Variational Inference with Gradient Flows

Nicholas Altieri
naltieri@eecs.berkeley.edu
University of California Berkeley

David Duvenaud
dduvenaud@seas.harvard.edu
Harvard University

Abstract

We present a variational inference method defined through gradient ascent on the likelihood function. This method can be used to improve existing posterior approximations or as part of a recognition network in variational autoencoders. Despite its simplicity, it proves to be competitive on standard benchmarks.

1 Introduction

Computing posterior distributions is at the core of many problems in machine learning, but is generally intractable. Variational inference casts this problem as an optimization: given a class of tractable probability distributions, we find the one closest to the true posterior by optimizing a lower bound on the marginal likelihood.

We propose a simple method to do variational inference or augment an existing variational method. Starting from an initial approximate distribution $q(z)$, we implictly define an improved distribution by updating samples $q(z)$ in the direction of the gradient of the target distribution $p(x, z)$. Our main contribution in this paper is augmenting inference networks with gradient ascent as well as Langevin dynamics.

2 Background

In variational inference, given a family of distributions $q_\phi(z|x)$ and a true, but intractable posterior $p(z|x)$, we wish to find the $q_\phi(z|x)$ that minimizes the KL-divergence to the true posterior. There has been a renewed interest in variational methods recently and in particular a promising class of variational distributions have emerged. In this family of work, we parameterize q_ϕ as a series of transition operators applied to an initial distribution. In particular, we let $q_\phi = T_{\phi_n} \circ \cdots \circ T_{\phi_1} \circ q_{\phi_0}(z)$, where n is a fixed number of steps. Two examples of such methods are [4] and [5].

In [4], the operator T_{ϕ_i} is a step of a sampler or a quasi-sampler (eg, Hamiltonian Monte Carlo or Metropolis Hasting), and the authors give a general framework to optimize the distributions defined by this family with variational inference. In particular they demonstrate the efficacy of optimizing a form of modified Hamiltonian Monte Carlo without an accept-reject step. In these methods is that for one must learn a reverse network for each step of the transition operator to compute estimate the change in entropy. In this way, one can see this as learning variational inference within variational inference.

In [5], the transition operator T_{ϕ_i} is a step of Normalizing Flow which is a learnable rank one invertible transformation, which when applied in a sequence, can model arbitrarily complex distributions.

In our case, the transition operator is a step of gradient ascent on the likelihood $p(x, z)$ with respect to z. Furthermore, the number of steps of our transition are determined at random by a policy.

The algorithmic method presented in this work is adapted from [3]. In [3], gradient descent with early stopping was interpreted as implicitly performing variational inference by considering the gradient step as a transition operator. The Jacobian of this operator was used to construct an estimate of the evidence lower bound. They then compared this bound to the cross-validation error when optimizing neural networks on classification and regression problems.

In this paper, we are interested in using gradient descent for variational inference in the context of variational autoencoders, where gradient descent is performed as part of the inference procedure. The parameters of this inference procedure (such as step sizes) can then be optimized as part of an outer loop to increase the evidence lower bound. In this context, it can be compared against other iterative recognition networks, such as Hamiltonian Variational Inference [4], or Normalizing Flows [5].

2.1 Variational Autoencoders

Our experiments consist of augmenting variational autoencoders. [1] Variational autoencoders specify a generative model, and a recognition (encoder) network. Given a dataset X, we model it with latent variables $Z \sim N(0, I)$ as follows. We first sample $Z_i \sim N(0, I)$, then sample $X_i \sim P_\theta(\cdot|Z)$. In addition, given a datapoint X_i, the output of the encoder network gives $Q_\phi(\cdot|X_i)$ that approximates $P_\theta(Z|X_i)$. We optimize the variational autoencoder by maximizing the variational lower bound $\mathcal{L}_{\theta,\phi}$ defined by P_θ and Q_ϕ.

3 Variational Inference using Gradient Ascent

The problem we are interested in solving is approximating a posterior distribution $p(z|x)$ by starting with a high entropy distribution $q_0(z)$ and doing incomplete gradient ascent on $p(z, x)$ with respect to z. Observe that if we took gradient ascent to completion, we would get degenerate solutions to local minima of $p(z, x)$. Given an initial distribution $q_0(z)$ we can view gradient descent as a transition operator making repeated modifications to the distribution, ie $z_{t+1} = T(z_t)$ where $T(z) = z - \alpha\nabla_z \log p(x, z)$, creating distributions $q_0, q_1, \dots, q_n$.

Recall that if $\mathcal{H}$ is the entropy function, then

$$\log p(x) \geq \mathcal{H}[q(z|x)] + \mathbb{E}_{q(z|x)} \log p(x, z) := \mathcal{L}[q] \tag{1}$$

If we wish to compute $\mathcal{L}[q_t]$, or an unbiased estimate, $\tilde{\mathcal{L}}[q_t]$, then we need to estimate both the likelihood term and the entropy term. We can exactly sample from q_t by simply running the optimizer for t steps starting from a random initialization, and we can use samples $z_1, \dots, z_n$ to form a Monte-Carlo estimate of the likelihood $\mathbb{E}_{q_t(z|x)} \log p(x, z)$. For the entropy, following [3], we observe that $\mathcal{H}[q_{t+1}] - \mathcal{H}[q_t] = \mathbb{E}_{q_t(z|x)} \log |J(z_t)|$ where J is the Jacobian induced by taking the gradient step. Consequently, we can write $\mathcal{L}[q_t] \approx \log p(z_t, x) + \sum_{t=0}^{T} \log |J(z_t)| + \mathcal{H}[q_0]$.

For gradient ascent in particular, we have $z_{t+1} = z_t + \alpha\nabla_z \log p(x, z)$ and consequently, if H_t is the Hessian of $\log p(x, z_t)$ with respect to z, then $\mathcal{H}[q_{t+1}] - \mathcal{H}[q_t] = \log |I - \alpha H_t|$

3.1 Estimating the Jacobian in high dimensions

Computing the log determinant is impractical for large-scale problems since it requires an $O(D^3)$ determinant computation. Fortunately, we can make a good approximation using Hessian-vector products, which can be performed in time proportional to evaluation of the gradient using reverse-mode differentiation.

In particular, [3] gives a local lowerbound for the log deteriminant:

$$\log |I - \alpha H| \geq -\alpha \text{Tr}(H) - \alpha^2 \text{Tr}(HH)$$

and show that one can compute an unbiased estimate of the lowerbound in linear time using hessian vector products. However, this lowerbound only holds for $\alpha\rho(H) \leq .68$ where ρ is the spectral radius of H. Consequently, we optimize our variational lowerbound using the lowerbound, but for fairness we evaluate the lowerbound using the exact log determinant. Additionally, we could use the algorithm outlined in [7]. This allows us to compute log-determinants in randomized linear time. For general non-singular matrices, they have an additive bound with high probability. However, for

Algorithm 1 Gradient Ascent with Entropy Estimates, following [3]

```
input: Weight initialization scale $\sigma_0$, step size $\alpha$, twice-differentiable negative log-likelihood $L(Z)$
initialize $Z_0 \sim N(\sigma_0, I_D)$
initialize $\mathcal{H}_0 = \frac{D}{2}(1 + \log 2\pi) + D \log \sigma_0$
for $t = 1$ to $T$ do
    $\mathcal{H}_t = \mathcal{H}_{t-1} + \log |I - \alpha H_{t-1}|$        ▷ Update entropy
    $Z_t = Z_{t-1} + \alpha \nabla L(Z_t)$        ▷ Update parameters
end for
output sample $Z_T$, entropy estimate $\mathcal{H}_T$
```

matrices with largest singular values less than one, they provide a multiplicative bound. Note that gradient ascent will not converge if $\|I - \alpha H\|_2 \geq 1$ and with sufficiently small stepsizes α we will have that $\|I - \alpha H\|_2 < 1$.

If Hessian-vector products are used when estimating the entropy, then computing gradients with respect to hyperparameters in the outer loop will require third-order gradient information.

3.2 Optimizing the lower bound

We optimize the above procedure by setting a fixed number of of gradient steps, then optimizing the lowerbound of variational autoencoder $\mathcal{L}[q_n]$ in an end to end fashion (including the stepsize hyperparameter).

3.3 Extension to Langevin Dynamics

Instead of just doing gradient descent down $p(z, x)$, we can also augment our network with Langevin Dynamics, which involves alternating between taking gradient steps and adding noise. For the gradient steps, we can use the methods mentioned above and for the noise steps, we can use the entropy power inequality. The entropy power inequality states that given two random variables X and Y, then

$$\mathcal{H}(X + Y) \geq \frac{1}{2} D \log(e^{2\mathcal{H}(X)/D} + e^{2\mathcal{H}(Y)/D})$$

Therefore the entropy of the distribution after adding noise is bounded below by:

$$\mathcal{H}(Z + \epsilon) \geq \frac{1}{2} D \log(e^{2\mathcal{H}(Z)/D} + e^{2\mathcal{H}(\epsilon)/D})$$

4 Experiments

Here we demonstrate the efficacy of variational inference using gradient descent.

4.1 Augmenting Inference Networks with GF and Langevin Dynamics

Here we show how we can improve an inference network with variational gradient ascent. We train a variational autoencoder where the encoder and decoder are two fully connected MLPs each having two hidden layers containing 300 nodes with a 20 dimensional latent space. Explicitly, $q_\phi(z|x) = N(\mu_\phi(x), \sigma_\phi(x))$ where μ_ϕ, σ_ϕ are neural networks and σ_ϕ is the standard deviations of a diagonal Gaussian. Additionally, $p_\theta(x|z)_{ij} = \text{Bern}(\rho_\theta(Z)_{ij})$, where ρ_θ is a matrix of probabilities defined by a neural network. We refer to this as a canonical variational autoencoder. In addition to a canonical variational autoencoder, we compare to a variational autoencoder augmented with normalizing flow, which has achieved results competitive with state of the art. [5] We compare three methods. The first is normalizing flow with 5-80, the second is a variational autoencoder augmented with 1-3 steps of gradient ascent, and finally, a variational autoencoder augmented with 1-2 steps of Langevin dynamics. We train on MNIST with minibatches of size 100 and optimize using ADAM [6]. It is also worth noting that the need to compute $\nabla_z \log p_\theta(x, z)$ in the gradient methods makes this slower per iteration than Normalizing Flows.

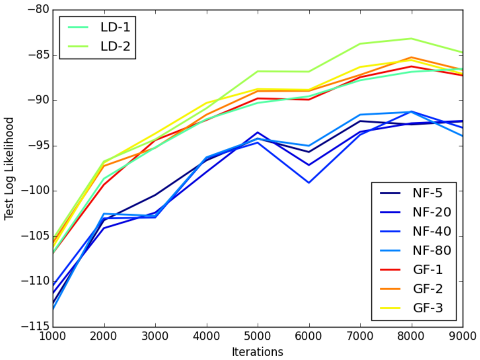

Figure 1: Normalizing, Gradient, and Langevin Flow

Furthermore, we have also trained GF-1 with latent dimensions equal to 40 and batch size equal to 1000. After 7000 iterations the variational lowerbound on the test set for this method was -79.73, the lowest we've seen in the literature.

5 Conclusion and Future Directions

We introduced a variational inference method based on taking gradient ascent steps on the likelihood and showed an unbiased way to estimate its variational lower bound. This results in a simple extension to inference networks that yields promising results.

This method could be extended as in [3] by performing non-linear warping to gradients, to better preserve entropy by reducing step-sizes when gradients are small, with the expense of having an extra hyperparameter.

6 References

[1] Kingma, Welling. Auto-Encoding Variational Bayes ICLR 2014 http://arxiv.org/pdf/1312.6114v10.pdf

[2] Danilo Jimenez Rezende, Shakir Mohamed, Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models ICML 2014 http://arxiv.org/pdf/1401.4082v3.pdf

[3] Dougal Maclaurin, David Duvenaud, Ryan P. Adams. Early Stopping is Nonparametric Variational Inference http://arxiv.org/pdf/1504.01344v1.pdf

[4] Tim Salimans, Diederik P. Kingma, Max Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap http://arxiv.org/pdf/1410.6460v4.pdf

[5] Danilo Jimenez Rezende, Shakir Mohamed. Variational Inference with Normalizing Flows http://arxiv.org/abs/1505.05770

[6] Diederik Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization http://arxiv.org/abs/1412.6980

[7] Insu Han, Dmitry Malioutov, Jinwoo Shin. Large-scale Log-determinant Computation through Stochastic Chebyshev Expansions http://arxiv.org/abs/1503.06394