Adversarial Variational Optimization of Non-Differentiable Simulators

Gilles Louppe and Kyle Cranmer

Tomorrow on the arXiv!

Likelihood-free assumptions

Operationally,

$$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}) \equiv \mathbf{z} \sim p(\mathbf{z}|\boldsymbol{\theta}), \mathbf{x} = g(\mathbf{z}; \boldsymbol{\theta})$$

where

- $\mathbf{z}$ provides a source of randomness;
- g is a non-differentiable deterministic function (e.g. a computer program).

Accordingly, the density $p(\mathbf{x}|\boldsymbol{\theta})$ can be written as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int_{\{\mathbf{z}:g(\mathbf{z};\theta)=\mathbf{x}\}} p(\mathbf{z}|\boldsymbol{\theta})\mu(d\mathbf{z})$$

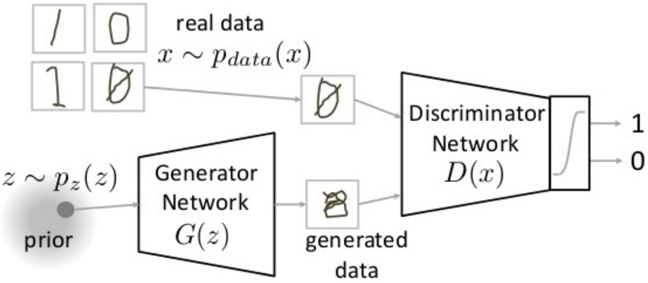
Evaluating the integral is often intractable.

Problem statement

Given observations $\mathbf{x} \sim p_r(\mathbf{x})$, we seek:

$$\boldsymbol{\theta}^* = \arg\min_{\theta} \rho(p_r(\mathbf{x}), p(\mathbf{x}|\boldsymbol{\theta}))$$

Generative Adversarial Networks



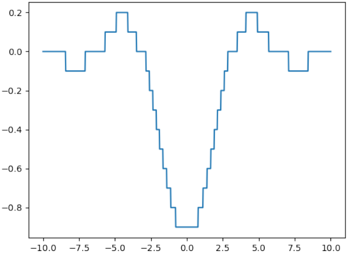
$$
\begin{aligned}
\mathcal{L}_W &= \mathbb{E}_{\tilde{\mathbf{x}} \sim p(\mathbf{x}|\boldsymbol{\theta})}[d(\tilde{\mathbf{x}}; \boldsymbol{\phi})] - \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[d(\mathbf{x}; \boldsymbol{\phi})] \\
\mathcal{L}_d &= \mathcal{L}_W + \lambda \Omega(\boldsymbol{\phi}) \\
\mathcal{L}_g &= -\mathcal{L}_W
\end{aligned}
$$

What if g isn't a neural net, but a non-differentiable simulator?

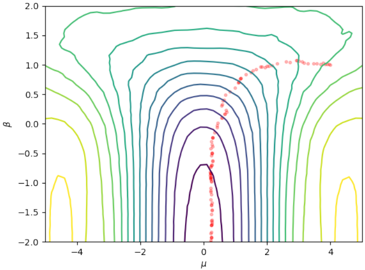
Variational Optimization

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \leq \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi})}[f(\boldsymbol{\theta})] = U(\boldsymbol{\psi})$$

$$\nabla_{\boldsymbol{\psi}} U(\boldsymbol{\psi}) = \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi})}[f(\boldsymbol{\theta}) \nabla_{\boldsymbol{\psi}} \log q(\boldsymbol{\theta}|\boldsymbol{\psi})]$$



Piecewise constant $-\frac{\sin(\mathbf{x})}{\mathbf{x}}$



$q(\boldsymbol{\theta}|\boldsymbol{\psi} = (\mu, \beta)) = \mathcal{N}(\mu, e^{\beta})$

Adversarial Variational Optimization

- Replace the generative network in a non-differentiable forward simulator $g(\mathbf{z}; \boldsymbol{\theta})$.
- With VO, optimize upper bounds of the adversarial objectives:

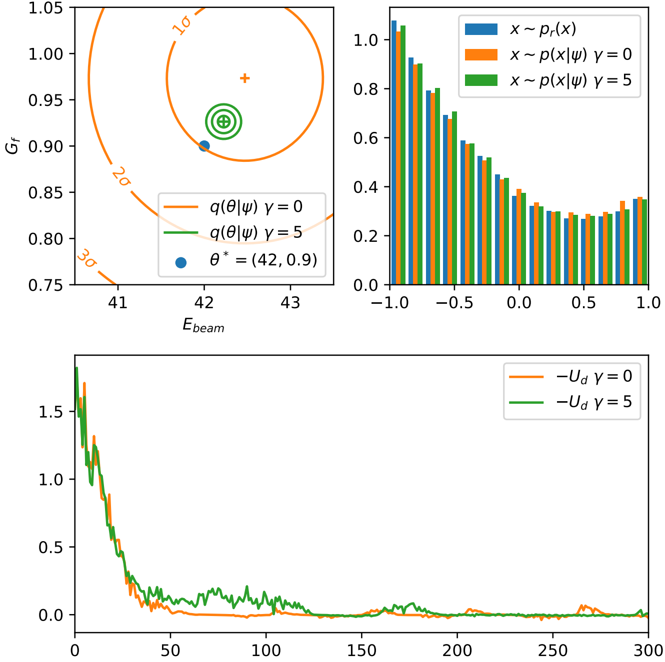
$$U_d = \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi})}[\mathcal{L}_d] \quad (1)$$

$$U_g = \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi})}[\mathcal{L}_g] \quad (2)$$

respectively over $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$.

Operationally, we get the marginal model:

$$\mathbf{x} \sim q(\mathbf{x}, \boldsymbol{\psi}) \equiv \boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{\psi}), \mathbf{z} \sim p(\mathbf{z}|\boldsymbol{\theta}), \mathbf{x} = g(\mathbf{z}; \boldsymbol{\theta})$$



1σ
2σ
3σ
$q(\theta|\psi)\ \gamma = 0$
$q(\theta|\psi)\ \gamma = 5$
$\theta^* = (42, 0.9)$
G_f
E_{beam}
$x \sim p_r(x)$
$x \sim p(x|\psi)\ \gamma = 0$
$x \sim p(x|\psi)\ \gamma = 5$
$-U_d\ \gamma = 0$
$-U_d\ \gamma = 5$

Summary

- We provide a new likelihood-free inference algorithm:
  - That works for non-differentiable forward simulators.
  - Combines adversarial training with variational optimization.

- Needs further validation on realistic simulators.