<INCOMPLETE>
Variational Inference: A Review for Optimizers

David Merrell
dmerrell@cs.wisc.edu

February, 2018

Abstract

This document is the **evil twin** of a recent paper by Blei et al. entitled *Variational Inference: A Review for Statisticians* [1]. In that paper, the authors present the foundational concepts of Variational Inference (VI) and list some open problems in VI research. It provides a useful starting point for anyone interested in the field.

Variational Inference is at the intersection of statistics and optimization. While the original *Review* paper is explicitly aimed at statisticians, this document attempts to fill a similar role for optimization researchers interested in VI. Requisite statistical background can be found in appendix A.

1 Introduction and Motivation

Exact Bayesian inference is intractable for most statistical models. A variety of approximate inference methods have been invented in response. Among these, sampling methods like MCMC have had a long and celebrated history.

More recently, Variational Inference (VI) methods have arisen as an alternative to the sampling approach. VI methods reduce the inference task to an optimization problem—they search within a family of approximating distributions for one that is "closest" to the true posterior in some sense.

More precisely, VI frames Bayesian inference as the following optimization task:

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} KL\big(q(z) \| p(z|x)\big), \tag{1}$$

where $\mathcal{Q}$ is a *variational family* of distributions, $p(z|x)$ is the exact posterior distribution, and KL denotes the Kullback-Leibler Divergence.

The goal of this document is to clarify the meaning of this optimization problem and to present foundational algorithms, as described by Blei et al. in their *Review* paper [1]. We do so by fully working out a concrete problem: inference on a Gaussian Mixture model. We will make generalizations along the way.

The document is structured as follows:

- We give a high-level view of the Variational Inference problem in section 2.
- We present the Gaussian Mixture model and view it through the lens of VI in section 3.
- We apply two foundational VI algorithms to the Gaussian mixture model in sections 4 and 5, showing our work in full detail.
- Requisite statistical background is summarized in Appendix A; the idea was to make this document self-contained, and accessible to optimization researchers.

2 A High-Level View of VI

Bayesian inference. Let $p(x, z)$ be a statistical model; a joint probability distribution where

- x are *observed variables*—data is instantiated from them.
- z are *unobserved variables*—these can include "latent variables" and model parameters.

One can think of a statistical model as the description of a process that is only partially observed. Given what we *can* observe about the process (e.g., its outputs), what can we infer about the unobserved parts of the process?

The typical goal of Bayesian inference is to compute the *posterior distribution*:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{2}$$

$$= \frac{p(x|z)p(z)}{\int p(x, z)dz} \tag{3}$$

That is: given some data x, update the distribution for hidden variables z. It turns out that, for all but the simplest models, the integral in the denominator either

- has no closed form, or
- is computationally intractable (e.g., exponential cost to compute).

The whole motivation of Variational Inference methods is that they sidestep this intractability. Instead of computing the exact posterior, they search through a *family* of distributions for one that is closest to the exact posterior.

The VI optimization problem. Let $\mathcal{Q}$ be a family of distributions—we refer to it as a *variational family*. We seek the member of $\mathcal{Q}$ with minimal KL-Divergence from the true posterior:

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} KL\Big(q(z) \Big\| \, p(z|x)\Big)$$

It turns out that this KL-Divergence is itself uncomputable:

$$KL\Big(q(z) \Big\| \, p(z|x)\Big) = \mathop{\mathbb{E}}_{z\sim q}\left[\log \frac{q(z)}{p(z|x)}\right].$$

Notice that $p(z|x)$—the posterior distribution—appears in this expression. The posterior is the thing we were trying to find in the first place. Even if we rewrite it as $\frac{p(z,x)}{p(x)}$, the denominator $p(x)$ is typically intractible to evaluate. So evaluating the KL-divergence directly is a futile effort.

However, rewriting $p(z|x)$ as $\frac{p(z,x)}{p(x)}$ does allow us to obtain a different quantity called the Evidence Lower Bound (ELBO).

$$ELBO(q; p) = \mathop{\mathbb{E}}_{z\sim q}\left[\log \frac{p(z, x)}{q(z)}\right]$$

$$= \log\big(p(x)\big) - KL\Big(q(z) \Big\| \, p(z|x)\Big)$$

ELBO is the negation of KL-Divergence, up to an added term that is constant with respect to q.

Although we cannot evaluate KL-Divergence, we can still minimize it by *maximizing* the ELBO. In short, maximizing ELBO is **equivalent** to minimizing KL-Divergence. Variational Inference is most commonly framed as ELBO maximization:

$$q^*(z) = \arg\max_{q \in \mathcal{Q}} ELBO(q; p)$$

We note in passing that the choice of variational family $\mathcal{Q}$ is very important for VI. Our goal is to find a *distribution* that makes an *integral functional* (i.e., ELBO) stationary. A judicious choice of $\mathcal{Q}$ reduces this infinite-dimensional calculus of variations problem to a finite-dimensional optimization problem without sacrificing too much fidelity. Intuitively, we want $\mathcal{Q}$ that is sufficiently complex to closely approximate $p(z|x)$, but simple enough to permit efficient optimization.

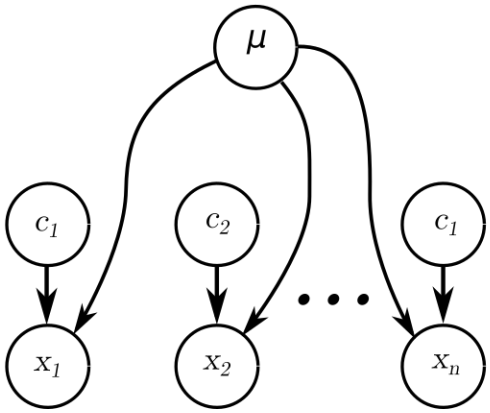

Figure 1: Bayesian network depicting a simple Gaussian Mixture model.

3 A Concrete Problem: Gaussian Mixtures

In this section, we present the classic Gaussian Mixture model. We show that exact inference is expensive for this model, and take some initial steps to prepare it for Variational Inference.

3.1 The Gaussian Mixture Model

Throughout this document, we will work on a classic statistical model called the Gaussian Mixture. It models data that appear in *clusters.* We depict it using a Bayes net in figure 1 and a probabilistic program in figure 2.

One can think of a Gaussian Mixture as a generative process: a vector $\vec{\mu}$ of K means is generated by a normal distribution. Then each data point is generated by *(i)* randomly selecting one of those means, and *(ii)* randomly drawing from a Gaussian distribution centered at that mean. This is written out explicitly in figure 2.

Figure 2: A pseudocode probabilistic program for our Gaussian Mixture model.

```
% Generate K means
for k=1:K
    mu(k) = normal(0,sigma^2);
end
% Generate the data
for i=1:n
    c(i) = categorical(1.0/K, ... , 1.0/K);  % assign a cluster
    x(i) = normal(mu(c(i)), 1.0);            % generate a data point
end
```

The Gaussian Mixture's PDF is given as follows:

$$p(\mathbf{x}, \mathbf{c}, \vec{\mu}) = p(\vec{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \vec{\mu}) \tag{4}$$

$$= N(\vec{\mu}; 0, \sigma^2 I) \prod_{i=1}^{n} DU(c_i; 1, K) N(x_i; \mu_{c_i}, 1) \tag{5}$$

where we have abused notation a bit: N denotes a gaussian PDF and DU denotes a discrete uniform PMF.

Now reconsider the Bayesian inference task, specifically for the Gaussian Mixture:

$$p(\mathbf{c}, \vec{\mu}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{c}, \vec{\mu})p(\mathbf{c}, \vec{\mu})}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x}, \mathbf{c}, \vec{\mu})}{p(\mathbf{x})}$$

The numerator is easy to evaluate; it's simply the PDF given in equation 5. The denominator is the source of troubles for exact inference:

$$\begin{aligned} p(\mathbf{x}) &= \int \sum_{i=1}^{n} \sum_{k=1}^{K} p(\mathbf{x}, c_i = k, \vec{\mu}) d\vec{\mu} \\ &= \int \left[p(\vec{\mu}) \sum_{i=1}^{n} \sum_{k=1}^{K} \prod_{j=1}^{n} p(c_i = k) p(x_j|\vec{\mu}, c_i = k) \right] d\vec{\mu} \\ &= \sum_{\mathbf{c}} p(\mathbf{c}) \int p(\vec{\mu}) \prod_{i=1}^{n} p(x_i|c_i, \vec{\mu}) d\vec{\mu} \end{aligned}$$

The sum over $\mathbf{c}$ in the last line is a sum over all K^n possible configurations for the cluster assignment vector, $\mathbf{c}$. It turns out that each term of the sum is a computable integral since $p(\vec{\mu})$ is a conjugate prior for $p(x_i|c_i, \vec{\mu})$. However, the number of terms grows exponentially with the size of the dataset; so exact inference is intractable.

We have shown that exact inference is prohibitively expensive for the Gaussian Mixture model. This calls for approximate inference—VI, for example. However, before we use Variational Inference on the Gaussian Mixture model we will do the following:

- choose a variational family $\mathcal{Q}$;
- compute its ELBO and think of how we might maximize that quantity.

3.2 Choosing a Variational Family $\mathcal{Q}$ for Gaussian Mixtures

One of the simplest and most common choices of variational family is the Mean Field Family. It's simply the family of joint distributions where all variables are assumed to be independent of each other:

$$\mathcal{Q}_{MFF} = \left\{ q \;\middle|\; q(z) = \prod_{j=1}^{m} q_j(z_j) \right\}$$

In the Gaussian Mixture's case, the hidden variables are $\vec{\mu}$ and $\mathbf{c}$. If we assume that each mean μ_k is normally distributed, and that each cluster assignment c_i is categorically distributed, we get the following Mean Field Family for the Gaussian Mixture problem:

$$q(\mathbf{c}, \vec{\mu}) = \prod_{k=1}^{K} q_k(\mu_k; m_k, s_k^2) \prod_{i=1}^{n} q_i(c_i; \varphi_i) \tag{6}$$

where m_k, s_k^2, φ_i are *variational parameters*; i.e., parameters that index our variational family.

Parenthetically: it turns out that the choice of normal $\vec{\mu}$ and categorical $\mathbf{c}$ is in fact optimal, given our model and our restriction to the Mean Field Family. We'll show how to derive this in section 4.2.

3.3 Evaluating the ELBO for Gaussian Mixtures

Equipped with a model $p(x, z)$ and a variational family $\mathcal{Q}$, we will consider the task of finding $q \in \mathcal{Q}$ that is "closest" to $p(z|x)$. Recall that VI poses this as ELBO-maximization.

$$q^*(\mathbf{c}, \vec{\mu}) = \arg\max_{q \in \mathcal{Q}} ELBO(q; p)$$

For our particular scenario, we have the following ELBO:

$$
\begin{aligned}
ELBO(q;p) &= \mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log \frac{p(\mathbf{x},\mathbf{c},\vec{\mu})}{q(\mathbf{c},\vec{\mu})}\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log \frac{p(\vec{\mu})\prod_i p(c_i)p(x_i|\vec{\mu},c_i)}{\prod_k q(\mu_k;m_k,s_k^2)\prod_i q(c_i;\varphi_i)}\right] \\
&= \sum_{k=1}^{K} \mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log p(\mu_k)\right] \\
&\quad + \sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log p(c_i)\right] + \mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log p(x_i;c_i,\mu)\right]\right) \\
&\quad - \sum_{i=1}^{n} \mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log q(c_i;\varphi_i)\right] - \sum_{k=1}^{K} \mathop{\mathbb{E}}_{\mathbf{c},\vec{\mu}\sim q}\left[\log q(\mu_k;m_k,s_k^2)\right]
\end{aligned}
$$

Each of these expectations can be evaluated, yielding a closed form for the ELBO:

$$
\begin{aligned}
ELBO(\mathbf{m},\mathbf{s},\varphi) = &-\frac{1}{2\sigma^2}\sum_{k=1}^{K}(m_k^2+s_k^2) \\
&- \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\varphi_{ik}\left[(m_k-x_i)^2+s_k^2\right] \\
&- \sum_{i=1}^{n}\sum_{k=1}^{K}\varphi_{ik}\log(\varphi_{ik}) \\
&+ \frac{1}{2}\sum_{k=1}^{K}\log(2\pi s_k^2) \quad + \text{const.}
\end{aligned}
$$

Hence, our problem consists of choosing variational parameters $\mathbf{m}, \mathbf{s}$, and φ that maximize this quantity. Note that this is a constrained optimization problem. The variables φ_{ik} represent probabilities, and must satisfy

$$
\varphi_{ik} \in [0,1] \qquad \forall i,k \qquad \text{and} \qquad \sum_{k=1}^{K}\varphi_{ik} = 1 \qquad \forall i.
$$

How do we maximize the ELBO? A naïve first effort is to find its stationary points. We obtain the following partial derivatives:

$$
\frac{\partial}{\partial m_k} ELBO = -\frac{m_k}{\sigma^2} - \sum_{i=1}^{n}\varphi_{ik}(m_k - x_i) \tag{7}
$$

$$
\frac{\partial}{\partial s_k^2} ELBO = \frac{1}{2s_k^2} - \frac{1}{2\sigma^2} - \frac{1}{2}\sum_{i=1}^{n}\varphi_{ik} \tag{8}
$$

$$
\frac{\partial}{\partial \varphi_{ik}} ELBO = \frac{1}{2}\left[(m_k-x_i)^2+s_k^2\right] - \log\varphi_{ik} - 1 \tag{9}
$$

Notice that $\frac{\partial}{\partial m_k} ELBO$ is a decreasing function of m_k; this implies that the ELBO is concave in the direction of m_k, when the other parameters are held fixed. The same is true for the other parameters—the ELBO is concave for each of them individually, when the rest are held fixed. This will be important when we discuss Coordinate Ascent Variational Inference in section 4.

Setting the partial derivatives to 0 yields a system of $(n+2)k$ nonlinear equations in $(n+2)k$ variables:

$$m_k = \frac{\sum_{i=1}^{n} \varphi_{ik} x_i}{\frac{1}{\sigma^2} + \sum_{i=1}^{n} \varphi_{ik}} \qquad \forall k \in [K] \tag{10}$$

$$s_k^2 = \frac{1}{\frac{1}{\sigma^2} + \sum_{i=1}^{n} \varphi_{ik}} \qquad \forall k \in [K] \tag{11}$$

$$\varphi_{ik} = \exp\left(-\frac{1}{2}\left[(m_k - x_i)^2 + s_k^2\right]) - 1\right) \qquad \forall k \in [K], \forall i \in [n] \tag{12}$$

Solving these simultaneously appears challenging; they may not even have a closed-form solution. Even so, our efforts have not been in vain. We have shown that the ELBO is smooth, with easily-computed derivatives. This means that a rich assortment of optimization techniques can be used to maximize the ELBO.

The original *Review* paper presented two such techniques. The first is called *Coordinate Ascent Variational Inference* (CAVI). As the name suggests, CAVI maximizes the ELBO in an iterative fashion, updating one variational parameter at a time. We dig into this in section 4

In their review, Blei et al. also describe gradient ascent methods, focusing on *Stochastic Variational Inference* (SVI). In essence, SVI is the application of stochastic gradient ascent to ELBO-maximization. Section 5 discusses this in some detail.

4 Coordinate Ascent VI

One ELBO-maximization technique available for the Mean Field Family is *coordinate ascent variational inference* (CAVI). Roughly speaking, CAVI iterates through the factor distributions q_i, updating each individually while holding the rest constant.

In this section, we apply CAVI to the Gaussian Mixture model. We will also spend some time generalizing our results to a broader class of models.

4.1 Coordinate Ascent VI for Gaussian Mixtures

In section 3 we computed the ELBO for the Gaussian Mixture model, under the Mean Field Family assumption. Furthermore, we computed the ELBO's partial derivatives; by setting them to 0, we obtained equations (10)-(12).

It turns out that we can use equations (10)-(12) to maximize the ELBO in an iterative fashion. After randomly initializing the variational parameters, we loop through them and update them according to equations (10)-(12), holding the other parameters fixed. Figure 4.1 illustrates this algorithm in more detail. Note that since the φ_{ik}s must be probabilities, we project them onto the simplex (i.e., normalize them) after updating each probability vector φ_i.

Recall from section 3 that the ELBO is concave with respect to each of its parameters when the others are held fixed. Since CAVI updates each parameter to its coordinate-wise stationary point, it follows that CAVI increases the ELBO with each update. We cannot guarantee convergence to a global maximum, but we *will* reach a local one.

4.2 Coordinate Ascent VI, Generalized

It turns out that our application of CAVI to the gaussian mixture model, using the mean field variational family, can be easily generalized to other statistical models. If we keep the mean field family assumption—but leave the model unspecified—we can derive a more general form for CAVI coordinate updates. Start-

```
randomly initialize variational parameters m, s, phi;

% Sweep through the parameters until convergence.
while ELBO has not converged
    % Update data-specific cluster assignments
    for i=1:n
        phi(i,:) = exp(-1.0 - 0.5*((m - x(i,:))^2 + s));
        % Just to be safe: enforce the fact that phi(i,:) sums to one
        phi(i,:) = phi(i,:) / sum(phi(i,:))
    end

    % Update cluster means and variances
    for k=1:K
        m(k) = phi(:,k)'*x(:,k) / (1.0/var + sum(phi(:,k)));
        s(k) = 1.0 / (1.0/var + sum(phi(:,k)));
    end
end
```

Figure 3: Pseudocode Coordinate Ascent Variational Inference (CAVI) for our gaussian mixture model. Observe that it uses equations (10)-(12) to update variational parameters, one at a time.

ing with the definition of the ELBO, we find

$$
\begin{aligned}
ELBO(q;p) &= \mathop{\mathbb{E}}_{z\sim q}\left[\log p(z,x)\right] - \mathop{\mathbb{E}}_{z\sim q}\left[q(z)\right] \\
&= \mathop{\mathbb{E}}_{j}\left[\mathop{\mathbb{E}}_{-j}\left[\log p(z_j, z_{-j}, x)\right]\right] - \mathop{\mathbb{E}}_{z\sim q}\left[\log \prod_i q_i(z_i)\right] \\
&= \mathop{\mathbb{E}}_{j}\left[\mathop{\mathbb{E}}_{-j}\left[\log p(z_j, z_{-j}, x)\right]\right] - \mathop{\mathbb{E}}_{j}\left[\log q_j(z_j)\right] + \text{const.} \\
&= \mathop{\mathbb{E}}_{j}\left[\log \exp\left\{\mathop{\mathbb{E}}_{-j}\left[\log p(z_j, z_{-j}, x)\right]\right\}\right] - \mathop{\mathbb{E}}_{j}\left[\log q_j(z_j)\right] + \text{const.} \\
&= -KL\left(q_j(z_j) \,\middle\|\, \exp\left\{\mathop{\mathbb{E}}_{-j}\left[\log p(z_j, z_{-j}, x)\right]\right\}\right) + \text{const.}
\end{aligned}
$$

In summary: under the mean field family, the ELBO is a negative KL divergence from each coordinate-wise distribution $q_j(z)$ to the distribution $\propto \exp\{\mathbb{E}_{-j}\left[\log p(z_j, z_{-j}, x)\right]\}$. An elementary property of KL divergence states that it is minimized when its arguments are equal; it follows that the ELBO is maximized when we set

$$
q_j(z_j) \propto \exp\left\{\mathop{\mathbb{E}}_{-j}\left[\log p(z_j, z_{-j}, x)\right]\right\}. \tag{13}
$$

Indeed, applying this to our gaussian mixture model recovers the entire CAVI scheme we devised for that model—including the parametric forms of the distributions, and the parameter updates:

TO DO: REDERIVE CAVI UPDATES FOR GAUSSIAN MIXTURE, USING 13

$$
q_k(\mu_k; m_k, s_k^2) \propto \exp\{\}
$$

$$
q_i(c_i; \varphi_i) \propto \exp\{\}
$$

Notice that equation 13 gave us the *forms* of the coordinate distributions, as well as the correct parameter updates. Specifically, we found that the $q_k(\mu_k)$ are normal distributions and the $q_i(c_i)$ are categorical distributions. We know from our derivation of equation 13 that these forms are optimal.

5 Stochastic Variational Inference

In section 3 we computed the ELBO for the Gaussian Mixture model and obtained its derivatives. Since we can evaluate derivatives, it seems natural to try gradient-based optimization methods to maximize the ELBO. In this section, we follow Blei et al.'s explanation of gradient ascent methods, with a focus on *Stochastic Variational Inference* (SVI), an application of stochastic gradient to the variational inference problem.

5.1 Stochastic Variational Inference for Gaussian Mixtures

For a concrete example, let's return to the Gaussian mixture model. Its derivatives were given in equations (7)-(9). In principle, we could use a straightforward gradient ascent method to climb the ELBO (accounting for the constraints on φ—e.g., by re-projecting the step).

As the size of our data increases, however, this becomes infeasible. For every data point, there is a cluster assignment parameter φ_i specific to that data point. Each partial derivative in equations (7)-(8) depends on every φ. Furthermore, the partial derivative w.r.t. m_k depends on every data point x_i directly. Hence, evaluating the full gradient—even just for the global parameters m_k and s_k^2—requires a scan through the full dataset. In modern analyses involving large volumes of data, this is prohibitively expensive.

Fortunately, we can get around this problem by applying a stochastic gradient method. Notice that the derivatives (7) and (8) depend on a *sum* over the data. It follows that we can easily compute unbiased estimates of the gradient for m_k and s_k^2 by selecting a data point x_i uniformly at random:

$$\begin{aligned}
-\frac{m_k}{\sigma^2} - n \cdot \mathop{\mathbb{E}}_i \left[\varphi_{ik}(m_k - x_i)\right] &= -\frac{m_k}{\sigma^2} - n\sum_{i=1}^{n} \frac{1}{n}\varphi_{ik}(m_k - x_i) \\
&= -\frac{m_k}{\sigma^2} - \sum_{i=1}^{n} \varphi_{ik}(m_k - x_i) \\
&= \frac{\partial}{\partial m_k} ELBO
\end{aligned}$$

$$\begin{aligned}
\frac{1}{2s_k^2} - \frac{1}{2\sigma^2} - \frac{1}{2} n \mathop{\mathbb{E}}_i[\varphi_{ik}] &= \frac{1}{2s_k^2} - \frac{1}{2\sigma^2} - \frac{1}{2} n \sum_{i=1}^{n} \frac{1}{n}\varphi_{ik} \\
&= \frac{1}{2s_k^2} - \frac{1}{2\sigma^2} - \frac{1}{2} \sum_{i=1}^{n} \varphi_{ik} \\
&= \frac{\partial}{\partial s_k^2} ELBO
\end{aligned}$$

These observations suggest a simple algorithm:

1. Choose one of the data points x_i uniformly at random;
2. update its corresponding assignment parameter φ_i according to equation 12 (taking care to normalize its components);
3. update the cluster parameters m_k and s_k^2 in a steepest ascent fashion, using our "noisy" gradient:

$$m_k \leftarrow m_k + \alpha \left[-\frac{m_k}{\sigma^2} - n\varphi_{ik}(m_k - x_i)\right]$$

$$s_k^2 \leftarrow s_k^2 + \alpha \left[\frac{1}{2s_k^2} - \frac{1}{2\sigma^2} - \frac{1}{2} n\varphi_{ik}\right]$$

4. Repeat until convergence.

However, we can do even better than this. In the *Review* paper, Blei et al. describe a stochastic optimization method that uses unbiased estimates of the *natural gradient*. In optimization parlance, they find unbiased estimates of the *Newton* update direction:

$$p^N = -(\nabla^2_{m,s_k^2} ELBO)^{-1} \nabla_{m,s_k^2} ELBO.$$

i.e., the direction obtained by premultiplying the gradient by the inverse Hessian. Statisticians call this the *natural gradient* for reasons described in Appendix A. Notice that we are only updating the "global" cluster parameters m_k and s_k^2 in this fashion.

This technique requires little additional work in the case of our gaussian mixture model. From the partial derivatives of the ELBO shown in equations (7)-(9), we see that the Hessian of the ELBO (w.r.t. m_k and s_k^2) is a diagonal matrix. It follows that its inverse is diagonal, and the Newton updates are

TO DO: WORK OUT THE NATURAL GRADIENT FOR GAUSSIAN MIXTURE. NATURAL GRADIENT FOR M_K SURPRISINGLY TRICKY—PHI APPEARS NONLINEARLY

$$m_k \leftarrow m_k + \left(\frac{1}{\sigma^2} + \sum_{i=1}^n \varphi_{ik}\right)^{-1} \left[-\frac{m_k}{\sigma^2} - \sum_{i=1}^n \varphi_{ik}(m_k - x_i)\right],$$

$$s_k^2 \leftarrow s_k^2 + \left(-\frac{1}{(s_k^2)^2}\right)^{-1} \left[\frac{1}{2s_k^2} - \frac{1}{2\sigma^2} - \frac{1}{2}\sum_{i=1}^n \varphi_{ik}\right]$$

or, after simplification,

$$m_k \leftarrow \frac{\sum_{i=1}^n \varphi_{ik} x_i}{\frac{1}{\sigma^2} + \sum_{i=1}^n \varphi_{ik}}$$

$$s_k^2 \leftarrow \frac{1}{2} s_k^2 \left[1 + s_k^2 \left(\frac{1}{\sigma^2} + \sum_{i=1}^n \varphi_{ik}\right)\right]$$

5.2 Stochastic Variational Inference, Generalized

We can generalize our derivation of stochastic variational inference to a broader class of statistical models—the *conditionally conjugate* models. This class includes our gaussian mixture, along with

Conditionally conjugate models have two defining features:

1. their hidden variables can be segregated into *local variables* (z) and *global variables* (β);
2. their *complete conditional distributions* are in the *exponential family*.

The distinction between local and global hidden variables can be understood in the following way: if the variable is specific to an individual data point, then it is *local* to that data point. If the variable influences all data points, then it is *global*.

Suppose we have a statistical model $p(z, x)$. Its *complete conditional distributions* are obtained by conditioning on all the variables except one. For example, $p(z_j | z_{-j}, x)$ is a complete conditional distribution.

Our gaussian mixture is one example of a conditionally conjugate model; its local variables are the cluster assignments, c_i (there is one cluster assignment for each data point). Its global variables are the cluster locations, $\vec{\mu}$ (every data point has some chance of being in each cluster). Moreover, it has the convenient property that all of its complete conditional distributions are in the exponential family.

We generalize stochastic variational inference to all conditionally conjugate models

TO DO: DERIVE NATURAL GRADIENT STEP FOR GENERAL CONDITIONALLY CONJUGATE MODELS

References

[1] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *arXiv:1601.00670v7*, 2017.

A A Statistics Primer

Variational Inference is at the intersection of statistics and optimization. As such, some knowledge of statistics goes a long way in this domain.

This section assumes undergraduate-level familiarity with probability. Beyond that, it gives the requisite background for making sense of VI.

A.1 Bayesian Inference

Let $p(x, z)$ be a statistical model—a joint probability distribution where

- x are *observed variables*—data is sampled from them.
- z are *unobserved variables*—these can include "latent variables" and model parameters.

One can think of a statistical model as the description of a process that is only partially observed. Given what we *can* observe about the process (e.g., its outputs), what can we infer about the unobserved parts of the process?

The typical goal of Bayesian inference is to compute the *posterior distribution*:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{14}$$

$$= \frac{p(x|z)p(z)}{\int p(x, z)dz} \tag{15}$$

That is: given some data x, update the distribution for hidden variables z. It turns out that, for all but the simplest models, the integral in the denominator either

- has no closed form, or
- is computationally intractable (e.g., exponential cost to compute).

Over the years, statisticians have come up with several families of models that, at the very least, result in closed forms. We present some of them in the following paragraphs—they are worth knowing, since they will appear when we describe VI methods in Section **??**.

Conjugate priors. Conjugate priors are an algebraic convenience that appears in Bayesian inference. Using a conjugate prior ensures that the update in equation 15 takes a closed form.

For certain likelihood functions $p(x|z)$, the prior distribution $p(z)$ over the hidden variables can be chosen such that the posterior distribution $p(z|x)$ is from the same family as $p(z)$.

For example: if a statistical model has binomial likelihood, and the prior is chosen to be a beta distribution, then the posterior will also be a beta distribution (with parameters updated to reflect the data).

Exponential family. The exponential family is the set of all distributions with the following form:

$$p(x|z) = h(x) \exp\left[\eta(z)^\top T(x) - A(z)\right]$$

It turns out that most of the "familiar" discrete and continuous distributions can be expressed in this way, through appropriate choice of h, η, T, and A.

The function $T(x)$ is of particular interest. It is usually called the *sufficient statistic* of the distribution; so called because when T is evaluated on a data set, the result is sufficient for estimating the hidden variables z.

The function $\eta(z)$ is also important; it's called the *natural parameter* of the distribution. Some descriptions of the exponential family remove η's dependence on z, yielding the form

$$p(x|\eta) = h(x) \exp\left[\eta^\top T(x) - A'(\eta)\right]$$

Distributions in the exponential family have many desirable properties; it's worth noting that every exponential family distribution has a conjugate prior, which is *also* in the exponential family.

Conditionally conjugate models. The authors of the original *Review* paper devote much of their discussion to a class of dubbed *conditionally conjugate models*. They are of interest because they generalize the class of conjugate models (models where variables have conjugate prior relationships with each other) while retaining algebraic convenience. In the VI setting, this means that coordinate updates can be derived in closed form with relative ease.

Despite these innovations, it turns out that approximate inference is still almost always necessary. Variational inference is one such approach, though it too can benefit from the convenient properties of conjugate priors and the exponential family.

A.2 Some Information Geometry

Variational Inference searches through a space of distributions for one that is "closest" to the exact posterior. This section defines some quantities that are useful for navigating that space and finding an optimal distribution.

KL divergence—Definition. Kullback-Liebler (KL) Divergence is one way to measure the "distance" from one distribution to another. The KL divergence of q from p is defined as follows:

$$KL(q\|p) = \mathop{\mathbb{E}}_{x\sim q}\left[-\log\left(\frac{p(x)}{q(x)}\right)\right] \tag{16}$$

$$= \mathop{\mathbb{E}}_{x\sim q}\left[-\log p(x)\right] - \mathop{\mathbb{E}}_{x\sim q}\left[-\log q(x)\right] \tag{17}$$

I've put "distance" in quotes because the KL divergence is not a metric. It is nonnegative for all p and q and takes the value 0 iff $p = q$; however, it is not symmetric, and does not satisfy the triangle inequality.

KL divergence—Intuition. KL divergence has a very concrete information-theoretic interpretation: both terms in equation 17 can be thought of as entropies.

Suppose you have a digital signal with symbol frequencies given by q, and you want to compress this signal as much as possible. If you used a compression scheme optimized for q, then it would compress your signal to an average of $\mathbb{E}_{x\sim q}\left[-\log q(x)\right]$ bits per symbol (the entropy of q).

But suppose the only compression scheme available to you was optimized for p-signals. Then it would compress your q-signal to an average of $\mathbb{E}_{x\sim q}\left[-\log p(x)\right]$ bits per symbol.

The difference between these two quantities gives the "inefficiency" of using p-optimal compression for a q-signal. This inefficiency is exactly the KL divergence of q from p.

Fisher information. Roughly speaking, Fisher information describes the sensitivity of a distribution to changes in its parameters. We'll denote it as $I(z)$; it typically appears as a matrix, with entries dependent on a model's parameters z.

Fisher information can be formulated in a few different ways; but most relevant for our purposes is its characterization as **the Hessian of the KL divergence.**

More specifically, the Fisher information matrix $I(z)$ is given by

$$I(z) = \left[\frac{\partial^2}{\partial t_i \partial t_j} KL\left(p(x|t)\|p(x|z)\right)\Big|_{t=z}\right]_{i,j}$$

Natural gradients. The *natural gradient* of the KL divergence is obtained by premultiplying its ordinary (Euclidean) gradient by the inverse Fisher information:

$$g_N(z) = I(z)^{-1} \cdot \nabla_z KL(p(x|z)\|p(x|z))$$

i.e., the gradient is transformed to account for the KL divergence's locally quadratic structure. This would, in essence, be the direction of a Newton update for minimizing KL divergence.

Evidence Lower Bound (ELBO). Recall that variational inference entails minimizing the KL divergence of $q(z)$ from $p(z|x)$. However, this KL divergence cannot be computed directly in practice. A related quantity called the Evidence Lower Bound (ELBO) is used instead. ELBO is defined as follows:

$$\begin{aligned} ELBO(q) &= \mathop{\mathbb{E}}_{z\sim q}\left[\log \frac{p(z,x)}{q(z)}\right] \\ &= \log\big(p(x)\big) - KL\Big(q(z)\Big\|\, p(z|x)\Big) \end{aligned}$$

This relationship between ELBO and KL divergence means that we can minimize KL divergence by maximizing ELBO.

A.3 Computer Representations

A fundamental part of computational statistics is the representation of probability distributions in software. The original *Review* paper [1] describes methods that operate on probability distributions, but is not always clear about how these operations are represented in a computer.

An examination of the paper's concrete examples reveals that they rely heavily on symbolic derivations of closed form distributions. At no point are full distributions represented by the computer—for example, by splines, discretizations, or other data structures. Instead, the computer only ever stores *parameters* of distributions, and performs all of its calculations using those parameters.

In some sense, this is an economical way to represent a distribution with a computer; most commonly-used distributions are fully defined by a small number of parameters (e.g., a normal distribution is defined by its mean and variance).

However, this compactness comes with the need for laborious symbolic derivations. Even if the symbolic derivations are performed by computer (e.g., using Mathematica), the process of derivation and implementation requires a high degree of expertise from the user.