1. What is variational inference?

- In Bayesian statistics, all inference about an unknown quantity is framed as a calculation about a posterior distribution
- But often, the posterior distributions we deal with are not easy to compute
- Instead we need to design algorithms to approximate them
- One such method to *approximate* a pdf is Variational Inference (VI)
  - This is in contrast to *sampling* which is usually done using Markov Chain Monte Carlo simulation

2. Problem setup

- Consider the joint density of latent variables $z = z_{1:m}$ and observations $x = x_{1:m}$, which takes the form $p(z, x) = p(z)\, p(x \mid z)$
- In Bayesian models, the latent variables govern the distribution of the data
  - The model draws latent variables from the *prior* density $p(z)$ and then relates them to observations using the *likelihood* $p(x \mid z)$
  - So we KNOW what $p(z)$ looks like, and we also know what $p(x|z)$ looks like.
- Inference in this setup means to compute the *posterior* distriubution $p(z \mid x)$ by conditioning on the observed data
- Q: Why do we ever care about the posterior distribution if you already know (or assume) the prior?
  - Because the prior can encompass a vast array of possibilities that could be true, while you are only interested in the unique set of possibilities that could be true conditional on the data/knowledge you have about the world!
    * devoid of any context in this vast space, it can be (and usually is) impossible to narrow down the scope of possibilities to something meaningful
  - You don't care what $p(infectivity_{omicron})$ looks in the abstract, devoid of any context - you know very little to substantiate any claim here
  - You do care about $p(infectivity_{omicron} \mid number\ of\ omicron\ covid\ cases\ seen)$ because this data informs the opinion you should hold about the variant
  - It is almost pointless to reason about all the potential possibilities for z
    * especially if it is multidimensional: curse of dimensionality?
  - Instead, you would rather isolate a local subregion of interest and understand what is happening there
- But in many cases, this posterior is difficult to compute analytically/exactly, so we need to approximate it instead
- The dominant paradigm for this approximate inference has been MCMC, which involves sampling an approximate distribution
  - First, construct an ergodic Markov chain on z whose stationary distribution is $p(z \mid x)$

  - Then sample from this chain to collect samples from the stationary distribution
  - Finally, approximate the posterior with an empiricial estimate constructed from the collected samples
- However, MCMC has its limitations
  - When datasets are large or models are complex, MCMC can take a long time to yield an approximate conditional
  - Q: why?
- In these settings, VI offers a good alternative approach to the problem of conducting inference

3. Variational inference

- The main idea behind VI is to use *optimization* instead of *sampling* to approximate a distribution
- First, posit a family of approximate densities, Q
  - This is a set of densities over the latent variables z
- Then, try and find a member of this family that minimizes the KL divergence to the *exact* posterior
  - $q^*(x) = argmin_{q(x) \, \epsilon \, Q} \; KL(q(z) \, || \, p(z \mid x))$
  - Note that KL divergence is just one of many possible metric for divergence (though it is arguably the most common such metric)
- Q: why do we not care to approximate $p(z|x)$ using $q(z|x)$ instead of $q(z)$?
- A: each $q(z) \, \epsilon \, Q$ is a candiate approximation to the *exact* conditional for the given state $X = x$. That is, we take the X to be given and held constant, and then find a good $q^*(z)$ for this specific conditional.
  - Each $X = x$ will have a different corresponding $q(z)$!
    * In the context of variational autoencoders, we are NOT learning parameters for one $q(z)$ distribution. Rather, we learn a function that generates a new set of hyperparameters for a new $q(z)$ for every new data point $X = x_{new}$ that we see!
  - The question is: for each new $X = x_{new}$ we encounter: what set of 'variational parameters' best parameterizes the distribution, which has a known shape since it comes from a known family Q, so it most closely resembles the posterior distribution we really care about?
- Finally, approximate the posterior using $q^*(x)$
- The reach of the family Q manages the complexity of the optimization problem: you want to choose a family that is both flexible enough to capture a density close to the real $P(z \mid x)$, and also simple enough for the optimization problem itself to be simple enough

3.1 When to use MCMC vs VI?

- MCMC approximates the posterior with samples from a chain; VI appriximates the posterior with the result of the optimization
- VI can take advantage to make optimization quicker, like stochastic optimization and distributed optimization
  - So, it is typically used in settings with large datasets, or scenarios where we want to explore many models
- MCMC is perhaps more suited for settings where want more precise samples at the tradeoff of a higher computational cost
- The relative accuracy of these two methods is still unknown
  - As a a general rule of thumb, VI tends to underestimate the variance of the posterior density, but this is often acceptable
- Also, these tools are not exclusively useful in a Bayesian setting! They prove useful tools any setting where we need to simulated from a density (MCMC) or approximate a density (VI)

3.2 Diving into VI

- The term 'variational inference' comes from the use of free 'variational parameters' to parameterize the assumed family of densities, Q, over the latent variables, z
- Assume latent variables $z = z_{1:m}$ and observed data $x = x_{1:m}$
- Then, the conditional density $p(z \mid x) = \dfrac{p(z,\ x)}{p(x)}$
- The denominator, $p(x)$, is the marginal density of the observed data, also called the *evidence*
- We know this evidence $p(x) = \int p(z,\ x)\, dz$
- If we could calculate this evidence, then calculating the conditional density we want would be trivial - however, in many models, this integral is unavailable in closed form, or has a computational complexity that is exponential in the number of samples
  - This is why inference in these models is a tough problem!

3.3 An example: Bayesian mixture of Gaussians

- Consider a (Bayesian) mixture of K univariate Gaussians, with sample size n
  - $\boldsymbol{\mu} = \{\mu_1,\ ...,\ \mu_K\}, k = 1,\ ...,\ K$
  - $p(\mu_k) \sim N\left(0,\ \sigma^2\right)$, where σ is a hyperparameter
  - $c_i \sim categorical(1/K,\ ...,\ 1/K),\ i = 1,\ ...,\ n$
    - $\mathbf{c} = \{c_1,\ c_2,\ ...,\ c_n\}$
  - $x_i \mid (c_i,\ \boldsymbol{\mu}) \sim N\left(c_i^T \boldsymbol{\mu},\ 1\right),\ i = 1,\ ...,\ n$
    - $\mathbf{x} = \{x_1,\ x_2,\ ...,\ x_n\}$

- The latent variables z in this model are hence c and $\boldsymbol{\mu}$
  - These latent variables are in turn generated by some pdf that is parameterized by hyperparameters
  - hyperparameters → latent variables → observed data
- The objective is: given $\mathbf{x}$, make an inference on $p(z \mid \mathbf{x})$.
- The joint density of latent and observed variables is hence

$$p(z,\ \mathbf{x})\ =\ p(\mathbf{c},\ \boldsymbol{\mu},\ \mathbf{x})\ =\ p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i)\ p(x_i \mid (c_i,\ \boldsymbol{\mu}))\ (0)$$

  - As a reminder, this is saying: for a given $\boldsymbol{\mu} = \mu^*,\ \mathbf{c}\ =\ c^*,$ and $\mathbf{x}\ =\ x_{1:n}$ - that is, for a given mean vector, given allocation of data points to clusters, and given observed data - what is the probability that you will see these given values values jointly across all three random variables
  - Q: What is $p(\boldsymbol{\mu})$?
    * I imagine it is just $\prod_{i=1}^{K} p(\mu_i)$?
  - Q: shouldn't we also account for $p(\mathbf{c})\ =\ \frac{1}{K^n}$?
    * This equation already does! $p(c_i)$ is always $\frac{1}{K}$, and over n samples, this multiplies out to the same
- So the evidence is then $p(x)\ =\ \int p(z,\ x)\ dz\ =\ \int p(x,\ c,\ \mu)\ d(c_i,\ \mu)$
  - But c_i is categorical, so we can sum over all its values
- So $p(\mathbf{x})\ =\ \int p(\boldsymbol{\mu}) \prod_{i=1}^{n} \sum_{c_i} p(c_i)\ p(x_i \mid (c_i,\ \mu))\ d\boldsymbol{\mu}$
  - Since there are K classes, the expanded product will take the form of (K term sum) $*$ (K term sum) ... , n times, since the product is over all samples
    * For example, $(a\ +\ b)\ (c\ +\ d)\ (e\ +\ f)$ when expanded out will yield $2 * 2 * 2\ =\ 8$ terms, or 2^3 terms
  - So, the expanded integral will feature K^n terms (expanding this product each of the n K-term sums will yield this many individual terms)
- So, the time complexity of evaluating this integral is $O\left(K^n\right)$, or exponential in n
  - This is prohibitively expensive for large enough n!
- Another way of writing this function would be as a sum of a solvable integral over every possible data point <> cluster allocation, i.e.

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

- But there are $O(K^n)$ such data point <> cluster allocation possibilities! So the time complexity of evaluating the integral still remains.

4.The Evidence Lower Bound

- In VI, we specify a family of densities Q over the latent variables. Each $q(z) \in Q$ is a candidate approximation to the exact condition we care about, i.e. $p(z|x)$.
- The goal is to find the best possible such $q(z)$, the one closest in KL divergence to the exact conditional. That is, we need to solve the problem
  $q^*(\mathbf{z}) = argmin_{q(\mathbf{z}) \in Q} KL(q(\mathbf{z}) \,||\, p(\mathbf{z}|\mathbf{x}))$.
- Q: Why do we care for $KL(q(z) \,||\, p(z|x))$ and not $KL(p(z|x) \,||\, q(z))$? Since KL divergence is not symmetric, why pick this over the alternative?
- A: https://bjlkeng.github.io/posts/variational-bayes-and-the-mean-field-approximation/
- If $p(x)$ is the true distribution, and $q(x)$ is the approximation distribution, the 'forward' KL divergence is defined as
  $KL(p(x)||q(x))$
  $= E_{p(x)}(log(p(x)) - E_{p(x)}(log(q(x)) = \sum p(x) log\left(\frac{p(x)}{q(x)}\right) = -\sum p(x) log\left(\frac{q(x)}{p(x)}\right)$
  - This is considered the 'forward' KL divergence.
  - If for some x, $q(x) \rightarrow 0$ but $p(x)$ does not, this will blow up the $log\left(\frac{p(x)}{q(x)}\right)$ term, therefore increasing the KL divergence.
  - And if we have high mass of $q(x)$ where there is low mass of $p(x)$, that does not impact the KL divergence much.
  -

	p(x) zero	p(x) non-zero
q(x) zero	ok	bad
q(x) non-zero	negligible	ok

  - So, to minimize the forward KL divergence, when we choose the approximate distribution $q(x)$, you want to 'cover' all the non-zero parts of $p(x)$ as much as possible - so we want $q(x)$ to place mass wherever $p(x)$ has mass, possibly at the expense of placing mass in $q(x)$ where $p(x)$ has none.
  - But this may not be very useful! For example, if $p(x)$ is multimodal, the $q(x)$ that minimizes forward KL divergence may look nothing like it.

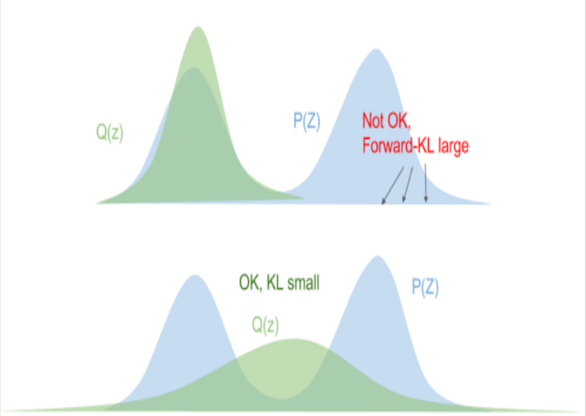

Figure 1: Forward KL Divergence (source: Eric Jang's Blog)

- Similarly, the 'reverse' KL divergence is defined as $KL(q(x)||p(x))$

$$= E_{q(x)}(log(q(x)) - E_{q(x)}(log(p(x)) = \sum q(x) log\left(\frac{q(x)}{p(x)}\right) = -\sum q(x) log\left(\frac{p(x)}{q(x)}\right)$$

  - If for some x, $q(x) \rightarrow 0$ but $p(x)$ does not, this will not impact the KL divergence much.
  - And if we have high mass of $q(x)$ where this is low mass of $p(x)$, it will blow up the $log\left(\frac{q(x)}{p(x)}\right)$ term, therefore increasing the KL divergence.
  -

	p(x) zero	p(x) non-zero
q(x) zero	ok	negligible
q(x) non-zero	bad	ok

  - So, to minimize the reverse KL divergence, when we choose the approximate distribution $q(x)$, you do not want $q(x)$ to place mass where $p(x)$ has no mass, possibly at the expense of also not placing any $q(x)$ mass when $p(x)$ has some.
  - Typically this means you tend to capture at least one mode in a multimodal distribution well, which can be useful
  -

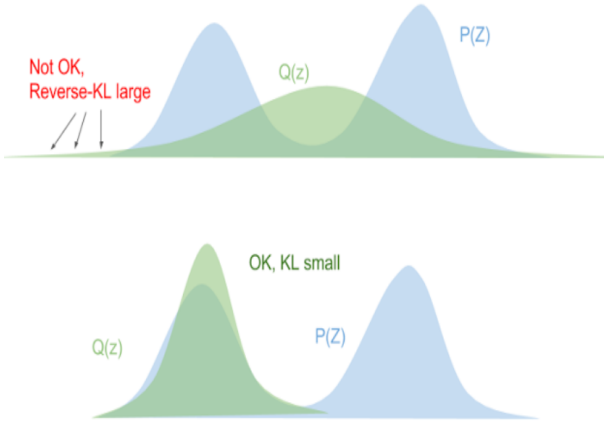

Figure 2: Reverse KL Divergence (source: Eric Jang's Blog)

•

4.1 Entropy

- Entropy is a measure of the amount of information in a probability distribution
  - The self information of an event $X = x$ is $I(x) = -log(p(x))$.
    * Intuitively, if there is a high probability that $X = x$, then we learn very little from observing this information.
    * Because $0 \leqq p(x) \leqq 1$, the log of any $p(x)$ in this range is negative
    * And, $log(x)$ is more negative closer for x closer to 0 than to 1
    * So $-log(x)$ is more when $p(x)$ is closer to 0
  - Q: is the reason information is $-log(p(x)) = log(1/p(x))$ simply that it has the mathematical properties necessary to fit this intuition? Or is there more to it?
    * If the behaviour you want is: design some $I(x)$ such that when $p(x)$ is high, $I(x)$ is low, and vice versa.
    * You could achieve the same behaviour through simply using $-p(x)$! or $exp(-p(x))$?
    * for $I(x) = -p(x)$
      · when $p(x) = 0, I(x) = 0$
      · when $p(x) = 1, I(x) = -1$
    * for $I(x) = exp(-p(x))$,
      · when $p(x) = 0, I(x) = exp(0) = 1$
      · when $p(x) = 1, I(x) = exp(-1) = 1/e$
  - The criteria Shannon wanted the measure to meet were:
    * when $p(x) = 1$, $I(x) = 0$
      · an easy way to ensure this is to use a log function: something like $log(p(x))$ would do it.

        - Is there any other way? naively: $I(x) = 1 - p(x)$ would as well...
    * when $p(x) < p(y)$, $I(x) > I(y)$, and vice versa
        - This would also be fulfilled by $I(x) = 1 - p(x)$
    * if two independent events are measured separately, the total amount of information contained is the sum of the self-informations of the invididual events: i.e. $I(x, y) = I(x) + I(y) \; for \; x \; independent \; of \; y$
        - if $I(x) = 1 - p(x)$, then
          $I(x, y) = 1 - p_{X,Y}(x, y) = 1 - p_X(x)p_Y(y) \neq 1 - p_X(x) + 1 - p_Y(y)$
    * The only function that fulfills these three criteria is $-log(p(x))$
    * $I_{X,Y}(x, y) = -log(p_{X,Y}(x, y)) = -log(p_X(x)p_Y(y))$
      $= -(log(p_X(x)) + log(p_Y(y))) = I_X(x) + I_Y(y)$
- Entropy is the expected value of the self-information of every event in a probability distribution: $H(x) = E_{p(x)}(I(x)) = -\sum p(x)log(p(x))$ for a discrete probability distribution

4.2 KL divergence, again

- KL divergence is a non symmetric measure of the difference between two probability distributions. Specifically, the Kullback-Leibler (KL) divergence of q(x) from p(x), denoted

DKL(p(x), q(x)), is is used to approximate p(x).

- http://hanj.cs.illinois.edu/cs412/bk3/KL-divergence.pdf

- KL divergence is not a distance metric because it is not symmetrical and does not satisfy triangle inequality
  - Q: What is the triangle inequality for metrics of this form?
  - It is the same as it would be for any distance metric.
    $d(p, q) \leqq d(p, r) + d(r, q)$ where p, q, r are probability distributions.
    * See here for an example proving that KL divergence does not fulfill this: https://ai.stackexchange.com/questions/18019/why-does-kl-divergence-not-satisfy-the-triangle-inequality
- When calculating a KL divergence from observational data for p and q, we usually need to apply smoothing by assigning some small ϵ pdf value to values that are in one distributoin but not the other
  - KL divergence is defined as $KL(p(x)||q(x)) = E_{p(x)}(log(p(x)) - E_{p(x)}(log(q(x))$
  - This is because when $p(x) \neq 0$ but $q(x) = 0$, the KL divergence is infinite
  - But given we are building these estimates from observation, we cannot ever really be sure that $q(x)$ is truly 0

4.3 ELBO, again

- To reiterate: The objective is to find $q^*(\mathbf{z}) = argmin_{q(z) \, \epsilon \, Q} \, KL(q(\mathbf{z}) \, || \, p(\mathbf{z}|\mathbf{x}))$, so we

can approximate $p(z|x)$.

- NOTE: in this section, the symbols z and x refer to the full vectors i.e. z should really be **z**, referring to all the latent variables at once, and x should really be **x**, referring to all the observed data at once
- But

$$
\begin{aligned}
&KL(q(z) \,||\, p(z|x)) \\
&= E_{q(z)}[log(q(z)] - E_{q(z)}[log(p(z|x)] \\
&= E_{q(z)}[log(q(z)] - E_{q(z)}\left[log\left(\frac{p(z,\, x)}{p(x)}\right)\right] \\
&= E_{q(z)}[log(q(z)] - E_{q(z)}[log(p(z,\, x)] + E_{q(z)}[log(p(x)] \\
&= E_{q(z)}[log(q(z)] - E_{q(z)}[log(p(z,\, x)] + log(p(x))
\end{aligned}
$$

which relies on calculating $log(p(x))$, which in turn means we need to calculate $p(x)$!

- But this is the problem we are trying to solve in the first place! To not have to calculate $p(x)$!
- So we cannot really optimize the KL divergence.
- But what if we find an equivalent metric to optimize over instead?
- Enter the ELBO, the negative of the first two terms of the equation above, which yields (1) and (2).
-

$$
\begin{aligned}
&ELBO(q(z)) \\
&= E[log(p(z,\, x))] - E[log(q(z)] \; (1) \\
&= -KL(q(z) \,||\, p(z|x)) + log(p(x)) \; (2) \\
&= E_{q(z)}[log(p(x|z))] - KL(q(z)||p(z)) \; (3,\ derived\ below)
\end{aligned}
$$

- Using the ELBO works because we know that $log(p(x))$ is constant for a given x for any value of z, i.e. it is independent of $q(z)$ (and constant across it)
  - Rewriting the KL divergence equation in terms of $log(p(x))$, we get $log(p(x)) = -KL(q(z) \,||\, p(z|x)) + E_{q(z)}[log(q(z)] - E_{q(z)}[log(p(z,\, x)]$
  - So, since $ELBO - KL = constant$: given a constant $log(p(x))$, maximizing the ELBO is equivalent to minimizing the KL divergence.
- Further, we can rewrite ELBO as a function of the latent variable prior $p(z)$.

$$
\begin{aligned}
&ELBO(q(z)) \\
&= E_{q(z)}[log(p(z,\, x))] - E_{q(z)}[log(q(z))] \\
&= E_{q(z)}[log(p(x|z)p(z))] - E_{q(z)}[log(q(z))] \\
&= E_{q(z)}[log(p(x|z))] + E_{q(z)}[log(p(z))] - E_{q(z)}[log(q(z))] \\
&= E_{q(z)}[log(p(x|z))] - KL(q(z)||p(z)) \; (3)
\end{aligned}
$$

- In other words, ELBO = expected log likelihood of the data observed - KL divergence between q(z) and the prior p(z)
- Intuitively, this suggests that maximizing the ELBO will encourage $q(z)$ densities that place their mass on configurations of the latent variables that:
  - explain the observed data, since the first term is an expected log likelihood

function
  - resemble the prior $p(z)$ closest, since the second term is the negative KL divergence between $q(z)$ and $p(z)$
- The ELBO gets its name because it serves as the lower bound for the log evidence function, $log(p(x))$
  - $log(p(x)) \; = \; KL(q(z) \; || \; p(z|x)) \; + E_{q(z)}[log(q(z)] \; - \; E_{q(z)}[log(p(z, \; x)]$
  - Since the KL divergence ≥ 0, this means that $log(p(x)) \; \geq \; ELBO(q(z))$
- Given that ELBO yields useful insight into the evidence function, can we use it for the purpose of model selection as well?
  - For example, when picking between different models, picking the model with the highest ELBO would mean picking the model with the highest lower bound on the evidence
- The first term of the ELBO expression (1) is the expected complete log likelihood, which is optimized by the Expectation Maximization algorithm (EM)
  - The EM algorithm was designed for finding maximum likelihood estiamtes in models with latent variables
  - It uses the fact that the ELBO = $log(p(x))$ when $q(z) \; = \; p(z|x)$, since

4.3 Expectation Maximization

Problem setup

- We have a statistical model that
  - generates known/observed data X
  - from latent unknown/unobserved data Z
  - using unknown/unobserved model parameters θ
  - so something along the lines of $X \; = \; f(Z, \; \theta)$
- For example, when we assume that data X is generated by a mixture model, we may find it useful to assume the existence of some latent variable Z that encodes which data cluster a given data point belongs to
- Our objective is: find the maximum likelihood estimate parameter set θ^*, i.e. the parameter set θ^* that maximizes $p(x \mid \theta)$
- Note this is a totally different setup than we have in variational inference!
  - in VI, we want to estimate $p(\theta|x)$, i.e. we want to understand the posterior distribution as a whole
  - in MLE, we want a point estimate θ^* that maximizes $p(x|\theta)$
  - in MAP, we want a point estimate θ^* that maximizes $p(x|\theta)p(\theta)$ for some assumed prior $p(\theta)$ over the parameter set
    * if the prior is uniform, then this is the same as MLE.
  - More here: https://towardsdatascience.com/mle-map-and-bayesian-inference-3407b2d6d4d9

- Q: Why do you ever want to estimate the posterior distribution when you can
- Why is this difficult?
  - https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
  - Finding a maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values, the parameters and the latent variables, and simultaneously solving the resulting equations. In statistical models with latent variables, this is usually impossible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice versa, but substituting one set of equations into the other produces an unsolvable equationbecause we don't have any analytical equations to describe
  - The EM algorithm proceeds from the observation that there is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work, but it can be proven in this context. Additionally, it can be proven that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a local maximum or a saddle point.
- Formalizing this a bit more, we have $p(x \mid \theta) = \int p(x \mid z, \theta)\, p(z \mid \theta)\, dz$, and we want to find the parameter set θ^* that maximizes this.
  - This is usually not solvable? But why? because we do not know what $p(z|\theta)$ looks like?
  - Q: can't we assume a prior distribution over $p(z)$? Like we do in the variational inference case anyway?
    * we do not observe z in this case either, but that does not stop us from assuming a prior over these values?
    * Do we want to avoid making any assumptions over what Z might look like at all? A totally non-parametric approach?
- <come back to this later>

5. The Mean-Field Variational Family

- We have just described how maximizing ELBO usually is the variational objective function
- Now, we will describe a variational family, Q, to complete the specification of the optimization problem
  - the complexity of this family determines the complexity of the optimization problem
- Here we will discuss the mean-field variational family, where the latent variables are

modelled as mutually independent. Each latent variable is governed by a distinct factor in the variational density

- This does NOT mean that we are assuming that the latent variables are independent of each other! It just means that we cannot capture any correlation structure between variables using a mean-field variational family.
  - To be clear, this DOES NOT mean that $p(z_j \mid \mathbf{z}_{-j}) = p(z_j)$. But it DOES mean that $q(z_j \mid z_{-j}) = q(z_j)$, because we are specifying variational factors of this form!
- A generic member of the mean-field variational family is $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$
  - each latent variable z_j is governed by its own variational factor, the density $q_j(z_j)$
- Note that this has nothing to do with the model specifications that specify how the hyperparameters generate the latent variables of interest!
  - the q_i functions here *could* share the same form as the $p_i(z_i)$ functions we specify in the model spec, but they do not have to!
  - Remember that $q(z)$ is an approzimation for $p(z|x)$ which does not need to have anything to do with $p(z)$
- In optimization, these variational factors are chosen to maximixe ELBO as described in equation (1), i.e. $ELBO(q(\mathbf{z})) = E[log(p(\mathbf{z}, \mathbf{x}))] - E[log(q(\mathbf{z})]$
- The variational family is not a model of the observed data $\mathbf{x}$: the ELBO connects the fitted variational density to the data and model.

5.1 Revisiting the example 3.3 - Bayesian mixture of Gaussians

- The mean-field variational family contains approximate posterior densities of the form

$$q(\boldsymbol{\mu}, \mathbf{c}) = q(\boldsymbol{\mu})\, q(\mathbf{c}) = \prod_{k=1}^{K} q\left(\mu_k;\, m_k,\, s_k^2\right) \prod_{i=1}^{n} q(c_i;\, \phi_i) \quad (4)$$

- Each latent variable is governed by its own variational factor.
- The first factor is a distribution on the *k*th mixture component's mean parameter
- The second factor is a distribution on the ith observation's mixture assignment, whose assignment probabilities are a k-vector ϕ_i
- We can go further, and assign parametric forms to these factors well.
  - Breaking up factor 1, we can parameterize it using a Gaussian distribution on the *k*th mixture component's mean parameter, with mean m_k and variance s_k^2. So each cluster has variational parameters specific to it.
    * Here I would guess that we expect to see $m_k \sim 0$ and $s_k^2 \sim \sigma^2$ for all the k clusters? Because that is indeed how the clusters are generated (via the prior on them)
  - Breaking up factor 2, we say that cluster assignments are categorical with cluster

probabilities specific to the ith data point.So each data point has a cluster assignment specific to it. For each data point, the assignment probabilities are a K-vector ϕ_i.

    * Here I guess ideally we get a one hot vector embedding which exactly identifies a cluster for each data point? Just like c_i specifies the cluster assignment for every corresponding data point

- So what we are saying is that
  - $p(\boldsymbol{\mu} \mid \mathbf{x}) \sim q(\boldsymbol{\mu}) = \prod_{i=1}^{k} q(\mu_i) = \prod_{i=1}^{k} N\left(m_k,\ s_k^2\right)$
- This completely specifies the VI problem for the mixture of Gaussians.
  - To reiterate, the objective is to estimate $p(\mathbf{z} \mid \mathbf{x}) = p(\mathbf{c},\ \boldsymbol{\mu} \mid \mathbf{x})$ by finding a suitable $q^*(\mathbf{z})$ that approximates $p(\mathbf{z} \mid \mathbf{x})$
  - To do so, we find $q^*(\mathbf{z})$
    $= argmax_q\ ELBO(q(\mathbf{z})) = E_{q(z)}[log(p(\mathbf{z},\ \mathbf{x}))] - E_{q(z)}[log(q(\mathbf{z})]$ from (1)
  - $p(\mathbf{z},\ \mathbf{x}) = p(\mathbf{c},\ \boldsymbol{\mu},\ x) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i)\, p(x_i \mid (c_i,\ \boldsymbol{\mu}))$ from (0)
  - $q(\mathbf{z}) = q(\boldsymbol{\mu},\ \mathbf{c}) = \prod_{k=1}^{K} q\left(\mu_k;\ m_k,\ s_k^2\right) \prod_{i=1}^{n} q(c_i;\ \phi_i)$ from (4)
- By solving this optimization, we should end up with values for
  - $m_k,\ s_k^2$ for each of the k mixtures, and
  - ϕ_i for every one of the i data points x_i
- So...how do we solve this optimization problem?

5.2 Understanding the mean-field variational family a bit more

- The mean-field family is expressive because it can capture any marginal density of the latent variables. But it cannot capture correlation between latent variables
  - Since we assume by definition that latent variables are mutually independent
- Assume the true posterior that we want to approximate, $p(z|x)$, is a 2-d Gaussian distribution with correlated density
  - This (obviously) means that the true latent variables are correlated
  - Trying to scope this out: the true data generating model is:
    * Two latent variables, parameterized by two hyperparameters. $\mathbf{z} = \{z_1,\ z_2\}$. $z_1 \sim g_1(h_1)$ and $z_2 \sim g_2(h_2)$. So $p(\mathbf{z}) = g_1(h_1)g_2(h_2)$
    * Observed $x_i \sim f(z_1,\ z_2)$. So $p(x_i \mid \mathbf{z} = z_1,\ z_2) = f(z_1,\ z_2)$
    * $\mathbf{x} = \{x_1,\ x_2,\ ...,\ x_n\}$
    * We want to estimate $p(\mathbf{z} \mid \mathbf{x})$ for a given $\mathbf{x}$. We will do this through VI using ELBO maximization, for which we need $p(\mathbf{z},\ \mathbf{x})$ and $q(\mathbf{z})$.

– From the model setup, we know

$$p(\mathbf{z},\ \mathbf{x}) = \ p(\mathbf{x} \mid \mathbf{z})\ p(\mathbf{z})\ = \prod_{i=1}^{n} f(z_1,\ z_2)\ g_1(h_i)\ g_2(h_2)$$

– And using a mean-field variational family, we specify the family $q(\mathbf{z})\ = \prod_{i=1}^{2} q_i(z_i)$

* What this means is: we assume that $p(z_i \mid \mathbf{x})$ for each latent variable z_i is independent of the other z_j

– But if we *know* that the true $p(\mathbf{z} \mid \mathbf{x})\ =\ N(\boldsymbol{\mu}_z,\ \boldsymbol{\Sigma}_z)$ for some non-diagonal covariance matrix, then we know that the mean-field variational family will not be able to capture the covariance structure between the latent variables!

6. Coordinate Ascent Mean-Field Variational Inference

- CAVI is an algorithm to solve the optimization problem specified by the VI model.
  - CAVI iteratively optimizes each factor of the mean-field variational density while holding the others fixed.

Remember: the intent is to find some algorithm that climbs/maximizes $ELBO(q)$, because this minimizes $KL(q(\mathbf{z}) \mid\mid p(\mathbf{z} \mid \mathbf{x}))$, which will give us the ideal $q^*(\mathbf{z})$ that best approximates $p(\mathbf{z} \mid \mathbf{x})$.

6.1 The CAVI algorithm

- For the jth latent variable z_j, the *complete conditional* is its conditional density given all of the other latent variables in the model and the observations, i.e. $p(z_j \mid \mathbf{z}_{-j},\ \mathbf{x})$.
  – As noted earlier, only the assumed approximate factors for the latent variables are independent of each other. The latent variables themselves need not be!
- Fix the other variational factors that are not relevant to the latent variable of interest. That is, fix all $q_l(z_l),\ l \neq j$.
- Then, the optimal $q_j(z_j)$ is proportional to the exponentiated expected log of the complete conditional. That is, $q_j^*(z_j)\ \propto\ \exp\{E_{-q_j}[log(p(z_j \mid \mathbf{z}_{-j},\ \mathbf{x})]\}$.
  – This expectation is over the (currently fixed) variational density over $\mathbf{z}_{-j}$, i.e.

$$\prod_{l \neq j} q_l(z_l)$$

- Proportional usually implies that modulo a linear(?) coefficient, the entities are the same
  – $q_j^*(z_j)\ =\ k \exp\{E_{-q_j}[log(p(z_j \mid \mathbf{z}_{-j},\ \mathbf{x})]\}$
  – Though I don't know if the term proportional here indicates a strictly linear relationship (vs some loose sense of them being in sync with each other).
- What does this really mean?

  - The overarching intent is get a $q_j(z_j)$ that closely approximates $p(z_j \mid \mathbf{x})$.
  - Note that z_j could be correlated with some other latent variables. So in reality $p(z_j \mid \mathbf{x}) = \int p(z_j \mid \mathbf{x}, \mathbf{z}_{-j})\, d(\mathbf{z}_{-j})$.
  - So, one way to approach a good $q_j(z_j)$ is to incorporate information from the other latent variables by conditioning on them.
  - For the most optimal $q_j(z_j)$, $KL\left(q_j^*(z_j) \mid\mid p(z_j \mid \mathbf{x})\right) = 0$. So then $E_{q_j^*}\left(log\left(q_j^*(z_j)\right)\right) = E_{q_j^*}(log(p(z_j \mid \mathbf{x}))) = E_{q_j^*}(E_{-q_j}(log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x}))))$
    * Note that the expected values are equal, but the functions they are calculating expectation over are not equal!
  - This implies that $q_j^*(z_j) \propto \exp\{E_{-q_j}(log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})))\}$
- But remember that the fundamental problem we are trying to solve is that we cannot compute conditionals of the form $p(\mathbf{z}|\mathbf{x})$! So we need to find another way to transform this conditional probability calculation into something we can work with.
- One thing we can work with is the joint distribution $p(\mathbf{z}, \mathbf{x})$! Indeed, we use exactly this in the $ELBO(q)$ computation.
- And we know that that the expected complete conditional over q_{-j} is proportional to the expected joint distribution over q_{-j}!
- That is, $E_{-q_j}(log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x}))) \propto E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x})))$
- Why is this true?
  - Simple Bayes rule. $p(A, B) = p(A \mid B)\, p(B)$. So $p(A \mid B) = \frac{p(A, B)}{p(B)} = \frac{p(A, B)}{\int p(A, B)\, dpA}$. So $p(A \mid B) \propto p(A, B)$!
- Similarly,

$$
\begin{aligned}
& E_{-q_j}(log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x}) \\
&= E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x}) - log(p(\mathbf{z}_{-j}, \mathbf{x})) \\
&= E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x})) - E_{-q_j}(log(p(\mathbf{z}_{-j}, \mathbf{x})) \\
&= E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x})) + const, because\ we\ currently\ fix\ q(\mathbf{z}_{-j})! \\
&\propto E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x}))
\end{aligned}
$$

- So $q_j^*(z_j) \propto \exp\{E_{-q_j}(log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})))\} \propto \exp\{E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x})))\}$ as well!
- Most importantly, we can calculate the joint!
- Conveniently, because the variational factors for each latent variable are independent, the expectation is taken across a joint distribution that does not involve the jth variational factor. So this this a valid coordinate update for $ELBO(q)$!
  - What would make something an invalid coordinate update?
  - Presumably some update rule that uses the probability distribution of the existing state of the coordinate we are trying to update

– But why is this bad?

6.2 Pseudocode for the CAVI algorithm

Input: A model $p(\mathbf{x}, \mathbf{z})$, a data set $\mathbf{x}$

Output: A variational density $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$

Initialize: Variational factors $q_j(z_j)$

- Q: what are they initialized to?

while ($ELBO(q)$ has not converged):

for $j \in \{1, ..., m\}$:

Set $q_j(z_j) \propto \exp\{E_{-j}[log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x}))]\} \leftarrow$ update step

Compute $ELBO(q) = E_q[log(p(\mathbf{z}, \mathbf{x})] + E_q[log(q(\mathbf{z}))]$

return $q(\mathbf{z})$

This eventually returns a local optimum.

Question: We know that this update step approaches the ideal $q_j^*(z_j)$. But why does this update step climb $ELBO(q)$?

6.3 Proof that this update step maximizes $ELBO(q)$

Rewrite $ELBO(q)$ as a function of the jth variational factor $q_j(z_j)$.

$$
\begin{aligned}
&ELBO(q_j) \\
&= E_q[log(p(\mathbf{z}, \mathbf{x}))] - E_q[log(q(\mathbf{z}))] \\
&= E_q[log(p(\mathbf{z}, \mathbf{x}))] - E_q[log(q_1(z_1)q_2(z_2)...q_m(z_m))] \\
&= E_q[log(p(\mathbf{z}, \mathbf{x}))] - E_{q_j}[log(q_j(z_j))] - \prod_{i \neq j} E_{q_i}[log(q_i(z_i))], \ by\ independence\ of\ latent\ variables \\
&= E_{q_j}[E_{-q_j}[log(p(z_j, \mathbf{z}_{-j}, \mathbf{x}) \mid z_j]] - E_{q_j}[log(q_j(z_j))] + const
\end{aligned}
$$

where we rewrite the first term using iterated expecations. $E[A] = E[E[A|B]]$

Recall that the optimal

$q_j^*(z_j) \propto \exp\{E_{-q_j}[log(p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})]\} \propto \exp\{E_{-q_j}[log(p(z_j, \mathbf{z}_{-j}, \mathbf{x}))]\}$

So $log\left(q_j^*(z_j)\right) \propto E_{-q_j}[log(p(z_j, \mathbf{z}_{-j}, \mathbf{x}))] = E_{-q_j}[log(p(\mathbf{z}, \mathbf{x}))]$

Also recall that $KL(p(x) \mid\mid q(x)) = E_{p(x)}[log(p(x))] - E_{p(x)}[log(q(x))]$

So, up to an added constant, the ELBO is simply (negative) KL divergence between $q_j(z_j)$ and $q_j^*(z_j)$!

That is, $ELBO(q(x)) = const - KL(q_j(z_j) \,||\, q_j^*(z_j))$!
So, to maximize $ELBO(q(x))$, we can minimize this KL divergence.
The way to minimize the KL divergence is to make $q_j(z_j)$ as similar to $q_j^*(z_j)$ as possible, which we can accomplish by setting $q_j(z_j) = q_j^*(z_j)$!

6.4 Things to keep in mind

Initialization. $ELBO(q)$ is usually a nonconvex objective function, so CAVI only guarantees convergence to a local optimum, which can be sensitive to init.

Assessing convergence. A good way to do this is to keep a holdout dataset, and use the current learnt $q(\mathbf{z})$ to conduct predictive inference on the holdout at every step.

Numerical stability. Work with log prob instead of probabilities directly. The log-sum-exp trick is useful.

$$log\left(\sum_i \exp(x_i)\right) = \alpha + log\left(\sum_i \exp(x_i - \alpha)\right)$$

Why?

- exponentiate both sides.
- $e^a + e^b = e^\alpha\left(e^{a-\alpha} + e^{b-\alpha}\right)$

Typically, α is set to $\max_i(x_i)$ to provide numerical stability.

7. A complete example: Bayesian Mixture of Gaussians

As earlier, consider a (Bayesian) mixture of K univariate Gaussians, with sample size n

- $\boldsymbol{\mu} = \{\mu_1, ..., \mu_K\}, k = 1, ..., K$
- $p(\mu_k) \sim N\left(0, \sigma^2\right)$, where σ^2 is a hyperparameter
- $c_i \sim categorical(1/K, ..., 1/K), i = 1, ..., n$
  - each c_i is an indicator K-vector
  - $\mathbf{c} = \{c_1, c_2, ..., c_n\} = \{\{0, 0, 1, ..., 0s\ for\ k\ elements\}, \{0, 1, 0, ..., 0s\ for\ k\ elements\}, ..$
- $x_i \mid (c_i, \boldsymbol{\mu}) \sim N\left(c_i^T\boldsymbol{\mu}, 1\right) = p(x_i \mid (c_i, \boldsymbol{\mu})), i = 1, ..., n$
  - $\mathbf{x} = \{x_1, x_2, ..., x_n\}$
  - We assume here that the observation variance = 1

We know:

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

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

Note that there are two types of variational parameters:

1. categorical parameters ϕ_i to approximate the posterior cluster assignment of the ith data point
   - ϕ_i is a vector such that $q(c_i = k) = \phi_{ik}$
2. Gaussian parameters m_k and s_k^2 to approximate the posterior of the kth mixture component.
   - $q(\mu_k) = N\left(m_k, s_k^2\right)$

Set $\mathbf{m} = \{m_1, m_2, ..., m_K\}$, $\mathbf{s^2} = \left\{s_1^2, s_2^2, ..., s_K^2\right\}$, $\boldsymbol{\phi} = \{\phi_1, \phi_2, ..., \phi_n\}$.

Then,

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

CAVI updates each of the $2K + n$ variational parameters in turn, by holding the others constant.

7.1 The variational density of the mixture assignments: the update rule for ϕ_i

Note: Here we consider the variational factor for c_i to be the jth variational factor. So

$$
\begin{aligned}
&q_j = q(c_i;\, \phi_i) \\
&q^*(c_i;\, \phi_i) \\
&\propto \exp\{E_{-q_j}(log(p(z_j, \mathbf{z}_{-j}, \mathbf{x})))\} \\
&= \exp\{E_{-q_j}(log(p(\mathbf{z}, \mathbf{x})))\} \\
&= \exp\left\{\sum_{k=1}^{K} E_{-q_j}\left[log(p(\mu_k));\, m_k,\, s_k^2\right] + \sum_{i=1}^{n} E_{-q_j}[log(p(c_i));\, \phi_i] + \sum_{i=1}^{n} E_{-q_j}\left[log(\, p(x_i \mid (c_i, \boldsymbol{\mu}))); \mathbf{m}, \mathbf{s}^2,\right.\right.
\end{aligned}
$$

The first term is independent of c_i, and therefore resolves to a constant in expectation.
The second is $-nlog(K)$, because $p(c_i) = 1/K$ for all c_i.

$$= \exp\left\{const + const + \sum_{i=1}^{n} E_{-q_j}\left[log(\, p(x_i \mid (c_i, \boldsymbol{\mu}))); \mathbf{m}, \mathbf{s}^2, \phi_i\right]\right\}$$

$$\propto \exp\left\{E_{-q_i}[log(p(x_i \mid c_i, \boldsymbol{\mu}))] + \sum_{j \neq i} E_{-q_j}[log(p(c_j)) + log(p(x_j \mid c_j, \boldsymbol{\mu}))]\right\}$$

As the second term is not dependent on ϕ_i, and we fully specify the distribution of all the variables used

$$= \exp\left\{E_{-q_j}\left[log(p(x_i \mid c_i, \boldsymbol{\mu})); \mathbf{m}, \mathbf{s}^2, \phi_i\right] + const\right\}$$

$$\propto \exp\left\{E_{-q_j}\left[log(p(x_i \mid c_i, \boldsymbol{\mu})); \mathbf{m}, \mathbf{s}^2, \phi_i\right]\right\}$$

We also know that c_i is an indicator variable. So we can write:

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

Then we can write:

$$q^*(c_i; \phi_i)$$

$$\propto \exp\{E_{-q_i}[log(p(x_i \mid c_i, \boldsymbol{\mu}))]\}$$

$$= \exp\left\{E_{-q_j}\left[\log \prod_{k=1}^{K} p(x_i \mid \mu_k)^{c_{ik}}\right]\right\}$$

$$= \exp\left\{E_{-q_j}\left[\sum_{k=1}^{K} c_{ik} \log p(x_i \mid \mu_k)\right]\right\}$$

$$= \exp\left\{\sum_{k=1}^{K} c_{ik}\, E_{-q_j}[\log p(x_i \mid \mu_k)]\right\}$$

From the model spec, we know that $p(x_i \mid \mu) = \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\dfrac{1}{2}\dfrac{(x_i - \mu_k)^2}{\sigma^2}\right\}$

and we know that $\sigma^2 = 1$.

$$= \exp\left\{\sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\log \frac{1}{\sigma\sqrt{2\pi}}\right] - \sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\frac{1}{2}\frac{(x_i - \mu_k)^2}{\sigma^2}\right]\right\}$$

As σ is independent of the variational factors q_{-j},

$$= \exp\left\{const + \sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\frac{-(x_i - \mu_k)^2}{2\sigma^2}\right]\right\}$$

$$\propto \exp\left\{\sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\frac{-\left(x_i^2 + \mu_k^2 - 2x_i\mu_k\right)}{2\sigma^2}\right]\right\}$$

$$\propto \exp\left\{\sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\frac{-\mu_k^2 + 2x_i\mu_k}{2\sigma^2}\right] - \sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\frac{-x_i^2}{2\sigma^2}\right]\right\}$$

$$= \exp\left\{\sum_{k=1}^{K} c_{ik} E_{-q_j}\left[\frac{-\mu_k^2 + 2x_i\mu_k}{2\sigma^2}\right] - const\right]$$

As $\sigma^2 = 1$,

$$q^*(c_i;\ \phi_i) \propto \exp\left\{\sum_{k=1}^{K} c_{ik}\left[x_i E_{-q_j}\left[\mu_k;\ \mathbf{m},\ \mathbf{s^2}\right] - \frac{1}{2} E_{-q_j}\left[\mu_k^2;\ \mathbf{m}\ ,\ \mathbf{s^2}\right]\right]\right\}$$

As the variational factor $q(\mu_k)$ is fixed for this iteration of the algorithm, which means we have values for the $m_k,\ s_k^2$ variational factors, we can compute this value!

Also, we know that c_i is a one-hot K-vector, so $c_{ik} = 0$ for all k-1 indices that do not correspond to the index of the cluster that x_i is drawn from, and $c_{ik} = 1$ for the one index that does.

So, from this form for the update for c_i, we can say for each element ϕ_{ik} of the vector ϕ_i that:

$$\phi_{ik}^* \propto \exp\left\{x_i E_{-q_j}\left[\mu_k;\ m_k,\ s_k^2\right] - \frac{1}{2} E_{-q_j}\left[\mu_k^2;\ m_k,\ s_k^2\right]\right\}$$

$$= \exp\left\{x_i m_k - \frac{1}{2}\left(s_k^2 - m_k^2\right)\right\}$$

7.2 The variational density of the mixture means/variances: the update rule for m_k and s_k^2

Note: Here we consider the variational factor for μ_k to be the kth variational factor. So $q_k = q\left(\mu_k;\ m_k,\ s_k^2\right)$

We start from the same place as we did in 7.1

$$q^*\left(\mu_k;\ m_k,\ s_k^2\right)$$
$$\propto \exp\{E_{-q_k}(log(p(z_j,\ \mathbf{z}_{-j},\ \mathbf{x})))\}$$
$$= \exp\{E_{-q_k}(log(p(\mathbf{z},\ \mathbf{x})))\}$$

$$= \exp\left\{\sum_{i=1}^{n} E_{-q_k}[log(p(c_i));\ \phi_i] + \sum_{j=1}^{K} E_{-q_k}\left[log(p(\mu_j));\ \mathbf{m},\ \mathbf{s^2}\right] + \sum_{i=1}^{n} E_{-q_k}\left[log(\ p(x_i \mid (c_i,\ \boldsymbol{\mu})));\ \mathbf{m},\ \mathbf{s^2},\ q\right.\right.$$

As the first term is independent of the variational factors m_k and s_k^2, it evaluates to a constant. So,

$$\propto \exp\left\{\sum_{j=1}^{K} E_{-q_k}\left[log(p(\mu_j));\ \mathbf{m},\ \mathbf{s^2}\right] + \sum_{i=1}^{n} E_{-q_k}\left[log(\ p(x_i \mid (c_i,\ \boldsymbol{\mu})));\ \mathbf{m},\ \mathbf{s^2},\ \phi_i\right]\right\}$$

For the second term, we know from 7.1 *that* $\log p(x_i \mid c_i,\ \boldsymbol{\mu}) = \sum_{k=1}^{K} c_{ik} \log p(x_i \mid \mu_k)$

$$= \exp\left\{\sum_{j\neq k}^{K} E_{-q_k}\left[log(p(\mu_j));\ \mathbf{m_{-k}},\ \mathbf{s^2_{-k}}\right] + E_{-q_k}\left[\log p(\mu_k); m_k,\ s_k^2\right] + \sum_{i=1}^{n} E_{-q_k}\left[\sum_{j=1}^{K} c_{ij} \log p(x_i \mid \mu_j)\ ;\ \mathbf{m_{-}}\right.\right.$$

As we fully specify the distributions necessary to evaluate the expectation in the first term, it evaluates
And as the second term is independent of the distributions over which the expectation is calculated, it jus

$$\propto \exp\left\{\log p(\mu_k) + \sum_{i=1}^{n} E_{-q_k}\left[\sum_{j=1}^{K} c_{ij} \log p(x_i \mid \mu_j)\ ;\ \mathbf{m_{-k}},\ \mathbf{s^2_{-k}},\ \phi_i\right]\right\}$$

For the first term, we know from the model spec that $\mu_k \sim N(0,\ \sigma^2)$, where σ is a hyperparameter.

So $p(\mu_k) = \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\dfrac{1}{2}\dfrac{{\mu_k}^2}{\sigma^2}\right\}$

And $\log p(\mu_k) = \log \dfrac{1}{\sigma\sqrt{2\pi}} - \dfrac{1}{2}\dfrac{{\mu_k}^2}{\sigma^2}$

$$= \exp\left\{\log \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} + \sum_{i=1}^{n} E_{-q_k}\left[\sum_{j=1}^{K} c_{ij} \log p(x_i \mid \mu_j);\ \mathbf{m_{-k}},\ \mathbf{s_{-k}}^2,\ \phi_i\right]\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} + \sum_{i=1}^{n} E_{-q_k}\left[c_{ik} \log p(x_i \mid \mu_k) + \sum_{j\neq k}^{K} c_{ij} \log p(x_i \mid \mu_j);\ \mathbf{m_{-k}},\ \mathbf{s_{-k}}^2,\ \phi_i\right]\right\}$$

As we fully specify the distributions for the variables used in the last term, it evaluates to a constant.

$$\propto \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} + \sum_{i=1}^{n} E_{-q_k}[c_{ik} \log p(x_i \mid \mu_k);\ \phi_i]\right\}$$

Note that there is no dependence on μ_k and s_k^2 because we exclude these in q_{-k}.
So, $\log p(x_i \mid \mu_k)$ is independent of the distributions we are calculating the expectation over.

$$= \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} + \sum_{i=1}^{n} E_{-q_k}[c_{ik};\ \phi_i] \log p(x_i \mid \mu_k)\right\}$$

A few things to note here:

1. E_{-q_k} only means that we are excluding a distribution over μ_k. It explicitly includes the distribution we need here, which is over c_i. $E_{-q_k}[c_{ik};\ \phi_i] \ = \ E_{q_{c_i}}[c_{ik};\ \phi_i]$

2. When we calculate an expected value, we sample values for the variables involved from the pdf specified, and evaluate the function at the values we have sampled. Sampling c_{ik} from q_{c_i} yields ϕ_{ik} by definition!

$$= \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} + \sum_{i=1}^{n} \phi_{ik} \log p(x_i \mid \mu_k)\right\}$$

And we know from the model spec that $p(x_i \mid \mu_k) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{(x_i - \mu_k)^2}{\sigma^2}\right\}$, where $\sigma^2 = 1$.

$$= \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} + \sum_{i=1}^{n} \phi_{ik}\left(\log \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2}(x_i - \mu_k)^2\right)\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} - \frac{1}{2}\sum_{i=1}^{n} \phi_{ik}\,(x_i - \mu_k)^2\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\frac{{\mu_k}^2}{\sigma^2} - \frac{1}{2}\sum_{i=1}^{n} \phi_{ik}\left(\mu_k^2 - 2x_i\mu_k\right)\right\}$$

$$q^*\left(\mu_k;\, m_k,\, s_k^2\right) \propto \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}+\sum_{i=1}^{n}\phi_{ik}\right)\mu_k^2 \;+\left(\sum_{i=1}^{n}\phi_{ik}x_i\right)\mu_k\right\}$$

This form reveals that the coordinate-optimal variational density of μ_k is a Gaussian.

- I don't understand the details here.

Question: so what are the update equations for the variational parameters m_k and s_k^2?
Try mapping the form of $q^*\left(\mu_k;\, m_k,\, s_k^2\right)$ to the known form of a normal distribution, $p\left(x;\, \mu,\, \sigma^2\right)$.

$$p\left(x;\, \mu,\, \sigma^2\right) \propto \exp\left\{-\frac{1}{2}\frac{(x_i \,-\, \mu\,)^2}{\sigma^2}\right\}$$

$$= \exp\left\{-\frac{1}{2}\frac{\left(x_i^2+\mu^2\, -2x_i\mu\,\right)}{\sigma^2}\right\}$$

$$= \exp\left\{-\frac{1}{2}\frac{x_i^2}{\sigma^2}-\frac{1}{2}\frac{\mu^2}{\sigma^2}+\frac{x_i\mu}{\sigma^2}\right\}$$

Then, mapping the similar entities:

$$-\frac{1}{2}\frac{x_i^2}{\sigma^2} \;\simeq\; -\;\frac{1}{2}\left(\frac{1}{\sigma^2}+\sum_{i=1}^{n}\phi_{ik}\right)\mu_k^2$$

$$so\ s_k^2 = \frac{1}{\frac{1}{\sigma^2}+\sum_{i=1}^{n}\phi_{ik}}$$

We won't get a term for m_k^2 I think? Because we keep removing the constants from the equation. But we may get something for m_k?
Mapping similar entities again:

$$\frac{\mu x_i}{\sigma^2} \simeq \left(\sum_{i=1}^{n} \phi_{ik} x_i \right) \mu_k$$

$$so \ \frac{m_k}{s_k^2} = \left(\sum_{i=1}^{n} \phi_{ik} x_i \right)$$

$$i.e. \ m_k = \left(\frac{\sum_{i=1}^{n} \phi_{ik} x_i}{\frac{1}{\sigma^2} + \sum_{i=1}^{n} \phi_{ik}} \right)$$

7.3 Summary

The update rules are:

1. $\phi_{ik}^* \propto \exp\left\{ x_i m_k - \frac{1}{2}\left(s_k^2 - m_k^2 \right) \right\}$

2. $s_k^2 \leftarrow \frac{1}{\frac{1}{\sigma^2} + \sum_{i=1}^{n} \phi_{ik}}$

3. $m_k \leftarrow \left(\frac{\sum_{i=1}^{n} \phi_{ik} x_i}{\frac{1}{\sigma^2} + \sum_{i=1}^{n} \phi_{ik}} \right)$

Notes from trying to explain things to myself:

1. When we fully specify the distributions for all the variables in a function, the expecation will be a constant
2. If the function is independent of the distributions we are using, the expectation will yield the function
3. if some of the variables in a function have specified distributions and others don't, then we get an expression that is not constant. Try and split it up so you isolate the constant parts from non constant parts

In an exponentiation, anything that you can isolate in a summed term of its own is a constant scaling factor

$$\exp xa + xb - yc - yd = e^{x(a+b) - y(c-d)} \neq \frac{e^{a+b} \cdot e^{x}}{e^{c-d} \cdot e^{y}} !!$$

$$\exp a - b/2 = e^{a - b/2} = \frac{e^{a}}{e^{b/2}} = \frac{e^{a}}{e^{b^{1/2}}}$$

My question is: how we do know when something is constant?
What is *not* constant? Is it the function evaluation itself? or the expectation over the function?
And what does it mean to be constant?

We have currently fixed all $q(\mathbf{z}_{-j})$. What does that mean?
It means that we have fixed all the appropriate variational parameters. So given a z_{-j} value, we can calculate the variational factor for it.
We do have a variational factor for z_j as well, but we are trying to update it.
Why is $E_{-q_j}[\log p(\mu_k)]$ constant? That is, how do we know that we know its value?
When we calculate an expected value over a function, you need to:
1. from the probability density function you are using to calculate expectation over the function of interest, generate a value for the variable
2. get the probability of this variable exhibiting the value you have picked in the probability density function you are using to calculate expectation over the function of interest
3. evaluate the function itself at the value you have generated
4. add prob_from_2 * function_eval_value to the running expected_value
5. repeat for every possible value for the variable
Assuming we know that the variational factor for μ_k is not the excluded factor in the fixed expectation,

$$E_{-q_j}[\log p(\mu_k)] \;=\; \int \log[p(\mu_k)]\; q(\mathbf{z}_{-j})\; d\mathbf{z}_{-j} \;=\; \int \log[p(\mu_k)]\; q\left(\mu_k;\; m_k^{+},\; s_k^{2\,+}\right)\; d\mu_k\; q(\mathbf{z}_{-j,\,-\mu_k})\; d\mathbf{z}_{-j,\,-\mu_k}$$

$$E_{-q_j}[10] \;=\; \int 10\; q(\mathbf{z}_{-j})\; d\mathbf{z}_{-j}$$

Q; Does $\int q(\mathbf{z}_{-j})\; d\mathbf{z}_{-j} \;=\; 1$? It really should...?

$$E_{-q_j}[\log p(x \mid c_i,\; \boldsymbol{\mu})]$$

$$= \int \log p(x \mid c_i,\; \boldsymbol{\mu})\; q\left(\boldsymbol{\mu};\; \mathbf{m},\; \mathbf{s}^2\right)\; d\boldsymbol{\mu}\;\; q(\mathbf{c}_{-j};\; \boldsymbol{\phi}_{-j})\; d\mathbf{c}_{-j}$$

$=$

What is $E_{p(x)}[5y]$? does this even make sense as as statement? → yes it does it is just $5y$
So you are left with a statement that is still in terms of its variables - you cannot resolve it to a constant