In-Depth Variational Inference Tutorial

Chris Xie

June 17, 2016

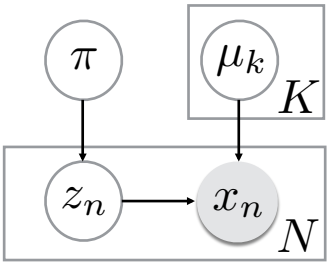
1 Introduction

This tutorial is an in-depth example of how to derive a variational inference (VI) algorithm for a basic graphical model. It was mainly for me, as I had recently learned VI and derived a VI algorithm and stochastic variational inference (SVI) algorithm for a basic model as practice before I started looking at more complex models. But for the reader that is not me, I assume that you have read Bishop's chapter on Variational Inference [1] (at least sections 10.1 and 10.2) and Hoffman's SVI paper [3] for a base understanding of the topic. It also helps to have read David Blei's lecture notes on variational inference.

The purpose of this tutorial is to not skip any details that books or papers skip, which will give readers a better understanding of why certain equations appear in certain papers. I derive in painful detail a VI and SVI algorithm on a simple d dimensional Gaussian mixture model where the variance is known. I also derive a Gibbs sampler (that pops out directly from the derivation of the VI algorithm) for comparison. We show some experimental results to empirically compare variational methods.

2 The Model

The model, as mentioned above, is a d-dimensional Gaussian mixture model with known variance. Since we are performing Bayesian approximate inference (as variational inference/variational Bayes is), our parameters are also random variables. Here is a graphic of the model in plate notation:



We can read off the conditional independence laws from this model. Note that $\{x_n\}_{n=1}^N$ is the observed data. Here I denote the probability distributions:

$$\pi \sim \text{Dir}(\alpha_0),\ p(\pi) = C(\alpha_0) \prod_{k=1}^K \pi_k^{\alpha_0 - 1}$$

$$Z_n|\pi \sim \text{Mult}(1, \pi),\ p(z_n|\pi) = \prod_{k=1}^K \pi_k^{z_{nk}}$$

$$\mu_k \sim N(\mu_0, \Sigma_0),\ p(\mu_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_0|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mu_k - \mu_0)^\intercal \Sigma_0^{-1}(\mu_k - \mu_0)\right)$$

$$X_n|z_n, \mu \sim N\left(\mu_{z_n}, \Sigma\right),\ p(x_n|z_n, \mu) = \prod_{k=1}^K \left[\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k)\right)\right]^{z_{nk}}$$

The distribution for π is a symmetric Dirichlet distribution with parameter α_0 (i.e. the parameter vector is $\alpha_0 \cdot \mathbf{1}$, where $\mathbf{1}$ is a vector of ones). $C(\alpha_0)$ is the normalizing constant of the Dirichlet distribution. μ_0, μ_k, x_n are d-dimensional vectors. Σ_0, Σ are $d \times d$ covariance matrices for the prior on μ_k and the data generating process, respectively.

2.1 Notation

When I write μ, that means the set of all μ_k's. Similarly, when I write z (or Z), I mean the set of all z_n's. z_n is a 1-hot vector, i.e. if a draw from the multinomial distribution selects j, then the j^{th} component is 1 and the rest are 0's. The known covariance here is Σ; we assume that is fixed. $p(\cdot)$ denotes a density function while $\mathbb{P}(\cdot)$ denotes a probability measure.

2.2 Joint Probability

The joint density of this model is

$$p(x, z, \pi, \mu) = p(x|z, \mu)p(z|\pi)p(\mu)p(\pi) \tag{1}$$

This can be read off of the directed graphical model. Some things to note:

$$p(x|z, \mu) = \prod_{n=1}^N p(x_n|z_n, \mu) \tag{2}$$

$$p(z|\pi) = \prod_{n=1}^N p(z_n|\pi) \tag{3}$$

$$p(\mu) = \prod_{k=1}^K p(\mu_k) \tag{4}$$

3 A Brief Review of Variational Inference

In variational inference, there is some probability model with latent variables and observed data. We would like to calculate the true posterior density of the latent variables given the data $p(z, \pi, \mu|x)$, but this is analytically and computationally intractable (typically due to the integral in the denominator, as is the case with most of Bayesian inference). Thus, the goal of variational inference is to approximate the true posterior $p(z, \pi, \mu|x)$ with a "variational" distribution $q(z, \pi, \mu)$, where q lives in a family of simpler, restricted distributions called Q. To compute the approximation, we seek to minimize the KL divergence between q and $p(z, \pi, \mu|x)$. We first derive the Evidence Lower BOund (ELBO), which will be denoted as $\mathcal{L}(q)$. It turns out the ELBO is very related to the KL divergence, as will be shown right after. Let x be our observed data.

$$\begin{aligned}
\ln p(x) &= \ln \int p(x, z, \pi, \mu) dz d\pi d\mu && (5)\\
&= \ln \int q(z, \pi, \mu) \frac{p(x, z, \pi, \mu)}{q(z, \pi, \mu)} dz d\pi d\mu && (6)\\
&= \ln \mathbb{E}_q \left[\frac{p(x, z, \pi, \mu)}{q(z, \pi, \mu)} \right] && (7)\\
&\geq \mathbb{E}_q \left[\ln \frac{p(x, z, \pi, \mu)}{q(z, \pi, \mu)} \right] && (8)\\
&= \mathbb{E}_q \left[\ln p(x, z, \pi, \mu) \right] - \mathbb{E}_q \left[\ln q(z, \pi, \mu) \right] && (9)\\
&:= \mathcal{L}(q) && (10)
\end{aligned}$$

Here we use Jensen's inequality for concave functions to switch the natural log and the expectation operator, which introduces the $\geq$ sign. This works because $\log(\cdot)$ is a concave function. Note the similarity to the derivation of Expectation Maximization (EM).

The ELBO is a lower bound on the marginal probability of the data x. We seek to find a q that maximizes this lower bound because we see that

$$\begin{aligned}
\mathrm{KL}(q(z, \pi, \mu) \;||\; p(z, \pi, \mu|x)) &= \mathbb{E}_q \left[\ln \frac{q(z, \pi, \mu)}{p(z, \pi, \mu|x)} \right] && (11)\\
&= \mathbb{E}_q[\ln q(z, \pi, \mu)] - \mathbb{E}_q \left[\frac{p(x, z, \pi, \mu)}{p(x)} \right] && (12)\\
&= \mathbb{E}_q[\ln q(z, \pi, \mu)] - \mathbb{E}_q[\ln p(x, z, \pi, \mu)] + \mathbb{E}_q[\ln p(x)] && (13)\\
&= -\mathcal{L}(q) + \ln p(x) && (14)
\end{aligned}$$

Our original goal was to minimize the KL divergence between q and p, but it turns out that this is equivalent to maximizing the ELBO w.r.t. q (the log marginal of the data is a constant w.r.t. q). So now our goal has been formulated as finding $q \in Q$ to maximize the ELBO.

3.1 Mean-Field Approximation

KL divergence is nonnegative, and equal to 0 when it's two arguments are equal, i.e. $q(z, \pi, \mu) = p(z, \pi, \mu|x)$. Setting q to be the true posterior defeats the goal in the first place since calculating the true posterior is analytically and computationally intractable. Thus, we like to make the assumption that the variational distribution factorizes in this way

$$q(z, \pi, \mu) = q(\pi) \prod_{n=1}^{N} q(z_n) \prod_{k=1}^{K} (\mu_k) \tag{15}$$

This is called a "mean-field approximation". We can also make a "structured" mean-field approximation by assuming

$$q(z, \pi, \mu) = q(z)q(\pi)q(\mu) \tag{16}$$

or by

$$q(z, \pi, \mu) = q(z)q(\pi, \mu) \tag{17}$$

For the model in section 2, we can assume the latter structured mean-field approximation (Eq (17)). We will show that assuming that actually results in it being equivalent to the full mean-field approximation (Eq (15)).

A note: for the variational distribution q, I suppress some notation; I should really write $q_z(z), q_{\pi,\mu}(\pi, \mu)$ to differentiate the different q's, but that clutters the notation.

Note that we are not restricting the analytical form of the variational distribution q, but instead restricting the flexibility and independence properties of q. In other words, a distribution is part of Q as long as it satisfies the independence assumptions. We could restrict the form of q to a parameterized distribution, and we will do so later for a specific purpose. But for now, let's not assume that for generality.

Recall that in traditional variational inference, maximizing the ELBO is done via coordinate ascent on $\mathcal{L}(q)$. Recall from [1] the formula for the traditional coordinate ascent algorithm:

$$\ln q^*(\pi, \mu) = \mathbb{E}_{q(z)}[\ln p(x, z, \pi, \mu)] + c \tag{18}$$

where c is some normalizing constant. Here, q^* is the exact distribution that maximizes the ELBO w.r.t. its coordinates (we piggyback off of the notation in [1]). Assuming the

structured mean field approximation in Eq (17), we get that

$$\begin{aligned}
\ln q^*(\pi,\mu) &= \mathbb{E}_{q(z)}[\ln \mathbb{P}(x,z,\pi,\mu)] + c && (19)\\
&= \mathbb{E}_{q(z)}[\ln \mathbb{P}(x,z|\pi,\mu)] + \ln \mathbb{P}(\pi,\mu) + c && (20)\\
&= \mathbb{E}_{q(z)}\left[\sum_{n=1}^{N} \ln \mathbb{P}(x_n,z_n|\pi,\mu)\right] + \ln \mathbb{P}(\pi) + \ln \mathbb{P}(\mu) + c && (21)\\
&= \sum_{n=1}^{N} \left(\mathbb{E}_{q(z)}\left[\ln \mathbb{P}(x_n|z_n,\mu)\right] + \mathbb{E}_{q(z)}\left[\ln \mathbb{P}(z_n|\pi)\right]\right) + \ln \mathbb{P}(\pi) + \ln \mathbb{P}(\mu) + c && (22)\\
&= \ln q^*(\pi) + \ln q^*(\mu) && (23)
\end{aligned}$$

(the constant c has been absorbed) Thus, by making the structured mean-field assumption, it turns out this is equivalent to making a more fine-grained structured mean-field approximation in Eq (16). We can actually break this down even further to the full mean-field approximation due to Eq 4 and $\ln \mathbb{P}(x,z|\pi,\mu) = \sum_{n=1}^{N} \ln \mathbb{P}(x_n,z_n|\pi,\mu)$. Thus, for this model, the structured mean-field assumption in Eq (17) is equivalent to the full mean-field assumption in Eq (15). Mathematically, these approximation assumptions are all the same:

$$q(z)q(\pi,\mu), \qquad q(z)q(\pi)q(\mu), \qquad q(\pi)\prod_{n=1}^{N} q(z_n)\prod_{k=1}^{K}(\mu_k)$$

4 Derivation of the Traditional Coordinate Ascent Updates

In order to derive the coordinate ascent updates, we can go about this in two different ways. In Blei's work [2], he uses the update rule (as derived in his lecture notes and in Bishop's chapter [1])

$$\ln q^*(\theta_j) = \mathbb{E}_{-j}[\ln p(x,\theta)] + c \tag{24}$$

where the $\mathbb{E}_{-j}$ means expectation w.r.t. to all latent variables except for j and $\theta = \{z,\pi,\mu\}$ is all of the latent variables (global and local). Note that c is a normalizing constant. Recall from Blei's lecture notes that the notation $q^*(\theta_j)$ simply means the best choice of $q(\theta_j)$ while holding $q(\theta_{-j})$ fixed.

We can often find the analytical form of $q^*(\theta_j)$ by evaluating the expectation. In the case where each complete conditional distribution $p(\theta_j|\theta_{-j},x)$ is in the exponential family, then we can easily evaluate the expectation and find the analytical form of $q^*(\theta_j)$ which also happens to be in the same exponential family, as described in Blei's lecture notes. Because we put conjugate prior distributions on our parameters of our model, we will end up with complete conditional distributions in the exponential family (recall that an exponential family, e.g. Gaussian, is described by a set of parameters). Since we know the form of $q^*(\theta_j)$, we will restrict $q(\theta_j)$ to be in this family of exponential distributions.

This is not the same as letting $q(\theta, x)$ be any distribution that satisfies the independence properties of Q; it is more restricted than that. However, still encompasses the coordinate ascent solutions and allows us to work with finite dimensional parameterizations.

To reflect the restriction of $q(\theta_j)$ to parameterized exponential family distributions, we can endow each distribution with a finite "variational" parameter:

$$q(\pi) = q(\pi|\nu) \tag{25}$$
$$q(\mu_k) = q(\mu_k|\lambda_k) \tag{26}$$
$$q(z_n) = q(z_n|\gamma_n) \tag{27}$$

Using this, we can derive the traditional coordinate ascent updates of our model of interest by taking derivatives of the ELBO w.r.t. the variational parameters and setting them to 0. Also, we can now perform a gradient descent as well. This setup will help us derive the SVI algorithm for this model.

A great exercise is to use Eq (24) to show that when a conditional distribution is in an exponential family, the corresponding variational distribution is in the same family (this is also shown in Blei's lecture notes). The next subsection details this for the variational distributions for the model.

4.1 Forms of Variational Distributions

As mentioned before, when we make the assumption of a conjugate prior on the latent variable, then we get that the posterior of this variable (same as complete conditional) is also in the same exponential family. For example, if $p(\theta_j|\theta_{-j}, x)$ is a Gaussian, we can simply set $q(\theta_j)$ to be parameterized as a Gaussian. In appendix A, we redundantly derive the exact forms of the variational distributions for the latent variables of our model in Section 2.

4.2 Derivatives of the ELBO

Here, we take derivatives of the ELBO to get the traditional coordinate ascent algorithm. Recall that the ELBO looks like

$$\mathcal{L}(q) = \mathbb{E}_q[\ln p(x, z, \pi, \mu)] - \mathbb{E}_q[\ln q(z, \pi, \mu)] \tag{28}$$

For each variational factor, note that we only need to look at the corresponding complete conditional and the variational factor. Everything else is a constant and will go to zero when we take derivatives. This will be more clear when you go through the examples. I also write the ELBO as a function of the variational parameter which I take the derivative with respect to.

4.2.1 Global Variable π

First, let's examine the complete conditional. Since we chose conjugate priors, things turn out nicely (and these exercises are good to try for yourself).

$$p(\pi|x,z,\mu) = p(\pi|z) \propto p(\pi)\prod_{n=1}^{N} p(z_n|\pi) = C(\alpha_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1}\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}} \tag{29}$$

$$= C(\alpha_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1+\sum_{n=1}^{N} z_{nk}} \tag{30}$$

This shows us that $\pi|z \sim \text{Dir}(\alpha')$, where $\alpha'_k = \alpha_0 + \sum_{n=1}^{N} z_{nk}$. In general, when you put a conjugate prior on a latent variable and observe samples of data, then the posterior will look like the prior has simply added some "virtual" data, and will live in the same distribution.

Looking at the corresponding complete conditional and the variational factor, we have

$$\mathcal{L}(\nu) = \mathbb{E}_q[\ln p(x,Z,\pi,\mu)] - \mathbb{E}_q[\ln q(Z,\pi,\mu)] \tag{31}$$

$$= \mathbb{E}_q[\ln p(\pi|x,Z,\mu) + \ln p(x,Z,\mu)] - \mathbb{E}_q[\ln q(\pi|\nu) + \ln q(Z,\mu)] \tag{32}$$

$$= \mathbb{E}_q[\ln p(\pi|Z)] - \mathbb{E}_q[\ln q(\pi|\nu)] + c \tag{33}$$

$$= \sum_{k=1}^{K}\left(\alpha_0 - 1 + \sum_{n=1}^{N}\mathbb{E}_{q(z_n)}[z_{nk}]\right)\mathbb{E}_{q(\pi)}[\ln\pi_k] - \nu^\intercal\mathbb{E}_{q(\pi)}[t(\pi)] + A_\pi(\nu) + c \tag{34}$$

$$= \mathbb{E}_{q(z)}[\alpha' - \mathbf{1}]^\intercal\nabla_\nu A_\pi(\nu) - \nu^\intercal\nabla_\nu A_\pi(\nu) + A_\pi(\nu) + c \tag{35}$$

$$= \left(\mathbb{E}_{q(z)}[\alpha' - \mathbf{1}] - \nu\right)^\intercal\nabla_\nu A_\pi(\nu) + A_\pi(\nu) + c \tag{36}$$

Recall Eq (128). We know from exponential family properties that the expectation of the sufficient statistics is equal to the gradient of the log normalizer. Thus, $\mathbb{E}_{q(\pi)}[\ln\pi] = \nabla_\nu A_\pi(\nu)$.

You should verify for yourself that

$$\nabla_x\left[(c-x)^\intercal\nabla_x f(x)\right] = \nabla_x^2 f(x)(c-x) - \nabla_x f(x) \tag{37}$$

$\nabla_x^2 f(x)$ is the hessian of f evaluated at x. We will use this to take derivatives of the ELBO.

Continuing, we have

$$\nabla_\nu\mathcal{L}(\nu) = \nabla_\nu^2 A_\pi(\nu)\left(\mathbb{E}_{q(z)}[\alpha' - 1] - \nu\right) - \nabla_\nu A_\pi(\nu) + \nabla_\nu A_\pi(\nu) \tag{38}$$

$$= \nabla_\nu^2 A_\pi(\nu)\left(\mathbb{E}_{q(z)}[\alpha' - 1] - \nu\right) \tag{39}$$

From exponential family properties, the Hessian of the log normalizer turns out to be the covariance matrix of the sufficient statistics. Thus, it is always PSD. Assuming it's PD, then setting this gradient to 0 gives us the update equation

$$\nu^* = \mathbb{E}_{q(z)}[\alpha' - 1] \tag{40}$$

and that's our update! This is exactly what we got in Section A.1. I will not cover this here, but it's really cool to see that the natural gradient of the ELBO is

$$\tilde{\nabla}_\nu \mathcal{L}(\nu) = \mathbb{E}_{q(z)}[\alpha' - 1] - \nu \tag{41}$$

when we use the symmetrized KL divergence as a distance metric to measure the distance between the variational parameters (Nick Foti has a great blog post about this here). This will be very useful for the SVI algorithm.

Note that the natural gradient is computationally much cheaper to calculate than the classic gradient.

4.2.2 Global Variable μ_k

First, let's examine the complete conditional.

$$p(\mu_k|x, z, \pi, \mu_{-k}) = p(\mu_k|x, z, \mu_{-k}) \propto p(\mu_k) \prod_{n=1}^{N} p(x_n|z_n, \mu) \tag{42}$$

It is easier to work in log space.

$$\ln p(\mu_k|x, z, \mu_{-k}) = \ln p(\mu_k) + \sum_{n=1}^{N} \ln p(x_n|z_n, \mu) + c \tag{43}$$

$$= \ln p(\mu_k) + \sum_{n=1}^{N} \sum_{j=1}^{k} z_{nj} \ln N(x_n; \mu_j, \Sigma) + c \tag{44}$$

$$= -\frac{1}{2}(\mu_k - \mu_0)^\intercal \Sigma_0^{-1}(\mu_k - \mu_0) + \sum_{n=1}^{N} z_{nk} \left(-\frac{1}{2}(x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k) \right) + c \tag{45}$$

$$= -\frac{1}{2}(\mu_k^\intercal \Sigma_0^{-1} \mu_k - 2\mu_0^\intercal \Sigma_0^{-1} \mu_k + \mu_0^\intercal \Sigma_0^{-1} \mu_0) \tag{46}$$

$$-\frac{1}{2} \sum_{n=1}^{N} z_{nk} \left[x_n^\intercal \Sigma^{-1} x_n - 2x_n^\intercal \Sigma^{-1} \mu_k + \mu_k^\intercal \Sigma^{-1} \mu_k \right] + c \tag{47}$$

$$= -\mu_k^\intercal \left(\frac{1}{2}\Sigma_0^{-1} + \frac{1}{2}\Sigma^{-1} \sum_{n=1}^{N} z_{nk} \right) \mu_k + \left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1} \sum_{n=1}^{N} z_{nk} x_n \right)^\intercal \mu_k + c \tag{48}$$

Due to conjugate priors, we know this will be Gaussian with some parameters $\tilde{\mu}, \tilde{\Sigma}$. Then we can set the above equal to

$$-\frac{1}{2}(\mu_k^\intercal \tilde{\Sigma}^{-1}\mu_k - 2\tilde{\mu}^\intercal\tilde{\Sigma}^{-1}\mu_k + \tilde{\mu}^\intercal\tilde{\Sigma}^{-1}\tilde{\mu}) \tag{49}$$

$$= -\frac{1}{2}(\mu_k - \tilde{\mu})^\intercal\tilde{\Sigma}^{-1}(\mu_k - \tilde{\mu}) \tag{50}$$

To solve for $\tilde{\mu}, \tilde{\Sigma}$, we can match coefficients. Matching the first coefficient, we have

$$-\frac{1}{2}\mu_k^\intercal\tilde{\Sigma}^{-1}\mu_k = -\mu_k^\intercal\left(\frac{1}{2}\Sigma_0^{-1} + \frac{1}{2}\Sigma^{-1}\sum_{n=1}^{N} z_{nk}\right)\mu_k \tag{51}$$

which gives us

$$\tilde{\Sigma}^{-1} = \Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} z_{nk} \tag{52}$$

Matching the second coefficient, we have

$$\mu_k^\intercal\tilde{\Sigma}^{-1}\tilde{\mu} = \mu_k^\intercal\left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}x_n\right) \tag{53}$$

giving us

$$\tilde{\mu} = \left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}\right)^{-1}\left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}x_n\right) \tag{54}$$

Thus, $\mu_k|x, z \sim N(\tilde{\mu}, \tilde{\Sigma})$. An interesting tidbit to note here is that $\mathbb{P}(\mu_k|x, z, \pi, \mu_{-k}) = \mathbb{P}(\mu_k|x, z)$. The Markov blanket for μ_k is $\{x, z, \mu_{-k}\}$, but given x and z, μ_{-k} gives you no information; you only need the x_n's that belong to cluster k.

Going back to the ELBO, we can now take the derivative w.r.t. λ_k. Using Eq (133),

$$\mathcal{L}(\lambda_k) = \mathbb{E}_q[\ln p(\mu_k|x, z, \pi, \mu_{-k})] - \mathbb{E}_q[\ln q(\mu_k|\lambda_k)] + c \tag{55}$$

$$= \mathbb{E}_{q(\mu_k,z)}[\ln p(\mu_k|x, z)] - \mathbb{E}_{q(\mu_k)}[\ln q(\mu_k|\lambda_k)] + c \tag{56}$$

$$= \mathbb{E}_{q(\mu_k,z)}\left[-\frac{1}{2}\mu_k^\intercal\tilde{\Sigma}^{-1}\mu_k + \tilde{\mu}^\intercal\tilde{\Sigma}^{-1}\mu_k\right] - \lambda_k^\intercal\mathbb{E}_{q(\mu_k)}[t(\mu_k)] + A_\mu(\lambda_k) + c \tag{57}$$

$$= \mathbb{E}_{q(z)}\left(\begin{bmatrix}\tilde{\mu}^\intercal\tilde{\Sigma}^{-1} & \left[-\frac{1}{2}\tilde{\Sigma}^{-1}\right]_{[:]}^\intercal\end{bmatrix}\right)\mathbb{E}_{q(\mu_k)}[t(\mu_k)] - \lambda_k^\intercal\mathbb{E}_{q(\mu_k)}[t(\mu_k)] + A_\mu(\lambda_k) + c \tag{58}$$

$$= \left(\mathbb{E}_{q(z)}[\tilde{\beta}] - \lambda_k\right)^\intercal\nabla_{\lambda_k}A_\mu(\lambda_k) + A_\mu(\lambda_k) + c \tag{59}$$

where $\tilde{\beta} = \begin{bmatrix} \tilde{\mu}^\intercal \tilde{\Sigma}^{-1} & \left[-\frac{1}{2}\tilde{\Sigma}^{-1}\right]_{[:]}^\intercal \end{bmatrix}^\intercal$, and $B_{[:]}$ means the stacked column vector by taking the matrix B and stacking into a vector (row-wise or column-wise doesn't matter since we are dealing with symmetric matrices). Taking derivatives, we get

$$\nabla_{\lambda_k} \mathcal{L}(\lambda_k) = \nabla^2_{\lambda_k} A_\mu(\lambda_k) \left(\mathbb{E}_{q(z)}[\tilde{\beta}] - \lambda_k \right) \tag{60}$$

The analysis of this is the same as the previous section. Again, note that the natural gradient is

$$\tilde{\nabla}_{\lambda_k} \mathcal{L}(\lambda_k) = \mathbb{E}_{q(z)}[\tilde{\beta}] - \lambda_k \tag{61}$$

Note that this analysis could be completed using the trace operator instead of the weird matrix stacking that I introduced (however, the derivatives would be more annoying to deal with).

4.2.3 Local Variable z_n

First, let's examine the complete conditional.

$$p(z_n | x, z_{-n}, \pi, \mu) \propto p(x_n | z_n, \mu) p(z_n | \pi) \tag{62}$$

Again, we work in log space.

$$\begin{aligned}
\ln p(z_n | x_n, \pi, \mu) &= \ln p(x_n | z_n, \mu) + \ln p(z_n | \pi) + c && (63) \\
&= \sum_{k=1}^K z_{nk} \left[-\frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left((x_n - \mu_k)^\intercal \Sigma^{-1} (x_n - \mu_k) \right) \right] && (64) \\
&\quad + \sum_{k=1}^K z_{nk} \ln \pi_k + c && (65) \\
&= \sum_{k=1}^K z_{nk} \left[-\frac{1}{2} (x_n - \mu_k)^\intercal \Sigma^{-1} (x_n - \mu_k) \right] + \sum_{k=1}^K z_{nk} \ln \pi_k + c && (66) \\
&= \sum_{k=1}^K z_{nk} \left[\ln \pi_k - \frac{1}{2} \left((x_n - \mu_k)^\intercal \Sigma^{-1} (x_n - \mu_k) \right) \right] + c && (67)
\end{aligned}$$

This shows us that $z_n | x_n, \pi, \mu \sim \text{Mult}(1, \pi')$ where $\pi'_k \propto \pi_k \exp\left(-\frac{1}{2}(x_n - \mu_k)^\intercal \Sigma^{-1} (x_n - \mu_k)\right)$. Note that we will have to normalize π' to make sure it sums to 1. This is because we did not deal with the details in the constant c. Further note that when you expand the squared term in the exponent, there is an $x_n^\intercal \Sigma x_n$ term in the expression for π'_k although $x_n^\intercal \Sigma x_n$ does depend on k. When π'_k is normalized, this term will have no effect. Verifying that is a quick and simple exercise.

Going back to the ELBO, we have

$$\mathcal{L}(\gamma_n) = \mathbb{E}_{q(z_n,\pi,\mu)}[\ln \mathbb{P}(z_n|x_n,\pi,\mu)] - \mathbb{E}_{q(z_n)}[q(z_n|\gamma_n)] + c \tag{68}$$
$$= \mathbb{E}_{q(\pi,\mu)}[\ln \pi']^\intercal \ \mathbb{E}_{q(z_n)}[z_n] - \gamma_n^\intercal \mathbb{E}_{q(z_n)}[t(z_n)] + A_z(\gamma_n) + c \tag{69}$$
$$= \ \left(\mathbb{E}_{q(\pi,\mu)}[\ln \pi'] - \gamma_n\right)^\intercal \nabla_{\gamma_n} A_z(\gamma_n) + A_z(\gamma_n) + c \tag{70}$$

Taking the gradient w.r.t. γ_n, we get

$$\nabla_{\gamma_n} \mathcal{L}(\gamma_n) = \nabla^2_{\gamma_n} A_z(\gamma_n) \left(\mathbb{E}_{q(\pi,\mu)}[\ln \pi'] - \gamma_n\right) \tag{71}$$

and the natural gradient is

$$\tilde{\nabla}_{\gamma_n} \mathcal{L}(\gamma_n) = \mathbb{E}_{q(\pi,\mu)}[\ln \pi'] - \gamma_n \tag{72}$$

4.3 The Updates

Before we can write out the analytical updates to the traditional coordinate ascent algorithm, we first need to compute the expectations in the gradients. Once we have those analytical gradients, then we can construct a batch variational inference algorithm and an SVI algorithm.

4.3.1 Global Variable π

Recall that the natural gradient w.r.t ν is

$$\tilde{\nabla}_{\nu} \mathcal{L}(\nu) = \mathbb{E}_q[\alpha' - 1] - \nu \tag{73}$$

The k^{th} element in this gradient vector is

$$\mathbb{E}_q \left[\alpha_0 + \sum_{n=1}^{N} z_{nk}\right] - 1 - \nu \tag{74}$$
$$= \alpha_0 + \sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}] - 1 - \nu \tag{75}$$
$$= \alpha_0 + \sum_{n=1}^{N} e^{\gamma_{nk}} - 1 - \nu \tag{76}$$

Note that γ_n is a k dimensional vector that is the natural parameters of a Categorical distribution (variant 1 on Wikipedia [4]) and γ_{nk} is the k^{th} component of that vector.

4.3.2 Global Variable μ_k

Recall that the natural gradient w.r.t. λ_k is

$$\tilde{\nabla}_{\lambda_k}\mathcal{L}(\lambda_k) = \mathbb{E}_{q(z)}[\tilde{\beta}] - \lambda_k \tag{77}$$

$$= \mathbb{E}_{q(z)}\left(\begin{bmatrix}\tilde{\Sigma}^{-1}\tilde{\mu}\\ \left[-\frac{1}{2}\tilde{\Sigma}^{-1}\right]_{[:]}\end{bmatrix}\right) - \lambda_k = \left(\begin{bmatrix}\mathbb{E}_{q(z)}\left(\tilde{\Sigma}^{-1}\tilde{\mu}\right)\\ -\mathbb{E}_{q(z)}\left[-\frac{1}{2}\tilde{\Sigma}^{-1}\right]_{[:]}\end{bmatrix}\right) - \lambda_k \tag{78}$$

Looking first at $\mathbb{E}_{q(z)}\left(\tilde{\Sigma}^{-1}\tilde{\mu}\right)$:

$$\mathbb{E}_{q(z)}\left(\tilde{\Sigma}^{-1}\tilde{\mu}\right) = \mathbb{E}_{q(z)}\left[\left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}\right)\left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}\right)^{-1}\left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}x_n\right)\right] \tag{79}$$

$$= \mathbb{E}_{q(z)}\left[\left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}x_n\right)\right] \tag{80}$$

$$= \left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}]x_n\right) \tag{81}$$

$$= \left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} e^{\gamma_{nk}}x_n\right) \tag{82}$$

Next, looking at $-\frac{1}{2}\mathbb{E}_{q(z)}\left[\tilde{\Sigma}^{-1}\right]$:

$$-\frac{1}{2}\mathbb{E}_{q(z)}\left[\tilde{\Sigma}^{-1}\right] = -\frac{1}{2}\mathbb{E}_{q(z)}\left[\left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} z_{nk}\right)\right] \tag{83}$$

$$= -\frac{1}{2}\left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}]\right) \tag{84}$$

$$= -\frac{1}{2}\left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} e^{\gamma_{nk}}\right) \tag{85}$$

Putting it all together, the natural gradient $\hat{\nabla}_{\lambda_k}\mathcal{L}(\lambda_k)$ is

$$\mathbb{E}_{q(z)}\left(\begin{bmatrix}\tilde{\Sigma}^{-1}\tilde{\mu}\\ \left[-\frac{1}{2}\tilde{\Sigma}^{-1}\right]_{[:]}\end{bmatrix}\right) - \lambda_k = \begin{bmatrix}\left(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N} e^{\gamma_{nk}}x_n\right)\\ \left[-\frac{1}{2}\left(\Sigma_0^{-1} + \Sigma^{-1}\sum_{n=1}^{N} e^{\gamma_{nk}}\right)\right]_{[:]}\end{bmatrix} - \lambda_k \tag{86}$$

Note that for the global updates (for π, λ_k), we only need to compute $\sum_{n=1}^{N} e^{\gamma_{nk}}$ and $\sum_{n=1}^{N} e^{\gamma_{nk}} x_n$.

4.3.3 Loval Variable z_n

Recall that the natural gradient w.r.t γ_n is

$$\tilde{\nabla}_{\gamma_n} \mathcal{L}(\gamma_n) = \mathbb{E}_{q(\pi,\mu)}[\ln \pi'] - \gamma_n \tag{87}$$

Since $\sum_{k=1}^{K} e^{\gamma_k} = 1$, we can work with unnormalized values of γ_k and then normalize afterwards. Recalling Eq (67), the (unnormalized) k^{th} component of the vector $\mathbb{E}_{q(\pi,\mu)}[\ln \pi']$ is

$$\mathbb{E}_{q(\pi,\mu_k)}\left[\ln \pi_k - \frac{1}{2}\left((x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k)\right)\right] \tag{88}$$

Again, I separate this into two parts, $\mathbb{E}_{q(\pi)}[\ln \pi_k]$ and $\mathbb{E}_{q(\mu_k)}\left[-\frac{1}{2}\left((x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k)\right)\right]$.

$$\begin{aligned}
\mathbb{E}_{q(\pi)}[\ln \pi_k] = \mathbb{E}_{q(\pi)}[t(\pi)]_k &= \frac{\partial}{\partial \nu_k} A_\pi(\nu) && (89)\\
&= \frac{\partial}{\partial \nu_k}\left(\sum_{l=1}^{K} \ln \Gamma(\nu_l + 1) - \ln \Gamma\left(\sum_{l=1}^{K}(\nu_l + 1)\right)\right) && (90)\\
&= \frac{\partial}{\partial \nu_k} \ln \Gamma(\nu_k + 1) - \frac{\partial}{\partial \nu_k} \ln \Gamma\left(\sum_{l=1}^{K}(\nu_l + 1)\right) && (91)\\
&= \frac{\partial}{\partial(\nu_k + 1)} \ln \Gamma(\nu_k + 1)(1) - \frac{\partial}{\partial\left(\sum_{l=1}^{K}(\nu_l + 1)\right)} \ln \Gamma\left(\sum_{l=1}^{K}(\nu_l + 1)\right)(1) && (92)\\
&= \frac{\partial}{\partial \alpha_k} \ln \Gamma(\alpha_k) - \frac{\partial}{\partial\left(\sum_{k=1}^{K}(\alpha_k)\right)} \ln \Gamma\left(\sum_{k=1}^{K}(\alpha_k)\right) && (93)\\
&= \psi(\alpha_k) - \psi(\hat{\alpha}) && (94)
\end{aligned}$$

We used the chain rule in Eq (92). Also recall that the canonical parameters are related to the natural parameters of the Dirichlet distribution by $\alpha_k = \nu_k + 1$. Lastly, I define $\hat{\alpha} = \sum_{k=1}^{K} \alpha_k$. $\psi(x)$ is called the digamma function and is defined as $\psi(x) = \frac{\partial}{\partial x} \ln \Gamma(x)$. There are many useful approximation implementations (highly accurate, might I add) online, so we will have no trouble numerically computing this.

Now, we look to the second term.

$$-\frac{1}{2}\mathbb{E}_{q(\mu_k)}\left[\left((x_n-\mu_k)^\intercal\Sigma^{-1}(x_n-\mu_k)\right)\right] \tag{95}$$

$$= -\frac{1}{2}\mathbb{E}_{q(\mu_k)}\left[x_n^\intercal\Sigma^{-1}x_n - 2x_n^\intercal\Sigma^{-1}\mu_k + \mu_k^\intercal\Sigma^{-1}\mu_k\right] \tag{96}$$

$$= -\frac{1}{2}\left(x_n^\intercal\Sigma^{-1}x_n - 2x_n\Sigma^{-1}\mathbb{E}_{q(\mu_k)}[\mu_k] + \mathbb{E}_{q(\mu_k)}[\mu_k^\intercal\Sigma^{-1}\mu_k]\right) \tag{97}$$

$$= -\frac{1}{2}\left(x_n^\intercal\Sigma^{-1}x_n - 2x_n\Sigma^{-1}\mathbb{E}_{q(\mu_k)}[\mu_k] + \left[\Sigma^{-1}\right]_{[:]}^\intercal\mathbb{E}_{q(\mu_k)}\left[\mu_k\mu_k^\intercal\right]_{[:]}\right) \tag{98}$$

Remember that $q(\mu_k|\lambda_k) \sim N(\hat{\mu}, \hat{\Sigma})$ is a (d-dimensional) normal distribution with natural parameter $\lambda_k = \begin{bmatrix}\lambda_{k1}\\ \lambda_{k2}\end{bmatrix} = \begin{bmatrix}\hat{\Sigma}^{-1}\hat{\mu}\\ \left[-\frac{1}{2}\hat{\Sigma}^{-1}\right]_{[:]}\end{bmatrix}$. Then

$$\mathbb{E}_{q(\mu_k)}\begin{bmatrix}\mu_k\\ \mu_k\mu_k^\intercal\end{bmatrix} = \begin{bmatrix}\hat{\mu}\\ \hat{\Sigma}+\hat{\mu}\hat{\mu}^\intercal\end{bmatrix} = \begin{bmatrix}-\frac{1}{2}\lambda_{k2}^{-1}\lambda_{k_1}\\ -\frac{1}{2}\lambda_{k2}^{-1} + \frac{1}{4}\lambda_{k2}^{-1}\lambda_{k1}\lambda_{k1}^\intercal\lambda_{k2}^{-1}\end{bmatrix} \tag{99}$$

Note that we are abusing notation here, and assuming that λ_{k2} is in its original matrix form as opposed to its d^2-dimensional stacked vector form. Again, this analysis could have been performed using the trace operator, but since I introduced this weird stacking notation I decided I'll stick with it.

Putting it all together, the k^{th} component of the natural gradient $\hat{\nabla}_{\gamma_n}\mathcal{L}(\gamma_n)$ is

$$\psi(\alpha_k) - \psi(\hat{\alpha}) - \frac{1}{2}\left(x_n^\intercal\Sigma^{-1}x_n - 2x_n\Sigma^{-1}\hat{\mu} + \left[\Sigma^{-1}\right]_{[:]}^\intercal\left[\hat{\Sigma}+\hat{\mu}\hat{\mu}^\intercal\right]_{[:]}\right) - \gamma_{nk} \tag{100}$$

As a reminder, after performing the update, you need to normalize $e^{\gamma_{nk}}$.

5 Computing the ELBO

Recall that traditional variational inference is a coordinate ascent method. We need to calculate the ELBO in order to determine convergence of the algorithm. Like EM, we should always see a monotonically increasing ELBO in our experiments.

Recall that

$$\mathcal{L}(q) = \mathbb{E}_q\left[\ln p(x,z,\pi,\mu)\right] - \mathbb{E}_q\left[q(z,\pi,\mu)\right] \tag{101}$$

$$= \mathbb{E}_{q(\pi)}[\ln p(\pi)] + \sum_{k=1}^{K}\mathbb{E}_{q(\mu_k)}[\ln p(\mu_k)] + \sum_{n=1}^{N}\mathbb{E}_{q(z_n,\pi)}[\ln p(z_n|\pi)] + \sum_{n=1}^{N}\mathbb{E}_{q(z_n,\mu)}[\ln p(x_n|z_n,\mu)] \tag{102}$$

$$- \mathbb{E}_{q(\pi)}[\ln q(\pi|\nu)] - \sum_{k=1}^{K}\mathbb{E}_{q(\mu_k)}[\ln q(\mu_k|\lambda_k)] - \sum_{n=1}^{N}\mathbb{E}_{q(z_n)}[\ln q(z_n|\gamma_n)] \tag{103}$$

I detail each one of the terms here. For the terms regarding the negative cross entropy, we have

$$\mathbb{E}_{q(\pi)}[\ln p(\pi)] = \ln C(\alpha_0) + (\alpha_0 - 1)\sum_{k=1}^{K}\mathbb{E}_{q(\pi)}[\ln \pi_k] = \ln C(\alpha_0) + (\alpha_0 - 1)\mathbf{1}^\intercal \mathbb{E}_{q(\pi)}[t(\pi)] \tag{104}$$

$$\mathbb{E}_{q(\mu_k)}[\ln p(\mu_k)] = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_0| - \frac{1}{2}\left(\left[\Sigma_0^{-1}\right]_{[:]}^\intercal \mathbb{E}_{q(\mu_k)}\left[\mu_k\mu_k^\intercal\right]_{[:]} - 2\mu_0^\intercal\Sigma_0^{-1}\mathbb{E}_{q(\mu_k)}[\mu_k] + \mu_0^\intercal\Sigma_0^{-1}\mu_0\right) \tag{105}$$

$$\mathbb{E}_{q(z_n,\pi)}[\ln p(z_n|\pi)] = \mathbb{E}_{q(z_n,\pi)}\left[\sum_{k=1}^{K} z_{nk}\ln\pi_k\right] = \sum_{k=1}^{K}\mathbb{E}_{q(z_n)}[z_{nk}]\mathbb{E}_{q(\pi)}[\ln\pi_k] \tag{106}$$

$$= \mathbb{E}_{q(z_n)}[t(z_n)]^\intercal\mathbb{E}_{q(\pi)}[t(\pi)] \tag{107}$$

$$\mathbb{E}_{q(z_n,\mu)}[\ln p(x_n|z_n,\mu)] = \mathbb{E}_{q(z_n,\mu)}\left[\sum_{k=1}^{K} z_{nk}\left(-\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}\left(x_n^\intercal\Sigma^{-1}x_n - 2x_n^\intercal\Sigma^{-1}\mu_k + \mu_k^\intercal\Sigma^{-1}\mu_k\right)\right)\right] \tag{108}$$

$$= -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| \tag{109}$$

$$-\frac{1}{2}\sum_{k=1}^{K}\mathbb{E}_{q(z_n)}[z_{nk}]\left(x_n^\intercal\Sigma^{-1}x_n - 2x_n^\intercal\Sigma^{-1}\mathbb{E}_{q(\mu_k)}[\mu_k] + \left[\Sigma^{-1}\right]_{[:]}^\intercal\mathbb{E}_{q(\mu_k)}\left[\mu_k\mu_k^\intercal\right]_{[:]}\right) \tag{110}$$

For each of the terms regarding the entropy of the variational distribution, we have

$$-\mathbb{E}_{q(\pi)}[\ln q(\pi|\nu)] = -\int q(\pi|\nu)\left[\ln h(\pi) + \nu^\intercal t(\pi) - A_\pi(\nu)\right]d\pi \tag{111}$$

$$= -\mathbb{E}_{q(\pi)}[\ln h(\pi)] - \nu^\intercal\mathbb{E}_{q(\pi)}[t(\pi)] + A_\pi(\nu) \tag{112}$$

$$= -\nu^\intercal\mathbb{E}_{q(\pi)}[t(\pi)] + A_\pi(\nu) \tag{113}$$

Note that we used the fact that $h(\pi) = 1$. We have already calculated the expected sufficient statistic in Section 4.3.3.

Recall that $q(\mu_k|\lambda_k) \sim N(\hat{\mu}, \hat{\Sigma})$, where $\lambda_k = \begin{bmatrix}\hat{\mu}^\intercal\hat{\Sigma} & -\frac{1}{2}\left[\hat{\Sigma}^{-1}\right]_{[:]}\end{bmatrix}^\intercal$. We will abuse notation again:

$$-\mathbb{E}_{q(\mu_k)}[\ln q(\mu_k|\lambda_k)] = -\mathbb{E}_{q(\mu_k)}[\ln h(\mu_k)] - \lambda_k^\intercal \mathbb{E}_{q(\mu_k)}[t(\mu_k)] + A_\mu(\lambda_k) \tag{114}$$

$$= \frac{d}{2}\ln(2\pi) - \lambda_k^\intercal \begin{bmatrix} \hat{\mu} \\ \left[\hat{\Sigma} + \hat{\mu}\hat{\mu}^\intercal\right]_{[:]} \end{bmatrix} - \frac{1}{4}\lambda_{k1}^\intercal \lambda_{k2}^{-1}\lambda_{k1} - \frac{1}{2}\ln|-2\lambda_{k2}| \tag{115}$$

$$= \frac{d}{2}\ln(2\pi) - \hat{\mu}^\intercal\hat{\Sigma}^{-1}\hat{\mu} + \frac{1}{2}\left[\hat{\Sigma}^{-1}\right]_{[:]}^\intercal \left[\hat{\Sigma} + \hat{\mu}\hat{\mu}^\intercal\right]_{[:]} + \frac{1}{2}\hat{\mu}^\intercal\hat{\Sigma}^{-1}\hat{\mu} + \frac{1}{2}\ln|\hat{\Sigma}| \tag{116}$$

Using the matrix stacking notation, recall that $\mathrm{Tr}(Axx^\intercal) = x^\intercal Ax = [A]_{[:]}^\intercal [xx^\intercal]_{[:]}$.

Thus,

$$\frac{1}{2}\left[\hat{\Sigma}^{-1}\right]_{[:]}^\intercal \left[\hat{\Sigma} + \hat{\mu}\hat{\mu}^\intercal\right]_{[:]} = \frac{1}{2}\left[\mathrm{Tr}\left(\hat{\Sigma}^{-1}\left(\hat{\Sigma} + \hat{\mu}\hat{\mu}^\intercal\right)\right)\right] = \frac{1}{2}\left(d + \hat{\mu}^\intercal\hat{\Sigma}^{-1}\hat{\mu}\right) \tag{117}$$

Plugging this back into the above, the entropy of the variational distribution for μ_k is

$$\frac{d}{2}\ln(2\pi e) + \frac{1}{2}\ln|\hat{\Sigma}| \tag{118}$$

For z_n,

$$-\mathbb{E}_{q(z_n)}[\ln q(z_n|\gamma_n)] = -\sum_{k=1}^{K}\mathbb{P}(z_{nk}=1)\ln\mathbb{P}(z_{nk}=1) = -\sum_{k=1}^{K} e^{\gamma_{nk}}\ln e^{\gamma_{nk}} \tag{119}$$

$$= -\sum_{k=1}^{K}\gamma_{nk}e^{\gamma_{nk}} \tag{120}$$

An interesting subtlety: Since we are using variant 1 [4], where the parameters are constrained to sum to 1, the expected sufficient statistics does not equal the derivative of the log normalizer because it is a curved exponential family. But since Z_n is a categorical random variable, the entropy is straightforward to calculate.

6 Algorithms

In this section we detail the batch VI (traditional VI), and derive an SVI algorithm along with a Gibbs Sampler for comparison.

We perform some experiments to compare the batch and stochastic VI methods on our model in Section 2.

Algorithm 1 Batch Variational Inference for Bayesian GMM

Require: Data $\{x_n\}_{n=1}^N$

Randomly initialize variational distributions $q(z_n), q(\pi), q(\mu_k)$ for all n, k

while ELBO not converged **do**

  for all n **do**

    Update $q(z_n|\gamma_n)$ by setting $\gamma_n = \mathbb{E}_{q(\pi,\mu)}[\ln \pi']$. Same as setting Eq (100) to 0. Normalize $e^{\gamma_{nk}}$.

  for all k **do**

    Update $q(\mu_k|\lambda_k)$ by setting $\lambda_k = \mathbb{E}_{q(z)} \begin{bmatrix} \tilde{\Sigma}^{-1}\tilde{\mu} \\ -\frac{1}{2}\tilde{\Sigma}^{-1} \end{bmatrix}$. Same as setting Eq (86) to 0.

  Update $q(\pi|\nu)$ by setting $\nu = \mathbb{E}_{q(z)}[\alpha' - 1]$. Same as setting Eq (76) to 0.

6.1 Batch Variational Inference Algorithm

In this section, we pull together all of the details derived in this article to piece together the iterative algorithm for VI. Recall that to maximize the ELBO, you perform coordinate ascent on it. This involves cycling through each of the variational distributions and performing the updates. Note that if the initialization of the variational distributions is deterministic, then the entire algorithm is deterministic. The resulting algorithm is detailed in Algorithm 1.

6.2 Stochastic Variational Inference Algorithm

In this article, we fixed the forms of the variational distributions $q(\theta_j)$ to be in the same family of the optimal distributions found via coordinate ascent $q^*(\theta_j)$. We did most of our derivation with this fix, which is the same setup used to derive SVI in [3]. I hope the reader noticed the parallels between this derivation and in Hoffman's paper.

Our model is in the form described by [3] where the observed data are i.i.d. conditioned on latent parameters. To derive the SVI algorithm, we perform stochastic natural gradient ascent on the ELBO. We do this by first choosing a single data point x_n at each iteration. Next, we perform the traditional coordinate ascent updates for this data point. We then apply a stochastic natural gradient update on the global latent variables as if the dataset was x_n copied N times. Let $\mathcal{L}_n(\cdot)$ be the ELBO assuming the dataset was x_n copied N times (this is a noisy ELBO). For the stochastic natural gradient ascent, we need a decaying step size ρ_i s.t. $\rho_i \to 0$, $\sum_{i=1}^{\infty} \rho_i = \infty$.

To take the derivative of $\mathcal{L}_n$, we could make N copies of the distribution $q(z_n)$ and plug that into existing code. Obviously there will be redundant calculations. The shortcut is to multiply the expected sufficient statistics of the complete conditional distribution by N which is described in [3]. In equations (76) and (86), the summands in the term $\sum_{n=1}^{N} \ldots$ would be the same, making it equal to N times the summand. To make this concrete,

Algorithm 2 Stochastic Variational Inference for Bayesian GMM

Require: Data $\{x_n\}_{n=1}^N$, step sizes $\{\rho_i\}_{i=1}^\infty$, minibatch size S

1: Randomly initialize variational distributions $q(z_n), q(\pi), q(\mu_k)$ for all n, k
2: **repeat**
3: Choose random minibatch of indices $\mathcal{I}$ s.t. $|\mathcal{I}| = S$
4: **for all** $n \in \mathcal{I}$ **do**
5: Update $q(z_n|\gamma_n)$ by setting $\gamma_n = \mathbb{E}_{q(\pi,\mu)}[\ln \pi']$. Same as setting Eq (100) to 0. Normalize $e^{\gamma_{nk}}$.
6: **for all** k **do**
7: Update $q(\mu_k|\lambda_k)$ by setting $\lambda_k^{(t+1)} = \lambda_k^{(t)} + \frac{\rho_i}{S} \sum_{n\in\mathcal{I}} \tilde{\nabla}_\nu \mathcal{L}_n(\nu)$. See Eq (121).
8: Update $q(\pi|\nu)$ by setting $\nu^{(t+1)} = \nu^{(t)} + \frac{\rho_i}{S} \sum_{n\in\mathcal{I}} \tilde{\nabla}_{\lambda_k} \mathcal{L}_n(\lambda_k)$. See Eq (122).
9: **until** forever

taking the natural gradient of the (noisy) ELBO for the global variational parameters would be

$$\tilde{\nabla}_\nu \mathcal{L}_n(\nu) = \alpha_0 + N e^{\gamma_{nk}} - 1 - \nu \tag{121}$$

$$\tilde{\nabla}_{\lambda_k} \mathcal{L}_n(\lambda_k) = \begin{bmatrix} \left(\Sigma_0^{-1}\mu_0 + N\Sigma^{-1} e^{\gamma_{nk}} x_n\right) \\ \left[-\frac{1}{2}\left(\Sigma_0^{-1} + N\Sigma^{-1} e^{\gamma_{nk}}\right)\right]_{[:]} \end{bmatrix} - \lambda_k \tag{122}$$

To perform minibatch stochastic natural gradient ascent, we simply choose S datapoints and average their noisy gradients. It can be shown that the expectation of this noisy natural gradient is equal to the full natural gradient of the normal ELBO (using the entire dataset). The resulting algorithm is detailed in Algorithm 2.

Note that for these types of models where the data is i.i.d. conditioned on the latent parameters, deriving an SVI algorithm is very simple once you've derived the batch VI algorithm; all that is left to do is figure out the gradients of the noisy ELBO.

6.3 Gibbs Sampler

Because we computed the complete conditionals while deriving the coordinate ascent updates in Section 4, we can easily construct a Gibbs sampler. A Gibbs sampler will give you samples from a target distribution (in our case, the posterior that we are interested in). Note that for these types of sampling algorithms, we can get samples from the true posterior distribution in the limit. In this sense, this algorithm is exact while the variational methods are not. However, we can't run our sampling algorithm forever, so we will have to cut it short and get an approximation. Typically, these methods take a very long time to "mix" (converge to the stationary/target distribution), espcially for large models. In this setting, variational methods are preferred due to their computational efficiency. We

Algorithm 3 Gibbs Sampler for Bayesian GMM

Require: Data $\{x_n\}_{n=1}^N$

Randomly initialize variables $z_n^{(0)}, \mu_k^{(0)}, \pi^{(0)}$ for all n, k
repeat
  for all n **do**
    Sample $z_n^{(t+1)} \sim p(z_n|z_{-n}, \mu, \pi, x) = \text{Mult}(1, \pi')$. See Eq (67).
  for all k **do**
    Sample $\mu_k^{(t+1)} \sim p(\mu_k|z, x) = N(\tilde{\mu}, \tilde{\Sigma})$. See Eqs (52) and (54).
  Sample $\pi^{(t+1)} \sim p(\pi|z) = \text{Dir}(\alpha')$. See Eq (29).
until forever

have already written out each complete conditional; thus, we have implicitly derived the algorithm. The resulting algorithm is detailed in Algorithm 3.

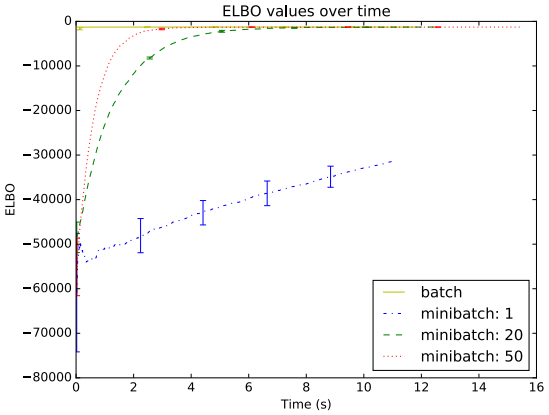
7 Experimental Comparison of VI and SVI

In this section we run some experiments on synthetic data generated from the model in Section 2. We run experiments on a 1-dimensional synthetic dataset and a 2-dimensional synthetic dataset. Both have $N = 1000$ datapoints each. We sample data from $K = 3$ clusters and use this K in our VI/SVI algorithms. We use $\Sigma = I, \alpha_0 = 1, \mu = 0, \Sigma_0 = 3I$. For VI, we run 5 trials with random initializations and 100 iterations for each run. For SVI, we try minibatches of 1, 20, and 50, and run 5 trials with random initializations and 500 iterations for each minibatch size. We use a step size of $\rho_t = \frac{1}{t+d}$, where d is a delay parameter. We set $d = 100$.

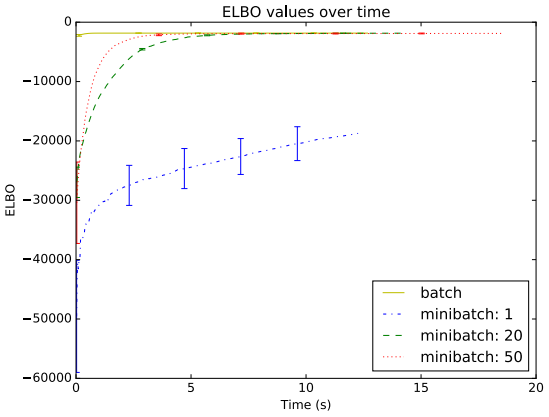
In Figure 1, we show plots of ELBO vs. time. We also show plots of the data along with posterior means which are colored relatively by their weights π. The darker the color, the heavier the weight.

Note that batch VI outperforms minibatch SVI. This is because N is relatively small. In big data settings when N is very large, the minibatch setting will outperform the batch setting since each iteration of the batch setting requires touching the entire dataset. Also note that the minibatch algorithms for sizes 20 and 50 reach the same ELBO as the batch setting.

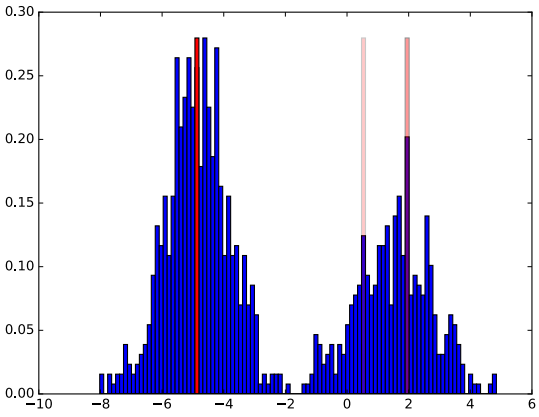
A rough implementation is available at my github.



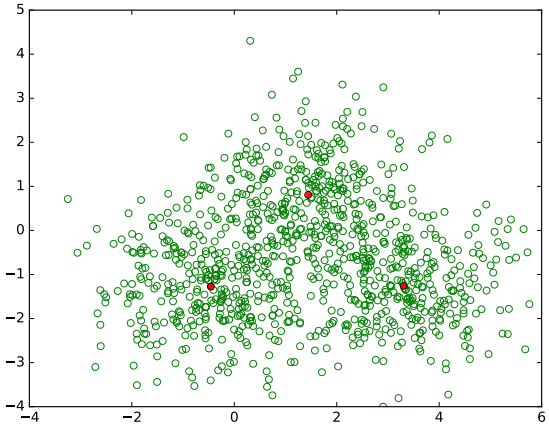
(a) ELBO vs. time for 1-dimensional synthetic data



(b) ELBO vs. time for 2-dimensional synthetic data



(c) Posterior means for for 1-dimensional synthetic data



(d) Posterior means for for 2-dimensional synthetic data

Figure 1: Experimental results for synthetic data

8 How to construct a VI and SVI algorithm

Here I detail the steps you need to take in order to construct a variational inference and/or stochastic variational inference algorithm:

1) Determine the model and the joint probability of all random variables associated with the model.

2) Construct a factorized approximation of the variational distribution (this can be a mean-field, or structured mean-field approximation).

3) For each hidden variable (local and global), find the form of the variational distribution associated with that variable by looking at the complete conditional distribution of that variable. Choosing a conjugate prior of an exponential family distribution makes this much easier to do. This determines the form of the variational distribution.

4) Find the natural gradient by taking the expectation of the log joint density w.r.t. all other hidden variables. Evaluate this expectation to find the exact computations you will need for each update.

5) Write out the ELBO computation.

6) Lastly, use the updates and the ELBO computation to write out either a batch variational inference algorithm or a stochastic variational inference algorithm. Use the ELBO to determine convergence.

9 Conclusion

In this tutorial, we presented an in-depth derivation of a variational inference algorithm on a simple Bayesian Gaussian mixture model. The motivation for this tutorial was to not skip any details of derivations that are absent in other introductory works. I hope that this presentation was helpful to the reader and that it furthered the reader's understanding of the topic.

Appendix

A Variational forms

A.1 $q(\pi|\nu)$

Using the joint probability, we have

$$\begin{aligned}
\ln q^*(\pi) &= \mathbb{E}_{q(z,\mu)}[\ln \mathbb{P}(x, Z, \pi, \mu)] + c && (123)\\
&= \ln \mathbb{P}(\pi) + \sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[\ln \mathbb{P}(Z_n|\pi)] + c && (124)\\
&= \ln C(\alpha_0) + \sum_{k=1}^{K} (\alpha_0 - 1) \ln \pi_k + \sum_{n=1}^{N} \mathbb{E}_{q(z_n)} \left[\sum_{k=1}^{K} z_{nk} \ln \pi_k \right] + c && (125)\\
&= \sum_{k=1}^{K} (\alpha_0 - 1) \ln \pi_k + \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}] \ln \pi_k + c && (126)\\
&= \sum_{k=1}^{K} \ln \pi_k \left[\alpha_0 - 1 + \sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}] \right] + c && (127)
\end{aligned}$$

Note that I have absorbed a bunch of stuff into the constant c. If we take the exponent of both sides, then we see that $q^*(\pi) \sim \text{Dir}(\alpha'), \alpha'_k = \alpha_0 + \sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}]$. Since we put a conjugate prior on π, the complete condition of π is a Dirichlet distribution, which caused the variational distribution of π also be a Dirichlet distribution. $\alpha' - 1$ is the natural parameter (ν) of this distribution. Thus,

$$q(\pi|\nu) = h(\pi) \exp(\nu^\intercal t(\pi) - A_\pi(\nu)), \qquad t(\pi) = \begin{bmatrix} \ln \pi_1 \\ \vdots \\ \ln \pi_k \end{bmatrix} \tag{128}$$

The sufficient statistics of Dirichlet distributions can be found on Wikipedia [4].

A.2 $q(\mu_k|\lambda_k)$

$$\ln q^*(\mu_k) = \mathbb{E}_{q(\pi,\mu_{-k},z)}[\ln \mathbb{P}(x, Z, \pi, \mu_{-k})] + c \tag{129}$$

$$= \ln \mathbb{P}(\mu_k) + \sum_{n=1}^{N} \mathbb{E}_{q(z_n,\mu_{-k})}[\ln \mathbb{P}(x_n|z_n, \mu)] + c \tag{130}$$

$$= \ln \mathbb{P}(\mu_k) + \sum_{n=1}^{N} \mathbb{E}_{q(z_n,\mu_{-k})} \left[\sum_{j=1}^{K} z_{jk} \ln \mathbb{P}(x_n|z_n, \mu_j)\right] + c \tag{131}$$

$$= -\frac{1}{2}(\mu_k - \mu_0)^\intercal \Sigma_0^{-1}(\mu_k - \mu_0) + \sum_{n=1}^{N} \mathbb{E}_{q(z_n)}[z_{nk}] \left(-\frac{1}{2}(x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k)\right) + c \tag{132}$$

This shows that $q^*(\mu_k)$ is a Gaussian distribution. Thus,

$$q(\mu_k|\lambda_k) = h(\mu_k) \exp\left(\lambda_k^\intercal t(\mu_k) - A_\mu(k)\right), \qquad t(\mu_k) = \begin{bmatrix} \mu_k \\ \mu_k \mu_k^\intercal \end{bmatrix} \tag{133}$$

Note that $\mu_k \mu_k^\intercal$ must be stacked into a vector in order for $t(\mu_k) \in \mathbb{R}^{d+d^2}$ to be a vector. Alternatively, we can leave it as a matrix and take the trace.

A.3 $q(z_n|\gamma_n)$

$$\ln q^*(z_n) = \mathbb{E}_{q(\mu)}[\ln \mathbb{P}(x_n|z_n, \mu)] + \mathbb{E}_{q(\pi)}[\ln \mathbb{P}(z_n|\pi)] + c \tag{134}$$

$$= \mathbb{E}_{q(\mu)} \left[\sum_{k=1}^{K} z_{nk} \left(-\frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln|\Sigma| - \frac{1}{2}(x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k)\right)\right] \tag{135}$$

$$+ \mathbb{E}_{q(\pi)} \left[\sum_{k=1}^{K} z_{nk} \ln \pi_k\right] + c \tag{136}$$

$$= \sum_{k=1}^{K} z_{nk} \left[-\frac{1}{2} \mathbb{E}_{q(\mu_k)} \left[(x_n - \mu_k)^\intercal \Sigma^{-1}(x_n - \mu_k)\right] + \mathbb{E}_{q(\pi)}[\ln \pi_k]\right] + c \tag{137}$$

This shows that $q^*(z_n)$ is a Multinomial (with 1 throw, sometimes denoted as Categorical) distribution. This is variant 1 as described on Wikipedia [4]. In this variant, we have that $\sum_{k=1}^{K} e^{\gamma_k} = 1$. Thus,

$$q(z_n|\gamma_n) = h(z_n) \exp(\gamma_n^\intercal t(z_n) - A_z(\gamma_n)), \qquad t(z_n) = z_n \tag{138}$$

References

[1] Christopher M Bishop. "Pattern Recognition". In: *Machine Learning* (2006).

[2] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational Inference: A Review for Statisticians". In: *arXiv preprint arXiv:1601.00670* (2016).

[3] Matthew D Hoffman et al. "Stochastic variational inference". In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 1303–1347.

[4] Wikipedia. *Exponential family — Wikipedia, The Free Encyclopedia*. [Online; accessed 24-March-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Exponential_family.