Variational Inference & Variational Autoencoders

Casey Meehan & Mary Anne Smart

CSE 254

May 2019

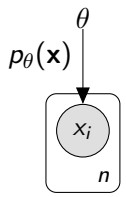
Overview

1. Variational Inference: A Review for Statisticians (Blei, Kucukelbir, & McAuliffe)
   - When and why use VI?
   - The Evidence Lower Bound (ELBO)
   - Mean Field Variational Inference
   - Comparison with older methods
2. Auto-Encoding Variational Bayes (Kingma & Welling)

Bayesian Modeling

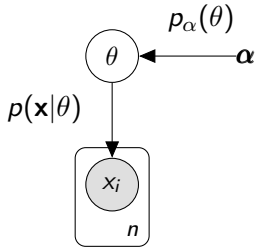
- Define a probabilistic model for your data in which the parameters are random variables
- Consider a simple coin-toss model, $x_i \in \{0, 1\}$:

Non-Bayesian



$X_i \sim \text{Bern}(\theta)$

Bayesian



$X_i \sim \text{Bern}(\theta)$

$\theta \sim \text{Beta}(\alpha_1, \alpha_2)$

MLE, MAP, Full Bayesian

Max Likelihood Est. (MLE)

- Latent θ is a numerical parameter, not random variable.
- Choose the θ^* that best explains the data
- Maximize log likelihood:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p_\theta(x_i) = \frac{\sum_{i=1}^{n} x_i}{n}$$

- likely to overfit without enough data:
- seeing 1 sample, $x_1 = 1$ implies $\theta^* = 1$

MLE, MAP, Full Bayesian

Maximum A Posteriori Estimate (MAP)

- Latent $\theta \sim \text{Beta}(\alpha_1, \alpha_2)$ is a random variable
- Choose the θ^* with maximum probability
- Maximize the log posterior:

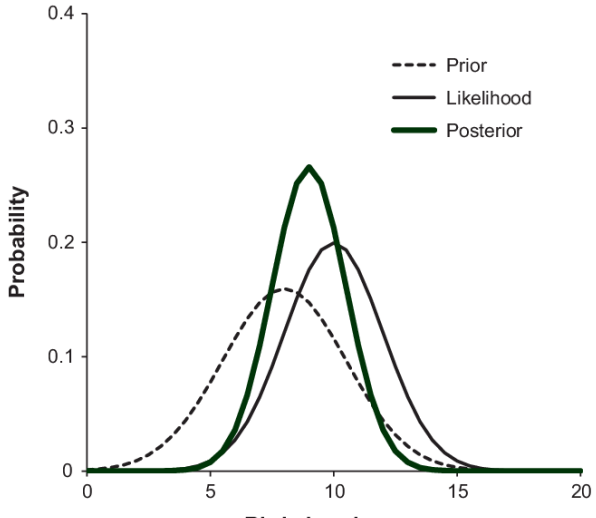
$$\begin{aligned}\theta^* &= \arg\max_\theta \Big[\log p(\theta|\mathbf{x})\Big] \\ &= \arg\max_\theta \Big[\log \frac{p(\mathbf{x}|\theta)}{p(\mathbf{x})}\Big] \\ &= \frac{\alpha_1 - 1 + \sum_{i=1}^n x_i}{n + \alpha_1 + \alpha_2 - 2}\end{aligned}$$

- less likely to overfit:
- Say, $\alpha_1 = \alpha_2 = 2$. Then seeing 1 sample, $x_1 = 1$ implies $\theta^* = \frac{2}{3}$

MLE, MAP, Full Bayesian

Full Bayesian
(goal of variational inference)

- Latent $\theta \sim \text{Beta}(\alpha_1, \alpha_2)$ is a random variable
- Find the full distribution of $\theta|\mathbf{x}, \boldsymbol{\alpha}$. Don't pick a single value



A VI Story...

1. we have a sample of n data points, $\mathbf{x} = \{x_i : i \in [1, n]\}$
2. we have latent vars, $\mathbf{z} = \{z_i : i \in [1, m]\}$,
3. we have a probabilistic model with a likelihood and a prior distribution

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

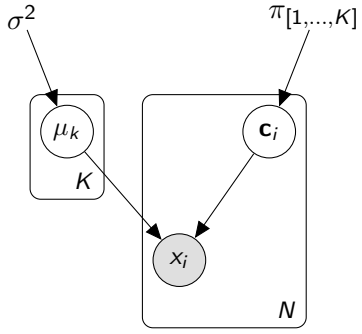
4. we want to be full Bayesian, but the posterior is intractable!

$$p(\mathbf{z}|\mathbf{x})$$

5. so, we pick a simpler class of distributions, $\mathcal{Q}$, to approximate $p(\mathbf{z}|\mathbf{x})$
6. we choose the optimal $q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$ by minimizing the reverse KL divergence from $p(\mathbf{z}|\mathbf{x})$
7. ...which is equivalent to maximizing the ELBO

Latent Variables can be Difficult to Infer

- Consider this Gaussian mixture model of N samples and K clusters:



$$\begin{aligned} \mu_k &\sim \mathcal{N}(0, \sigma^2) \\ \mathbf{c}_i &\sim \text{Cat}(\pi_1, \ldots, \pi_K) \\ x_i|\mathbf{c}_i, \boldsymbol{\mu} &\sim \mathcal{N}(\mathbf{c}_i^\top \boldsymbol{\mu}, \mathbf{I}) \end{aligned}$$

Latent Variables can be Difficult to Infer

- The posterior probability requires computing the evidence...

$$p(\mathbf{x}) = p(x_1, x_2, \ldots, x_n)$$

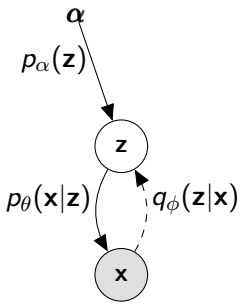
$$p(\boldsymbol{\mu}, \mathbf{c}|\mathbf{x}) = \frac{p(\boldsymbol{\mu}, \mathbf{c}, \mathbf{x})}{p(\mathbf{x})} = \frac{\prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(\mathbf{c}_i) p(\mathbf{x}_i|\mathbf{c}_i, \boldsymbol{\mu})}{p(\mathbf{x})}$$

- ...and it's a pain to compute

$$\begin{aligned} p(\mathbf{x}) &= \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(x_i|\boldsymbol{\mu}) d\boldsymbol{\mu} \\ &= \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} \sum_{\mathbf{c}_i} p(\mathbf{c}_i) p(x_i|\mathbf{c}_i, \boldsymbol{\mu}) d\boldsymbol{\mu} \implies \mathcal{O}(K^n) \end{aligned}$$

A general framework for variational inference

- Generally, let $\mathbf{z}$ be our set of latent variables, $\mathbf{x}$ our observed data
- Variational inference approximates $p(\mathbf{z}|\mathbf{x})$ with some $q(\mathbf{z}|\mathbf{x})$



- In the case of the GMM above...

$\mathbf{z} = \{\mu_1, \ldots, \mu_K, \mathbf{c}_1, \ldots, \mathbf{c}_n\}$
$\mathbf{x} = \{x_1, \ldots, x_n\}$
$\boldsymbol{\alpha} = \{\sigma^2, \pi_1, \ldots, \pi_K\}$
$\boldsymbol{\theta} = \{\varnothing\}$
$\boldsymbol{\phi}$ not defined yet

The ELBO: choosing $q_\phi(z|x)$ with KL divergence

- $q_\phi(\mathbf{z}|\mathbf{x})$ belongs to set of distributions $\mathcal{Q}$ parameterized by $\phi \in \Phi$.
- VI chooses $q_\phi(\mathbf{z}|\mathbf{x})$ by maximizing the Evidence Lower BOund
- This is akin to finding parameters $\phi \in \Phi$ that minimize the KL distance between $p(\mathbf{z}|\mathbf{x})$ and $q_\phi(\mathbf{z}|\mathbf{x})$.

$$\min_\phi \Big[D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\big) \Big]$$

Deriving the ELBO

Factor the KL divergence; let $\mathbb{E}_q = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}$

$$D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\big) \tag{1}$$

$$= \mathbb{E}_q\big[\log q_\phi(\mathbf{z}|\mathbf{x})\big] - \mathbb{E}_q\big[\log p(\mathbf{z}|\mathbf{x})\big] \tag{2}$$

$$= \mathbb{E}_q\big[\log q_\phi(\mathbf{z}|\mathbf{x})\big] - \Big(\mathbb{E}_q\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] + \mathbb{E}_q\big[\log p_\alpha(\mathbf{z})\big] - \log p(\mathbf{x})\Big) \tag{3}$$

$$= D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\alpha(\mathbf{z})\big) - \mathbb{E}_q\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] + \log p(\mathbf{x}) \tag{4}$$

So the evidence is bounded by the ELBO:

$$\begin{aligned}\log p(\mathbf{x}) &= \mathbb{E}_q\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\alpha(\mathbf{z})\big) + D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\big) \\ &\geq \mathbb{E}_q\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\alpha(\mathbf{z})\big)\end{aligned}$$

Anatomy of the ELBO?

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_q\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\alpha(\mathbf{z})\big) \\ &= \text{likelihood} \quad + \quad \text{regularizer}\end{aligned}$$

- The first term encourages $q_\phi(\mathbf{z}|\mathbf{x})$ to place mass on $\mathbf{z}$ values that explain $\mathbf{x}$ according to the likelihood, $p_\theta(\mathbf{x}|\mathbf{z})$
- The second term encourages $q_\phi(\mathbf{z}|\mathbf{x})$ to be close to the prior distribution, $p_\theta(\mathbf{z})$, effectively regularizing the approximate posterior

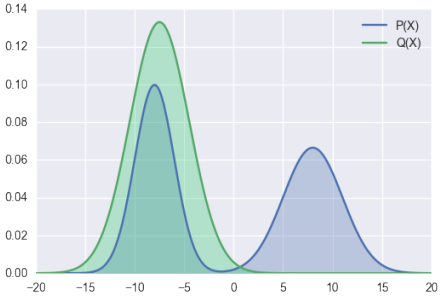
Reverse KL (used) vs. Forward KL (not used)

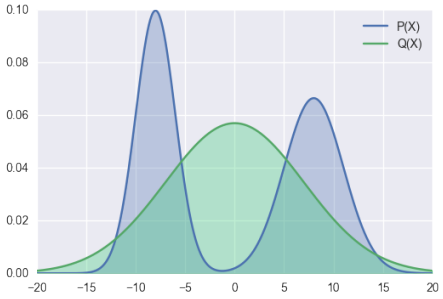
Reverse KL zero 'forces':

$$\begin{aligned}&D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\big) \\ &\quad = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\end{aligned}$$

Forward KL zero 'avoids':

$$\begin{aligned}&D_{KL}\big(p_\theta(\mathbf{z}|\mathbf{x})||q_\phi(\mathbf{z}|\mathbf{x})\big) \\ &\quad = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]\end{aligned}$$





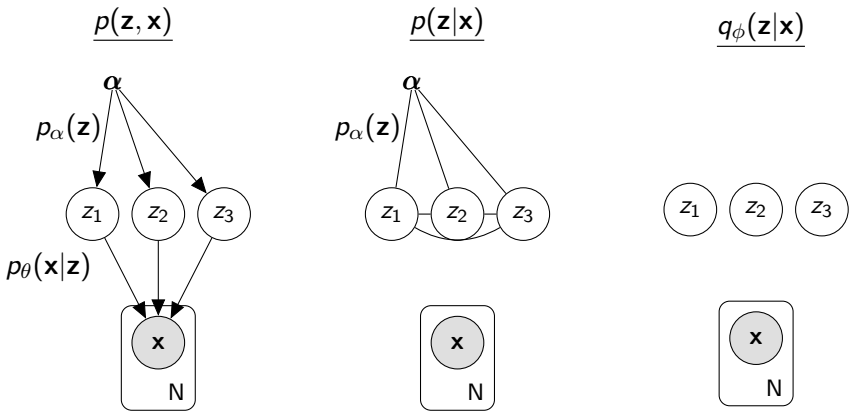
img credit: Eric Jang

Consequence: support of posterior may be under-represented

Mean Field Variational Inference

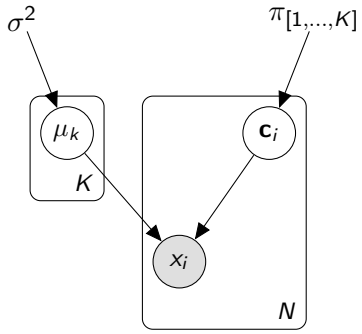
- Mean-field variational inference assumes $q_\phi(\mathbf{z}|\mathbf{x})$ completely factors:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^{M} q_i(z_i|\mathbf{x})$$



Mean Field on GMM example

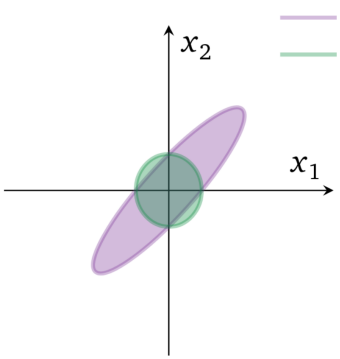
- A MF approximation for our GMM applies gaussians for the μ_k's and categoricals for the $\mathbf{c}_i$'s:



$$q_\phi(\mu_k|\mathbf{x}) = \mathcal{N}(\mu_k; m_k, s_k^2)$$
$$q_\phi(\mathbf{c}_i|\mathbf{x}) = \text{Cat}(\mathbf{c}_i; \psi_1, \ldots, \psi_k)$$

- these are optimal forms of mean-field for GMM

Mean Field captures marginal, not covariate characteristics



Exact Posterior

Mean-field Approximation

$$p(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^{M} p(z_i|z_{i-1}, \ldots, z_1, \mathbf{x})$$

vs

$$q_\phi(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^{M} q_i(z_i|\mathbf{x})$$

Coordinate Ascent Variational Inference (CAVI)

- One method to optimize our variational bound is by updating each $q_i(z_i|\mathbf{x})$ while keeping all other Z_{-i} constant.
- The update will look a lot like EM, except we're updating an approximation:

$$q_i(z_i|\mathbf{x}) \propto \exp\left\{\mathbb{E}_{q_{-i}}\left[\log p(z_i, \mathbf{Z}_{-i}, \mathbf{x})\right]\right\}$$

- Note that the expectation over q_{-i} only needs to include the markov blanket of i, not all z_i's

Coordinate Ascent Variational Inference (CAVI)

- CAVI iteratively updates each coordinate: $q_i(z_i|\mathbf{x})$

$$\text{ELBO} = \mathbb{E}_q\big[\log p(\mathbf{z},\mathbf{x})\big] - \mathbb{E}_q\big[\log q_\phi(\mathbf{z}|\mathbf{x})\big] \tag{1}$$

$$= \log p(\mathbf{x}) + \mathbb{E}_q\big[\log p(\mathbf{z}|\mathbf{x})\big] - \sum_{i=1}^{M}\mathbb{E}_{q_i}\big[\log q_i(z_i|\mathbf{x})\big] \tag{2}$$

$$= \text{const} + \sum_{i=1}^{M}\Big(\mathbb{E}_q\big[\log p(z_i|z_{1:(i-1)},\mathbf{x})\big] - \mathbb{E}_{q_i}\big[\log q_i(z_i|\mathbf{x})\big]\Big) \tag{3}$$

$$\text{ELBO}_i = \mathbb{E}_{q_i}\Big[\mathbb{E}_{q_{-i}}\big[\log p(z_i|z_{-i},\mathbf{x})\big]\Big] - \mathbb{E}_{q_i}\big[\log q_i(z_i|\mathbf{x})\big] \tag{i}$$

$$= \int q_i(z_i)\Big(\mathbb{E}_{q_{-i}}\big[\log p(z_i|z_{-i},\mathbf{x})\big] - \log q_i(z_i|\mathbf{x})\Big)dz_i \tag{ii}$$

Coordinate Ascent Variational Inference (CAVI)

$$\text{ELBO}_i = \mathbb{E}_{q_i}\Big[\mathbb{E}_{q_{-i}}\big[\log p(z_i|\mathbf{z}_{-i},\mathbf{x})\big]\Big] - \mathbb{E}_{q_i}\big[\log q_i(z_i|\mathbf{x})\big] \tag{i}$$

$$= \int q_i(z_i)\Big(\mathbb{E}_{q_{-i}}\big[\log p(z_i|\mathbf{z}_{-i},\mathbf{x})\big] - \log q_i(z_i|\mathbf{x})\Big)dz_i \tag{ii}$$

$$\frac{d}{dq_i(z_i)}\text{ELBO}_i = \mathbb{E}_{q_{-i}}\big[\log p(z_i|\mathbf{z}_{-i},\mathbf{x})\big] - \log q_i(z_i|\mathbf{x}) - 1 = 0 \tag{iii}$$

$$\tag{5}$$

solve with Lagrange multipliers:

$$q_i(z_i|\mathbf{x}) \propto \exp\Big\{\mathbb{E}_{q_{-i}}\big[\log p(z_i|\mathbf{Z}_{-i},\mathbf{x})\big]\Big\}$$
$$\propto \exp\Big\{\mathbb{E}_{q_{-i}}\big[\log p(z_i,\mathbf{Z}_{-i},\mathbf{x})\big]\Big\}$$

How Does VI Compare to Older Methods like EM?

- Doesn't Expectation Maximization (EM) compute latent variables?

$$\text{ELBO} = \mathbb{E}_q\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\alpha(\mathbf{z})\big)$$

  - EM sets $q_\phi(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$ and iterates to find fixed latent params, $\mathbf{z}$
  - VI, does not assume posterior is tractable – needed for bayesian settings, where latent params are not fixed, but distributed

How Does VI Compare to Older Methods like MCMC?

- MCMC: produce samples that asymptotically approach $p(\mathbf{z}|\mathbf{x})$
  - More accurate, with better guarantees, but computationally very expensive
- VI: run optimization to analyitically approximate $p(\mathbf{z}|\mathbf{x})$
  - VI is generally faster, and scales nicely for large datasets
  - Take advantage of stochastic and distributed optimization methods

Open Questions in VI

- VI is not as well understood as MCMC, open areas include
  - Theory to bound accuracy of VI for certain classes of models
  - Improved optimization methods for massive data
  - Developing VI approximations that work for a wide class of models
  - Improving posterior representation without overly complex family of q_ϕ's

Questions on General Variational Inference?

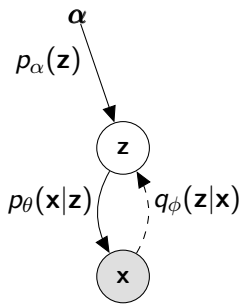
¯_(ツ)_/¯

Auto-Encoding Variational Bayes (Kingma & Welling)

- SGVB (Stochastic Gradient Variational Bayes) estimator
- AEVB (Auto-Encoding Variational Bayes) algorithm
- Variational Auto-encoder

VAE Terminology

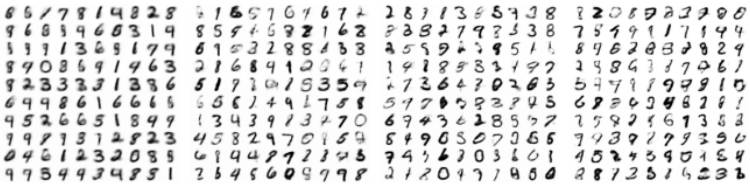
- $q_\phi(\mathbf{z}|\mathbf{x})$ is our ***encoder***. Given some $\boldsymbol{x}$ in our observed space, how is $\boldsymbol{z}$ distributed in latent space?
- $p_\theta(\mathbf{x}|\mathbf{z})$ is our ***decoder***. Given some $\boldsymbol{z}$ in latent space, how is $\boldsymbol{x}$ distributed in the original space?



Experiments

- $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are learned jointly using a neural network.
- Once we are done training the network, we can generate samples from the learned distribution $p_\theta(\mathbf{x}|\mathbf{z})$.
- Datasets
  - MNIST: training set of 60,000 28x28 pixel images of handwritten digits
  - Frey Face: almost 2000 20x28 images of Brendan Frey's face, taken from sequential video frames

MNIST Generated Samples

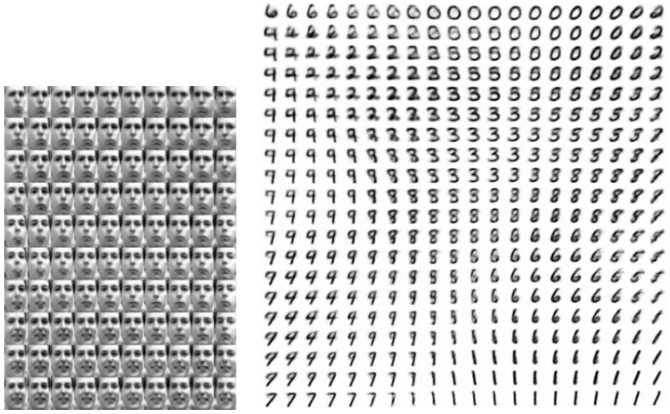


(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

Figure: (Kingma & Welling, 2013)

Visualizing Learned 2-D Manifolds



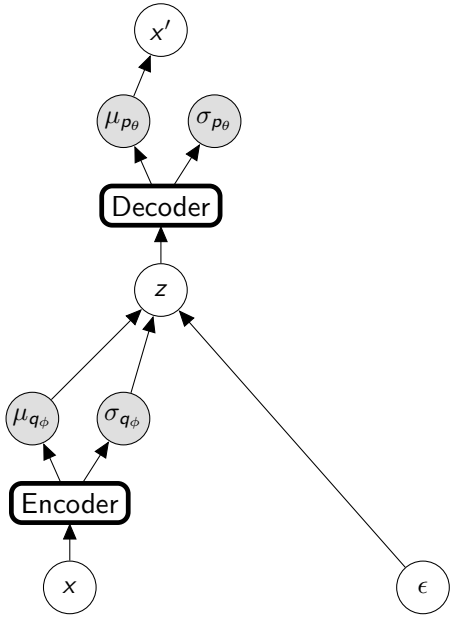
(a) Learned Frey Face manifold (b) Learned MNIST manifold

Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

VAE Example

- Let the prior over the latent variables be the centered isotropic multivariate Gaussian:
  $p_\alpha(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$.
- Let $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ be a multivariate Gaussian with a diagonal covariance structure:
  $q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I})$
  where μ and σ are outputs of our encoder network, a neural network with one hidden layer.
- Let $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ also be a multivariate Gaussian parameterized by a neural network (our decoder network) with one hidden layer.

VAE Diagram



Reparameterization Trick

- We can often reparameterize $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$ as $\boldsymbol{z} = g(\boldsymbol{x}, \boldsymbol{\epsilon})$, where g is some deterministic, differentiable function, and $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ is a noise variable.
- Gaussian example:
  We want to sample $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\boldsymbol{I})$.
  Let $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$.
  Let $g(\boldsymbol{x}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$.
  We can write: $\boldsymbol{z} = g(\boldsymbol{x}, \boldsymbol{\epsilon})$
- Instead of sampling $\boldsymbol{z}$ from q_ϕ directly, we sample $\boldsymbol{\epsilon}$ from $\mathcal{N}(0, \boldsymbol{I})$ and apply the differentiable function g.

Applying the Reparameterization Trick

$$
\begin{aligned}
& E_{q_\phi(z|x)}[f(z)] \\
&= \int q_\phi(z|x) f(z) dz \\
&= \int p(\epsilon) f(z) d\epsilon \\
&= \int p(\epsilon) f(g(x, \epsilon)) d\epsilon \\
&= E_{p(\epsilon)}[f(g(x, \epsilon))]
\end{aligned}
$$

When can we apply the reparameterization trick?

- *When q belongs to any location-scale family*

  e.g. Gaussian, Laplace

- *When q has a tractable inverse CDF*

  Let g be the inverse CDF and let $\boldsymbol{\epsilon} \sim \mathcal{U}(0, \boldsymbol{I})$.

  e.g. Exponential, Cauchy, Logistic

- *When we can express $z \sim q$ as some compositional transformation of auxiliary variables*

  e.g. Log-Normal, Dirichlet

Variational Inference & the ELBO

Recall that in VI, we select $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to maximize the ELBO (Evidence Lower BOund):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = -D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\alpha(\boldsymbol{z})) + E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})}[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})]$$

Why differentiate the ELBO?

- If we want to train the neural network from the earlier example through backprop, we need a differentiable objective function.
- Even if we don't intend to parameterize p_θ and q_ϕ with a neural network, the ability to use stochastic gradient ascent techniques to optimize the ELBO can help VI scale to large datasets.

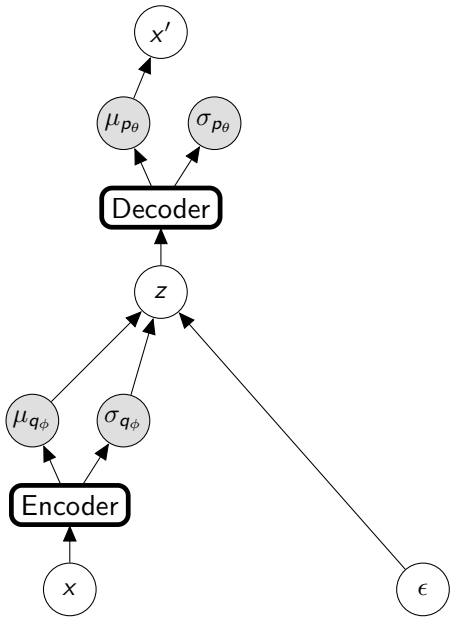
Auto-Encoding Variational Bayes Algorithm

Algorithm 1 Minibatch version of the AEVB algorithm.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\boldsymbol{X}^M \leftarrow$ Random minibatch of M datapoints
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from $p(\boldsymbol{\epsilon})$
  $\boldsymbol{g} \leftarrow \nabla_{\theta,\phi} \tilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}^M, \boldsymbol{\epsilon})$
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients (e.g. SGD or Adagrad)
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

$\tilde{\mathcal{L}}^M(\theta, \phi; X^M, \epsilon) = \frac{N}{M} \sum_{i=1}^{N} \tilde{\mathcal{L}}(\theta, \phi; x^{(i)})$
where $\tilde{\mathcal{L}}$ is the differentiable estimator of the ELBO that we will derive next. ($\tilde{\mathcal{L}}^M$ is the minibatch version.)

VAE Diagram



Stochastic Gradient Variational Bayes Estimator

- ELBO (Evidence Lower BOund)

  $\mathcal{L}(\boldsymbol{\theta}, \phi; \boldsymbol{x}^{(i)}) = -D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\alpha(\boldsymbol{z})) + E_{q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})}[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})]$
- How to differentiate with respect to ϕ and θ?
- In practice, we can often obtain closed form expression for the first term (KL-divergence).
- For the second term we need to do some kind of estimation by sampling.

Deriving the SGVB Estimator

$E_{q_\phi(z|x^{(i)})}[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})] \approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)})$

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

This gives us our SGVB estimator:

$\mathcal{L}(\boldsymbol{\theta}, \phi; \boldsymbol{x}^{(i)}) \approx -D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\alpha(\boldsymbol{z})) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)})$

Back to Our Original Example

- Let the prior over the latent variables be the centered isotropic multivariate Gaussian:
  $p_\alpha(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$.
- Let $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ be a multivariate Gaussian with a diagonal covariance structure:
  $q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I})$
  where μ and σ are outputs of our encoder network, a neural network with one hidden layer.
- Claim:

  $-D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\alpha(\boldsymbol{z}))$

  $= \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\boldsymbol{\sigma}_j^{(i)})^2) - (\boldsymbol{\mu}_j^{(i)})^2 - (\boldsymbol{\sigma}_j^{(i)})^2)$

Deriving an Expression for the KL-Divergence

$-D_{KL}(Q||P) = -\int q(z)\log\frac{q(z)}{p(z)}dz$

$= \int(q(z)\log p(z)dz - \int q(z)\log q(z)dz$

$= \int \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)\log\mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I}) - \int \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)\log\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

$= [-\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2)] - [-\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(1 + \log\sigma_j^2)]$

$= \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2)$

Note: J denotes the dimensionality of z.

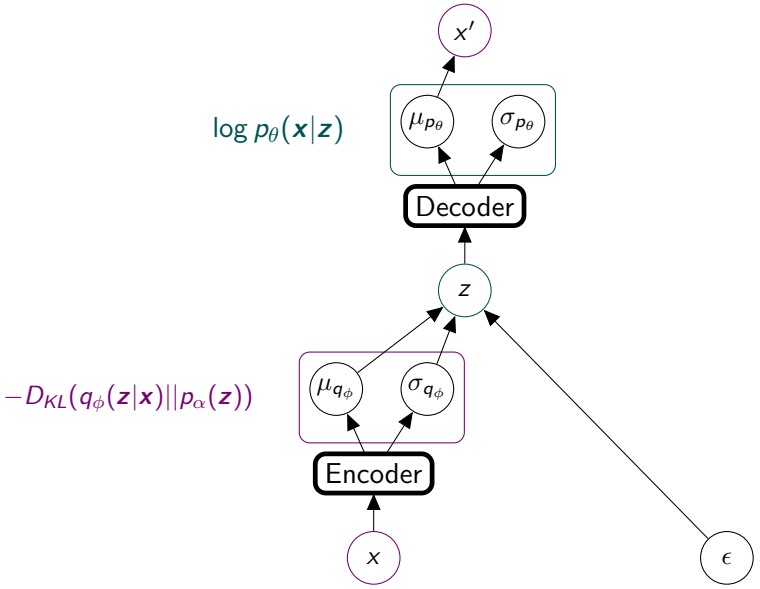
Final Result

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)})$$

$$\approx \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2\right) + \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\boldsymbol{x}^{(i)} | \boldsymbol{z}^{(i,l)})$$

where $\boldsymbol{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ and $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \boldsymbol{I})$

VAE Diagram Revisited



See Also

- These are the specific papers we talked about:

  Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes. In *The 2nd International Conference on Learning Representations (ICLR)*, 2013.

  David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: a review for statisticians. *arXiv:1601.00670*, 2016.

- This is a helpful tutorial that we also consulted:

  C. Doersch. Tutorial on variational autoencoders. *arXiv:1606.05908*, 2016.

- Many variations of the standard VAE exist. For example:
  - Conditional VAEs
  - VAE-GANs