13 Generative Models

Goal
An overview of recent generative models. GAN, VAE, Autoregressive, Normalizing flow.

Alert 13.1: Convention
Gray boxes are not required hence can be omitted for unenthusiastic readers.
This note is likely to be updated again soon.

Definition 13.2: Density estimation
The central problem of this note is to estimate a density function (or more generally a probability measure), through a finite training sample. Formally, we are interested in estimating a probability measure \( \chi \) from a (non)parametric family \( \{ \lambda_\theta \}_{\theta \in \Theta} \). A typical approach is to minimize some statistical divergence between a noisy version \( \hat{\chi} \) and \( \lambda_\theta \):

\[
\inf_{\theta \in \Theta} D(\hat{\chi}, \lambda_\theta).
\]

However, the minimization problem above may not always be easy to solve, and alternative (indirect) strategies have been developed.

Definition 13.3: Exponential Family Distribution
The exponential family distributions have the following density form (here we also condition on some covariate \( x \), for later use):

\[
p(y|x) = h(y) \exp \left( \langle \eta(x), T(y) \rangle - A(\eta(x)) \right),
\]

where \( T(y) \) is the sufficient statistics, \( \eta \) is the natural parameter, \( A \) is the log-partition function, and \( h \) represents the base measure. For example:

- Gaussian: \( \eta = (\mu, -\frac{1}{2\sigma^2}) \), \( T(y) = (y, y^2) \) and \( A(\eta) = \frac{\|\mu\|^2}{2\sigma^2} + \log |\sigma| \), where we use the same variance parameter \( \sigma \) for simplicity.
- Bernoulli: \( \eta_k = \log \frac{p_k}{1-p_k} \), \( T(y) = y \) and \( A(\eta) = \sum_{k=1}^c \log(1 + \exp(\eta_k)) \), where we assume \( y_k \)'s are independent.
- Categorical: \( \eta = [\log \frac{p_1}{p_c}, \ldots, \log \frac{p_{c-1}}{p_c}, 0] \), \( T(y) = [1_{y=1}, \cdots, 1_{y=c}] \) and \( A(\eta) = -\log p_c \), where of course \( \sum_k p_k = 1 \).

Exercise 13.4: Mean parameter and moments
Prove that for the exponential family distribution,

\[
\nabla A(\eta) = E(T) \\
\n\nabla^2 A(\eta) = E(TT^\top) - E(T)E(T)^\top.
\]

Exercise 13.5: Marginal and Conditional of Exponential Family
Let \( p(x, y) \) be a joint distribution from the exponential family. Prove the following:

- The marginal \( p(x) \) need not be from the exponential family.
• The conditional \( p(y|x) \) is also from the exponential family.

**Exercise 13.6: Exponential family approximation under KL**

Let \( p(x) \) be an arbitrary distribution and \( q(x) \) from the exponential family with sufficient statistics \( T \) and log-normalization function \( A \). Then,

\[
\eta^* = \arg\min_{\eta} \text{KL}(p, q)
\]

is given by moment-matching:

\[
E_p T(X) = E_q T(X) = \nabla A(\eta), \quad \text{i.e.,} \quad \eta = \nabla A^*(E_p T(X)).
\]

**Definition 13.7: Conditional Density Networks (Bishop 1994)**

Bishop (1994) was one of the first to use neural networks for modeling probability density functions, where the goal is to predict a target vector \( y \in \mathbb{R}^c \) given some input vector \( x \in \mathbb{R}^d \). We model the conditional density using exponential family, with (some) natural parameters \( \eta(x) = f(x; w) \) parameterized by a neural net with weights \( w \), which is then tuned by maximizing the log-likelihood:

\[
\max_w \prod_{i=1}^n p(y_i|x_i)p(x_i) = \max_w \frac{1}{n} \sum_{i=1}^n (f(x_i; w), T(y_i)) - A(f(x_i; w)).
\]

If our network \( f \) ignores the input \( x \), then at optimality we have

\[
f(x; w^*) = (\nabla A)^{-1}(\bar{T}), \quad \text{where} \quad \bar{T} = \frac{1}{n} \sum_{i=1}^n T(y_i), \quad \text{hence} \quad (\nabla A)(f(x; w^*)) \to E(T(y)) \text{ as } n \to \infty.
\]

On the other hand, if our network \( f \) is sufficiently expressive, then at optimality we have

\[
f(x_i; w^*) = (\nabla A)^{-1}(T(y_i)), \quad \text{hence} \quad (\nabla A)(f(x; w^*)) \to E(T(y)|x) \text{ as } n \to \infty.
\]

The interested minds can try to unwrap the above general procedure for the Gaussian and Bernoulli distributions, where our network outputs converge to the (“best” approximation of the) conditional expectation \( E(y|x) \).


**Definition 13.8: Mixture Density Networks (Bishop 1994)**

Modeling the conditional density as a member in the exponential family is analytically appealing, but it may not be always realistic. A natural extension is to use mixtures (as we saw in tensor networks):

\[
p(y|x) = \sum_{j=1}^r \lambda_j(x)p_j(y|x), \quad \text{where} \quad p_j(y|x) := h(y) \exp \{ \langle \eta_j(x), T(y) \rangle - A(\eta_j(x)) \}, \quad (13.1)
\]

where the mixing coefficients \( \lambda_j(x) \geq 0 \) and \( \sum_j \lambda_j(x) = 1 \), i.e. \( \lambda \in \Delta_{r-1} \). Typically, we choose each mixing component \( p_j \) from the same exponential family, i.e. \( h, T, A \) does not change w.r.t. \( j \). The mixing coefficients \( \lambda(x) \) and natural parameters \( \eta(x) \) are modeled as the output of some neural network \( f(x; w) \), with \( x \) as input and \( w \) as the tuning weights. Since mixtures can approximate any density function, by using a large \( r \) and sufficiently expressive network \( g(x; w) \), Bishop (1994, page 7) claimed that the mixture density network can approximate any conditional density function.

To be concrete, for the Gaussian case, we partition the network output into three parts \( g(x; w) = \)
Consider the functions \( g_1, g_2, g_3 \), where we set
\[
\lambda(x) = \text{softmax}(g_1(x; w)) \\
\mu(x) = g_2(x; w) \\
\sigma(x) = \exp(g_3(x; w)).
\]

As usual we minimize the negative log-likelihood (NLL):
\[
\min_w - \frac{1}{n} \sum_{i=1}^{n} \log \left( \sum_{j=1}^{r} \lambda_j(x_i) p_j(y_i | x_i) \right).
\]

The derivative w.r.t. \( g_1 \) can be calculated straightforwardly:
\[
\frac{\partial \text{NLL}}{\partial \lambda_j} = - \frac{1}{n} \sum_{i=1}^{n} \frac{p_j(y_i | x_i)}{p(y_i | x_i)}, \quad \frac{\partial \lambda}{\partial g_1} = \text{diag}(\lambda) - \lambda \lambda^\top.
\]

If we introduce
\[
\pi_j(y, x) = \frac{\lambda_j(x) p_j(y | x)}{p(y | x)},
\]
then we have the succinct formula
\[
\frac{\partial \text{NLL}}{\partial g_1} = - \frac{1}{n} \sum_{i=1}^{n} \left[ \pi(y_i, x_i) - \lambda(x_i) \right] =: \hat{E}(\lambda - \pi).
\]

Similarly, we can derive the derivative w.r.t. \( \eta_j \):
\[
\frac{\partial \text{NLL}}{\partial \eta_j} = \frac{1}{n} \sum_{i=1}^{n} \pi_j(y_i, x_i) [E_{p_j}(T(y)) - T(y_i)] =: \hat{E}_{\pi_j} [E_{p_j}(T(y)) - T(y)],
\]
from which we can easily obtain the derivative w.r.t. \( g_2 \) and \( g_3 \). Again, the interested minds can specialize the above formula for the Gaussian case. As \( n \) goes to infinity (and assuming our network is sufficiently expressive), equating the gradients to 0 gives the equilibrium:
\[
\lambda(x) = E[\pi(y, x) | x] \\
E_{p_j}(T(y) | x) = \frac{E[\pi_j(y, x) T(y) | x]}{E[\pi_j(y, x) | x]}.
\]


Remark 13.9: Advantage of mixture density networks (MDN)

Once we obtain an estimate of the mixture density network (MDN) in (13.1), we can compute many statistics from it. For instance, we can make a prediction using its mean and quantify our prediction uncertainty using its variance (as in conventional Bayesian statistics). We can also compute its mode as the most probable outcome. In particular, assuming the mixing components separate sufficiently well, the maximum in \( y \) can be approximated as
\[
\max_j \lambda_j(x) \max_y p_j(y | x).
\]

Bishop (1994) gave a compelling argument for mixture density networks: when we try to solve an inverse problem, where the output for a given input may have multiple possibilities, then using a standard neural network amounts to using the average of all possibilities, which itself may not be a sensible output. On the other hand, a mixture density network can capture the different possibilities of the output through multiple
mixture components. Moreover, since the mixing coefficient $\lambda(x)$ depends on the input $x$, MDN is adaptive: depending on the input $x$, the network can decide to display $1, 2, \ldots, r$ mixture components, corresponding to the number of possibilities of the output. See (Bishop 1994, §5) for an illuminating example.


Example 13.10: Simulating distributions

Suppose we want to simulate a Gaussian distribution with mean $u$ and covariance $S$. The typical approach is to first simulate the standard Gaussian distribution (with zero mean and identity covariance) and then perform transformation:

$$\text{If } Z \sim \mathcal{N}(0, I), \text{ then } X = T(Z) := u + S^{1/2}Z \sim \mathcal{N}(u, S).$$

Similarly, if we want to simulate a $\chi^2$ distribution with zero mean and degree $d$:

$$\text{If } Z \sim \mathcal{N}(0, I_d), \text{ then } X = T(Z) := \sum_{j=1}^{d} Z_j^2 \sim \chi^2(d).$$

In fact, we can simulate any distribution $F$ on $\mathbb{R}$ as follows:

$$\text{If } Z \sim \mathcal{N}(0, 1), \text{ then } X = T(Z) := F^{-1}({\Phi(Z)}) \sim F, \text{ where } F^{-1}(t) = \min\{x : F(x) \geq t\},$$

and $\Phi$ is the cumulative distribution function of standard normal.

Theorem 13.11: Transforming to any probability measure

Let $\mu$ be a diffuse (Borel) probability measure on a polish space $Z$ and similarly $\nu$ be any (Borel) probability measure on another polish space $X$. Then, there exist (measurable) maps $T : Z \to X$ such that

$$\text{If } Z \sim \mu, \text{ then } X := T(Z) \sim \nu.$$  

Recall that a (Borel) probability measure is diffuse iff any single point has measure 0. For less mathematical readers, think of $Z = \mathbb{R}^p$, $X = \mathbb{R}^d$, $\mu$ and $\nu$ as probability distributions on the respective Euclidean spaces.

Definition 13.12: (Strictly) convex function

A real-valued function $f : C \to \mathbb{R}$ is convex if

- its domain $C := \text{dom}(f)$ is a convex set, and
- for all $\lambda \in (0, 1)$, for all $x, z \in C$:
  $$f(\lambda x + (1 - \lambda)z) \leq \lambda f(x) + (1 - \lambda) f(z).$$

We call $f$ strictly convex if the above inequality is strict whenever $x \neq z$.

If $f$ is twice differentiable (and $C$ is open), then $f$ is convex iff its Hessian $\nabla^2 f$ is always positive semidefinite. If $\nabla^2 f$ is always positive definite, then $f$ is strictly convex, whereas the converse is not true (why?).

Exercise 13.13: Calculus for convexity

Prove the following:

- If $f$ and $g$ are convex, then for any $\alpha, \beta \geq 0$, $\alpha f + \beta g$ is also convex;
- If $f_t$ is convex for all $t \in T$, then $f := \sup_{t \in T} f_t$ is convex; (what about infimum?)
• If \( f : C \rightarrow \mathbb{R} \) is convex, then the perspective function \( g(x, t) := tf(x/t) \) is convex on \( C \times \mathbb{R}^+ \);

• If \( f(x, z) \) is convex, then for any \( x, f_x := f(x, \cdot) \) is convex and similarly for \( f_z := f(\cdot, z) \). (what about the converse?)

**Definition 13.14: Fenchel conjugate function**

For any real-valued function \( f : \text{dom} \rightarrow \mathbb{R} \) we define its Fenchel conjugate function as:

\[
f^*(x^*) := \sup_{x \in \text{dom} f} \langle x, x^* \rangle - f(x).
\]

According to one of the above exercises, \( f^* \) is always a convex function.

If \( \text{dom} f \) is nonempty and closed, and \( f \) is continuous, then

\[
f^{**} := (f^*)^* = f.
\]

**Definition 13.15: \( f \)-divergence**

Let \( f : \mathbb{R}_+ \rightarrow \mathbb{R}_+ \) be a strictly convex function with \( f(1) = 0 \). We define the following \( f \)-divergence to measure the closeness of two probability measures \( \mu \) and \( \nu \):

\[
D_f(\nu, \mu) = \begin{cases} 
\int f\left(\frac{d\nu}{d\mu}\right) \, d\mu, & \text{if } \nu \ll \mu, \text{ i.e. } \nu \text{ is absolutely continuous w.r.t. } \mu, \\
\infty, & \text{otherwise}
\end{cases}
\]

When there exists a third probability measure \( \lambda \) so that \( \nu \ll \lambda \) and \( \mu \ll \lambda \), we can identify \( \nu \) and \( \mu \) with the densities \( p := \frac{d\nu}{d\lambda} \) and \( q := \frac{d\mu}{d\lambda} \), respectively, in which case the \( f \)-divergence simplifies to:

\[
D_f(p, q) := D_f(\nu, \mu) = \int qf(p/q) \, d\lambda.
\]

For two random variables \( Z \sim \mu \) and \( X \sim \nu \), we sometimes abuse the notation to mean

\[
D_f(X, Z) := D_f(\nu, \mu).
\]

**Exercise 13.16: Properties of \( f \)-divergence**

Prove the following:

• \( D_f(\nu, \mu) \geq 0 \), with 0 attained iff \( \nu = \mu \).

• If \( \mu \ll \nu \) and \( \nu \ll \mu \), then \( D_f(\nu, \mu) = D_g(\mu, \nu) \), where \( g(t) = t \cdot f(1/t) \). (Is \( g \) convex?)

**Definition 13.17: \( f \)-GAN (Nowozin et al. 2016)**

Following Nowozin et al. (2016), we summarize the main idea of \( f \)-GANs as follows:

• **Generator:** Let \( \mu \) be a fixed reference probability measure on space \( Z \) (usually the standard normal distribution) and \( Z \sim \mu \). Let \( \nu \) be any target probability measure on space \( X \) and \( X \sim \nu \). Let \( T \subseteq \{T : Z \rightarrow X\} \) be a class of transformations. According to Theorem 13.11 we know there exist transformations \( T \) (which may or may not be in our class \( T \)) so that \( T(Z) \sim X \sim \nu \). Our goal is to approximate such transformations \( T \) using our class \( T \).
Loss: We use the $f$-divergence to measure the closeness between the target $X$ and the transformed reference $T(Z)$:

$$\inf_{T \in \mathcal{T}} D_f(X, T(Z)).$$

In fact, any loss function that allows us to distinguish two probability measures can be used, as we will see shortly. However, we face an additional difficulty here: the densities of $X$ and $T(Z)$ (w.r.t. a third probability measure $\lambda$) are not known to us (especially the former) so we cannot naively evaluate the $f$-divergence in (13.2).

**Discriminator:** A simple variational reformulation will resolve the above difficulty! Indeed,

$$D_f(X \| T(Z)) = \int f \left( \frac{d\nu}{d\tau}(x) \right) d\tau(x) \quad (T(Z) \sim \tau)$$

$$= \int \sup_{s \in \text{dom}(f^*)} \left[ \frac{d\nu}{d\tau}(x) - f^*(s) \right] d\tau(x) \quad (f^{**} = f)$$

$$\geq \sup_{s \in \mathcal{S}} \int [S(x) \frac{d\nu}{d\tau}(x) - f^*(S(x))] d\tau(x) \quad (\mathcal{S} \subseteq \{ S : \mathcal{X} \to \text{dom}(f^*) \})$$

$$= \sup_{s \in \mathcal{S}} E[S(X)] - E[f^*(S(T(Z)))] \quad \text{(equality if } f' \left( \frac{d\nu}{d\tau} \right) \in \mathcal{S}),$$

so our estimation problem reduces to the following minimax zero-sum game:

$$\inf_{T \in \mathcal{T}} \sup_{s \in \mathcal{S}} E[S(X)] - E[f^*(S(T(Z)))].$$

By replacing the expectations with empirical averages we can (approximately) solve the above problem with classic stochastic algorithms.

**Reparameterization:** The class of functions $\mathcal{S}$ we use to test the difference between two probability measures in the $f$-divergence must have their range contained in the domain of $f^*$. One convenient way to enforce this constraint is to set

$$\mathcal{S} = \sigma \circ \mathcal{U} := \{ \sigma \circ U : U \in \mathcal{U} \}, \quad \sigma : \mathbb{R} \to \text{dom}(f^*), \quad \mathcal{U} \subseteq \{ U : \mathcal{X} \to \mathbb{R} \},$$

where the functions $U$ are unconstrained and the domain constraint is enforced through a fixed “activation function” $\sigma$. With this choice, the final $f$-GAN problem we need to solve is:

$$\inf_{T \in \mathcal{T}} \sup_{U \in \mathcal{U}} E[\sigma \circ U(X)] - E[(f^* \circ \sigma)(U(T(Z)))]$$

Typically we choose an increasing $\sigma$ so that the composition $f^* \circ \sigma$ is “nice.” Note that the monotonicity of $\sigma$ implies the same monotonicity of the composition $f^* \circ \sigma$ (since $f^*$ is always increasing as $f$ is defined only on $\mathbb{R}_+$). In this case, we prefer to pick a test function $U$ so that $U(X)$ is large while $U(T(Z))$ is small. This choice aligns with the goal to “maximize target and minimize transformed reference,” although the opposite choice would work equally well (merely a sign change).


**Remark 13.18: $f$-GAN recap**

To specify an $f$-GAN, we need:

- A reference probability measure $\mu$: should be easy to sample and typically we use standard normal;
- A class of transformations (generators): $\mathcal{T} \subseteq \{ T : \mathcal{Z} \to \mathcal{X} \};$
• An increasing convex function \( f^* : \text{dom}(f^*) \to \mathbb{R} \) with \( f^*(0) = 0 \) and \( f^*(s) \geq s \) (or equivalently an \( f \)-divergence);
• An increasing activation function \( \sigma : \mathbb{R} \to \text{dom}(f^*) \) so that \( f^* \circ \sigma \) is “nice”;
• A class of unconstrained test functions (discriminators): \( \mathcal{U} \subseteq \{ U : X \to \mathbb{R} \} \) so that \( S = \sigma \circ \mathcal{U} \).

Example 13.19: GAN (Goodfellow et al. 2014)

As an example, we show how to recover the original GAN as a special case of \( f \)-GAN. As Goodfellow et al. (2014) pointed out, GAN tries to optimize the Jensen-Shannon divergence:

\[
JS(\mu, \nu) = KL(\mu, \mu + \nu) + KL(\nu, \mu + \nu)
\]

\[
f(s) = s \log s - (s + 1) \log(s + 1)
\]

\[
f^*(t) = -\log(1 - \exp(t))
\]


Definition 13.20: Expectation-Maximization (EM) (Dempster et al. 1977)

We formulate EM under the density estimation formulation in (13.2), except that we carry out the procedure in a lifted space \( X \times Z \) where \( Z \) is the space that some latent random variable \( Z \) lives in. Importantly, we do not observe the latent variable \( Z \); it is “artificially” constructed to aid our job. We fit our model with a prescribed family of joint distributions

\[
\tilde{\lambda}_\theta(dx, dz) = \zeta_\theta(dz)D_\theta(dx|z) = \lambda_\theta(dx)e_\theta(dz|x), \quad \theta \in \Theta.
\]

In EM, we typically specify the joint distribution \( \tilde{\lambda}_\theta \) explicitly, and in a way that the posterior distribution \( e_\theta(dz|x) \) can be easily computed. Similarly, we “lift” \( \chi(dx) \) (our target of estimation) to the joint distribution

\[
\tilde{\gamma}(dx, dz) = \tilde{\chi}(dx)e(dx|x).
\]

Then, we minimize the discrepancy between the joint distributions \( \tilde{\lambda}_\theta \) and \( \tilde{\gamma} \), which is an upper bound of the discrepancy of the marginals \( KL(\tilde{\chi}, \lambda_\theta) \):

\[
\inf_{\theta \in \Theta} \inf_{e(dx|x)} KL(\tilde{\gamma}(dx, dz), \tilde{\lambda}_\theta(dx, dz)).
\]

Note that there is no restriction on \( E \) (and do not confuse it with \( e_\theta \), which is given).

The EM algorithm proceeds with alternating minimization:

• (E-step) Fix \( \theta_t \), we solve \( E_{t+1} \) by

\[
\inf_{e} KL(\tilde{\gamma}(dx, dz), \tilde{\lambda}_{\theta_t}(dx, dz)) = KL(\tilde{\chi}, \lambda_{\theta_t}) + E_{\tilde{\chi}}KL(E, e_{\theta_t}),
\]

which leads to the “closed-form” solution:

\[
E_{t+1} = e_{\theta_t}.
\]
\[ \inf_{\theta} \text{KL}(\tilde{\gamma}_{t+1}(dx, dz), \tilde{\lambda}_\theta(dx, dz)) = \text{KL}(\tilde{\chi}, \lambda) + E_{\tilde{\chi}} \text{KL}(E_{t+1}(\mathcal{E}), \mathcal{E}_\theta). \]

For the generalized EM algorithm, we need only decrease the above (joint) KL divergence if finding a (local) minima is expensive. It may be counter-intuitive that minimizing the sum of two terms above can be easier than minimizing the first likelihood term only!

Obviously, the EM algorithm monotonically decreases our (joint) KL divergence \[ \text{KL}(\tilde{\gamma}, \tilde{\lambda}_\theta). \] Moreover, due to construction the EM algorithm also descends the likelihood term:

\[
\text{KL}(\hat{\chi}, \lambda)_{t+1} \leq \text{KL}(\tilde{\gamma}_{t+1}, \tilde{\lambda}_{\theta_{t+1}}) \leq \text{KL}(\tilde{\gamma}_{t+1}, \tilde{\lambda}_{\theta_t}) = \text{KL}(\hat{\chi}, \lambda_{t}).
\]


**Exercise 13.21: EM for exponential family**

Prove that the M-step of EM simplifies to the following, if we assume the joint distribution \( \tilde{\lambda}_\theta \) is from the exponential family with natural parameter \( \theta \), sufficient statistics \( T \) and log-normalization function \( A \):

\[
\theta_{t+1} = \nabla A'(E_{t+1}(T(X))).
\]

**Definition 13.22: Variational Inference (e.g. Blei et al. 2017)**

Variational inference has been a popular tool for Bayesian analysis. We are given explicitly a statistical model

\[
\tilde{\lambda}(dx, dz) = \lambda(dx)\mathcal{E}(dz|x) = \zeta(dz)\mathcal{D}(dx|z).
\]

Unlike in EM, we assume the posterior distribution \( \mathcal{E}(dz|x) \) (or the marginal distribution, a.k.a. evidence, \( \lambda(dx) \)) cannot be computed efficiently, and we aim to find a tractable approximation from a parametric family \( \{\mathcal{E}_\phi(dz|x) : \phi \in \Phi\} \):

\[
\inf_{\phi \in \Phi} \text{KL}(\tilde{\gamma}_\phi, \tilde{\lambda}),
\]

where as usual

\[
\tilde{\gamma}_\phi(dx, dz) := \hat{\chi}(dx)\mathcal{E}_\phi(dz|x).
\]

Importantly, here \( \hat{\chi} \) is the joint empirical distribution of \((X_1, \ldots, X_n)\) based on a single sample \((x_1, \ldots, x_n)\). In contrast, in EM \( \hat{\chi} \) is the empirical distribution of \( X \) based on \( n \) samples \([x_1, \ldots, x_n]\).


**Definition 13.23: Evidence Lower BOund (ELBO)**

Using the equality

\[
\text{KL}(\tilde{\gamma}_\phi, \tilde{\lambda}) = \text{KL}(\hat{\chi}, \lambda) + E_{\hat{\chi}} \text{KL}(\mathcal{E}_\phi(dz|x), \mathcal{E}(dz|x))
\]

Yaoliang Yu 83

–Version 0.0–November 12, 2019–
and expanding from the definition of the left-hand side, we have
\[
\mathbb{E}_\hat{x} \log \lambda(dx) = \mathbb{E}_\hat{x}\KL(\mathcal{E}_\phi(dz|x), \mathcal{E}(dz|x)) + \mathbb{E}_\hat{x}\log d\hat{x}(dx) - \KL(\hat{\gamma}_\phi, \hat{\lambda}) \\
lower[5pt]\text{(expected log evidence)}
\]
\[
= \mathbb{E}_\hat{x}\KL(\mathcal{E}_\phi(dz|x), \mathcal{E}(dz|x)) + \mathbb{E}_\gamma\log d\mathcal{D}(dx|x) - \mathbb{E}_x\KL(\mathcal{E}_\phi(dz|x), \zeta(dz)) \\
\text{ELBO}
\]
\[
\geq 0
\]
leading to the "closed-form" solution (cf. Exercise 13.25)
\[
\log d\mathcal{E}_j^*(dz_j|x) \propto \mathbb{E}_{\hat{\gamma}_j} \log \mathbb{E}(dz_j|x, z_{\not\equiv j}) \propto \mathbb{E}_{\hat{\gamma}_j} \log d\hat{\lambda}(dx, dz), \quad \text{where } \mathcal{E}_{\hat{\gamma}_j}(dz_j|x) := \prod_{k \neq j} \mathcal{E}_k^*(dz_k|x),
\]
(13.4)
followed by the projection:
\[
\inf_{\phi_j \in \Phi_j} \KL(\mathcal{E}_j^*, \mathcal{E}_j)
\]

We immediately see that maximizing the ELBO is equivalent as minimizing the KL divergence between "lifted" joint densities.

**Definition 13.24: Mean Field Approximation (Peterson and Anderson 1987)**

To solve the variational minimization (13.3) tractably, we usually restrict the variational approximation \(\mathcal{E}_\phi\) to a factorized family:
\[
\mathcal{E}_\phi(dz|x) = \prod_{j=1}^m \mathcal{E}_j^{\phi_j}(dz_j|x),
\]
which is known as the mean field approximation. We can then rewrite the joint KL divergence:
\[
\inf_{\phi_1 \in \Phi_1} \cdots \inf_{\phi_m \in \Phi_m} \KL(\hat{\gamma}_\phi, \hat{\lambda}).
\]
Again, we employ alternating minimization: for the \(j\)-th factor \(\mathcal{E}_j^{\phi_j}\), we fix all other factors \(\mathcal{E}_k^{\phi_k}, k \neq j\), and solve
\[
\inf_{\phi_j \in \Phi_j} \KL(\hat{\gamma}_\phi, \hat{\lambda}) = \KL(\hat{\gamma}_\phi, \hat{\lambda}) + \mathbb{E}_\hat{\gamma}_j \KL(\hat{\gamma}_j, \mathcal{E}_j)(dz_j|x, \mathcal{E}(dz_j|x)) + \mathbb{E}_{\hat{\gamma}_j} \KL(\mathcal{E}_j^{\phi_j}(dz_j|x), \mathcal{E}(dz_j|x, z_{\not\equiv j})),
\]
leading to the "closed-form" solution (cf. Exercise 13.25)
\[
\log d\mathcal{E}_j^*(dz_j|x) \propto \mathbb{E}_{\hat{\gamma}_j} \log \mathbb{E}(dz_j|x, z_{\not\equiv j}) \propto \mathbb{E}_{\hat{\gamma}_j} \log d\hat{\lambda}(dx, dz), \quad \text{where } \mathcal{E}_{\hat{\gamma}_j}(dz_j|x) := \prod_{k \neq j} \mathcal{E}_k^*(dz_k|x),
\]
(13.4)
followed by the projection:
\[
\inf_{\phi_j \in \Phi_j} \KL(\mathcal{E}_j^*, \mathcal{E}_j)
\]

In practice, the variational family \(\mathcal{E}_\phi\) is chosen so that the "closed-form" solution \(\mathcal{E}_j^*\) in (13.4) is easy to compute.


**Exercise 13.25: KL under Factorization**

Prove that the solution of the following KL projection
\[
\arg\min_{q_1(\cdot)} \KL(q_1(z)q_2(x), p(z, x)]
\]
is given by
\[
q_1^*(z) \propto \exp[\mathbb{E}_{q_2(x)} \log p(z|x)] \propto \exp[\mathbb{E}_{q_2(x)} \log p(z, x)], \quad \text{i.e.}
\]
\[
\log q_1^*(z) \propto \mathbb{E}_{q_2(x)} \log p(z|x) \propto \mathbb{E}_{q_2(x)} \log p(z, x).
\]
Exercise 13.26: Variational Inference for Exponential Family

Suppose each (complete) conditional $d\mathcal{E}(dz_j|x,z_{\setminus j}) = h(z_j) \exp[\eta_j(z_{\setminus j}, x)^\top T_j(z_j) - A_j(\eta_j(z_{\setminus j}, x))]$ belongs to some exponential family (in particular, if the joint $\lambda$ is so), then the closed-form solution in (13.4) is:

$$d\mathcal{E}_j^*(dz_j|x) = h(z_j) \exp[[E_{z_{\setminus j}} \eta_j(z_{\setminus j}, x)]^\top T_j(z_j) - A_j[E_{z_{\setminus j}} \eta_j(z_{\setminus j}, x)]]$$

which has the same form as the (complete) conditional. Thus, if we choose the variational family $\mathcal{E}_j^{\phi_j}$ also from the same family as the (complete) conditional (with $\phi_j$ being the natural parameter), then we may simply perform the update:

$$\phi_j \leftarrow E_{E_{z_{\setminus j}} \eta_j(z_{\setminus j}, x)}$$

Definition 13.27: Auto-Encoding Variational Bayes (AEVB, Kingma and Welling 2014)

The main idea behind AEVB can be summarized as follows: We choose an arbitrary latent marginal distribution $\zeta(dz)$ and we fix the target data distribution $\chi(dx)$. Now we choose a probabilistic decoder $\mathcal{D}_\theta(dx|z)$ and a probabilistic encoder $\mathcal{E}_\phi(dz|x)$ so that the distance between the (joint) probability measures $\zeta \mathcal{D}_\theta$ and $\chi \mathcal{E}_\phi$ on $X \times Z$ is minimized. If we achieve the absolute minimum 0, then we know the marginals of both $\zeta \mathcal{D}_\theta$ and $\chi \mathcal{E}_\phi$ are $\zeta$ and $\chi$.

In more details, let

$$\tilde{\lambda}_\theta(dx,dz) = \zeta(dz)\mathcal{D}_\theta(dx|z) = \lambda_\theta(dx)\mathcal{E}_\phi(dz|x),$$

where $\mathcal{E}_\phi : \mathcal{B}(Z) \times X \rightarrow \mathbb{R}_+$ is a transition kernel (i.e. conditional distribution, or probabilistic encoder) and similarly $\mathcal{D}_\theta : \mathcal{B}(X) \times Z \rightarrow \mathbb{R}_+$ is a probabilistic decoder. Similarly define $\tilde{\gamma}_\phi(dx,dz) = \chi(dx)\mathcal{E}_\phi(dz|x)$. Then,

$$\text{KL}(\tilde{\gamma}_\phi, \tilde{\lambda}_\theta) = E_{\tilde{\gamma}_\phi} \log \frac{d\tilde{\gamma}_\phi}{d\tilde{\lambda}_\theta} = E_\chi \log d\chi + E_{\tilde{\gamma}_\phi} \log d\mathcal{E}_\phi(dz|x) - E_{\tilde{\gamma}_\phi} \log d\tilde{\lambda}_\theta$$

$$= \text{KL}(\chi, \lambda_\theta) + E_\chi \left[ \text{KL}(\mathcal{E}_\phi(dz|x), \mathcal{D}_\theta(dz|x)) \right].$$

Note that in AEVB, we choose the latent distribution $\zeta$ and the decoder $\mathcal{D}_\theta$, thus it is easy to compute and sample $\tilde{\lambda}_\theta$. By dropping $Z$ we can readily sample $\lambda_\theta$ as well. However, obtaining a closed-form for the marginal $\lambda_\theta$ by integrating out the latent variable $Z$ may be computationally challenging. Thus, directly minimizing the distance $\text{KL}(\chi, \lambda_\theta)$ may not be possible when both the data distribution $\chi$ and the marginal $\lambda_\theta$ are only available through samples. Instead, AEVB minimizes the upper bound $\text{KL}(\tilde{\gamma}_\phi, \tilde{\lambda}_\theta)$ on the lifted space:

$$\inf_{\phi,\theta} \text{KL}(\tilde{\gamma}_\phi, \tilde{\lambda}_\theta) = -E_{\tilde{\gamma}_\phi} \log d\tilde{\lambda}_\theta + E_{\tilde{\gamma}_\phi} \log d\mathcal{E}_\phi(dz|x) + E_\chi \log d\chi$$

(13.5)

$$= -E_{\tilde{\gamma}_\phi} \log d\mathcal{D}_\theta(dx|z) + E_\chi \left[ \text{KL}(\mathcal{E}_\phi(dz|x), \zeta(dz)) \right] + E_\chi \log d\chi.$$  

(13.6)
Unlike GANs which solve a min-max problem, AEVB solves a joint min-min problem, which might explain why in practice AEVB tends to be numerically more stable than GANs.


Remark 13.28: Push-forward reparameterization

To solve (13.5) with stochastic gradient methods, we need to compute an unbiased estimate of its gradient. The gradient w.r.t. $\theta$ can be easily computed (assuming differentiation and expectation interchanges):

$$E_{\tilde{\gamma}^\phi} \nabla_\theta \log \mathcal{D}_\theta(dx|z).$$

To compute the gradient w.r.t. $\phi$ we can use the log-trick:

$$\nabla_\phi E_{\tilde{\gamma}^\phi} f^\phi(X, Z) = E_{\tilde{\gamma}^\phi} [f^\phi \nabla_\phi \log(f^\phi d_{\tilde{\gamma}^\phi})].$$

A different strategy is to use push-forward to reparameterize the encoder

$$\mathcal{E}_\phi(dz|x) = [T^\phi(x)]_{\#}\mu, \text{ i.e. } Z = T^\phi(X, U) \text{ where } U \sim \mu,$$

and $\mu$ is chosen by convenience (e.g. standard normal or uniform). Thus, we can rewrite

$$E_{\tilde{\gamma}^\phi} f^\phi(X, Z) = E_{\chi \otimes \mu} f^\phi(X, T^\phi(X, U)),$$

which for the AEVB objective (13.5) corresponds to

$$f^\phi(x, z) = -\log d\mathcal{L}_\theta(dx, dz) + \log d\mathcal{E}_\phi(dz|x).$$

When the regularization term in (13.6) can be computed analytically, we will use the objective (13.6) instead, in which case we will set

$$f(x, z) = -\log d\mathcal{D}_\theta(dx|z),$$

which is now independent of the parameter $\phi$. 