# Variants of Variational Inference

**Scribe Notes, 03/04/2016**

**Ryan Dew**

Plan:

- VI in general
- Stochastic VI
- Black box VI
- Reparametrization trick

# Part 1: VI in General

Joint distribution: $p(z, x)$

$z$ = latent variables $x$ = data

Inference: compute $p(z|x)$

In VI, we specify a variational family, $q(z; \lambda)$, a distribution of the hidden variables parametrized by $\lambda$, the variational parameters.

(Aside: both these methods and MCMC were derived from ideas in statistical physics. MCMC took off in stats, while MCMC and VI took off in ML; there is renewed interest in VI now due to concerns about scaling. Dave suggests that VI may be even more useful as a generic algorithm for inference on any model.)

Key idea of VI: we're looking to find

$$\lambda^* = \arg\min_{\lambda} KL(q(z, \lambda)||p(z|x)).$$

In essence, we set up a family of distributions, indexed by $\lambda$, which we search over to find the closest match to the intractable posterior we're interested in, then use $q(z, \lambda^*)$ as a proxy for the true posterior.

Recall:

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

where the denominator is the "evidence". This part is hard to compute, which is why we need to do approximate inference. Unfortunately, the evidence plays a role in calculating the KL divergence too. So at first glance, we haven't solved a problem. More specifically:

$$KL(q(z)||p(z|x)) = E_q[\log q(Z)] - E_q[\log p(Z|x)] = E[\log q(Z)] - E[] \log p(Z, x)] + \log p(x)$$

Instead of computing that, we derive a quantity called the ELBO:

$$ELBO(q) = E[\log p(Z, x)] - E[\log q(Z)]$$

which is the negative KL plus $\log p(x)$, and we maximize that. ELBO stands for Evidence Lower BOund. We can decompose this expression: the first term is the expected log joint, which is alternatively written $E[\log p(Z)] + E[\log p(x|Z)]$. This encourages the optimization to place $q$'s mass on the likelihood, regularized by the prior (which is what we want). The second term is the entropy of $q$, which encourages $q$ to be diffuse

during the optimization. Since KL divergence is positive, the ELBO is indeed a lower bound on $p(x)$ (just re-arrange terms and note KL is positive).

Question: why bother using ELBO? It's the same as above line (p(x) not dependent on $\lambda$).

Question: ... Answer: Mean field is very limiting, forces variables to be independent, can't capture posterior correlations.

In especially neural nets/deep learning, they write the ELBO differently:

$$ELBO(q) = KL(q(Z))||p(Z)) + E[\log p(x|Z)]$$

which makes it more like a regularization problem.

This is all very similar to the historic derivation of the EM algorithm. This gives insight as to how VI came about (see Jordan et al. 1999). Recall the EM algorithm (Dempster 1977): need to compute $\log p(x) = \log \int p(x, z)dz$. To work with this, first do this trivially:

$$\begin{aligned}
\log \int p(x, z)dz &= \log \int p(x, z)\frac{q(z)}{q(z)} \\
&\geq \int q(z) \log \frac{p(x, z)}{q(z)}dz \\
&= E_q[\log p(x, z)] - E_q[\log q(z)]
\end{aligned}$$

where the second line comes from Jensen's inequality. Reminder: Jensen's inequality says that, for any convex function, the convex combination of two function values, is greater than the function of that convex combination of the inputs:

$$\lambda f(a) + (1 - \lambda)f(b) \geq f(\lambda a + (1 - \lambda)b)$$

Note that this last line is exactly the ELBO explained above.

**How do we choose q?**

So we may have a very multimodal $p(z|x)$, how will $q(z)$ estimate it? When $q(z)$ is high, KL cares about mismatch between q and the joint, but if $q(z)$ is low, we don't care as much (asymmetric nature of KL). This may lead to underestimation of the variance. Important note: the variational family has to have the same support as the joint, otherwise KL will blow up.

How do we pick a variational family $q(\cdot)$? Generally, we pick a mean field family:

$$q(z) = \prod_{j=1}^{d} q_j(z_j)$$

The graphical model of the variational family is just a bunch of independent variables none of which are observed (a node indexed j in a box with index d, possibly with a square $\lambda_j$ node pointing to $z_j$). This allows you to capture *marginals*, but not any correlation between the $z$'s (which ultimately might actually mess up the marginals... (discussed in a Question)).

Note: this only assumes independence. There is no assumption about the family of these $q_j$'s yet. They will be in the exponential family if the original conditionals are in the exponential family. We will get to this.

The parameters of our optimization are the distributions $q_j$ themselves. Generally we parametrize these distributions so that the $\lambda$'s are what we search over. The coordinate updates we get from this optimization are:

$$q_j^*(z_j) \propto \exp\{E_{-j}[\log p(z_j|z_{-j}, x)]\}$$

where the $-j$ notation means everything except index $j$. What is this thing? The $\log p(z_j|z_{-j}, x)$ term is the complete conditional. (Note: from here, we get a connection to Gibbs sampling.) So we take an expectation with respect to all of the other nodes, so the term is just a function of $z_j$, and in the end, we can renormalize this whole thing to get a distribution for $z_j$. Since this is proportional, we can re-write as:

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

Why is this optimal?

$$ELBO(q_j) = E_j[E_{-j}[\log p(z_j, Z_-j, x)]] - E_j[\log q_j(z_j)] + \text{Const.}$$

From this, it's obvious that this will be minimized when $q_j$ equals the above.

**Aside: Open Problems in VI**

Before we proceed, a discussion of the open problems:

- the issue of support and heavy vs. light tails; q mismatches p in the tails
- related to the first point, mean field tends to underestimate variances
- active research area: beyond the mean field family (aside: always try mean field first; any issue you have in mf you will have in a more complex procedure)
  - solutions: normalizing flow, variational models (from Dave's group)
- checking approximate inference (general issue, not necessarily specific to VI)
- alternative divergences (expectation propagation = KL(p||q) rather than q||p)
- statistical properties of aspects of VI
- understanding the optimization problem (open problem in many fields, how to avoid finding local optima and stuff like that)

**Practical implementation**

1. Initialization is a huge problem, less so in stochastic (see following)
2. Always work in log space to avoid computational problems
3. To assess convergence, monitor the ELBO until it flattens.

# Stochastic Variational Inference

For SVI to work, we need to have a certain model form (see the SVI paper for the DAG). In general, we need to have data, $x_i$ which are driven by local variables, $z_i$, and by global variables, $\beta$. Furthermore, the $z_i$ are determined by $\beta$. This describes a ton of models in ML. One downside of usual VI is that we need to evaluate each item of the "box" to do estimation; we have to evaluate the parameters governing *each* document before updating global parameters. Stochastic variational inference allows us to do this much more quickly.

Key assumption of SVI: every complete conditional is in the exponential family:

$$p(\beta|x, z) = h(\beta) \exp\{\eta(x, z)'\beta - a(\eta(x, \beta))\}$$

Note that the arguments to $\eta$ can be very very high dimensional (e.g. number of documents...).

$$p(z_i|x_i, \beta)] = h(z_i) \exp\{\eta(x_i, \beta)'z - a(\eta(x_i, \beta))\}$$

Note that dependence on other x's goes away due to Bayes ball. In general: $\eta(x, z) = \alpha + \sum_i t(x_i, z_i)$.

Assume a mean field: $q(\beta, z) = q(\beta; \lambda) \prod_i q(z_i; \phi_i)$. Then we get the following $q^*$:

$$\begin{aligned} q_i^*(z_i) &\propto \exp\{E_\lambda[\log h(z_i) + \eta(x_i, \beta)'z - a(x_i, \beta)]\} \\ &\propto \exp\{\log h(z_i) + E[\eta(x_i, \beta)]'z_i)\} \end{aligned}$$

Very important note: just going from the form of the optimal update described above, we've discovered that the optimal $q$ is in the *same exponential family*!

(Admin note: $j$ from above has changed to $i$ here just by accident)

Generic VI algorithm for this:

1. Init $\lambda$
2. Repeat:
   - For each data point i: $\phi_i = E_\lambda[\eta(x_i, \beta)]$. End for.
   - Set $\lambda = \alpha + \sum_i E_{\phi_i}[t(z_i, x_i)]$

   ... until ELBO converges.

This has connections to EM again, and also marches through all of the variational parameters. Inefficient because you have sum over each i. Two ideas that lead to a solution: stochastic optimization and natural gradients.

Idea behind natural gradients: searching around in a parameter space, moving one way may lead to a much different change in KL divergence than moving in another way. But we can scale this in some way that gets rid of this behavior (see the SVI paper). We can use this to refine our optimization problem instead of using coordinate ascent (gradient ascent instead).

First, compute the optimal local parameter as before. Then define a natural gradient as:

$$\mathcal{L}(\lambda) = \mathcal{L}_{ELBO}(\lambda, \phi^*)$$

(the ELBO with the optimal $\phi$ subbed in). Then the natural gradient (estimate?) is given by:

$$\hat{\nabla}_\lambda \mathcal{L} = (\alpha + \sum_i E_{\phi_i}[t(x_i, Z)]) - \lambda)]$$

We still have to evaluate over all $i$. We can get around them by thinking of a valid *noisy* gradient:

$$\tilde{\nabla}_\lambda \mathcal{L} = \alpha + nE_{\phi_j^*}[t(x_j, z_j)] - \lambda$$

where $j \sim \text{Unif}(1, 2, \ldots, n)$. Basically, we compute the single sampled point as the whole sample. Finally, we set the global parameter with a step size:

$$\lambda = (1 - \rho_t)\lambda + \rho_t \hat{\lambda}$$

The step has to be set according to two conditions (sum of steps diverges, but sum of squares converges).

Scribe note: this algorithm here is messy, but it's very clear in the SVI paper.

# Black Box VI

We are following through Dave's slides for this part.

Problem: What do we do if the models aren't in the exponential family?

Goal: VI algorithm that can reuse any variational family analysis but does not require any model-specific calculations beyond computing the log-joint.

Solution: form noisy gradients from samples from q that doesn't involve model-specific computation. Use the "score trick" to write the gradient as an expectation. Then we write this as a noisy gradient using the ideas from before.

Requirements for doing this:

- Sampling from $q(\beta, z)$
- Evaluating gradient of log $q(\beta, z)$
- Something else... see slides posted on Dave's website.

Unfortunately it doesn't work too well due to getting trapped in a part of the ELBO. How to make it work: Rao-Blackwellization, control variates, and AdaGrad algorithm.