# 1 Introduction

## 1.1 Problem Statement

In machine learning, particularly for high dimensional applications such as image analysis, it is often desirable to build generative models, so that we can represent the data in lower dimensions via representation learning, and generate new data similar to the examples in our dataset. Assume our dataset $X = \{x^{(i)}\}_{i=1}^{N}$ is $N$ i.i.d. samples of random variables $x$. Also assume $x$ can be generated by a stochastic process from a latent continuous random variable $z$. These models involve mapping the dataset x to lower dimensional latent prior $z$ (e.g. $z \sim N(\mu, \Sigma)$) then simulating from the prior $p_\theta(z)$ to generate new data through a decoder $p_\theta(x|z)$; $\theta$ represents the parameters of the distribution, typically by a neural network. In this particular field, there are three main problems to solve:

1. Estimation of $\theta$, so that we can actually generate new data $x$
2. Evaluation of the posterior density $p(z|x) = \frac{p(z)p(x|z)}{p(x)} = \frac{p(x|z)p(z)}{\int_z p(x,z)dz}$, so we can encode our data $x$ in an efficient representation $z$
3. Marginal inference of $x$ ie. evaluating $p(x)$, so it can be used as a prior for other tasks

However, with high dimensional data, calculating $p(x)$ by marginalising over z is intractable, meaning typical methods such as MCMC or expectation maximization can be computationally expensive or time-consuming. Also, with a large dataset, batch optimization such as Monte Carlo Expectation Maximization is too slow. To circumvent this problem, we apply implicit variational inference, estimating the true posterior $p_\theta(z|x)$ with another implicit distribution $q_\phi(z|x)$, a stochastic generator/encoder parametrized by another neural network: given data $x$ it outputs a distribution representing $z$ that could have generated the $x$. We then want to minimize the KL divergence between the two distributions:

$$KL(q(z|x)||p(z|x)) = E_{q(z|x)} \log \frac{q(z|x)}{p(z|x)}$$

Applying Bayes' law to $p(z|x)$, the KL divergence becomes

$$KL(q(z|x)||p(z|x)) = E_{q(z|x)} \left[ \log \frac{q(z|x)}{p(x|z)p(z)} + \log p(x) \right]$$

Again, $p(x)$ is intractable, so we cannot simply calculate the KL divergence. However, since $\log p(x)$ is constant, we can rearrange the expression to form:

$$\log p(x) = KL(q(z|x)||p(z|x)) - E_{q(z|x)} \log \frac{q(z|x)}{p(x|z)p(z)},$$

and we can therefore minimize the KL divergence by minimizing the rightmost term, which we call the evidence lower bound:

$$ELBO = E_{q(z|x)} \log \frac{q(z|x)}{p(x|z)p(z)} = -KL(q(z|x)||p(z)) + E_{q(z|x)} \log p(x|z)$$

via back-propagation and stochastic gradient descent. The KL divergence acts as a regularizer whilst the likelihood term represents the reconstruction error. $E_{q(z|x)} \log p(x|z)$ is the probability density of $x$ under the model given $z$ (I'm not entirely sure how to estimate it but I think we sample a few $x$'s from the dataset, and generate $z$'s with $q(z|x)$ then find the mean of something??).

We have defined $q(z|x)$ as an implicit distribution: a black box stochastic process parameterized by a neural network. The advantage of using an implicit posterior approximation as opposed to a typical explicit exponential distribution is the ability to model any dependencies within the data (normally $q(z|x) = N(\mu(x), \Sigma(x))$ where $\Sigma(x)$ is a diagonal matrix). The ELBO optimization process involves estimating $KL[q(z|x)||p(z)]$, which, in an explicit case, is easily computable via

$$KL[q(z|x)||p(z)] = KL[N(\mu_0, \Sigma_0)||N(\mu_1, \Sigma_1)] = \frac{1}{2}(tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\frac{det\Sigma_1}{det\Sigma_0}$$

where k is the dimensionality of the distribution.
However, with an implicit $q(z|x)$, we don't have any parameters, only a method of generating samples, so we must find an alternative way of estimating $E_{q(z|x)} \log \frac{q(z|x)}{p(z)}$. Our method of ratio estimation is an adversarial density ratio estimation approach. This involves taking samples from $p(z)$ and $q(z|x)$, labelling the $p(z)$ samples with $y = 1$ and the $q(z|x)$ samples with $y = 0$, then alternating optimization of a discriminator function $D(z; \theta_d) = P(y = 1|z)$, outputting the probability that a sample is from $q(z|x)$, and a generator function $G(\epsilon; \theta_g), \epsilon \sim N(0, I)$, creating $q(z|x)$ samples that imitate $p(z)$. This generator function is different (I think??) to the previous encoder function parameterized by $\phi$. Overall, we want to minimize the Bernoulli loss (other losses can be used):

$$\begin{aligned} L(\theta_d, \theta_g) &= E_{p(z|y)p(y)}[-y \log D(z; \theta_d) - (1 - y)\log(1 - D(z; \theta_d))] \\ &= \pi E_{p(z)}[-\log D(z; \theta_d)] + (1 - \pi)E_{q(z|x)}[-\log(1 - D(z; \theta_d))] \text{ where } \pi = P(y = 1) \\ &= \pi E_{p(z)}[-\log D(z; \theta_d)] + (1 - \pi)E_{q(z|x)}[-\log(1 - D(G(\epsilon; \theta_g); \theta_d))] \end{aligned}$$

This is achieved by alternating minimization of the ratio loss and generative loss:

$$\text{Ratio loss: } \min_{\theta_d} \pi E_{p(z)}[-\log D(x; \theta_d)] + (1 - \pi)E_{q(z|x)}[-\log(1 - D(z; \theta_d))]$$

$$\text{Generative loss: } \min_{\theta_g} E_{q(z|x)}[\log(1 - D(G(\epsilon; \theta_g); \theta_d)]$$

The trained discriminator can be used to find the log ratio.
The disadvantage of this approach is that $p(z|x)$ is not estimated explicitly, making inference very difficult.
An alternative approach to estimating $p(z|x)$ with an implicit $q(z|x)$ is to use the adversarial approach on the two distributions directly, but this gives us even less information to perform inference, as we won't have an estimate for the ELBO.
Our goal is to find better ways of estimating $log\frac{q(z|x)}{p(z)}$, either improving the speed or accuracy of the computation, or finding a method that provides more information of $p(z|x)$. We will implement the improved log ratio into common implicit variational models such as VAEs and GANs, run these on common benchmarks such as MNIST and ImageNet, then compare the results against current state-of-the-art models.

# 2 Learning

## 2.1 Variational Inference

### 2.1.1 Context

In Bayesian statistics, a common problem is to estimate posterior densities, so that we may perform inference to determine an unknown parameter. Consider a set of unknown, latent variables $\mathbf{Z} = \{z_i\}_{i=1}^M$ and a dataset of known variables $\mathbf{X} = \{x_i\}_{i=1}^N$. These sets have a joint density of $P(\mathbf{Z}, \mathbf{X})$. In the Bayesian framework, inference is often performed on the posterior density (the distribution of the parameters $\mathbf{Z}$ after the data $\mathbf{X}$ is observed) $P(\mathbf{Z}|\mathbf{X})$, which, after applying Bayes' theorem, can be written as:

$$P(\mathbf{Z}|\mathbf{X}) = \frac{P(\mathbf{Z})P(\mathbf{X}|\mathbf{Z})}{P(\mathbf{X})} = \frac{P(\mathbf{Z})P(\mathbf{X}|\mathbf{Z})}{\int_{\mathcal{Z}} P(\mathbf{Z}, \mathbf{X}) d\mathbf{Z}}$$

where

- $P(\mathbf{Z})$ is the prior distribution: the initial distribution of $\mathbf{Z}$ before the data $\mathbf{X}$ is observed. This can be initialised to represent our initial beliefs, or it can be parametrised randomly,
- $P(\mathbf{X}|\mathbf{Z})$ is the likelihood: the distribution of data $\mathbf{X}$ conditioned on the parameters $\mathbf{Z}$,
- $P(\mathbf{X}) = \int_{\mathcal{Z}} P(\mathbf{Z}, \mathbf{X}) d\mathbf{Z}$ is the marginal likelihood, or the evidence: the density of the data averaged across all possible parameter values.

If the evidence integral $P(\mathbf{X}) = \int_{\mathcal{Z}} P(\mathbf{Z}, \mathbf{X}) d\mathbf{Z}$ is impossible or difficult to compute (possibly because it is unavailable in closed form or the dimensionality is too high), then we are unable to evaluate the posterior density. Traditionally, MCMC(Markov Chain Monte Carlo) methods overcome this obstacle by constructing a Markov chain that converges to the stationary distribution $P(\mathbf{Z}|\mathbf{X})$, then sampling from the chain to create an empirical estimate for the posterior distribution. However, these methods rely on the speed of convergence, which can be slow for large datasets or complex models. When faced with these issues or when desiring a faster computation, one may instead apply variational inference, an alternative approach to density estimation.

### 2.1.2 Introduction to Variational Inference

Variational inference chooses another distribution $Q(\mathbf{Z})$ from a select family of variational distributions (approximate densities) $\mathcal{Q}$ to serve as an approximation to $P(\mathbf{Z}|\mathbf{X})$, and then minimizes the divergence between the two distributions in an optimization problem:

$$Q^*(\mathbf{Z}) = \underset{Q(\mathbf{Z}) \in \mathcal{Q}}{\arg\min}\, D(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X})) \tag{1}$$

where $D$ denotes an f-divergence (a measure of how divergent two probability distributions are, it is minimized if $Q = P$). This results in an analytical approximation to the posterior density. Additionally, a lower bound for the marginal likelihood of the dataset is derived, which can be used as a model selection criterion. Due to the stochastic nature of the optimization, variational inference methods can be much faster than MCMC, but the solution is only locally optimal as there is no guarantee of global convergence.

### 2.1.3 Derivation of the ELBO

The most common f-divergence used in variational inference is the KL(Kullback-Leibler) divergence, defined as the expected logarithmic difference between two distrbutions $Q$ and $P$ with respect to $Q$:

$$KL(Q||P) = \int_{-\infty}^{\infty} Q(x) \log \frac{Q(x)}{P(x)} dx = \mathbb{E}_{Q(x)} \left[ \log \frac{Q(x)}{P(x)} \right].$$

Note that the KL divergence is not symmetric. We use the reverse KL divergence instead of the forward KL divergence $KL(P||Q)$ because it leads to an expectation maximization algorithm as opposed to an expectation propagation algorithm.
Using this expression, we can rewrite equation (1) as:

$$\begin{aligned}
Q^*(\mathbf{Z}) &= \arg\min_{Q(\mathbf{Z}) \in \mathbb{Q}} KL(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X})) \\
&= \arg\min_{Q(\mathbf{Z}) \in \mathbb{Q}} \mathbb{E}_{Q(\mathbf{Z})}[\log Q(\mathbf{Z}) - \log P(\mathbf{Z}|\mathbf{X})] \\
&= \arg\min_{Q(\mathbf{Z}) \in \mathbb{Q}} \mathbb{E}_{Q(\mathbf{Z})} \left[ \log Q(\mathbf{Z}) - \log \frac{P(\mathbf{X}|\mathbf{Z})P(\mathbf{Z})}{P(\mathbf{X})} \right] \\
&= \arg\min_{Q(\mathbf{Z}) \in \mathbb{Q}} \left( \mathbb{E}_{Q(\mathbf{Z})}[\log Q(\mathbf{Z}) - \log P(\mathbf{X}|\mathbf{Z}) - \log P(\mathbf{Z})] + \log P(\mathbf{X}) \right).
\end{aligned}$$

Note in the last line $\mathbb{E}_{Q(\mathbf{Z})}[P(\mathbf{X})] = P(\mathbf{X})$ as it is not dependent on $Q(\mathbf{Z})$. Also note that the KL divergence is dependent on $P(\mathbf{X})$, which we have determined to be intractable, so this optimization problem cannot be solved in this form. This issue is resolved by rearranging the KL divergence expression as follows:

$$\begin{aligned}
KL(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X})) &= \mathbb{E}_{Q(\mathbf{Z})}[\log Q(\mathbf{Z}) - \log P(\mathbf{X}|\mathbf{Z}) - \log P(\mathbf{Z})] + \log P(\mathbf{X}) \\
\log P(\mathbf{X}) - KL(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X})) &= -\mathbb{E}_{Q(\mathbf{Z})}[\log Q(\mathbf{Z}) - \log P(\mathbf{X}|\mathbf{Z}) - \log P(\mathbf{Z})] \\
&= \mathbb{E}_{Q(\mathbf{Z})}[\log P(\mathbf{X}|\mathbf{Z})] - \mathbb{E}_{Q(\mathbf{Z})}[\log Q(\mathbf{Z}) - \log P(\mathbf{Z})] \\
&= \mathbb{E}_{Q(\mathbf{Z})}[\log P(\mathbf{X}|\mathbf{Z})] - KL(Q(\mathbf{Z})||P(\mathbf{Z})). \qquad (2)
\end{aligned}$$

We refer to $\log P(\mathbf{X}) - KL(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X}))$ as $ELBO(Q)$ (evidence lower bound), as it is equal to the marginal probability of the data subtracted by a constant error term. This error term $KL(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X}))$ becomes 0 when $Q(\mathbf{Z}) = P(\mathbf{Z}|\mathbf{X})$, maximizing the ELBO. Note that since $P(\mathbf{X})$ is constant, maximizing the $ELBO$ is equal to minimizing the KL divergence between $Q(\mathbf{Z})$ and $P(\mathbf{Z}|\mathbf{X})$, and that the expression on line (2) is entirely computable. We can therefore rewrite our optimization problem from equation (1) as:

$$\begin{aligned}
Q^*(\mathbf{Z}) &= \arg\min_{Q(\mathbf{Z}) \in \mathbb{Q}} D(Q(\mathbf{Z})||P(\mathbf{Z}|\mathbf{X})) \\
&= \arg\max_{Q(\mathbf{Z}) \in \mathbb{Q}} ELBO(Q) \\
&= \arg\max_{Q(\mathbf{Z}) \in \mathbb{Q}} \left( \mathbb{E}_{Q(\mathbf{Z})}[\log P(\mathbf{X}|\mathbf{Z})] - KL(Q(\mathbf{Z})||P(\mathbf{Z})) \right).
\end{aligned}$$

### 2.1.4 Mean-Field Variational Family

The family of variational distributions $\mathbb{Q}$ is typically a 'mean-field variational family', in which the distribution $Q(\mathbf{Z})$ factorizes over the latent variables $\{z_i\}_{i=1}^M$:

$$Q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(z_i). \qquad (3)$$

The individual factors $q_i(z_i)$ can take any form. We want to choose these factors so that $ELBO(Q)$ is maximized. To derive an expression for the optimal factor $q_i^*(z_i)$, we substitute equation (3) into the $ELBO$, factor out a specific $q_j(z_j)$ and equate the functional derivative of the resulting Lagrangian equation with 0. Firstly, we express $ELBO(Q)$ in an integral form as follows:

$$\begin{aligned} ELBO(Q) &= \mathbb{E}_{Q(Z)}[\log P(\mathbf{X}|\mathbf{Z})] - KL(Q(\mathbf{Z})||P(\mathbf{Z})) \\ &= \mathbb{E}_{Q(Z)}[\log P(\mathbf{X}|\mathbf{Z}) + \log P(\mathbf{Z}) - \log Q(\mathbf{Z})] \\ &= \mathbb{E}_{Q(Z)}[\log P(\mathbf{X}, \mathbf{Z}) - \log Q(\mathbf{Z})] \\ &= \int_{\mathbf{Z}} Q(\mathbf{Z})(\log P(\mathbf{X}, \mathbf{Z}) - \log Q(\mathbf{Z}))d\mathbf{Z}. \end{aligned}$$

Substituting $Q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(z_i)$ and factoring out $q_j(z_j)$ yields:

$$\begin{aligned} ELBO(Q) &= \int_{\mathbf{Z}} \left[\prod_{i=1}^{M} q_i(z_i)\right] \left(\log P(\mathbf{X}, \mathbf{Z}) - \sum_{i=1}^{M} \log q_i(z_i)\right) d\mathbf{Z} \\ &= \int_{z_j} q_j(z_j) \left(\int_{\mathbf{z}_{-j}} \log P(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i(z_i) d\mathbf{z}_{-j}\right) dz_j \\ &\quad - \int_{z_j} q_j(z_j) \left(\int_{\mathbf{z}_{-j}} \left[\prod_{i \neq j} q_i(z_i)\right] \sum_{i=1}^{M} q_i(z_i) d\mathbf{z}_{-j}\right) dz_j \\ &= \int_{z_j} q_j(z_j) \mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X}, \mathbf{Z})] dz_j - \int_{z_j} q_j(z_j) \log q_j(z_j) \left(\int_{\mathbf{z}_{-j}} \prod_{i \neq j} q_i(z_i) dz_{-j}\right) dz_j \\ &\quad - \int_{z_j} q_j(z_j) \left(\int_{\mathbf{z}_{-j}} \left[\prod_{i \neq j} q_i(z_i)\right] \sum_{i \neq j} q_i(z_i) d\mathbf{z}_{-j}\right) dz_j \\ &= \int_{z_j} q_j(z_j) \mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X}, \mathbf{Z})] dz_j - \int_{z_j} q_j(z_j) \log q_j(z_j) dz_j \\ &\quad - \int_{\mathbf{z}_{-j}} \left[\prod_{i \neq j} q_i(z_i)\right] \sum_{i \neq j} q_i(z_i) d\mathbf{z}_{-j} \qquad (4) \\ &= \int_{z_j} q_j(z_j) \left(\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X}, \mathbf{Z})] - \log q_j(z_j)\right) dz_j + \text{const.} \qquad (5) \end{aligned}$$

The term in line (4) becomes a constant as it does not depend on $q_j(z_j)$. Now our Lagrangian equation with the constraint that $q_i(z_i)$ are probability density functions is:

$$ELBO(Q) - \sum_{i=1}^{M} \lambda_i \int_{z_i} q_i(z_i) dz_i = 0$$

or using our expression for $ELBO(Q)$ in line (5),

$$\int_{z_j} q_j(z_j) \left(\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X}, \mathbf{Z})] - \log q_j(z_j)\right) dz_j - \sum_{i=1}^{M} \lambda_i \int_{z_i} q_i(z_i) dz_i + \text{const} = 0. \qquad (6)$$

Using the Euler-Lagrange equation (need to put this in), we then take the functional derivative of (6) with respect to $q_j(z_j)$ (in this case, the partial derivative with respect to $q_j(z_j)$ of the expression

inside the integral):

$$\begin{aligned}\frac{\partial ELBO(q)}{\partial q_j(z_j)} &= \frac{\partial}{\partial q_j(z_j)}\left[q_j(z_j)\left(\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})] - \log q_j(z_j)\right) - \lambda_j q_j(z_j)\right] \\ &= \mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})] - \log q_j(z_j) - 1 - \lambda_j \end{aligned} \tag{7}$$

Equating expression (7) to 0 and letting $1 + \lambda_j$ be a constant (as it is independent of $z$), we have:

$$\begin{aligned}\log q_j^*(z_j) &= \mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})] - \text{const} \\ q_j^*(z_j) &= \frac{e^{\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})]}}{\text{const}} \\ &= \frac{e^{\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})]}}{\int e^{\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})]} dz_j}.\end{aligned}$$

The normalization constant on the denominator can be easily derived by observing $q_j^*(z_j)$ as a density. Lastly, we derive a simpler expression of $q_j^*(z_j)$ by observing that terms independent of $z_j$ can be treated as a constant:

$$\begin{aligned}q_j^*(z_j) &\propto \exp\left(\mathbb{E}_{\mathbf{z}_{-j}}[\log P(\mathbf{X},\mathbf{Z})]\right) \\ &\propto \exp\left(\mathbb{E}_{\mathbf{z}_{-j}}[\log P(z_j|\mathbf{z}_{-j},\mathbf{X})]\right).\end{aligned} \tag{8}$$

This expression can be used in an expectation-maximization algorithm, in which the $q_j^*(z_j)$ is evaluated and iterated from $j = 1 \dots M$. This particular algorithm is called coordinate ascent variational inference (CAVI) (Algorithm 1):

**Data:** Dataset $\mathbf{X}$ and Model $\mathrm{P}(\mathbf{X},\mathbf{Z})$
**Result:** Approximate density $Q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(z_i)$
**begin**
  Initialize random variational factors $q_j(z_j)$;
  **while** *ELBO(Q) has not converged* **do**
    **for** $j = 1$ **to** $m$ **do**
      Set $q_j(z_j) \propto \exp(\mathbb{E}[\log P(z_j|\mathbf{z}_{-j},\mathbf{X})])$;
    **end**
    Calculate $ELBO(Q) = \mathbb{E}[\log P(\mathbf{Z},\mathbf{X})] - \mathbb{E}[\log Q(\mathbf{Z})]$;
  **end**
  Return $Q(\mathbf{Z})$;
**end**

**Algorithm 1:** Coordinate Ascent Variational Inference (CAVI)

### 2.1.5 Example: Bayesian mixture of Gaussians

To illustrate the variational inference approach, we will use the Bayesian mixture of Gaussians example from (Blei, 2018/16 idk).
Consider the hierarchical model

$$\begin{aligned}\mu_k &\sim N(0,\sigma^2), && k = 1,\dots,K, \\ c_i &\sim \text{Categorical}\left(\frac{1}{K},\dots,\frac{1}{K}\right), && i = 1,\dots,n, \\ x_i|c_i,\boldsymbol{\mu} &\sim N(c_i^\top \boldsymbol{\mu}, 1), && i = 1,\dots,n.\end{aligned}$$

This is a Bayesian mixture of univariate Gaussian random variables with unit variance. In this model, we draw $K$ $\mu_k$ variables from a prior Gaussian distribution $N(0, \sigma^2)$ ($\sigma^2$ is a hyperparameter), forming the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^\top$. We then generate an indicator vector $c_i$ of length $K$ from a prior categorical distribution. This vector has zeros for every element except for one element, where it is a 1. Each element has equal probability $1/K$ of being the element that contains the 1. The transpose of this $c_i$ is then multiplied by $\boldsymbol{\mu}$, essentially choosing one of the $\boldsymbol{\mu}$ elements at random. We then draw $x_i$ from the resulting $N(c_i^\top \boldsymbol{\mu}, 1)$.
Here, our latent variables are $\mathbf{z} = \{\mathbf{c}, \boldsymbol{\mu}\}$. Assuming $n$ samples, our joint density is

$$p(\boldsymbol{\mu}, \mathbf{c}, \mathbf{x}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \boldsymbol{\mu}). \tag{9}$$

From this, we derive the marginal likelihood

$$p(\mathbf{x}) = \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} \sum_{c_i} p(c_i) p(x_i | c_i, \boldsymbol{\mu}) d\boldsymbol{\mu}.$$

This integral is intractable, as the time complexity of evaluating it is $\mathcal{O}(K^n)$, which is exponential in $K$. To evaluate the posterior distribution over the latent variables $p(\boldsymbol{\mu}, \mathbf{c}|\mathbf{x})$, we would have to apply variational inference, approximating it with a variational distribution $q(\boldsymbol{\mu}, \mathbf{c})$. We will assume this distribution follows the mean-field variational family:

$$q(\boldsymbol{\mu}, \mathbf{c}) = \prod_{k=1}^{K} q(\mu_k; m_k, s_k^2) \prod_{i=1}^{n} q(c_i; \boldsymbol{\phi_i}).$$

In this distribution, we have $K$ Gaussian factors with mean $\mu_k$ and variance $s_k^2$, and $n$ categorical factors with index probabilities defined by the vector $\boldsymbol{\phi_i}$, such that

$$\begin{aligned} \mu_k &\sim N(m_k, s_k^2), && k = 1, \ldots, K, \\ x_i &\sim \text{Categorical}(\boldsymbol{\phi_i}), && i = 1, \ldots, n. \end{aligned}$$

Using this and equation (9), we can derive $ELBO(q)$:

$$\begin{aligned} ELBO(q) &= \mathbb{E}[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}[\log q(\mathbf{z})] \\ &= \mathbb{E}[\log p(\boldsymbol{\mu}, \boldsymbol{c}, \mathbf{x})] - \mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{c})] \\ &= \sum_{i=1}^{K} \mathbb{E}[\log p(\mu_k); m_k, s_k^2] + \sum_{i=1}^{k} \left( \mathbb{E}[\log p(c_i); \boldsymbol{\phi}_i] + \mathbb{E}[\log p(x_i | c_i, \boldsymbol{\mu}); \boldsymbol{\phi}_i, \mathbf{m}, \mathbf{s}^2] \right) \\ &\quad - \sum_{k=1}^{K} \mathbb{E}[\log q(\mu_k; m_k, s_k^2)] - \sum_{i=1}^{n} \mathbb{E}[\log q(c_i; \boldsymbol{\phi}_i)] \end{aligned}$$

From equation (8), we derive the optimal categorical factor

$$q^*(c_i; \boldsymbol{\phi}_i) \propto \exp\left( \log p(c_i) + \mathbb{E}[\log p(x_i | c_i, \boldsymbol{\mu}); \mathbf{m}, \mathbf{s}^2] \right). \tag{10}$$

Now since $c_i$ is an indicator vector,

$$p(x_i | c_i, \boldsymbol{\mu}) = \prod_{k=1}^{K} p(x_i | \mu_k)^{c_{ik}}.$$

We can now evaluate the second term of equation (10):

$$\begin{aligned}\mathbb{E}\left([\log p(x_i|c_i,\boldsymbol{\mu});\mathbf{m},\mathbf{s}^2]\right) &= \sum_{k=1}^{K} c_{ik}\mathbb{E}[\log p(x_i|\mu_k);m_k,s_k^2] \\ &= \sum_{k=1}^{K} c_{ik}\mathbb{E}[-(x_i-\mu_k)^2/2;m_k,s_k^2] + \text{const} \\ &= \sum_{k=1}^{K} c_{ik}\left(\mathbb{E}[\mu_k;m_k,s_k^2]x_i - \mathbb{E}[\mu_k^2;m_k,s_k^2]/2\right) + \text{const.}\end{aligned}$$

In each line, terms constant with respect to $c_{ik}$ have been taken out of the expression. By proportionality, we have the variational update

$$\phi_{ik} \propto \exp\left(\mathbb{E}[\mu_k;m_k,s_k^2]x_i - \mathbb{E}[\mu_k^2;m_k,s_k^2]/2\right).$$

**Data:** Data $\mathbf{x}$, Number of Gaussian components $K$, Hyperparameter value $\sigma^2$
**Result:** Optimal variational factors $q(\mu_k;m_k,s_k^2)$ and $q(c_i;\boldsymbol{\phi_i})$

**begin**
  Randomly initialize parameters $\mathbf{m},\mathbf{s}^2$ and $\boldsymbol{\phi}$;
  **while** *ELBO has not converged* **do**
    **for** $i = 1$ **to** $n$ **do**
      Set $\phi_{ik} \propto \exp\left(\mathbb{E}[\mu_k;m_k,s_k^2]x_i - \mathbb{E}[\mu_k^2;m_k,s_k^2]/2\right)$;
    **end**
    **for** $k = 1$ **to** $K$ **do**
      Set $m_k = \frac{\sum_i \phi_{ik}x_i}{1/\sigma^2+\sum_i \phi_{ik}}$;
      Set $s_k^2 = \frac{1}{1/\sigma^2+\sum_i \phi_{ik}}$;
    **end**
    Compute $ELBO(\mathbf{m},\mathbf{s}^2,\boldsymbol{\phi})$;
  **end**
  Return $q(\mathbf{m},\mathbf{s}^2,\boldsymbol{\phi})$;
**end**

**Algorithm 2:** CAVI Algorithm for Bayesian mixture of Gaussians

### 2.1.6 References

To be organised properly and moved to the end later:
*https* : *//en.wikipedia.org/wiki/Variational*$_B$*ayesian*$_m$*ethods*
http://bjlkeng.github.io/posts/variational-bayes-and-the-mean-field-approximation/
https://arxiv.org/pdf/1601.00670.pdf
Pattern recognition and machine learning by Bishop (2006) pages 461-dunno
https://www.cs.cmu.edu/ epxing/Class/10708-17/notes-17/10708-scribe-lecture13.pdf

## 2.2 Neural Networks

### 2.2.1 Motivation

To discuss brief history, relation to neuroscience.

### 2.2.2 Overall Structure

Insert picture, discuss input output layers, bias/intercept nodes

### 2.2.3 Node Structure

internal structure of node, activation function e.g. sigmoid or tanh

### 2.2.4 Back-Propagation

Give algorithm for training neural network.

### 2.2.5 Example: MNIST Classification

Show exact pseudocode algorithm for classifying MNIST samples with results/pictures. (I have already coded a basic NNet MNIST classifier at home).

## 2.3 Autoencoding Variational Bayes

Kingma 2013

### 2.3.1 Problem Context

Introduce notations, similar to VB but with NN parametrization. Discuss problems with intractability and large dataset, introduce the 3 main things we want to solve.

### 2.3.2 Structure

Diagram of VAE

### 2.3.3 Variational Bound

Take ELBO from VB section but parametrize it with NN, discuss the typical useless Monte Carlo gradient estimator.

### 2.3.4 Algorithm

Propose better estimator of lower bound and its derivatives, derive AEVB algorithm.

### 2.3.5 Reparametrization Trick

Explain the trick and show changes in structure.

### 2.3.6 Example: Variational Autoencoder (for MNIST? or cats :) or human faces D: )

Show pseudocode and diagram for VAE (if different from before) and show output from own VAE code (gonna have to code my own VAE).

## 2.4 Density Ratio Estimation

Mohamed 2016, Sugiyama textbook, Goodfellow 2014, Huszar 2017

### 2.4.1 Context: Implicit Generative Models

Describe these models and the inference on them

### 2.4.2 Loss Functions

Choice and derivation of loss functions

### 2.4.3 Class Probability Estimation

Adversarial Classifier, algorithm

### 2.4.4 Divergence Minimisation

### 2.4.5 Ratio Matching

### 2.4.6 Moment Matching

### 2.4.7 Denoising

## 2.5 Likelihood-Free Variational Inference

Tran 2017 (This section may go before Density Ratio Estimation)

### 2.5.1 Context

Lack of likelihood introduces additional intractability...

### 2.5.2 Variational Bound

Similar to VB section but expression is very different and there is alot of intractability.

### 2.5.3 Ratio Estimation

Derive loss function and minimize it, estimating the ratio in the process

### 2.5.4 Algorithm

Put everything together in an algorithm (LFVI)