Lecture 27: Physics-informed deep neural networks

Professor Ilias Bilionis

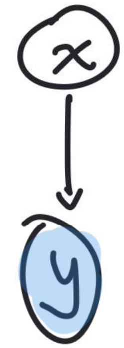
Introduction to probabilistic programming

PREDICTIVE
SCIENCE LABORATORY

Problem definition

Model parameters : x

Data : y



Prior : $x \sim p(x)$

Likelihood : $y \mid x \sim p(y \mid x)$

Posterior : ? $p(x \mid y) = \dfrac{p(y \mid x)\, p(x)}{Z}$?

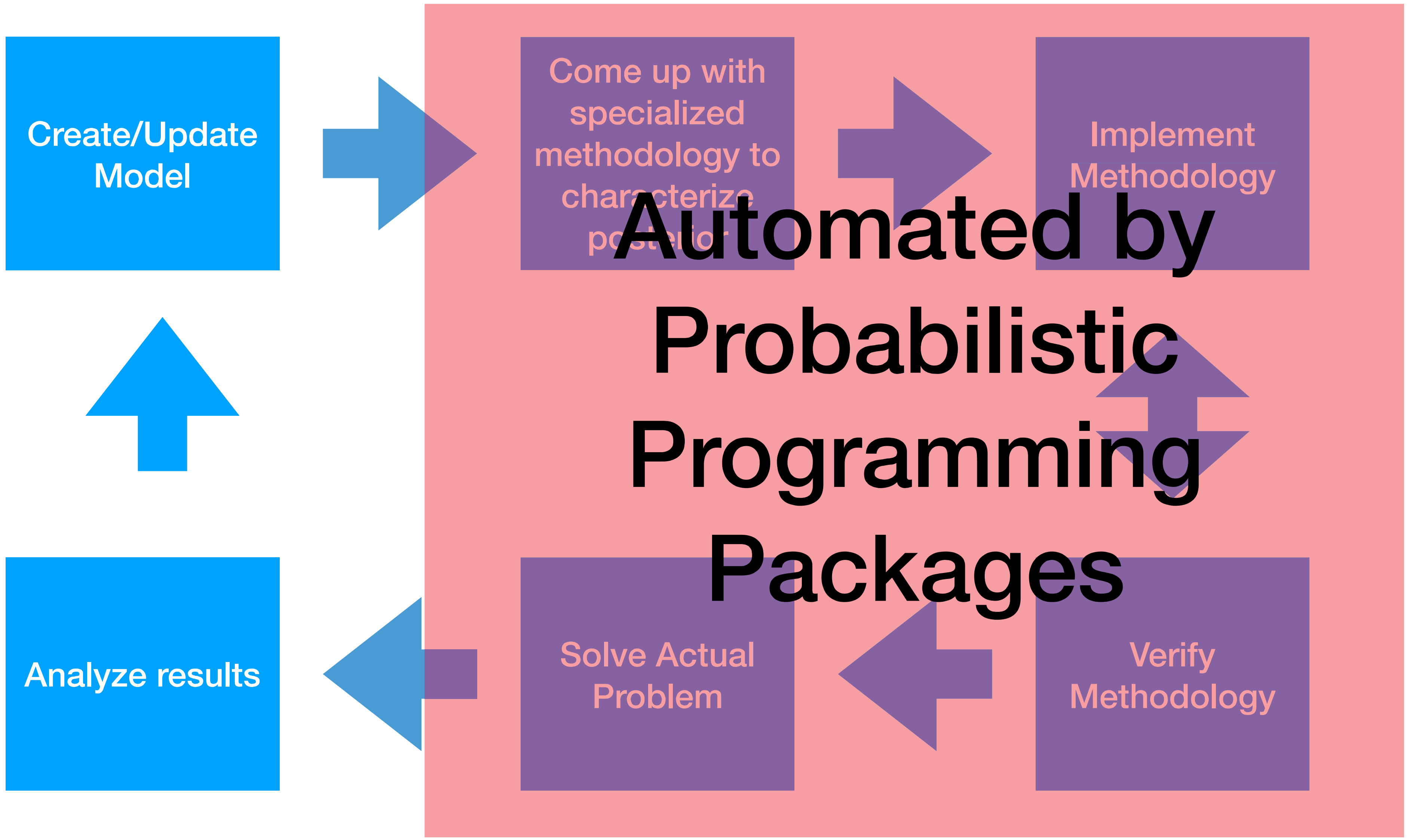
? $Z = \int p(y \mid x)\, p(x)\, dx$?

How do we characterize the posterior?

- Analytical
- Perturbation methods
- Sampling methods
- Variational inference
- Approximate Bayesian computation
- …

PREDICTIVE
SCIENCE LABORATORY

Probabilistic programming automates Bayesian inference



Probabilistic programming packages

BUGS

PyMC3

Stan

PYRO

Edward

TensorFlow

TensorfFlow Probability

Example: Hierarchical model to code

$$
\begin{aligned}
\lambda_1 &\sim \text{Exp}(\lambda_1|\alpha),\\
\lambda_2 &\sim \text{Exp}(\lambda_2|\alpha),\\
\tau &\sim \text{DiscreteUniform}([1851, 1852, \dots, 1961]),\\
\lambda &= \begin{cases} \lambda_1 & \text{if } t < \tau \\ \lambda_2 & \text{if } t \geq \tau \end{cases}\\
\text{obs}_i &\sim \text{Poisson}(\lambda_i)
\end{aligned}
$$

```python
disaster_model = pm.Model()
lower_year, upper_year = years.min(), years.max()
alpha = 1.

with disaster_model:
    # define the prior
    lambda_1 = pm.Exponential("lambda_1", lam=alpha)
    lambda_2 = pm.Exponential("lambda_2", lam=alpha)
    tau      = pm.DiscreteUniform('tau', lower=lower_year, upper=upper_year)
    lmbda = pm.Deterministic('lambda', tt.switch(tau >= years, lambda_1, lambda_2))

    # define the likelihood
    x = pm.Poisson('x', lmbda, observed=disasters)
```

```python
with disaster_model:
    trace = pm.sample(draws=40000, progressbar=True)
```