JointDistributions in TensorFlow Probability

Dan Piponi, Dave Moore, Christopher Suter, Junpeng Lao, Joshua V. Dillon

JointDistributions represent directed graphical models in TensorFlow Probability. They:
- Extend and generalize the interface of univariate Distributions.
- Provide a shared representation for both sampling and log prob queries.
- Abstract multiple flavors of model specification behind a common interface.
- Support vectorized sampling and inference.

You can use them today to build models and run inference at scale, in TensorFlow or JAX.

The code matches the math.

```
\[
\begin{align*}
  s &\sim \text{InverseGamma}(3, 2) \\
  m &\sim \text{Normal}(0, 1) \\
  x &\sim \text{Normal}(m, s)
\end{align*}
\]
```

```
simple_model = tfd.JointDistributionSequential(dict(
    s=tfd.InverseGamma(3., 2.),  # s
    m=tfd.Normal(0., 1.),        # m
    lambda m, s: tfd.Normal(m, s),  # x
))
```

```
s, m, x = simple_model.sample()
```

A unified interface.

```
# Draw a prior sample and evaluate its log density.
s, m, x = simple_model.sample()
simple_model.log_prob(s, m, x)
```

```
# Draw predictive samples given known 's'.
_, m, x = simple_model.sample(sample_shape=[100], s=2.0)
```

```
# Inspect conditional distributions.
(_, m_dist, x_dist), _ = simple_model.sample_distributions(s=2.0)
```

Complicated things are simple.

```
alpha = tfp.util.TransformedVariable(
    init_alpha, tfb.Softplus())
beta = tf.Variable(init_beta)
```

```
@tfd.JointDistributionCoroutineAutoBatched
def latent_dirichlet_allocation():
    n = yield tfd.Poisson(rate=avg_doc_length)
    theta = yield tfd.Dirichlet(concentration=alpha)
    z = yield tfd.Multinomial(total_count=n, probs=theta)
    w = yield tfd.Multinomial(total_count=x, logits=beta)
```

Discussion

- JDs deliberately focus on deterministic control flow, for easy vectorization.
- JD models may refer to trainable parameters as `tfVariables`, as in our LDA example. Variables are automatically tracked and may be accessed as `alpha, beta = lda.trainable_variables`
- Most TFP inference APIs take a callable specifying a `target_log_prob_fn`; joint distribution methods integrate seamlessly. TFP also provides utilities to generate fully-factorized or structured variational distributions from joint distribution models.
- Like most of TFP, joint distributions are supported in both TensorFlow and JAX backends: import `tensorflow_probability as tfp` or from `tensorflow_probability.substrates import jax as tfp`

Contact

https://www.tensorflow.org/probability/
Reach out to us on our Google group with questions or feedback: tfprobability@tensorflow.org