Probabilistic programming with Edward

John Reid

Biostatistics Unit,
School of Clinical Medicine,
Cambridge University

April 25, 2017

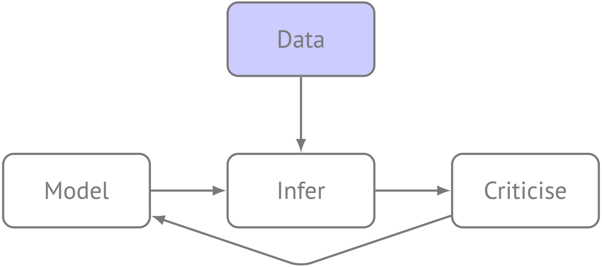
George E.P. Box (1919 - 2013)

An iterative process for science:

1. Build a model of the science
2. Infer the model given data
3. Criticize the model given data

Box and Hunter, 1962, 1965; Box and Hill, 1967; Box, 1976, 1980

Box’s Loop



Edward is a library designed around this loop.
(Box, 1976, 1980; David M. Blei, 2014)

Edward is a probabilistic programming language, designed for fast experimentation and research (Tran et al., 2017).

Modelling

- Composable Turing-complete language of random variables.
- Examples: Graphical models, neural networks, probabilistic programs.
- Many data types, tensor vectorization, broadcasting, 3rd party support.

Inference

- Composable language for hybrids, message passing, data subsampling.
- Examples: Black box VI, Hamiltonian MC, stochastic gradient MCMC, generative adversarial networks.
- Infrastructure to develop your own algorithms.

Criticism

- Examples: Scoring rules, hypothesis tests, predictive checks.

Built on TensorFlow (features distributed computing, GPUs, autodiff).

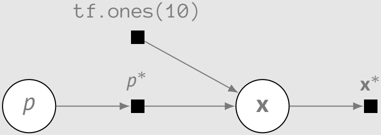
```
# DATA
x_data = np.array([0, 1, 0, 0, 0, 0, 0, 0, 0, 1])

# MODEL
p = Beta(a=1.0, b=1.0)
x = Bernoulli(p=tf.ones(10) * p)

# VARIATIONAL DISTRIBUTION
qp_a = tf.nn.softplus(tf.Variable(tf.random_normal([])))
qp_b = tf.nn.softplus(tf.Variable(tf.random_normal([])))
qp = Beta(a=qp_a, b=qp_b)

# INFERENCE
inference = ed.KLqp({p: qp}, data={x: x_data})
inference.run(n_iter=500)

# CRITICISM
x_post = ed.copy(x, {p : qp})
def T(xs, zs):
  return tf.reduce_mean(xs[x_post])
ed.ppc(T, data={x_post: x_data})
```

Model code

```
p = Beta(a=1.0, b=1.0)
x = Bernoulli(p=tf.ones(10) * p)
```

The random variables p and x are represented by tensors p^* and x^* in the tensorflow computation graph

Computation graph



Random variables are equipped with methods for likelihoods $\log(x|p)$, expectations $\mathbb{E}_{p(x|p)}[x]$, and sampling $\sim p(x|p)$.

Graph can be executed by `x.value()` which returns the tensor x^* and simulates the generative process.

Key concept is compositionality:

- Graphs can contain arbitrary tensorflow constructs
- Tensorflow conditional evaluations permit construction of nonparametric processes
- Graphs can interface with third party tensorflow libraries, e.g. Keras for deep learning

Deep generative model

```
from edward.models import Bernoulli, Normal
from keras.layers import Dense

z = Normal(mu=tf.zeros([N, d]), sigma=tf.ones([N, d]))
h = Dense(256, activation='relu')(z.value())
x = Bernoulli(logits=Dense(28 * 28)(h))
```

[1] G. E. P. Box and William G. Hunter. “A Useful Method for Model-Building”. In: *Technometrics* 4.3 (1962), pp. 301–318. JSTOR: 1266570.

[2] G. E. P. Box and William G. Hunter. “The Experimental Study of Physical Mechanisms”. In: *Technometrics* 7.1 (1965), pp. 23–42. JSTOR: 1266125.

[3] G. E. P. Box and W. J. Hill. “Discrimination among Mechanistic Models”. In: *Technometrics* 9.1 (1967), pp. 57–71. JSTOR: 1266318.

[4] George E. P. Box. “Science and Statistics”. In: *Journal of the American Statistical Association* 71.356 (1976), pp. 791–799. JSTOR: 2286841.

[5] George E. P. Box. “Sampling and Bayes’ Inference in Scientific Modelling and Robustness”. In: *Journal of the Royal Statistical Society. Series A (General)* 143.4 (1980), pp. 383–430. JSTOR: 2982063.

[6] David M. Blei. “Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models”. In: *Annual Review of Statistics and Its Application* 1.1 (2014), pp. 203–232.

[7] Dustin Tran et al. “Deep Probabilistic Programming”. In: (Jan. 13, 2017). arXiv: 1701.03757 [cs, stat].