Hierarchical Implicit Models and Likelihood-Free Variational Inference

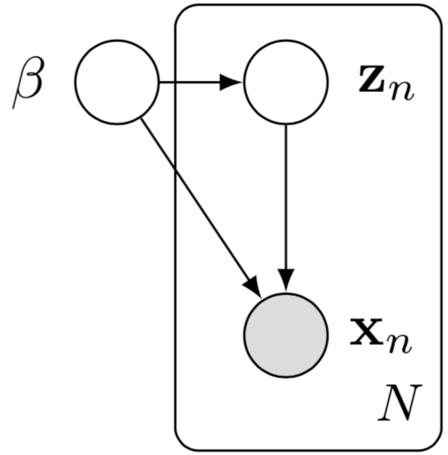
Sebastian Wagner-Carena

arXiv:1702.08896

Hierarchical Implicit Models (HIMs)

- The building blocks of HIMs are the same hierarchical Bayesian models we often use in astronomy:

$$p(\mathbf{x}, \mathbf{z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\mathbf{x}_n \mid \mathbf{z}_n, \boldsymbol{\beta}) p(\mathbf{z}_n \mid \boldsymbol{\beta}) \longrightarrow$$



- We often make x_n conditionally independent of β given z_n

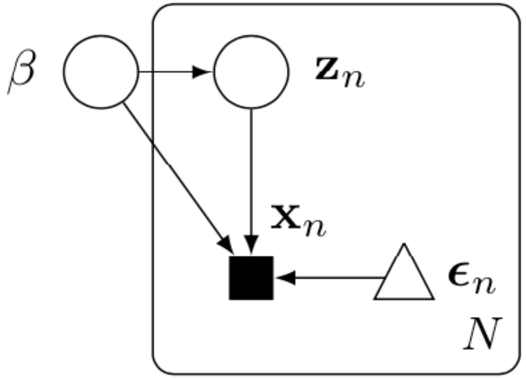
Hierarchical Implicit Models (HIMs)

- A HIM does not assume that we have access to the exact likelihood but does assume that we can sample from it (for example using a simulation).

$$\mathcal{P}(\mathbf{x}_n \in A \,|\, \mathbf{z}_n, \boldsymbol{\beta}) = \int_{\{g(\boldsymbol{\epsilon}_n \,|\, \mathbf{z}_n, \boldsymbol{\beta}) = \mathbf{x}_n \in A\}} s(\boldsymbol{\epsilon}_n)\, \mathrm{d}\boldsymbol{\epsilon}_n$$

with

$$\mathbf{x}_n = g(\boldsymbol{\epsilon}_n \,|\, \mathbf{z}_n, \boldsymbol{\beta}), \quad \boldsymbol{\epsilon}_n \sim s(\cdot)$$



- Notice that this integral is still intractable. However, we can now calculate anything the involves an expectation value of x_n conditioned on z_n, β

Variational Inference

- To approximate our posterior – $p(z, \beta | x)$ – we will use variational inference with an approximating family q
- We want an objective that fits two criteria:
  - **Scalability**: We can get an unbiased estimate of the objective by sampling a subset of the data (i.e. a batch)
  - **Admits implicit local approximations**: The objective must not require having an explicit form for the function $q(z_n | x_n, \beta)$. It should be calculable with samples of z_n
- The Kullback-Leibler (KL) divergence meets both of these criteria

KL Objective

- Minimizing the KL divergence between q and the posterior is equivalent of maximizing the evidence lower bound (ELBO):

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{\beta}, \mathbf{z} \mid \mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}, \boldsymbol{\beta}) - \log q(\boldsymbol{\beta}, \mathbf{z} \mid \mathbf{x})]$$

- We restrict our choice of q such that it can factorize:

$$q(\boldsymbol{\beta}, \mathbf{z} \mid \mathbf{x}) = q(\boldsymbol{\beta}) \prod_{n=1}^{N} q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\beta})$$

- Substituting this in we get:

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{\beta})}[\log p(\boldsymbol{\beta}) - \log q(\boldsymbol{\beta})] + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\beta}) q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\beta})}[\log p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta}) - \log q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\beta})]$$

Ratio Estimation

- We know we can't evaluate $p(x_n . z_n|\beta)$ and we do not want to restrict ourselves to being able to evaluate $q(x_n, z_n|\beta)$. So let's subtract a constant value from our loss:

$$\log q(x_n) = \log q(x_n, z_n|\beta) - \log q(z_n|x_n, \beta)$$

- This gives:

$$\mathcal{L} \propto \mathbb{E}_{q(\boldsymbol{\beta})}[\log p(\boldsymbol{\beta}) - \log q(\boldsymbol{\beta})] + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\beta})q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\beta})} \left[\log \frac{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})}{q(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})} \right]$$

- This final term is a ratio for which we can use ratio estimation techniques

Ratio Estimation (2)

- We introduce a ratio function (usually a neural network) that models the probability that a sample belongs to p given a sample from p or q:

$$\sigma(r(\mathbf{x}_n, \mathbf{z}_n, \boldsymbol{\beta}; \boldsymbol{\theta}))$$

- We connect this to a "proper scoring rule" loss function. The example they offer is:

$$\mathcal{D}_{\log} = \mathbb{E}_{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})}[-\log \sigma(r(\mathbf{x}_n, \mathbf{z}_n, \boldsymbol{\beta}; \boldsymbol{\theta}))] + \mathbb{E}_{q(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})}[-\log(1 - \sigma(r(\mathbf{x}_n, \mathbf{z}_n, \boldsymbol{\beta}; \boldsymbol{\theta})))]$$

- Where the gradients can be calculated using Monte Carlo sampling. Minimizing the loss with a sufficiently expressive function should give:

$$r^*(\mathbf{x}_n, \mathbf{z}_n, \boldsymbol{\beta}) = \log p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta}) - \log q(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})$$

Minimizing the KL Objective

- Assuming we have an optimal ratio function, then we can use this ratio estimator in our loss:

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{\beta} \mid \mathbf{x})}[\log p(\boldsymbol{\beta}) - \log q(\boldsymbol{\beta})] + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\beta} \mid \mathbf{x}) q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\beta})}[r(\mathbf{x}_n, \mathbf{z}_n, \beta)]$$

- We now introduce a global and local transformation:

$$\boldsymbol{\beta} = T_{\text{global}}(\boldsymbol{\delta}_{\text{global}}; \boldsymbol{\lambda}), \quad \boldsymbol{\delta}_{\text{global}} \sim s(\cdot)$$

$$\mathbf{z}_n = T_{\text{local}}(\boldsymbol{\delta}_n, \mathbf{x}_n, \boldsymbol{\beta}; \boldsymbol{\phi}), \quad \boldsymbol{\delta}_n \sim s(\cdot)$$

Minimizing the KL Objective (2)

- Assuming we have an optimal ratio function, then we can use this ratio estimator in our loss:

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{\beta} \mid \mathbf{x})}[\log p(\boldsymbol{\beta}) - \log q(\boldsymbol{\beta})] + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\beta} \mid \mathbf{x}) q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\beta})}[r(\mathbf{x}_n, \mathbf{z}_n, \beta)]$$

- We then have the update rules

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L} = \mathbb{E}_{s(\boldsymbol{\delta}_{\text{global}})}[\nabla_{\boldsymbol{\lambda}}(\log p(\boldsymbol{\beta}) - \log q(\boldsymbol{\beta}))]] + \sum_{n=1}^{N} \mathbb{E}_{s(\boldsymbol{\delta}_{\text{global}}) s_n(\boldsymbol{\delta}_n)}[\nabla_{\boldsymbol{\lambda}} r(\mathbf{x}_n, \mathbf{z}_n, \beta)]$$

$$\nabla_{\boldsymbol{\phi}} \mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\beta}) s(\boldsymbol{\delta}_n)}[\nabla_{\boldsymbol{\phi}} r(\mathbf{x}_n, \mathbf{z}_n, \boldsymbol{\beta})]$$

The final algorithm

Algorithm 1: Likelihood-free variational inference (LFVI)

Input : Model $\mathbf{x}_n, \mathbf{z}_n \sim p(\cdot \mid \boldsymbol{\beta}), p(\boldsymbol{\beta})$
Variational approximation $\mathbf{z}_n \sim q(\cdot \mid \mathbf{x}_n, \boldsymbol{\beta}; \boldsymbol{\phi}), q(\boldsymbol{\beta} \mid \mathbf{x}; \boldsymbol{\lambda})$,
Ratio estimator $r(\cdot; \boldsymbol{\theta})$

Output: Variational parameters $\boldsymbol{\lambda}$, $\boldsymbol{\phi}$

Initialize $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$, $\boldsymbol{\phi}$ randomly.

while *not converged* **do**

Compute unbiased estimate of $\nabla_{\boldsymbol{\theta}} \mathcal{D}$ (Eq.6), $\nabla_{\boldsymbol{\lambda}} \mathcal{L}$ (Eq.8), $\nabla_{\boldsymbol{\phi}} \mathcal{L}$ (Eq.9).

Update $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$, $\boldsymbol{\phi}$ using stochastic gradient descent.

end