Variational Bayes Made Easy

Mohammad Emtiyaz Khan emtiyaz.khan@riken.jp
RIKEN Center for Advanced Intelligence Project
1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

Abstract

Variational Bayes is a popular method for approximate inference but its derivation can be cumbersome. To simplify the process, we give a 3-step recipe to identify the posterior form by explicitly looking for linearity with respect to expectations of well-known distributions. We can then directly write the update by simply "reading-off" the terms in front of those expectations. The recipe makes the derivation easier, faster, shorter, and more general.

1. Introduction

Since its introduction in the early 90s (Hinton and Van Camp, 1993; Saul et al., 1996; Jaakkola and Jordan, 1996), variational Bayes (VB) has become a prominent method for approximate inference but, despite significant progress in probabilistic programming, deriving VB updates remains cumbersome for many. The main source of difficulty is the need to derive closed-form expressions for the integrals. For example, consider observed data $\mathbf{y}$ and latent vector $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_K)$ for a model $p(\mathbf{y}, \mathbf{z})$, and assume that we seek a mean-field posterior approximation: $p(\mathbf{z}|\mathbf{y}) \approx q(\mathbf{z}) = \prod_i q_i(\mathbf{z}_i)$. Then, a stationary point $q^*(\mathbf{z})$ of the evidence lower-bound (ELBO) satisfies the following (Bishop, 2006, Eq. 10.9),

$$\log q_i^*(\mathbf{z}_i) = \mathbb{E}_{q_{\setminus i}^*}\left[\log p(\mathbf{y}, \mathbf{z})\right] + \text{const.}, \tag{1}$$

where the integral in the right hand side marginalizes out the rest of the variables (denoted by $\mathbf{z}_{\setminus i}$) using the marginal $q_{\setminus i}^*(\mathbf{z}_{\setminus i})$; see a proof in App. A.

For conjugate-exponential (CE) family models, the integrals have closed-form expressions, but deriving them can still be cumbersome. It is not uncommon to find papers where these derivations take many pages of work. With such long derivations, it is easy to lose the main intuition and elegance of the updates. Difficulty increases when non-conjugate factors are also present. For such cases, despite using automatic differentiation, we still need to make several difficult choices: Which optimizer to choose and are they compatible with the conjugate updating? How to set the learning rate so that overall updates converge? And, when using coordinate-wise updates, in what sequence should we update the variables? Here, we present a recipe to derive the updates that simplify these difficulties.

Our recipe consists of 3-steps based on an alternate way to express Eq. 1. We assume an exponential-family form for $q_{\boldsymbol{\lambda}_i}(\mathbf{z}_i) \propto \exp\left(\langle \mathbf{T}_i(\mathbf{z}_i), \boldsymbol{\lambda}_i \rangle\right)$ with natural parameter $\boldsymbol{\lambda}_i$, sufficient statistics $\mathbf{T}_i(\cdot)$, and a constant base-measure. We can then write Eq. 1 in a form shown below that uses the pair $(\boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$ where $\boldsymbol{\mu}_i(\boldsymbol{\lambda}_i) = \mathbb{E}_{q_{\boldsymbol{\lambda}_i}}[\mathbf{T}_i(\mathbf{z}_i)]$ is the expectation parameter:

$$\boldsymbol{\lambda}_i^* = \nabla_{\boldsymbol{\mu}_i} \, \mathbb{E}_{q_{\boldsymbol{\lambda}}}\left[\log p(\mathbf{y}, \mathbf{z})\right]\big|_{\boldsymbol{\mu}_i = \boldsymbol{\mu}_i^*}. \tag{2}$$

Here, we marginalize with respect to $q_{\boldsymbol{\lambda}}(\mathbf{z})$ where $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_K)$ and denote by $(\boldsymbol{\lambda}_i^*, \boldsymbol{\mu}_i^*)$ the natural and expectation parameter pair for the optimal $q_i^*(\mathbf{z}_i)$. A proof is given in App. A by using results from Khan and Lin (2017), along with an extension for non-constant base-measures. The advantage of this expression is that we do not have to think about the distributional forms of q_i or any integrals. Rather, we can simply look up the definition of $(\boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$; see Table 1 for a list.

We give an example below. For notational ease, we will write $\mathbb{E}_{q_{\boldsymbol{\lambda}}}[\cdot]$ as $\mathbb{E}_q[\cdot]$.

Ex 1 (A simple mixture model) *We consider two components $p_a(\mathbf{y})$ and $p_b(\mathbf{y})$ for an observation $\mathbf{y}$ where mixture indicator $z \in \{0, 1\}$ is sampled from a Bernoulli prior,*

$$p(\mathbf{y}|z) = p_a(\mathbf{y})^z p_b(\mathbf{y})^{1-z}, \textit{ where } p(z) = \pi_0^z(1-\pi_0)^{1-z}.$$

The posterior over z can be obtained by using Bayes' rule, $p(z=1|\mathbf{y}) = \frac{\pi_0 p_a(\mathbf{y})}{\pi_0 p_a(\mathbf{y}) + (1-\pi_0)p_b(\mathbf{y})}$, but let us suppose that we do not know the posterior and we want to recover it from Eq. 2.

Denoting the posterior by $q(z)$, we can figure out its form by expanding

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{y}, z)] &= \mathbb{E}_q[\log p(\mathbf{y}|z) + \log p(z)] \\
&= \mathbb{E}_q\left[z \log p_a(\mathbf{y}) + (1-z)\log p_b(\mathbf{y}) + z \log \pi_0 + (1-z)\log(1-\pi_0)\right] \\
&= \underbrace{\mathbb{E}_q(z)}_{=\mu} \underbrace{\log \frac{\pi_0 p_a(\mathbf{y})}{(1-\pi_0)p_b(\mathbf{y})}}_{\textit{Coeff. in front of } \mu} + \underbrace{\log[(1-\pi_0)p_b(\mathbf{y})]}_{\textit{constant}}.
\end{aligned} \tag{3}$$

The last line suggests to choose $q(z)$ with expectation parameter $\mu = \mathbb{E}_q(z)$. From Table 1, we find Bernoulli distribution to have this expectation parameter, therefore we set $q(z) \propto \pi_1^z(1-\pi_1^{1-z})$ where $\pi_1 > 0$ is the probability of $z=1$ under q.

Having figured out the form of q, we can now use Eq. 2 to find its parameter. For this we first look up from Table 1 the natural parameter: $\lambda = \log \frac{\pi_1}{1-\pi_1}$ and set it according to Eq. 2. It is clear from Eq. 3 that $\mathbb{E}_q[\log p(\mathbf{y}, z)]$ is linear in μ, therefore the gradient $\nabla_\mu \mathbb{E}_q[\log p(\mathbf{y}, z)]$ is simply the coefficient in front of μ. This gives us

$$\lambda^* = \log \frac{\pi_0 p_a(\mathbf{y})}{(1-\pi_0)p_b(\mathbf{y})} \qquad \Longrightarrow \; \pi_1^* = \frac{\pi_0 p_a(\mathbf{y})}{\pi_0 p_a(\mathbf{y}) + (1-\pi_0)p_b(\mathbf{y})}. \tag{4}$$

where in the second equality we rewrite the update in terms of π_1. This recovers the posterior $p(z=1|\mathbf{y})$ obtained with the Bayes' rule.

The procedure is an instance of our 3-step recipe shown below where we identify the posterior form by explicitly looking for linearity with respect to $\boldsymbol{\mu}$ and use it to update $\boldsymbol{\lambda}$:

1. Identify all q_i at once, by looking for linearity of $\mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})]$ w.r.t. some $\boldsymbol{\mu}_i$.
2. Compute $\nabla_{\boldsymbol{\mu}_i} \mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})]$ (hint: for CE, just read-off the coefficient in front of $\boldsymbol{\mu}_i$).
3. Update $\boldsymbol{\lambda}_i \leftarrow (1-\rho_i)\boldsymbol{\lambda}_i + \rho_i \nabla_{\boldsymbol{\mu}_i} \mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})]$ with a learning rate ρ_i (often set to 1).

For most cases, there is no need to derive any integrals: we just expand $\mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})]$ as a multi-linear function of all $\boldsymbol{\mu}_i$ at once and look-up the rest from a table (similar to Table 1).

We can then directly write the update by simply "reading-off" the coefficient in front of $\boldsymbol{\mu}_i$, which is now a function of the $\boldsymbol{\mu}_{\backslash i}$ and observation $\mathbf{y}$.

The steps differ fundamentally from the standard way of deriving VB where each q_i is identified *separately.* Such derivations can be very long and tedious because we need to investigate each node separately by looking for a known distributional form for each of them (Blei et al., 2017, App. A)(Bishop, 2006, Sec. 10.2). The distributional form is not used anyways because the updates are ultimately written and implemented using the parameters of the distributions. Our recipe makes the direct use of the $(\boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$ pair, and leads to an easier, faster, and shorter derivation where all nodes are handled altogether.

The recipe is also more general and covers a wide-variety of cases, including conjugate, non-conjugate, and even deterministic factors. This is due to Step 3 which uses the Bayesian learning rule (BLR) (Khan and Rue, 2021) that contains many algorithms as special cases. Below, we give additional sub-steps for Step 3 to derive many variants:

(3a) Derive Bayes-rule by using $\rho_i = 1$; rewrite it in your parameter of choice (see Ex 1).

(3b) Derive Coordinate Ascent VI (CAVI) and variational message passing (VMP) by setting $\rho_i = 1$ and doing a coordinate-wise update (see Ex 2 and Ex 3).

(3c) Derive Stochastic VI (SVI) by setting $\rho_i = 1$ for the local variables and updating the global variable after a local update on any node i (see Ex 4).

(3d) Derive updates of deterministic nodes (and Laplace's approximation) by invoking the delta method to approximate the expectation (see Ex 5).

(3e) In presence of the non-conjugate terms, just use the gradient and simplify it using reparameterization trick whenever feasible (Khan and Rue, 2021) (see Ex 6).

Another benefit is that, Steps 3a-3d do not only not require any integrals but we do not even need any derivatives either; due to linearity we just need the coefficients in front. With Step 3d, *even MAP estimates can be derived without any derivatives.* Moreover, there is no need to use different optimizers for conjugate and non-conjugate parts, and the learning rate and update schedule need not be set carefully because the update converges under fairly general conditions (Khan et al., 2016). In contrast, coordinate-wise update may not always converge and require the solution to be unique for each coordinate (Paquet, 2014).

2. Examples

Let us see another example with a slightly more complex model where a mean-field approximation is required. We will see that, unlike standard derivation where each factor requires a separate expansion of the log-joint, we can identify both distributions simultaneously. Another advantage is that we do not have to start with a mean-field structure, rather we can figure this out by simply looking for linearity.

Ex 2 (A mixture-model with two levels) *We will add an additional latent variable:* $\pi_0 \rightarrow z_i \rightarrow \mathbf{y}_i$ *where we have multiple* vector *observations* $\mathbf{y}_i$*, and two sets of latent* *variables: z_i and π_0. We assume the following model with a beta prior over π_0,*

$$p(\mathbf{Y}|\mathbf{z}) = \prod_{i=1}^{N} p_a(\mathbf{y}_i)^{z_i} p_b(\mathbf{y}_i)^{1-z_i}, \; p(z_i|\pi_0) = \pi_0^{z_i}(1-\pi_0)^{1-z_i}, \; p(\pi_0) \propto \pi_0^{\alpha_0 - 1}(1-\pi_0)^{\beta_0 - 1}$$

Our goal is to estimate $p(\mathbf{z}, \pi_0|\mathbf{Y})$ but the model is not a conjugate one, therefore we have to use a mean-field approximation for tractability.

In the 1^{st} step we can figure out q for both z_i and π_0 at once by using linearity in the expected log-joint; there is no need to decide the factorization yet. We show this below where the second-line is obtained by using Eq. 3,

$$\begin{aligned}
&\mathbb{E}_q[\log p(\mathbf{Y}|\mathbf{z})p(\mathbf{z})p(\pi_0)] = \mathbb{E}_q\left[\sum_{i=1}^{N}(\log p(\mathbf{y}_i|z_i) + \log p(z_i)) + \log p(\pi_0)\right] \\
&= \mathbb{E}_q\left[\sum_{i=1}^{N}\left(z_i \log\frac{\pi_0 p_a(\mathbf{y}_i)}{(1-\pi_0)p_b(\mathbf{y}_i)} + \log[(1-\pi_0)p_b(\mathbf{y}_i)]\right) + \log\left(\pi_0^{(\alpha_0-1)}(1-\pi_0)^{(\beta_0-1)}\right)\right] \\
&= \sum_{i=1}^{N}\underbrace{\mathbb{E}_q(z_i)}_{=\mu_i}\left(\underbrace{\mathbb{E}_q\begin{bmatrix}\log\pi_0 \\ \log(1-\pi_0)\end{bmatrix}}_{=\boldsymbol{\mu}_0} + \begin{bmatrix}\log p_a(\mathbf{y}_i) \\ \log p_b(\mathbf{y}_i)\end{bmatrix}\right)^{\top}\begin{bmatrix}+1 \\ -1\end{bmatrix} + \sum_i \log p_b(\mathbf{y}_i) \\
&\qquad\qquad\qquad + \underbrace{\mathbb{E}_q\begin{bmatrix}\log\pi_0 \\ \log(1-\pi_0)\end{bmatrix}}_{=\boldsymbol{\mu}_0}^{\top}\begin{bmatrix}\alpha_0 - 1 \\ N + \beta_0 - 1\end{bmatrix}, \qquad (5)
\end{aligned}$$

where in the last step we assumed that $q(\mathbf{z}, \pi_0) = \prod_i q(z_i)q(\pi_0)$ to get linearity with respect to both μ_1 and $\boldsymbol{\mu}_0$. Using Table 1, we have $q(z_i)$ as Bernoulli and $q(\pi_0)$ as Beta.

After this, the 2^{nd} step is straightforward to simply read the coefficient in front of μ_i and $\boldsymbol{\mu}_0$ respectively, and use them in the 3^{rd} step with $\rho_i = 1$. This is shown below in both natural parameters and posterior parameters of Bernoulli (denoted by $\pi_{i,1}$) and Beta distributions (denoted by (α_1, β_1)) obtained by using Table 1:

$$\begin{aligned}
\lambda_i &\leftarrow \left(\boldsymbol{\mu}_0 + \begin{bmatrix}\log p_a(\mathbf{y}_i) \\ \log p_b(\mathbf{y}_i)\end{bmatrix}\right)^{\top}\begin{bmatrix}1 \\ -1\end{bmatrix} \implies \pi_{i,1} \leftarrow \frac{e^{\mathbb{E}_q[\log\pi_0]}p_a(\mathbf{y})}{e^{\mathbb{E}_q[\log\pi_0]}p_a(\mathbf{y}) + e^{\mathbb{E}_q[\log(1-\pi_0)]}p_b(\mathbf{y})} \\
\boldsymbol{\lambda}_0 &\leftarrow \sum_{i=1}^{N}\mu_i\begin{bmatrix}+1 \\ -1\end{bmatrix} + \begin{bmatrix}\alpha_0 - 1 \\ N + \beta_0 - 1\end{bmatrix} \implies \begin{bmatrix}\alpha_1 \\ \beta_1\end{bmatrix} \leftarrow \begin{bmatrix}\alpha_0 + \sum_i \mathbb{E}_q(z_i) \\ \beta_0 + \sum_i(1 - \mathbb{E}_q(z_i))\end{bmatrix} \qquad (6)
\end{aligned}$$

The recipe identifies the distributions for all nodes at once by looking for linearity; see the last line in Eq. 5. In addition, it reuses the already-known integrals by looking-up the required expectations. For instance, in the example above, we need to find $\mathbb{E}_q(z_i)$, $\mathbb{E}_q(\log\pi_0)$, and $\mathbb{E}_q(\log(1-\pi_0))$, whose expressions can be found in Wikipedia pages for Bernoulli and Beta distributions.

We are now ready to discuss the most-commonly used example for VB, which is the Gaussian mixture model. Unlike the standard derivation covered in (Bishop, 2006), we can derive all the updates with just a single expansion of the expected log-joint. The example clearly demonstrates the usefulness of the recipe in simplifying the derivation.

Ex 3 (Gaussian mixture model) *The model uses a Gaussian likelihood for the two components and a Gaussian-Wishart prior on their means and covariances,*

$$
\begin{aligned}
p_a(\mathbf{y}_i) &= \mathcal{N}(\mathbf{y}_i|\mathbf{m}_a, \mathbf{S}_a^{-1}), & p(\mathbf{m}_a, \mathbf{S}_a) &= \mathcal{N}\left(\mathbf{m}_a|0, (\gamma_0\mathbf{S}_a)^{-1}\right)\mathcal{W}(\mathbf{S}_a|\mathbf{W}_0, \nu_0)\\
p_b(\mathbf{y}_i) &= \mathcal{N}(\mathbf{y}_i|\mathbf{m}_b, \mathbf{S}_b^{-1}), & p(\mathbf{m}_b, \mathbf{S}_b) &= \mathcal{N}\left(\mathbf{m}_b|0, (\gamma_0\mathbf{S}_b)^{-1}\right)\mathcal{W}(\mathbf{S}_b|\mathbf{W}_0, \nu_0)
\end{aligned}
$$

for scalars $\gamma_0 > 0$ and $\nu_0 > D-1$, and positive-definite matrix $\mathbf{W}_0$. For this, we simply need to make two changes in Eq. 5. First, we replace $\log p_a(\mathbf{y}_i)$ and $\log p_b(\mathbf{y}_i)$ by their expected values. We show one of them below in terms of the required expectations,

$$
\begin{aligned}
\mathbb{E}_q[\log p_a(\mathbf{y}_i)] &= \tfrac{1}{2}\mathbb{E}_q[\log|\mathbf{S}_a|] - \tfrac{1}{2}\mathbb{E}_q\left[(\mathbf{y}_i - \mathbf{m}_a)^\top\mathbf{S}_a(\mathbf{y}_i - \mathbf{m}_a)\right] + c\\
&= \tfrac{1}{2}\underbrace{\mathbb{E}_q[\log|\mathbf{S}_a|]}_{\boldsymbol{\mu}_{p_a,1}} - \tfrac{1}{2}\,Tr\big(\mathbf{y}_i\mathbf{y}_i^\top\underbrace{\mathbb{E}_q[\mathbf{S}_a]}_{\boldsymbol{\mu}_{p_a,2}}\big) + \mathbf{y}_i^\top\underbrace{\mathbb{E}_q[\mathbf{S}_a\mathbf{m}_a]}_{\boldsymbol{\mu}_{p_a,3}} - \tfrac{1}{2}\underbrace{\mathbb{E}_q[\mathbf{m}_a^\top\mathbf{S}_a\mathbf{m}_a]}_{\boldsymbol{\mu}_{p_a,4}} + c.
\end{aligned}
$$

The term is linear in terms of the expectation-parameter $\boldsymbol{\mu}_{p_a,1:4}$ which corresponds to the Gaussian-Wishart distribution (Table 1); see the Wikipedia pages for their expressions. Second, we need to add the contribution of the prior, which is also linear in $\boldsymbol{\mu}_{p_a,1:4}$,

$$
\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{m}_a, \mathbf{S}_a)] &= \tfrac{1}{2}(\nu_0 - D)\mathbb{E}_q[\log|\mathbf{S}_a|] - \tfrac{1}{2}\,Tr\big(\mathbf{W}_0^{-1}\mathbb{E}_q[\mathbf{S}_a]\big) - \tfrac{1}{2}\mathbb{E}_q[\mathbf{m}_a^\top(\gamma_0\mathbf{S}_a)\mathbf{m}_a] + c\\
&= \tfrac{1}{2}(\nu_0 - D)\boldsymbol{\mu}_{p_a,1} - \tfrac{1}{2}\,Tr\big(\mathbf{W}_0^{-1}\boldsymbol{\mu}_{p_a,3}\big) - \tfrac{1}{2}\gamma_0\boldsymbol{\mu}_{p_a,4} + c.
\end{aligned}
$$

Then, we expand the expected log-joint by using model definition in the first line, then by plugging Eq. 5 to get the second line, and do rearrangement afterward to get linearity,

$$
\begin{aligned}
&\mathbb{E}_q[\log p(\mathbf{Y}|\mathbf{z}, \mathbf{m}_{a:b}, \mathbf{S}_{a:b})p(\mathbf{z})p(\pi_0)p(\mathbf{m}_{a:b}, \mathbf{S}_{a:b})]\\
&= \mathbb{E}_q[\log p(\mathbf{Y}|\mathbf{z}, \mathbf{m}_{a:b}, \mathbf{S}_{a:b})p(\mathbf{z})p(\pi_0)] + \mathbb{E}_q[\log p(\mathbf{m}_{a:b}, \mathbf{S}_{a:b})]\\
&= \sum_{i=1}^{N}\mu_i\left(\boldsymbol{\mu}_0 + \begin{bmatrix}\mathbb{E}_q(\log p_a(\mathbf{y}_i))\\ \mathbb{E}_q(\log p_b(\mathbf{y}_i))\end{bmatrix}\right)^\top\begin{bmatrix}+1\\ -1\end{bmatrix} + \sum_i\mathbb{E}_q[\log p_b(\mathbf{y}_i)] + \boldsymbol{\mu}_0^\top\begin{bmatrix}\alpha_0 - 1\\ N + \beta_0 - 1\end{bmatrix}\\
&\qquad + \mathbb{E}_q[\log p(\mathbf{m}_a, \mathbf{S}_a)] + \mathbb{E}_q[\log p(\mathbf{m}_b, \mathbf{S}_b)]\\
&= \sum_{i=1}^{N}\mu_i\left(\boldsymbol{\mu}_0 + \begin{bmatrix}\frac{1}{2}\boldsymbol{\mu}_{p_a,1} - \frac{1}{2}\,Tr\left(\mathbf{y}_i\mathbf{y}_i^\top\boldsymbol{\mu}_{p_a,2}\right) + \mathbf{y}_i^\top\boldsymbol{\mu}_{p_a,3} - \frac{1}{2}\boldsymbol{\mu}_{p_a,4}\\ \frac{1}{2}\boldsymbol{\mu}_{p_b,1} - \frac{1}{2}\,Tr\left(\mathbf{y}_i\mathbf{y}_i^\top\boldsymbol{\mu}_{p_b,2}\right) + \mathbf{y}_i^\top\boldsymbol{\mu}_{p_b,3} - \frac{1}{2}\boldsymbol{\mu}_{p_a,4}\end{bmatrix}\right)^\top\begin{bmatrix}+1\\ -1\end{bmatrix}\\
&\qquad + \sum_i\left(\tfrac{1}{2}\boldsymbol{\mu}_{p_b,1} - \tfrac{1}{2}\,Tr[\mathbf{y}_i\mathbf{y}_i^\top\boldsymbol{\mu}_{p_b,2}] + \mathbf{y}_i^\top\boldsymbol{\mu}_{p_b,3} - \tfrac{1}{2}\boldsymbol{\mu}_{p_b,4}\right) + \boldsymbol{\mu}_0^\top\begin{bmatrix}\alpha_0 - 1\\ N + \beta_0 - 1\end{bmatrix}\\
&\qquad + \tfrac{1}{2}(\nu_0 - D)\boldsymbol{\mu}_{p_a,1} - \tfrac{1}{2}\,Tr\left(\mathbf{W}_0^{-1}\boldsymbol{\mu}_{p_a,2}\right) - \tfrac{1}{2}\gamma_0\boldsymbol{\mu}_{p_a,4}\\
&\qquad + \tfrac{1}{2}(\nu_0 - D)\boldsymbol{\mu}_{p_b,1} - \tfrac{1}{2}\,Tr\left(\mathbf{W}_0^{-1}\boldsymbol{\mu}_{p_b,2}\right) - \tfrac{1}{2}\gamma_0\boldsymbol{\mu}_{p_b,4} + const.
\end{aligned}
$$

The last equation looks complicated but it is linear with respect to each $\mu_i, \boldsymbol{\mu}_0, \boldsymbol{\mu}_{p_a}, \boldsymbol{\mu}_{p_b}$, therefore we can simply look up the coefficient in front to write the update. The updates of $q(\pi_0)$ remain same as before because the coefficient is unchanged. For $q(z_i)$, the coefficient takes an expectation over $\log p_a$ and $\log p_b$, giving us the following,

$$\pi_{i,1} \leftarrow \frac{e^{\mathbb{E}_q[\log \pi_0] + \mathbb{E}_q[\log p_a(\mathbf{y})]}}{e^{\mathbb{E}_q[\log \pi_0] + \mathbb{E}_q[\log p_a(\mathbf{y})]} + e^{\mathbb{E}_q[\log(1-\pi_0)] + \mathbb{E}_q[\log p_b(\mathbf{y})]}} \tag{7}$$

The update for the mean and covariance of the component is also obtained in a straightforward manner. For $q(\mathbf{m}_a, \mathbf{S}_a)$, for instance, we set its 4 natural parameters (given in Table 1) to the coefficients in front of $\boldsymbol{\mu}_{p_a,1}$ to $\boldsymbol{\mu}_{p_a,4}$ respectively, and rearrange to get

$$\begin{aligned} \begin{bmatrix} \frac{1}{2}(\nu_a - D) \\ -\frac{1}{2}(\mathbf{W}_a^{-1} + \gamma_a \mathbf{m}_a \mathbf{m}_a^\top) \\ \gamma_a \mathbf{m}_a \\ -\frac{1}{2}\gamma_a \end{bmatrix} &= \begin{bmatrix} \frac{1}{2}\sum_i \mu_i + \frac{1}{2}(\nu_0 - D) \\ -\frac{1}{2}\sum_i \mu_i \mathbf{y}_i \mathbf{y}_i^\top - \frac{1}{2}\mathbf{W}_o^{-1} \\ \sum_i \mu_i \mathbf{y}_i \\ -\frac{1}{2}\sum_i \mu_i - \frac{1}{2}\gamma_0 \end{bmatrix} \\ \implies \begin{bmatrix} \nu_a \\ \mathbf{W}_a^{-1} \\ \mathbf{m}_a \\ \gamma_a \end{bmatrix} &= \begin{bmatrix} \sum_i \mu_i + \nu_0 \\ \sum_i \mu_i \mathbf{y}_i \mathbf{y}_i^\top - \frac{(\sum_i \mu_i \mathbf{y}_i)(\sum_i \mu_i \mathbf{y}_i)^\top}{\sum_i \mu_i + \gamma_0} + \mathbf{W}_0^{-1} \\ \frac{1}{\gamma_a}\sum_i \mu_i \mathbf{y}_i \\ \sum_i \mu_i + \gamma_0 \end{bmatrix} \end{aligned} \tag{8}$$

We encourage the reader to verify that these updates are same as ones derived in (Bishop, 2006, Eqs 10.49, 10.58, 10.60-10.63), but unlike that derivation, all updates here are derived by using just one expansion of expected log-joint, which leads to a shorter and much faster derivation.

A few additional examples are given in App. B.

3. Future Work

One drawback of the recipe is that it depends heavily on the $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ parameterization. There are several works that extend beyond this, for example, to mixture distributions (Lin et al., 2019), structured Gaussian distributions (Lin et al., 2021), and also to transformation families (Kıral et al., 2023). Such approaches have shown to be useful in deriving existing algorithms, as well as designing new ones (Khan and Rue, 2021). We believe that such extensions exist for generic distributions and that it is possible to derive all sorts of updates in the same fashion as described in this paper.

Acknowledgments

I would like to thank Dr. Negar Safinianaini and other members of our group for discussions which not only compelled me to write this paper but also allowed me to see the value of techniques I have been using to derive VB for many years.

References

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Annual Conference on Computational Learning Theory*, pages 5–13, 1993.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

T. Jaakkola and M. Jordan. A variational approach to Bayesian logistic regression problems and their extensions. In *International conference on Artificial Intelligence and Statistics*, 1996.

M. E. Khan and W. Lin. Conjugate-computation variational inference: converting variational inference in non-conjugate models to inferences in conjugate models. In *International Conference on Artificial Intelligence and Statistics*, pages 878–887, 2017.

M. E. Khan and H. Rue. The Bayesian learning rule. *arXiv preprint arXiv:2107.04562*, 2021.

M. E. Khan, R. Babanezhad, W. Lin, M. Schmidt, and M. Sugiyama. Faster stochastic variational inference using proximal-gradient methods with general divergence functions. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.

E. M. Kıral, T. Möllenhoff, and M. E. Khan. The Lie-group Bayesian learning rule. In *International conference on Artificial Intelligence and Statistics*, 2023.

W. Lin, M. E. Khan, and M. Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. 2019.

W. Lin, F. Nielsen, M. E. Khan, and M. Schmidt. Tractable structured natural-gradient descent using local parameterizations. In *International Conference on Machine Learning*, pages 6680–6691. PMLR, 2021.

U. Paquet. On the convergence of stochastic variational inference in Bayesian networks. *NIPS Workshop on variational inference*, 2014.

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

J. Winn and C. M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694, 2005.

A. VB Stationary Point, The Linearity Property and Equivalence of Eqs. 1 and 2

VB stationary point: We first give a short proof of the stationarity condition Eq. 1, which essentially follows by rewriting the ELBO as a function of q_i alone and expressing it as a Kullback-Leibler divergence (KLD) term,

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})] + \sum_i \mathbb{E}_q[-\log q_i(\mathbf{z}_i)] \\ &= \mathbb{E}_{q_i}\Big[\mathbb{E}_{q_{/i}}[\log p(\mathbf{y}, \mathbf{z})] - \log q_i(\mathbf{z}_i)\Big] + \text{const.} \\ &= \mathbb{D}_{\text{KL}}[q_i(\mathbf{z}_i) \,\|\, \tilde{p}_i(\mathbf{z}_i)] + \text{const.}\end{aligned} \tag{9}$$

where $\tilde{p}_i(\mathbf{z}_i) \propto \exp[\mathbb{E}_{q_{\setminus i}}[\log p(\mathbf{y}, \mathbf{z})]]$ is a distribution where the rest $\mathbf{z}_{\setminus i}$ is marginalized out. The minimum occurs when the two arguments in the divergence are equal, which gives us the condition in Eq. 1.

The linearity property: This is due to the fact that the CE terms in log-joint are multi-linear in sufficient statistics (Winn and Bishop, 2005, p. 668). Below, we give a formal statement which is the basis of Step 1.

Theorem 1 *If the conditional $p(\mathbf{z}_i|\mathbf{z}_{\setminus i}, \mathbf{y})$ is conjugate to $q_i(\mathbf{z}_i)$, that is, if there exists $\boldsymbol{\eta}_i(\cdot)$ such that the log-conditional can be expressed in terms of $\mathbf{T}_i(\mathbf{z}_i)$ as follows,*

$$\log p(\mathbf{z}_i|\mathbf{z}_{\setminus i}, \mathbf{y}) = \langle \mathbf{T}_i(\mathbf{z}_i)\,,\, \boldsymbol{\eta}_i(\mathbf{z}_{\setminus i}, \mathbf{y}_i)\rangle + \textit{const.}, \tag{10}$$

then $\mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})] = \langle \boldsymbol{\mu}_i\,,\, \mathbb{E}_{q_{\setminus i}}\left[\boldsymbol{\eta}_i(\mathbf{z}_{\setminus i}, \mathbf{y}_i)\right]\rangle + \textit{const.}$, which is linear with respect to $\boldsymbol{\mu}_i$.

This also justifies Step 2 because the gradient with respect to $\boldsymbol{\mu}_i$ becomes

$$\nabla_{\boldsymbol{\mu}_i}\mathbb{E}_q[\log p(\mathbf{y}, \mathbf{z})] = \mathbb{E}_{q_{\setminus i}}\left[\boldsymbol{\eta}_i(\mathbf{z}_{\setminus i}, \mathbf{y}_i)\right] \tag{11}$$

which is simply the term in front of $\boldsymbol{\mu}_i$.

Equivalence: Due to Eq. 2, we can show that, at a fixed point, the optimal natural parameter is equal to the coefficient in front of $\boldsymbol{\mu}_i$,

$$\boldsymbol{\lambda}_i^* = \nabla_{\boldsymbol{\mu}_i}\, \mathbb{E}_q\left[\log p(\mathbf{y}, \mathbf{z})\right]\big|_{\boldsymbol{\mu}_i = \boldsymbol{\mu}_i^*} = \mathbb{E}_{q^*_{\setminus i}}\left[\boldsymbol{\eta}_i(\mathbf{z}_{\setminus i}, \mathbf{y}_i)\right], \tag{12}$$

Using this, we can show that Eq. 2 leads to Eq. 1 by using the definition of $q_i^*(\mathbf{z}_i)$,

$$\begin{aligned}\log q_i^*(\mathbf{z}_i) &= \langle \mathbf{T}(\mathbf{z}_i), \boldsymbol{\lambda}_i^*\rangle + c, \\ &= \langle \mathbf{T}(\mathbf{z}_i), \mathbb{E}_{q^*_{\setminus i}}\left[\boldsymbol{\eta}_i(\mathbf{z}_{/i}, \mathbf{y}_i)\right]\rangle + c \\ &= \mathbb{E}_{q^*_{\setminus i}}\left[\log p(\mathbf{y}, \mathbf{z})\right] + c',\end{aligned} \tag{13}$$

where the second and third line follow by using Eq. 12 and Eq. 10 respectively (c and c' are constants that do not depend on $\mathbf{z}_i$). The derivation is reversible and we can also derive Eq. 2 from Eq. 1, making the two statements equivalent whenever the posterior is a conjugate-exponential family.

Extension for non-constant base-measures: If the exponential-family includes a base-measure $h_i(\mathbf{z}_i)$ as shown below,

$$q_i(\mathbf{z}_i) \propto h_i(\mathbf{z}_i) \exp\left(\langle \mathbf{T}_i(\mathbf{z}_i), \boldsymbol{\lambda}_i \rangle\right),$$

then the condition in Eq. 2 is modified to the following,

$$\boldsymbol{\lambda}_i^* = \nabla_{\boldsymbol{\mu}_i} \mathbb{E}_q \left[\log p(\mathbf{y}, \mathbf{z}) - \log h_i(\mathbf{z}_i)\right]\big|_{\boldsymbol{\mu}_i = \boldsymbol{\mu}_i^*}, \tag{14}$$

The proof is exactly the same so we omit it.

B. Additional Examples

We now discuss Sub-steps (3b)-(3f) and show that, by using the BLR in Step 3, we can derive many other types of algorithms from the VB updates. It makes sense to start from a more Bayesian approach and make approximation when required, rather than doing it the other way. Using the BLR in Step 3 does exactly this. For example, as we saw in the previous examples, we can write a CAVI or VMP style update as special cases by choosing $\rho_i = 1$, and doing coordinate-wise updates. But the same approach can be used to derive SVI (Hoffman et al., 2013), described below for completeness.

Ex 4 (SVI updates for the mixture model with two levels) *Instead of a coordinate-wise update that goes through all z_i, we can pick randomly just one z_i, do the update with $\rho_i = 1$ using Eq. 4, and after this we update the global variable π_0 with $\rho_0 < 1$,*

$$\begin{aligned} \boldsymbol{\lambda}_0 &\leftarrow (1-\rho_0)\boldsymbol{\lambda}_0 + \rho_0 \sum_{i=1}^N \mu_i \begin{bmatrix} +1 \\ -1 \end{bmatrix} + \begin{bmatrix} \alpha_0 - 1 \\ N + \beta_0 - 1 \end{bmatrix} \\ \implies \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} &\leftarrow (1-\rho_0) \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} + \rho_0 \begin{bmatrix} \alpha_0 + \sum_i \mathbb{E}_q(z_i) \\ \beta_0 + N - \sum_i \mathbb{E}_q(z_i) \end{bmatrix} \end{aligned} \tag{15}$$

Updating π_0 after every z_i update can speed up convergence (Hoffman et al., 2013).

Now, we give another example on matrix factorization where we show how to derive updates for deterministic nodes and also for expectation maximization. We derive the well-known alternating least-squares and probabilistic PCA (Tipping and Bishop, 1999). Both of these are obtained by using the delta method where the expectation parameter is approximated by using its mean: $\mathbb{E}_q(\mathbf{z}\mathbf{z}^\top) \approx \mathbf{m}\mathbf{m}^\top$, where $\mathbf{m}$ is the mean of q. More details on such delta method is given in Khan and Rue (2021).

Ex 5 *Matrix factorization and latent factor models aim to fit data matrix $\mathbf{Y}$ of size $N \times D$ by using two sets of factors $\mathbf{U}$ and $\mathbf{V}$ respectively of sizes $N \times K$ and $D \times K$ where K is the number of factors. The likelihood and the prior are given by,*

$$p(\mathbf{Y}|\mathbf{U}, \mathbf{V}) = \prod_{i=1}^N \prod_{j=1}^D \mathcal{N}(y_{ij} | \mathbf{u}_i^\top \mathbf{v}_j, \mathbf{I}), \quad p(\mathbf{u}_i) = \mathcal{N}(\mathbf{u}_i | 0, \mathbf{I}/\delta_u), \quad p(\mathbf{v}_j) = \mathcal{N}(\mathbf{v}_j | 0, \mathbf{I}/\delta_v)$$

where y_{ij} is the ij'th entry, $\mathbf{u}_i$ is the i'th row of $\mathbf{U}$, and $\mathbf{v}_j$ is the j'th row of $\mathbf{V}$. The most popular procedure is to use the alternating least-squares (ALS) procedure, but there is also expectation maximization (Tipping and Bishop, 1999) and VMP Paquet (2014). These can all be derived by using our recipe for VB, as we show now.

Following the 1^{st} step, we expand the expected log-joint and look for linearity,

$$\mathbb{E}_q[\log p(\mathbf{Y},\mathbf{U},\mathbf{V})] = \mathbb{E}_q\left[\sum_{i=1}^{N}\left(\sum_{j=1}^{D} -\frac{1}{2}(y_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 - \frac{\delta_v}{2}\mathbf{v}_j^\top\mathbf{v}_j\right) - \frac{\delta_u}{2}\mathbf{u}_i^\top\mathbf{u}_i\right] + const$$

$$= \sum_{i=1}^{N}\left(\sum_{j=1}^{D} -\frac{1}{2}\left(Tr\left(\underbrace{\mathbb{E}_q(\mathbf{u}_i\mathbf{u}_i^\top)}_{=\boldsymbol{\mu}_{u_i}^{(2)}}\underbrace{\mathbb{E}_q(\mathbf{v}_j\mathbf{v}_j^\top)}_{=\boldsymbol{\mu}_{v_j}^{(2)}}\right) - 2y_{ij}\underbrace{\mathbb{E}_q(\mathbf{u}_i)}_{=\boldsymbol{\mu}_{u_i}^{(1)}}{}^\top\underbrace{\mathbb{E}_q(\mathbf{v}_j)}_{\boldsymbol{\mu}_{v_j}^{(1)}}\right)\right.$$
$$\left. - \frac{\delta_v}{2} Tr\underbrace{\mathbb{E}_q\left(\mathbf{v}_j\mathbf{v}_j^\top\right)}_{=\boldsymbol{\mu}_{v_j}^{(2)}}\right) - \frac{\delta_u}{2} Tr\underbrace{\mathbb{E}_q\left(\mathbf{u}_i\mathbf{u}_i^\top\right)}_{=\boldsymbol{\mu}_{u_i}^{(2)}} + const$$

Here, we use mean-field approximation for all $\mathbf{u}_i$ and $\mathbf{v}_j$ because we want linearity with respect to all. The sufficient statistics correspond to Gaussian (third row in Table 1).

The 2^{nd} step is to simply read off as the coefficients in the front, and, by using the natural parameters given in Table 1, we can directly write the update from the 3^{rd} step. The first two updates below use the natural parameters denoted by $(\mathbf{S}_{u_i}\mathbf{m}_{u_i}, -\mathbf{S}_{u_i}/2)$ corresponding to $(\boldsymbol{\mu}_{u_i}^{(1)}, \boldsymbol{\mu}_{u_i}^{(2)})$, while the next two updates do the same for the variable $\mathbf{v}_j$,

$$\mathbf{S}_{u_i}\mathbf{m}_{u_i} \leftarrow (1-\rho_i)\mathbf{S}_{u_i}\mathbf{m}_{u_i} + \rho_i \sum_{j=1}^{D} \boldsymbol{\mu}_{v_j}^{(1)} y_{ij}$$
$$\mathbf{S}_{u_i} \leftarrow (1-\rho_i)\mathbf{S}_{u_i} + \rho_i\left(\sum_{j=1}^{D}\boldsymbol{\mu}_{v_j}^{(2)} + \delta_u \mathbf{I}_K\right)$$
$$\mathbf{S}_{v_j}\mathbf{m}_{v_j} \leftarrow (1-\rho_j)\mathbf{S}_{v_j}\mathbf{m}_{v_j} + \rho_j \sum_{i=1}^{N} \boldsymbol{\mu}_{u_i}^{(1)} y_{ij}$$
$$\mathbf{S}_{v_j} \leftarrow (1-\rho_j)\mathbf{S}_{v_j} + \rho_j\left(\sum_{i=1}^{N}\boldsymbol{\mu}_{u_i}^{(2)} + \delta_v \mathbf{I}_K\right).$$

We can then specialize these updates to derive ALS, EM, VB etc.

1. *Setting $\rho_i = \rho_j = 1$ and updating posteriors of $\mathbf{U}$ and $\mathbf{V}$ in alternate iterations (that is coordinate wise updates), we get the VMP update.*

$$\begin{aligned} \mathbf{S}_{u_i}\mathbf{m}_{u_i} &\leftarrow \sum_{j=1}^{D} \boldsymbol{\mu}_{v_j}^{(1)} y_{ij}, & \mathbf{S}_{u_i} &\leftarrow \sum_{j=1}^{D} \boldsymbol{\mu}_{v_j}^{(2)} + \delta_u \mathbf{I}_K \\ \mathbf{S}_{v_j}\mathbf{m}_{v_j} &\leftarrow \sum_{i=1}^{N} \boldsymbol{\mu}_{u_i}^{(1)} y_{ij}, & \mathbf{S}_{v_j} &\leftarrow \sum_{i=1}^{N} \boldsymbol{\mu}_{u_i}^{(2)} + \delta_v \mathbf{I}_K. \end{aligned} \tag{16}$$

2. *We can get the probabilistic PCA updates (Tipping and Bishop, 1999), where we only compute the posterior wrt $\mathbf{U}$ and assume $\mathbf{V}$ to be deterministic. We can do this by making two more additional approximations,*

   (a) denote $\mathbf{m}_{v_j}$ by $\hat{\mathbf{v}}_j$, and

   (b) use the delta approximation for $\boldsymbol{\mu}_{v_j}^{(2)} = \mathbb{E}_q[\mathbf{v}_j\mathbf{v}_j^\top] \approx \mathbb{E}_q[\mathbf{v}_j]\mathbb{E}_q[\mathbf{v}_j]^\top = \hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^\top$.

   With these, the update in Eq. 16 *reduces to*

$$\begin{aligned} \mathbf{S}_{u_i}\mathbf{m}_{u_i} \leftarrow \sum_{j=1}^{D} \hat{\mathbf{v}}_j y_{ij}, \quad \mathbf{S}_{u_i} \leftarrow \sum_{j=1}^{D} \hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^\top + \delta_u \mathbf{I}_K \\ \hat{\mathbf{v}}_j \leftarrow \left(\sum_{i=1}^{N} \boldsymbol{\mu}_{u_i}^{(2)} + \delta_v \mathbf{I}_K \right)^{-1} \sum_{i=1}^{N} \boldsymbol{\mu}_{u_i}^{(1)} y_{ij}. \end{aligned} \tag{17}$$

   These are equivalent to those derived in Bishop (2006, Eqs. 25-17).

3. *Similarly, we get the ALS updates by further*

   (a) denoting $\mathbf{m}_{u_i}$ by $\hat{\mathbf{u}}_i$, and

   (b) adding the delta approximation: $\boldsymbol{\mu}_{u_i}^{(2)} = \mathbb{E}_q[\mathbf{u}_i\mathbf{u}_i^\top] \approx \mathbb{E}_q[\mathbf{u}_i]\mathbb{E}_q[\mathbf{u}_i]^\top = \hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^\top$.

   With these, Eq. 17 *reduces to the following ALS scheme,*

$$\begin{aligned} \hat{\mathbf{u}}_i &\leftarrow \left(\sum_{j=1}^{D} \hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^\top + \delta_u \mathbf{I}_K \right)^{-1} \sum_{j=1}^{D} \hat{\mathbf{v}}_j y_{ij}, \\ \hat{\mathbf{v}}_j &\leftarrow \left(\sum_{i=1}^{N} \hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^\top + \delta_v \mathbf{I}_K \right)^{-1} \sum_{i=1}^{N} \hat{\mathbf{u}}_i^\top y_{ij}, \end{aligned} \tag{18}$$

No derivatives are used to derive ALS (a MAP estimation procedure). It is also possible to update all quantities in parallel by using all learning rates < 1, and convergence is less of an issue unlike VMP (Paquet, 2014).

Finally, we briefly discuss the inclusion of non-conjugate terms.

Ex 6 *Suppose that, in Ex 1, we decided to use a non-conjugate prior, say, a logit-normal distribution with mean* m, $p(\pi_0) \propto \frac{1}{\pi_0(1-\pi_0)} \exp\left[-\frac{(logit(\pi_0)-m)^2}{2}\right]$. *Still, the derivation proceeds in the same way by expanding* $\mathbb{E}_q[\log p(\mathbf{Y}|\mathbf{z})p(\mathbf{z})p(\pi_0)] =$

$$\left(\sum_{i=1}^{N} \underbrace{\mathbb{E}_q(z_i)}_{=\mu_i} \underbrace{\mathbb{E}_q\left[\begin{array}{c} \log \pi_0 \\ \log(1-\pi_0) \end{array}\right]^\top}_{=\boldsymbol{\mu}_0^\top} \underbrace{\left[\begin{array}{c} p_a(\mathbf{y}_i) \\ p_b(\mathbf{y}_i) \end{array}\right]}_{\textit{Coeff. in front}}\right) + N \underbrace{\mathbb{E}_q[\log(1-\pi_0)]}_{\mu_0^{(2)}} + \underbrace{\mathbb{E}_q[\log p(\pi_0)]}_{\textit{non-conjugate term}} .$$

and combine the last two terms by collecting the linear terms together,

$$\underbrace{\mathbb{E}_q\left[\begin{array}{c} \log \pi_0 \\ \log(1-\pi_0) \end{array}\right]^\top}_{=\boldsymbol{\mu}_0^\top} \underbrace{\left[\begin{array}{c} -1 \\ N-1 \end{array}\right]}_{\textit{Coeff. in front}} + \underbrace{\mathbb{E}_q\left[-\frac{(logit(\pi_0)-m)^2}{2}\right]}_{\textit{non-conjugate term}} .$$

We will now just have to add the last term to the derivative of $\nabla_{\boldsymbol{\mu}_0}$, *and rewrite the update. The form of the update does not change much, rather the non-conjugate prior can simply be written as a "pseudo" conjugate Beta prior,*

$$\boldsymbol{\lambda}_0 \leftarrow (1-\rho_0)\boldsymbol{\lambda}_0 + \rho_0 \left(\left[\begin{array}{c} \hat{\alpha}_0 - 1 \\ N + \hat{\beta}_0 - 1 \end{array}\right] + \sum_{i=1}^{N} \mu_i \left[\begin{array}{c} p_a(\mathbf{y}_i) \\ p_b(\mathbf{y}_i) \end{array}\right]\right).$$

where $\hat{\alpha}_0 = \nabla_{\mu_0^{(1)}} \mathbb{E}_q\left[-\frac{1}{2}(logit(\pi_0)-m)^2\right]$ *and* $\hat{\beta}_0 = \nabla_{\mu_0^{(2)}} \mathbb{E}_q\left[-\frac{1}{2}(logit(\pi_0)-m)^2\right]$ *are the parameters of the pseudo beta prior. The gradients automatically convert the non-conjugate prior form into a conjugate one, which is due to Khan and Lin (2017).*

The elegance of the update above is not a coincidence, rather it is by design. Khan and Lin (2017) linearize the whole VB objective with respect to all $\boldsymbol{\mu}$. This linearizes the non-conjugate terms while preserving the linearity of the conjugate terms; see (Khan and Lin, 2017, Lemma 1). This is a remarkable property of such linearizations. The BLR builds on this but it is the linearization which enables generalization to all sorts of algorithms. This insight is under-appreciated so far, but we do hope that the examples shown in this paper motivates many readers to exploit such (Bayesian) linearization procedure.

Table 1: Exponential-family used in this paper. For Gaussian-Wishart, we use two sets of variables: $\mathbf{z}_1$ is a real vector and $\mathbf{Z}_2$ is a positive-definite matrix. We do not give the expressions for the expectation parameters explicitly but these can be found in Wikipedia, with a more exhaustive list covering many other distributions.

Name	Distribution $q(\mathbf{z})$	Expectation Param $\boldsymbol{\mu}$	Natural param $\boldsymbol{\lambda}$
Bernoulli	$\propto \pi^z(1-\pi^{1-z})$	$\mathbb{E}_q(z)$	$\log \frac{\pi}{1-\pi}$
Beta	$\propto z^{\alpha-1}e^{-\beta z}$	$\mathbb{E}_q \begin{bmatrix} \log z \\ \log(1-z) \end{bmatrix}$	$\begin{bmatrix} \alpha - 1 \\ \beta - 1 \end{bmatrix}$
Gaussian	$\propto e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m})^\top \mathbf{S}(\mathbf{z}-\mathbf{m})}$	$\mathbb{E}_q \begin{bmatrix} \mathbf{z} \\ \mathbf{z}\mathbf{z}^\top \end{bmatrix}$	$\begin{bmatrix} \mathbf{S}\mathbf{m} \\ -\frac{1}{2}\mathbf{S} \end{bmatrix}$
Gaussian-Wishart	$\propto$ $\lvert\mathbf{Z}_2\rvert^{\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{z}_1-\mathbf{m})^\top \gamma \mathbf{Z}_2 (\mathbf{z}_1-\mathbf{m})}$ $\lvert\mathbf{Z}_2\rvert^{\frac{\nu-D-1}{2}} e^{-\frac{1}{2}\mathrm{Tr}(\mathbf{W}^{-1}\mathbf{Z}_2)}$	$\mathbb{E}_q \begin{bmatrix} \log\lvert\mathbf{Z}_2\rvert \\ \mathbf{Z}_2 \\ \mathbf{Z}_2\mathbf{z}_1 \\ \mathbf{z}_1^\top \mathbf{Z}_2 \mathbf{z}_1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2}(\nu - D) \\ -\frac{1}{2}(\mathbf{W}^{-1} + \gamma\mathbf{m}\mathbf{m}^\top) \\ \gamma\mathbf{m} \\ -\frac{1}{2}\gamma \end{bmatrix}$