NN/VI/Autoencoder Reading Group Notes

Evan Ott
evan.ott@utexas.edu

Last updated: September 11, 2017

Outline

- Variational inference (James)
- Stochastic variational inference (Michael)
- NN and backpropagation (Yuguang)
- Introduction to TensorFlow (Mo and Evan) – actual hello world (like a single layer perceptron or a linear model) – leave out complicated things like dropout, etc.
- Classical autoencoders (Mauricio)
- Variational autoencoders (Jennifer)

We'll have a GitHub page / Google Doc for papers, examples, talks, code, etc.

2017-09-11: Variational Inference (James)

"Variational Bayes for Idiots"

Today is a basic introduction to variational inference.

Observed data X and hidden variables Z (could be parameters for the whole dataset – like means and variances of mixture components of a GMM – or the per-data-point hidden variable – like an indicator for each datapoint being in a cluster).

$$p(x,z) = p(x|z) \cdot p(z)$$
$$\text{Bayes: } p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$
$$\text{Problem: } p(x) = \int_Z p(x,z)dz$$

$p(x)$ is challenging or impossible to compute.

Idea:

- posit some family of approximations, $\mathcal{Q}$
- Find the member of $\mathcal{Q}$ that is "closest" to $p(z|x)$ in KL-divergence (or other measure)

  Formally:
  $$q^*(z) = \arg\min_{q(z)\in\mathcal{Q}} KL\left(q(z)\|p(z|x)\right) \tag{1}$$
  where
  $$\begin{aligned} KL\left(q(z)\|p(z|x)\right) &= \mathbb{E}_q(z)\left[\log\frac{q(z)}{p(z|x)}\right] \\ &= \mathbb{E}_q\left[\log q(z)\right] - \mathbb{E}_q\left[\log p(z|x)\right] \end{aligned}$$

This is a calculus of variations problem (like the brachistochrome problem), where this will eventually lead to a vector optimization problem that's standing for a functional optimization problem.

This also will give us point estimates that are often better than running a Gibbs sampler, and be faster. KL will favor putting probability mass where $p(z|x)$ is large, so that the variance of the approximated distribution

is often smaller than the variance of the true posterior. If we flipped the ratio in equation **??**, we would instead be looking at Expectation Propagation.

Example: One-Hot Encoding

$$\begin{aligned}(x_i|\mu, c_i) &\sim N(c_i^\top \mu, 1)\\ \text{e.g., } c_i &= (0,0,1,0,0) \in \{0,1\}^k\\ \mu &\in \mathbb{R}^k\\ \mu_k &\sim N(0, \tau^2)\end{aligned}$$

Joint of data, parameters:

$$p(\mu, c, x) = p(\mu)\prod_{i=1}^{N} p(c_i)p(x_i|\mu, c_i)$$

Marginal or "Evidence": $p(x) = \int_{\mu,c} p(x, \mu, c)p(\mu, c)d\mu dc$

ELBO

ELBO: Evidence lower bound.

$\mathcal{Q}$: family of approximations.

Each $q(z) \in \mathcal{Q}$ is a candidate.

$\arg\min_{q(z)\in\mathcal{Q}} KL\left(q(z)\|p(z|x)\right)$ is not computable:

$$\mathbb{E}_q\left[\log q(z)\right] - \mathbb{E}_q\left[\log p(z|x)\right] = \mathbb{E}_q\left[\log q(z)\right] - \mathbb{E}_q\left[\log p(x, z) - \log p(x)\right]$$

But we can't compute $p(x)$, so we can't compute it, but the term is not dependent on the choice of z. So, we ignore that term, and want to maximize its negative (the ELBO):

$$\begin{aligned}ELBO(q) &= \mathbb{E}_q\left[\log p(x, z)\right] - \mathbb{E}_q\left[\log q(z)\right]\\ &= -KL\left(q(z)\|p(z|x)\right) + \log(p(x))\\ &\leq \log(p(x))\end{aligned}$$

So we can re-write this as:

$$\begin{aligned}ELBO(q) &= \mathbb{E}_q\left[\log p(x|z)\right] + \mathbb{E}_q\left[\log p(z)\right] - \mathbb{E}_q\left[\log q(z)\right]\\ &= \mathbb{E}_q\left[\log p(x|z)\right] + KL\left(q(z)\|p(z)\right)\end{aligned}$$

So we're making the likelihood as large as possible, and making the approximate distribution that penalizes moving away from the prior. In other words, be close to the data, and don't stray far from the prior.

Note

Carlos asked a question about not having to do the expectation of the log posterior with respect to q to get something more like a MAP estimate. But a reasonable compromise is doing a maximum marginal a posteriori estimate where we marginalize over the local variables in the model (the cluster assignments in a GLM), but keep the global parameters to find the MAP. That's a nice sort of Bayesian thing to do.

Mean-field family

This is a family of approximations where the correlation structure is completely independent.

$$\mathcal{Q} = \left\{ q(z) : q(z) = \prod_{j=1}^{M} q_j(z_j) \right\}$$

This is not a model of the data (there's no x in that). But we're going to connect it to the data through the ELBO. Now, we have options for each q_j, where we might take a nice parametric form. We might have Gaussians for location parameters or inverse gammas for scale parameters. But for some models where the complete conditionals are exponential, we can find optimal approximation to take for each q_j.

Coordinate ascent

Not totally different from Gibbs sampling, only we're doing optimization instead of probabilistic draws. We'll do one latent variable at a time, such as all the components one after another, then each mean in a GMM. Then, start over.

Consider the jth latent variable z_j with complete conditional $p(z_j|z_{-j}, x)$. With z_{-j} fixed, the optimal q_j (that makes ELBO as large as possible) is of the form:

$$q_j^*(z_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(z_j|z_{-j}, x)\right]\right\} \tag{2}$$

Or, can write in terms of the joint (the constants will go away). The expectation is with respect to the variational density over all the other parameters, i.e.,

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

So, we set $q_j^*(z_j)$ for each j, cycle through, update, etc. until it converges.