(p \parallel q) := D\left(p \,\middle\|\, \frac{p+q}{2}\right) + D\left(q \,\middle\|\, \frac{p+q}{2}\right).$$

3.2 Variational Principle for f-Divergence

Recall that in VAE, the variational relaxation was based on the nonnegativity of KL divergence, which is essentially a consequence of Jensen's inequality, or even the convexity of a certain function

[2] To see this, note that we have

$$D_f(p(\mathbf{x}) \parallel q(\mathbf{x})) \geq f\left(\mathbb{E}_{q(\mathbf{x})}\left[\frac{p(\mathbf{x})}{q(\mathbf{x})}\right]\right) = f(1) = 0,$$

by Jensen's inequality. Since f is strictly convex, the equality holds if and only if $p(\cdot) = q(\cdot)$.

at the lowest level. Again, we can derive a variational principle from the first principle, i.e., the convexity of f.

Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a differentiable convex function. Then, by the definition of convexity, we have

$$f\Big(\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\Big) \geq f(r) + f'(r)\Big(\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})} - r\Big)$$

for any $r \geq 0$, for each $\mathbf{x}$; here, the equation holds if and only if $r = \frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}$. Equivalently, for any function $r_\psi : \mathcal{X} \to \mathbb{R}_{\geq 0}$,

$$f\Big(\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\Big) \geq f(r_\psi(\mathbf{x})) + f'(r_\psi(\mathbf{x}))\Big(\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})} - r_\psi(\mathbf{x})\Big)$$

holds for any $\mathbf{x}$. Taking an expectation on both sides with respect to $p_\theta(\mathbf{x})$, we have

$$\begin{aligned} D_f(q_{\mathsf{data}}(\mathbf{x}) \parallel p_\theta(\mathbf{x})) &= \mathbb{E}_{p_\theta(\mathbf{x})}\Big[f\Big(\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\Big)\Big] \\ &\geq \mathbb{E}_{p_\theta(\mathbf{x})}\Big[f(r_\psi(\mathbf{x})) + f'(r_\psi(\mathbf{x}))\Big(\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})} - r_\psi(\mathbf{x})\Big)\Big] \\ &= \mathbb{E}_{q_{\mathsf{data}}(\mathbf{x})}[f'(r_\psi(\mathbf{x}))] + \mathbb{E}_{p_\theta(\mathbf{x})}[f(r_\psi(\mathbf{x})) - r_\psi(\mathbf{x})f'(r_\psi(\mathbf{x}))]. \end{aligned}$$

Note that the inequality holds with equality if and only if $r_\psi(\cdot) = \frac{q_{\mathsf{data}}(\cdot)}{p_\theta(\cdot)}$, i.e., when $r_\psi(\mathbf{x})$ exactly matches the density ratio $\frac{q_{\mathsf{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}$. Hence, maximizing the lower bound with respect to $r_\psi(\cdot)$ can be understood as fitting the density ratio model to the underlying ratio. This leads to the following variational characterization of the f-divergence:

$$D_f(q_{\mathsf{data}}(\mathbf{x}) \parallel p_\theta(\mathbf{x})) \geq \max_\psi \Big\{\mathbb{E}_{q_{\mathsf{data}}(\mathbf{x})}[f'(r_\psi(\mathbf{x}))] + \mathbb{E}_{p_\theta(\mathbf{x})}[f(r_\psi(\mathbf{x})) - r_\psi(\mathbf{x})f'(r_\psi(\mathbf{x}))]\Big\}.$$

Note that the inequality becomes tight if the function $r_\psi(\cdot)$ is expressive enough to capture the true density ratio.

Finally, taking the minimization over θ, we obtain

$$\min_\theta D_f(q_{\mathsf{data}}(\mathbf{x}) \parallel p_\theta(\mathbf{x})) \geq \min_\theta \max_\psi \Big\{\mathbb{E}_{q_{\mathsf{data}}(\mathbf{x})}[f'(r_\psi(\mathbf{x}))] + \mathbb{E}_{p_\theta(\mathbf{x})}[f(r_\psi(\mathbf{x})) - r_\psi(\mathbf{x})f'(r_\psi(\mathbf{x}))]\Big\}.$$

The minimax optimization problem on the right hand side is the well-known f-GAN objective [7].

We note that while the standard arguments for the variational lower bound for f-divergence are based on Fenchel duality [6, 7], in this note we derive from an even simpler principle, which is the definition of differentiable convex functions. This is much easier to state and understand, covering most, if not all, of the interesting examples.

Here, the *density ratio model* $r_\psi(\cdot)$ plays a role of the variational encoder $q_\varphi(\mathbf{z}|\mathbf{x})$ in VAE. If we define

$$D_\psi(\mathbf{x}) := \frac{r_\psi(\mathbf{x})}{1 + r_\psi(\mathbf{x})} \in [0, 1],$$

it is the standard parameterization of the auxiliary variable popularly known as a *discriminator*. With this parameterization and for $f(r) = r \log r - (1 + r) \log(1 + r)$, it recovers the most famous vanilla GAN [3].

References

[1] Lawrence Cayton. Algorithms for manifold learning. Technical report, University of California San Diego, 2008.

[2] Imre Csiszár and Paul C Shields. Information theory and statistics: A tutorial. *Found. Trends Commun. Inf. Theory*, 1(4):417–528, 2004.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inf. Proc. Syst.*, volume 27, 2014.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *Int. Conf. Learn. Repr.*, 2014.

[5] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *Found. Trends Mach. Learn.*, 12(4):307–392, 2019.

[6] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans. Inf. Theory*, 56(11): 5847–5861, 2010.

[7] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Adv. Neural Inf. Proc. Syst.*, volume 29, pages 271–279, 2016.