1 Gradient Ascent

Say we want to maximise an arbitary function $F(x, y)$, where the partial derivatives $\delta(F(x, y))/\delta x$ and $\delta(F(x, y))/\delta y$ can not be found analytically. Then gradient ascent is a simple search algorithm that can be used to find a local maximum.

Gradient ascent has a simple mechanism, start at an arbitary point, calculate numerical partial derivatives for each variable then step in the direction of that derivative. Each step is

$$\begin{bmatrix} x^{(t)} \\ y^{(t)} \end{bmatrix} = \begin{bmatrix} x^{(t-1)} \\ y^{(t-1)} \end{bmatrix} + p_t \begin{bmatrix} \frac{\delta F(x,y)}{\delta x}\Big|_{(x,y)=(x^{(t-1)},y^{(t-1)})} \\ \frac{\delta F(x,y)}{\delta y}\Big|_{(x,y)=(x^{(t-1)},y^{(t-1)})} \end{bmatrix}$$

and the algorithm is guaranteed to converge to a local minimum with

$$\begin{aligned} \sum_{t=1}^{\infty} p_t &= \infty \\ \sum_{t=1}^{\infty} p_t^2 &< \infty. \end{aligned}$$

$p_t = t^{-a}, 0.5 < a \leq 1$ satisfies this condition.

Intuitively, if the function is increasing, the algorithm steps forwards, and if it is decreasing the algorithm steps backwards. Smaller and smaller steps allow it to zig zag between either side of the maximum as it converges.

2 Stochastic Gradient Ascent

In the derivation for variational bayes, we had

$$\ln(p(y)) = KL[q(\theta)||p(\theta|y)] + \mathcal{L}(q, y),$$

where the Evidence Lower Bound (ELBO) is

$$\mathcal{L}(q, y) = \int_{\theta} q(\theta) \ln\left(\frac{p(y, \theta)}{q(\theta)}\right) d\theta. \tag{1}$$

As $KL[q(\theta)||p(\theta|y)]$ is non-negative, maximising $\mathcal{L}(q, y)$ with respect to a distribution $q(\theta)$ is equivalent to minimising the KL Divergence between

$p(\theta|y)$ and $q(\theta)$. If $q(\theta)$ factorises into $\prod_i q(\theta_i)$, and the likelihood and prior form exponential family conjugate pairs, then variational bayes maximises $\mathcal{L}(q, y)$ with the solutions

$$q(\theta_i) \propto \exp(E_{\setminus i}(ln(p(y, \theta)))) \text{ for all } i.$$

However, (1) makes no assumption about the form of $q(\theta)$, and gradient ascent can be used to find the maximum of $\mathcal{L}(q, y)$ for a much more flexible choice of q.

With λ as the vector of hyperparameters in the approximate distribution $q(\theta|\lambda)$, we require the partial derivative of $\mathcal{L}(q, y)$ with respect to λ. Paisley, Blei and Jordan (2012) introduce the stochastic gradient ascent approach, assuming the conditions to exchange the order of differentiation and integration hold.

$$\begin{aligned}
\nabla_\lambda \mathcal{L} &= \nabla_\lambda \int_\theta q(\theta|\lambda) \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta \\
&= \int_\theta \nabla_\lambda [q(\theta|\lambda) \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right)] d\theta \\
&= \int_\theta q(\theta|\lambda) \nabla_\lambda [(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))] d\theta + \int_\theta \nabla_\lambda [q(\theta|\lambda)] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta \\
&= -\int_\theta \nabla_\lambda [\ln(q(\theta|\lambda)] q(\theta|\lambda) d\theta + \int_\theta \nabla_\lambda [q(\theta|\lambda)] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta \\
&= -\int_\theta \frac{\nabla_\lambda q(\theta|\lambda)}{q(\theta|\lambda)} q(\theta|\lambda) d\theta + \int_\theta \nabla_\lambda [q(\theta|\lambda)] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta \\
&= -\nabla_\lambda \int_\theta q(\theta|\lambda) d\theta + \int_\theta \nabla_\lambda [q(\theta|\lambda)] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta \\
&= \int_\theta \nabla_\lambda [q(\theta|\lambda)] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta
\end{aligned}$$

This expression can be further simplified by the identity $\nabla_\lambda [q(\theta|\lambda)] = \nabla_\lambda [\ln(q(\theta|\lambda))] q(\theta|\lambda)$.

$$\begin{aligned}
\nabla_\lambda \mathcal{L} &= \int_\theta \nabla_\lambda [q(\theta|\lambda)] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) d\theta \\
&= \int_\theta \nabla_\lambda [\ln(q(\theta|\lambda))] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right) q(\theta|\lambda) d\theta \\
&= E_q [\nabla_\lambda [\ln(q(\theta|\lambda))] \left(\ln(p(y,\theta)) - \ln(q(\theta|\lambda))\right)]. \qquad (2)
\end{aligned}$$

We can compute a noisy but unbiased estimate of (2) by

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{N} \sum_{n=1}^{N} \nabla_\lambda [\ln(q(\theta_n|\lambda))](\ln(p(y, \theta_n)) - \ln(q(\theta_n|\lambda))) \tag{3}$$

with $\theta_n \sim q(\theta|\lambda)$.

To use stochastic gradient ascent to maximise $\mathcal{L}$ we just need to be able to evaluate the log-joint density and each log-approximate density, as well as calculate partial derivatives of each log-approximate density within a Monte-Carlo approximation.

3 Variance Reduction

The Monte-Carlo approximation is unbiased but can have a large variance if a small amount of draws is chosen. We can reduce the variance by instead sampling another variable with the same expected value but a reduced variance.

For the scalar θ case, consider the function

$$\hat{f}(\theta) = f(\theta) - a(g(\theta) - E[g(\theta)]) \tag{4}$$

Paisley, Blei and Jordan (2012) call the function g a control variate. Note that

$$E[\hat{f}(\theta)] = E[f(\theta)]$$

and

$$Var(\hat{f}) = Var(f) + a^2 Var(g) - 2aCov(f, g) \tag{5}$$

The variance of $\hat{f}$ is minimised at

$$a^* = \frac{Cov(f, g)}{Var(g)}$$

A similar approach holds for a vector θ.

Ranganath, Gerrish and Blei (2014) suggest the following function g

$$g(\theta) = \nabla_\lambda [\ln(q(\theta_n|\lambda))]$$

as it has a high covariance with (2) and thus efficiently reduces variance in (5).

If the joint and approximating distributions factorise, a Rao-Blackwellisation approach can be used to ignore the parts of the function that do not depend on the θ_i of interest in that step of the algorithm and further reduce variance. More details are in Ranganath, Gerrish and Blei (2014).

4 Copula Variational Inference

Using stochastic gradient ascent, we are no longer restricted to the mean-field assumption $q(\theta) = \prod_i q(\theta_i)$, and Tran, Blei and Airoldi (2015) suggest a copula model,

$$q(\theta|\lambda,\eta) = \prod_i q(\theta_i|\lambda)c(Q(\theta_1|\lambda),\ldots,Q(\theta_D|\lambda)|\eta) \text{ for } i = 1,\ldots D. \qquad (6)$$

The copula function is factorised into a sequence of pairs $c(Q(\theta_i|\lambda), Q(\theta_j|\lambda)|\eta)$ using a vine copula and a stochastic gradient ascent algorithm alternates between maximising $\mathcal{L}$ with respect to the variational parameters λ and the copula parameters η. The algorithm is not particularly fast to converge but allows and extremely flexible approximation with strong predictive abilities. If data was streaming, and the model was just incrementing $q(\theta_t)$ to $q(\theta_{t+1})$ repeatedly it may be faster.