- Iterative Amortized Inference
- Josehn Marino, Yisong Yue, Stephen Mandt

The Two Variational Inference Approach

Consider an approximate posterior of form $q_\lambda(z|x)$. Here, λ is a function of x.

Here we only consider the **inference** problem. That is, given θ and x, find λ.

Variational Expectation Maximization (EM) via Gradient Ascent

This approach optimizes λ for **each** $x^{(i)}$ separately. Optimization is straightforward:

$$\lambda^{(i)} \leftarrow \lambda^{(i)} + \alpha \nabla_\lambda \mathcal{L}(x^{(i)}, \lambda; \theta)$$

There are several problems here:

- $\nabla_\lambda \mathcal{L}$ is intractable. It must be evaluated stochastically, which is expensive (?)
- This process has to be repeated for each x.
- We have to set α.

Amortized Inference Models

Here we will assume that a global parameter ϕ exists such that $p_\theta(z|x)$ can be reasonable well approximated with $q_\phi(z|x)$. Basically, we are modelling λ as $\lambda = f(x; \phi)$.

With some tricks, this allowed optimization with SGD in minibatches, and is scalable.

We will refer to this as *standard inference models*.

Iterative Amortized Inference

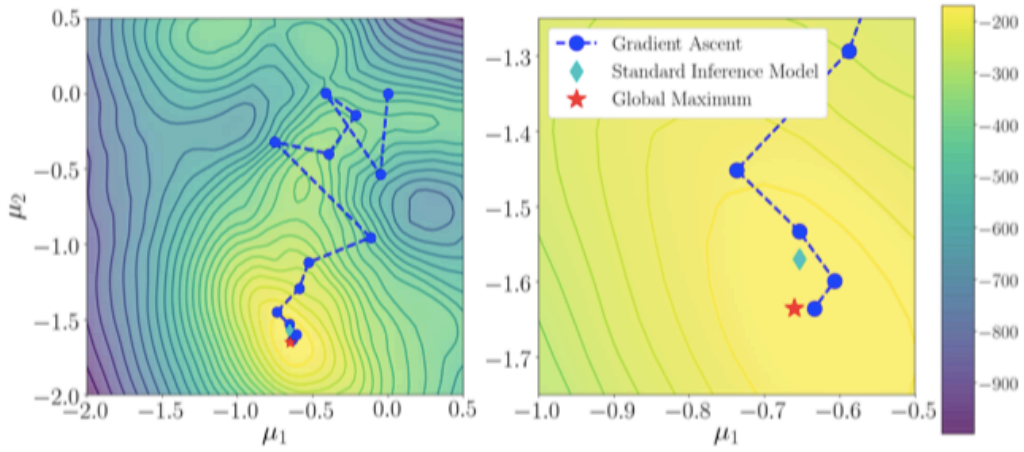

Figure 1. **Visualizing the amortization gap.** Optimization surface of $\mathcal{L}$ (in *nats*) for a 2-D latent Gaussian model and an MNIST data example. Shown on the plots are the optimal estimate (MAP), the output of a standard inference model, and an optimization trajectory of gradient ascent. The plot on the right shows an enlarged view near the optimum. Conventional optimization outperforms the standard inference model, exhibiting an amortization gap. With additional latent dimensions or more complex data, this gap could become larger.

The author has compare the two optimization method and draw the following conclusion:

- Gradient ascent requires many iterations and is sensitive to step-size. However, it will reach global optimal (of course)
- Standard inference model is fast, but failed to reach global optimum.

The second is known as an **amortization gap**.

Remark: I think this is due to the assumption that $q_\phi(z|x)$ can approximate $p_\theta(z|x)$ globally. The relationship between λ and x maybe be able to be model by a function f.

Learning to Iteratively Optimize

Now, we want to use standard inference model as a basic framework. However, we would like to go beyond this direct mapping between x and λ.

Remember that we keep θ fixed. In this case, a simple idea is just to backprop on ϕ. It seems that there are previous work on this?

A more advanced idea proposed in this paper is to let the network *learn* to optimized λ, given $\nabla\lambda$ and other information.

Iterative Inference Model

The iterative model will be called f, parametrized by ϕ. Let $\mathcal{L}_t^{(i)} = \mathcal{L}(x^{(i)}, \lambda_t^{(i)}; \theta)$ as the ELBO, the update rule is given by

$$\lambda_{t+1}^{(i)} \leftarrow f_t(\nabla_\lambda \mathcal{L}_t^{(i)}, \lambda_t^{(i)}; \phi)$$

At the end of this iteration, we can expend theses steps and obtain a computation graph. When evaluate the final ELBO with $\lambda_T^{(i)}$, we will get gradient backproped to ϕ, as well as for θ.

The algorithm for one iteration:

Algorithm 1 Iterative Amortized Inference

Input: data $\mathbf{x}$, generative model $p_\theta(\mathbf{x}, \mathbf{z})$, inference model f
Initialize $t = 0$
Initialize $\nabla_\phi = 0$
Initialize $q(\mathbf{z}|\mathbf{x})$ with $\boldsymbol{\lambda}_0$
repeat
  Sample $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})$
  Evaluate $\mathcal{L}_t = \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}_t; \theta)$
  Calculate $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t$ and $\nabla_\phi \mathcal{L}_t$
  Update $\boldsymbol{\lambda}_{t+1} = f_t(\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t, \boldsymbol{\lambda}_t; \phi)$
  $t = t + 1$
  $\nabla_\phi = \nabla_\phi + \nabla_\phi \mathcal{L}_t$
until $\mathcal{L}$ converges
$\theta = \theta + \alpha_\theta \nabla_\theta \mathcal{L}$
$\phi = \phi + \alpha_\phi \nabla_\phi$

More theories and details

Some derivations are tricky for me now. I may update this part later.