Hierarchical Implicit Models & Likelihood-Free Variational Inference

Dustin Tran†, Rajesh Ranganath†, David Blei†
†Columbia University, ‡Princeton University

TL;DR

- Implicit models encompass theories of the physical world.
- Implicit models are limited due to lack of latent structure and scalable inference.
- We develop hierarchical implicit models (HIMs). They combine the idea of implicit densities with hierarchical Bayesian models.
- We develop likelihood-free variational inference (LFVI). It is a scalable algorithm for HIMs and enables implicit densities as flexible posterior approximations.
- We scale simulators in ecology to unprecedented sizes.

Hierarchical Implicit Models

\[
x_i = g_\epsilon(x_{i-1}, \beta), \quad \epsilon_i \sim \mathcal{N}(0, 1),
\]

1. Hierarchical models play an important role in sharing statistical strength across examples.
2. A broad class of hierarchical Bayesian models can be written as a joint distribution,
   \[
p(x, z, \beta) = p(\beta) \prod_i p(x_i | z_i, \beta)p(z_i | \beta). \tag{1}
   \]
   \(x_i\) is an observation, \(z_i\) are latent variables associated to that observation (local), \(\beta\) are latent variables shared across observations (global).
3. HIM combine this idea with implicit densities: define a function \(g\) that takes in random noise \(\epsilon_i \sim \mathcal{N}(0, 1)\) and outputs \(x_i = g(\epsilon_i, \beta)\).
4. The induced likelihood is
   \[
P(x_i \in A | z_i, \beta) = \int_{g(\epsilon_i, z_i, \beta)} p(\epsilon_i) d\epsilon_i.
   \]
   This integral is typically intractable.
5. Example: Physical Simulators. For prey and predator populations \(x_1, x_2 \in \mathbb{R}\) respectively, one process is
   \[
dx_1/dt = \beta_1 x_1 - \beta_2 x_1 x_2 + \epsilon_1, \quad \epsilon_1 \sim \mathcal{N}(0, 10),
   \]
   \[
dx_2/dt = -\beta_2 x_1 + \beta_3 x_2 + \epsilon_2, \quad \epsilon_2 \sim \mathcal{N}(0, 10),
   \]
   Lognormal priors are placed over \(\beta\).
6. Example: Bayesian Generative Adversarial Network. The implicit model for a generative adversarial network (GAN) is
   \[
x_i = g(\epsilon_i; \theta), \quad \epsilon_i \sim \mathcal{N}(0, 1),
   \]
   We make GANS amenable to Bayesian analysis by placing a prior on the parameters \(\theta\).

Lotka-Volterra Predator-Prey Simulator

Bayesian GAN

LotVI achieves more accurate results and scales to unprecedented sizes.

KL Variational Objective

Classical VI maximizes the ELBO,
\[
\mathcal{L} = \mathbb{E}_q(\log p(x, z, \beta) - \log q(\beta, z | x)).
\]
Substitute in factorizations,
\[
\mathcal{L} = \mathbb{E}_q(\log p(\beta) - \log q(\beta)) + \sum_{n=1}^N \mathbb{E}_z(q_{x_n}(z_n | \beta)) \log \frac{p(x_n, z_n | \beta)}{q(z_n | \beta)}.
\]
This objective presents difficulties: the local densities \(p(x_n, z_n | \beta)\) and \(q(z_n | \beta)\) are both intractable.

Trick: Density Ratio Estimation

Let \(q(x_n)\) be the empirical distribution on \(x\). Subtract \(\log q(x_n)\) from the ELBO,
\[
\mathcal{L} \propto \mathbb{E}_q(\log p(x_n, z_n | \beta) - \log q(\beta)) + \sum_{n=1}^N \mathbb{E}_z(q_{x_n}(z_n | \beta)) \log \frac{p(x_n, z_n | \beta)}{q(z_n | \beta)}
\]
Train \(r(\cdot; \theta)\) by minimizing a loss function,
\[
r(\mathcal{L})(\ell) = \mathbb{E}_z(q_{x_n}(z_n | \beta)) \log \frac{r(z_n, x_n | \beta; \theta)}{q(z_n | \beta)} \log [1 - r(z_n, x_n | \beta; \theta)]
\]
If \(r(\cdot; \theta)\) is sufficiently expressive, minimizing the loss returns the optimal function,
\[
r^*(z_n, x_n | \beta) = \log p(x_n, z_n | \beta) - \log q(z_n | \beta).
\]
As we minimize \(X\), we use \(r(\cdot; \theta)\) as a proxy to the log ratio in \(X\). Note \(r\) estimates the log ratio; it’s of direct interest and more numerically stable than the ratio.

New KL Variational Objective

Optimizing the ELBO involves substituting in the ratio estimator,
\[
\mathcal{L} = \mathbb{E}_q(\log p(x_n, z_n | \beta) - \log q(\beta)) + \sum_{n=1}^N \mathbb{E}_z(q_{x_n}(z_n | \beta)) [r(z_n, x_n | \beta)]
\]
All terms are tractable. We can calculate gradients to optimize the variational family \(q\) using reparameterization gradients.

Recipe: Injecting Noise into Hidden Units

How do you build an implicit model? Inject noise!
For sequences \(x = (x_1, \ldots, x_t)\), write an RNN,
\[
x_{t+1} = g(x_t, x_{t-1}, \epsilon_t), \quad \epsilon_t \sim \mathcal{N}(0, 1),
\]
\[
x_t = g(x_t, \epsilon_t), \quad \epsilon_t \sim \mathcal{N}(0, 1),
\]
The \(g\) functions are dense layers with ReLUs and layer norm. Standard normal priors are placed over all weights and biases.