Notes on Variational Inference with Normalizing Flows

1 Summary of Paper

- We have a posterior that we want to sample from, but it is too complex. Use an approximating $q_\phi(\theta|\mathbf{x}) \in Q$ instead (standard VI).
- The usual mean-field assumption is too simplifying. Most cases have posteriors that are complex enough to not be in Q, and hence we will never achieve it even in asymptotic limits. Also, there is clear evidence that richer Q results in better results, because a limited $q \in Q$ tends to introduce an underestimate in the variance and a bias in the MAP.
- To make q complex enough, we can sample from mean-field assumed $\mathbf{z}_0 \sim q_0(\mathbf{z}|\mathbf{x})$, then put $\mathbf{z}_0$ through a normalizing flow. This is a set of transformations of a PDF through invertible mappings. Since it is normalizing, the final q_K is still a valid PDF. The flow is parameterized by λ_k, e.g. $\lambda_k = \{\mathbf{w}_k \in \mathbb{R}^D, \mathbf{u}_k \in \mathbb{R}^D, b \in \mathbb{R}\}$ for planar flows.
- We can use the standard introduction of latent variable $\mathbf{z}$, and work on maximizing the lower bound ELBO instead. For proof of concept illustrations, the paper only focusses on inferring $\mathbf{z}$ rather than θ.
- Generative process: assume the latent variables $\mathbf{z}$ are defined using a DLGM, where each layer l models $\mathbf{z}_l$. We get the final likelihood of observations $p_\theta(\mathbf{x}|\mathbf{z})$ by assuming some distribution for this and parameterizing with a DNN.

2 Theory Questions

- Two alternatives were given to make q richer. Using a mixture model was not scalable because it is too computationally expensive. What is the issue with using structured mean-field approximations?

  Does not capture multi-modality, because it is still a single Gaussian.

- Is this accurate: Note that since we have latent variable $\mathbf{z}_n$ for each $\mathbf{x}_n$, the posterior for $q(\mathbf{z}|\mathbf{x})$ factorises over all observations. Inference networks are used for amortized VI, where instead of learning ϕ_n for each q_n, since

$$q_\phi(\mathbf{z}|\mathbf{x}) = \prod_{n=1}^{N} q_{\phi_n}(\mathbf{z}_n|\mathbf{x}_n)$$

we represent $\phi = \text{NN}(\mathbf{x}; \mathbf{w})$, and hence only need to learn $\mathbf{w}$, which are shared across all observations. It then allows us to build an inverse map from observations $\mathbf{x}$ to latent variables $\mathbf{z}$ using global parameters rather than local parameters. Since this does not depend on N, it scales better.

Yes.

- When specifying a distribution to get the likelihood from the latent $\mathbf{z}$ (that come from the DLGM), why is $\mathbf{x}$ only conditioned on $\mathbf{z}_1$? What about $\mathbf{z}_2, ..., \mathbf{z}_L$?

  DLGM modelling assumption. The likelihood $p_\theta(\mathbf{x}|\mathbf{z}_1)$ is parameterized by θ, and is inferred in the algorithm.

- The ELBO is written with $\mathbb{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))$, but Section 3 states that this goes to 0 when $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$?

  Expectation is taken into $\mathbb{D}_{KL}$.

- What properties must maps in normalizing flows have that preserve PDFs?

  Invertibility.

3 Implementation Questions

- In the algorithm, why do we need $\mathcal{F}(\mathbf{x}) \approx \mathcal{F}(\mathbf{x}, \mathbf{z}_k)$, and how do we code this?

  Because we cannot work with the marginalized $\mathbf{x}$ (due to intractable integral), so we need to work with the joint instead. Basically the lower bound ELBO we derived.

- In the algorithm, is $\phi = \{\mu_0, \mathbf{\Sigma}_0, \lambda_1, ..., \lambda_K\}$, where $\mu_0, \mathbf{\Sigma}_0$ are the parameters of q_0 and $\lambda_1, ..., \lambda_K$ are the parameters of the K flows? If not, how else do we learn λ_k?

  $\phi = \{\lambda_1, ..., \lambda_K\}$, since we assume the initial form of q_0.

- In the algorithm, what is θ that we are optimizing over? I thought this paper only concentrated on inferring $\mathbf{z}$. Is this just the parameters of the likelihood we specified in the DLGM, ie in $p_\theta(\mathbf{x}|\mathbf{z})$?

  Yes, θ is the parameters of the likelihood we specified in the DLGM.

- How many NNs are there in total? There is one for each layer in the DLGM, there is one for the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. Is there one for the inference network over q? Any more?

  For toy example, do not need any NN. Just assume single layer DLGM, hence just $\mathbf{z} = \mathbf{z}_1$, and use some simple likelihood, e.g. $\mathbf{x}_n = \mathbf{A}\mathbf{z}_n + \mathbf{b}$. Then see if normalizing flow can learn this known $\theta = \{\mathbf{A}, \mathbf{b}\}$. Don't use amortized VI/inference network for toy example, just do basic VI.

- In the expression for free energy $\mathcal{F}^{\beta_t}(\mathbf{x})$, is $p(\mathbf{x}, \mathbf{z}_K) \propto \exp[-U(\mathbf{z})]$?

  Yes.

- What is the maxout activation function? I cannot understand what the windows do.

  Some fancy activation function that prevents blowing up for gradients. No need to use it.