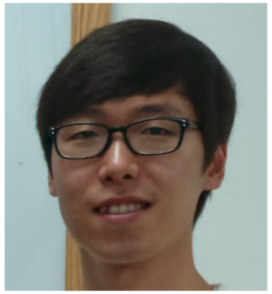
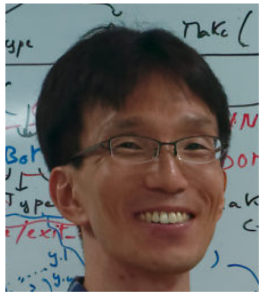
Towards Verified Stochastic Variational Inference for Probabilistic Programs



Wonyeol Lee[1] Hangyeol Yu[1] Xavier Rival[2] Hongseok Yang[1]

[1]KAIST, South Korea [2]INRIA/ENS/CNRS, France

POPL 2020

Probabilistic Programming

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```

Probabilistic Programming

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```

Probabilistic Programming

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample(“z”, Normal(0., 5.))
  if (z > 0):  pyro.sample(“x”, Normal( 1., 1.), obs=0.)
  else:        pyro.sample(“x”, Normal(-2., 1.), obs=0.)
```

Probabilistic Programming

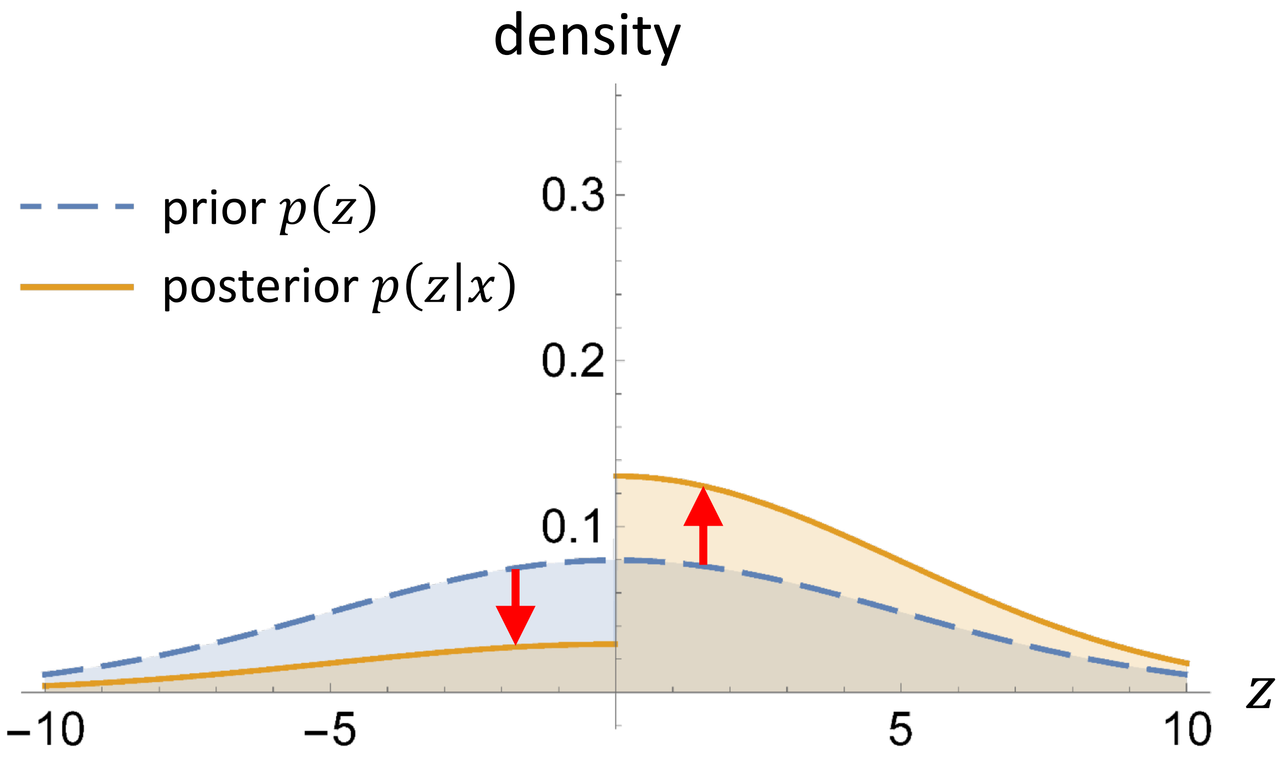
- Example 1:

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):   pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:         pyro.sample("x", Normal(-2., 1.), obs=0.)
```

Probabilistic Programming

- Example 1:

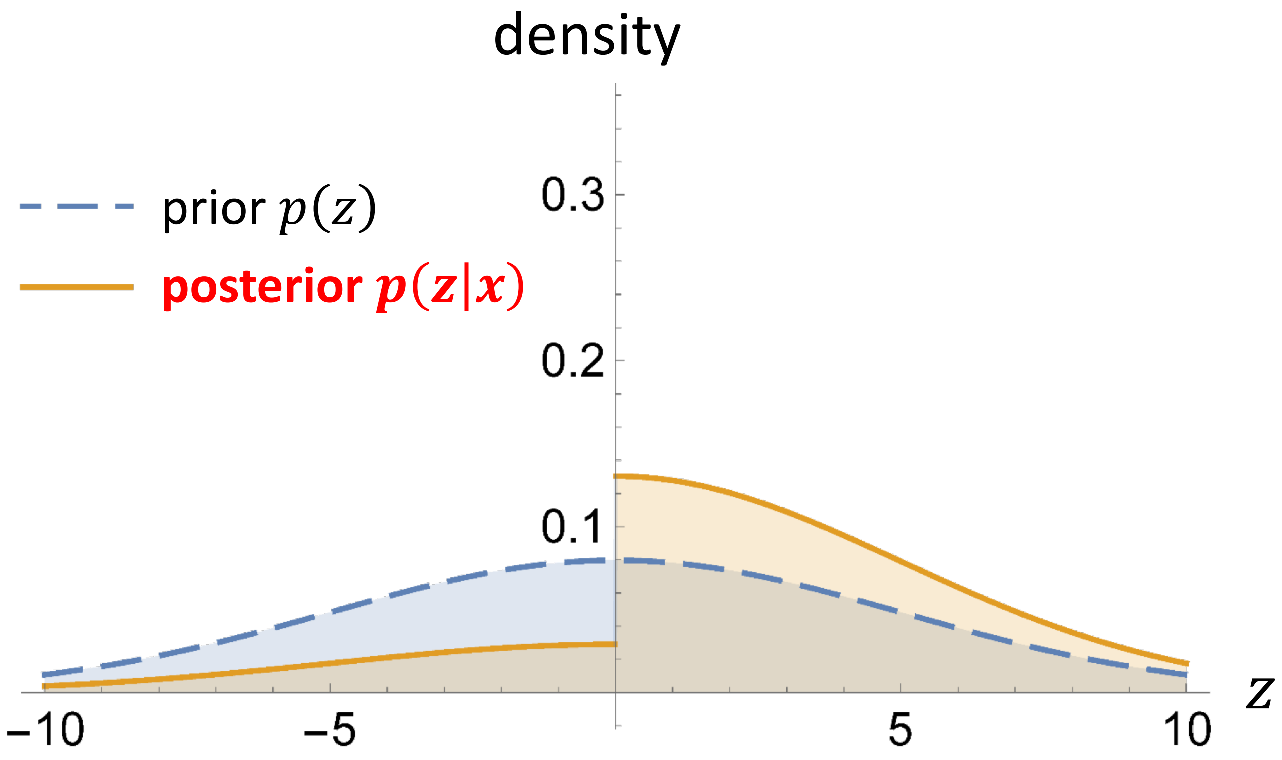
```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```



Probabilistic Programming

- Example 1:

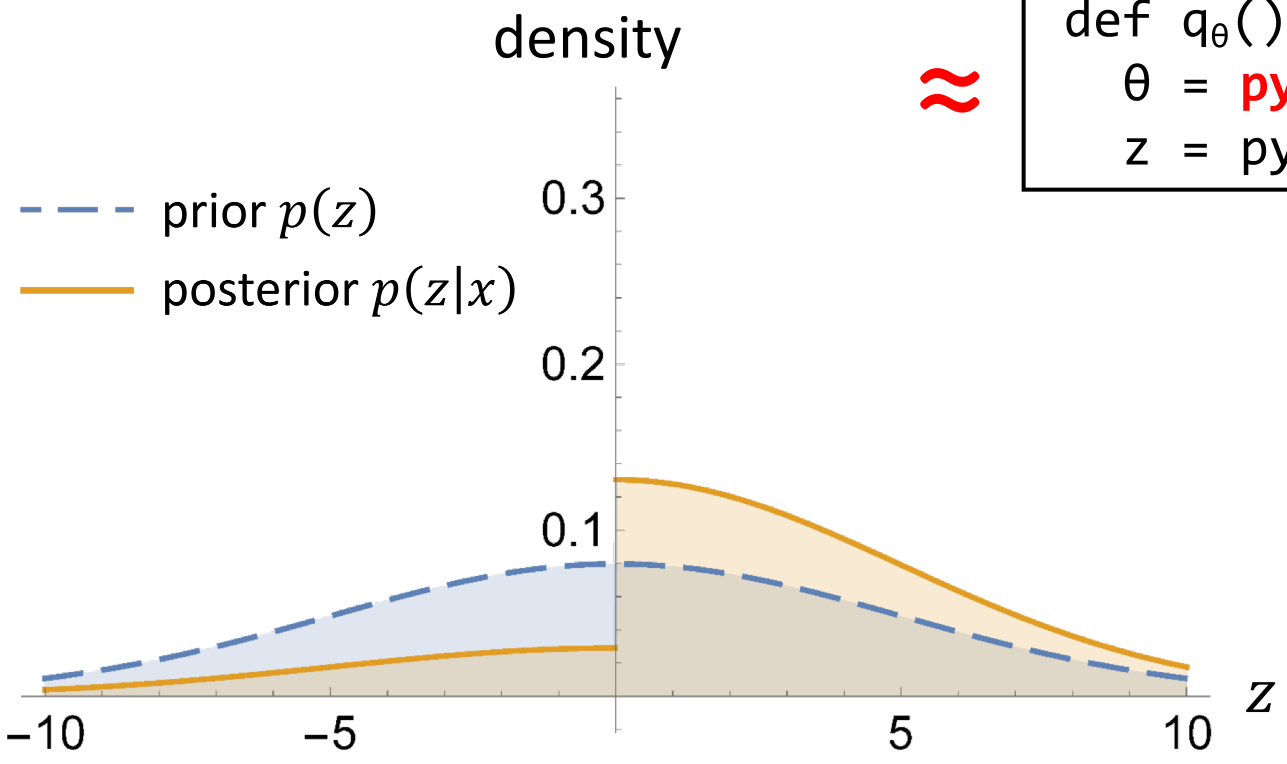
```
def p(): # model_eg1
  z = pyro.sample(“z”, Normal(0., 5.))
  if (z > 0):  pyro.sample(“x”, Normal( 1., 1.), obs=0.)
  else:        pyro.sample(“x”, Normal(-2., 1.), obs=0.)
```



Stochastic Variational Inference

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```



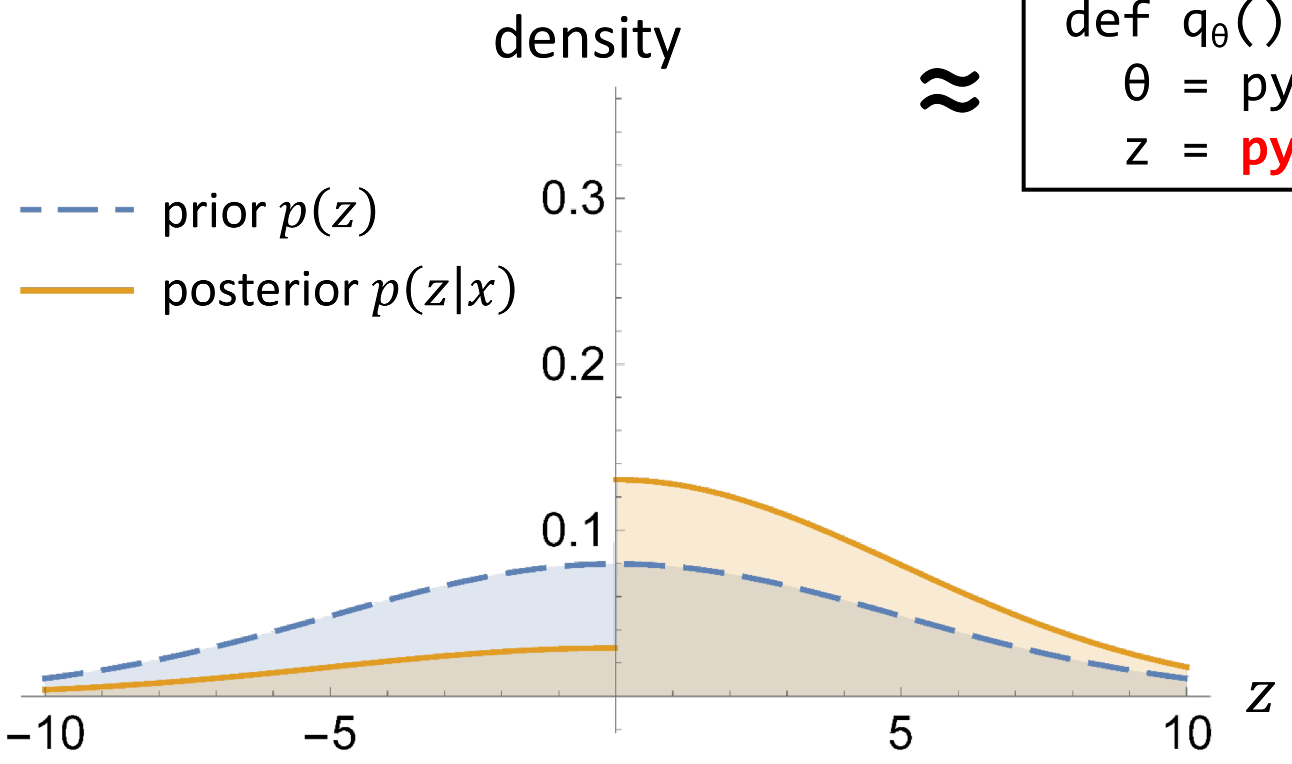
≈

```
def qθ(): # guide_eg1
  θ = pyro.param("θ", 0.)
  z = pyro.sample("z", Normal(θ, 1.))
```

Stochastic Variational Inference

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample(“z”, Normal(0., 5.))
  if (z > 0):  pyro.sample(“x”, Normal( 1., 1.), obs=0.)
  else:        pyro.sample(“x”, Normal(-2., 1.), obs=0.)
```



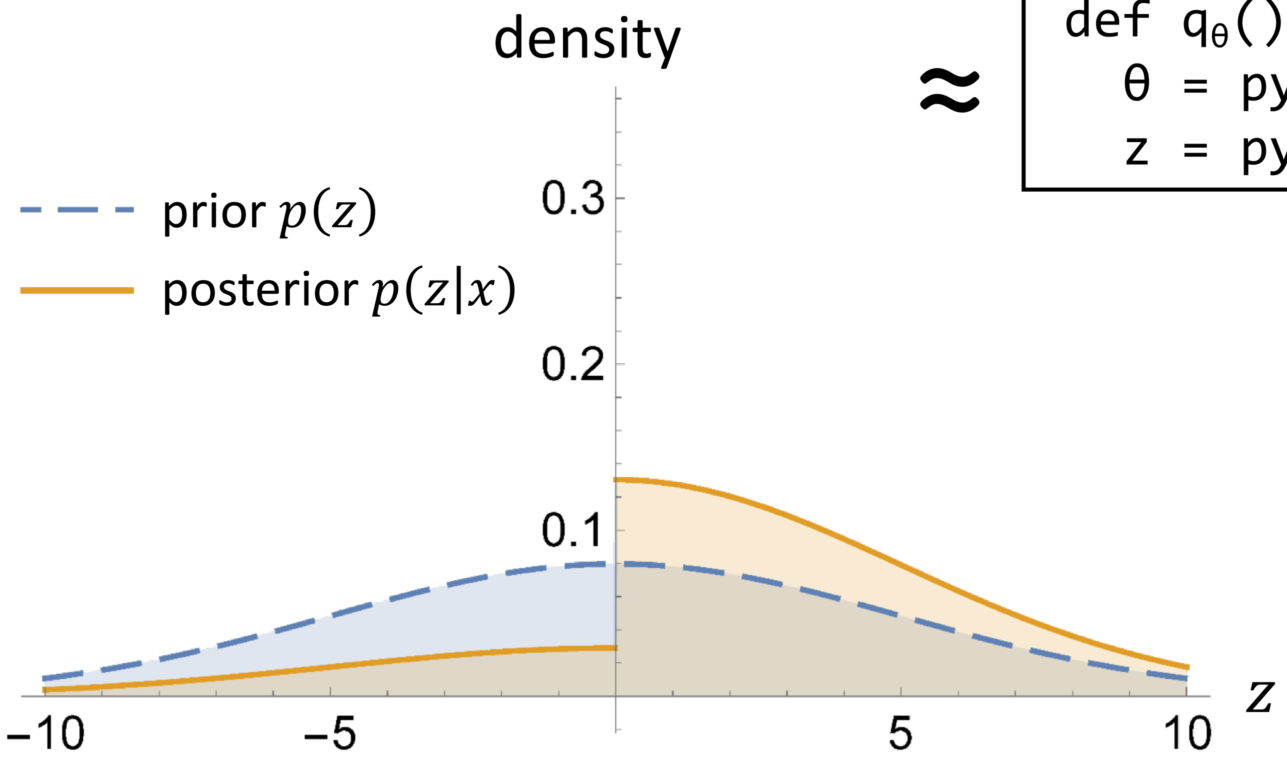
≈

```
def qθ(): # guide_eg1
  θ = pyro.param(“θ”, 0.)
  z = pyro.sample(“z”, Normal(θ, 1.))
```

Stochastic Variational Inference

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```



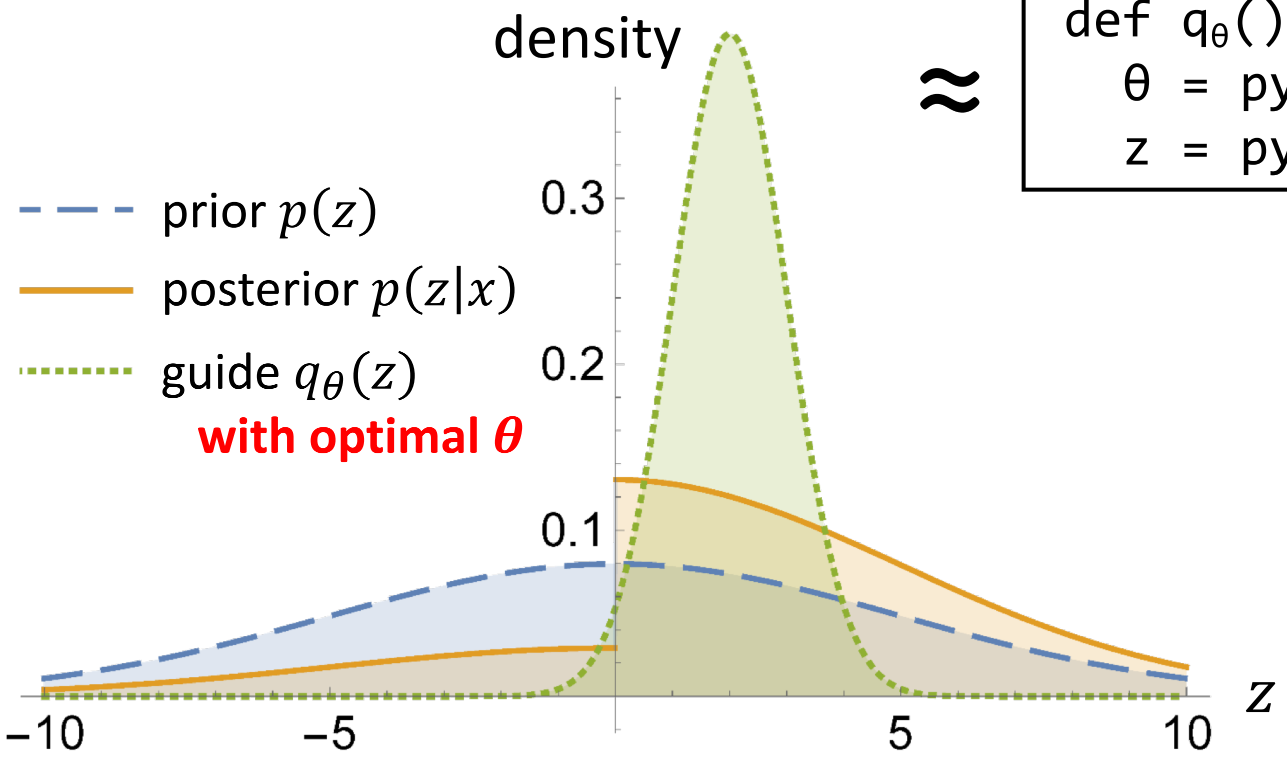
$\approx$

```
def qθ(): # guide_eg1
  θ = pyro.param("θ", 0.)
  z = pyro.sample("z", Normal(θ, 1.))
```

Stochastic Variational Inference

- Example 1:

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```



≈

```
def qθ(): # guide_eg1
  θ = pyro.param("θ", 0.)
  z = pyro.sample("z", Normal(θ, 1.))
```

Stochastic Variational Inference

- Typical optimization objective:

$$\operatorname{argmin}_\theta \underbrace{\mathrm{KL}[q_\theta(z)\,||\,p(z|x)]}_{\triangleq\, \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}.$$

Stochastic Variational Inference

- Typical optimization objective:

$$\operatorname{argmin}_\theta \underbrace{\mathrm{KL}_\theta}_{\triangleq \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}.$$

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

Stochastic Variational Inference

- Typical optimization objective:

$$\mathrm{argmin}_\theta \underbrace{\mathrm{KL}_\theta}_{\triangleq\, \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}.$$

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

- Optimization by stochastic gradient descent:

$$\theta_{n+1} \leftarrow \theta_n - 0.01\times(\nabla_\theta \mathrm{KL}_\theta)|_{\theta=\theta_n}.$$

Stochastic Variational Inference

- Typical optimization objective:

$$\mathrm{argmin}_\theta \underbrace{\qquad \mathrm{KL}_\theta \qquad}_{\triangleq\, \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}.$$

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

- Optimization by stochastic gradient descent:

$$\theta_{n+1} \leftarrow \theta_n - 0.01 \times (\nabla_\theta \mathrm{KL}_\theta)|_{\theta=\theta_n}.$$

Stochastic Variational Inference

- Typical optimization objective:

$$\operatorname{argmin}_\theta \underbrace{\mathrm{KL}_\theta}_{\triangleq\, \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}.$$

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

- Optimization by stochastic gradient descent:

$$\theta_{n+1} \leftarrow \theta_n - 0.01\times\widehat{(\nabla_\theta \mathrm{KL}_\theta)|_{\theta=\theta_n}}.$$

Stochastic Variational Inference

- Typical optimization objective:

$$\operatorname{argmin}_\theta \overbrace{\quad\quad\quad\quad}^{\mathrm{KL}_\theta} \triangleq \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]$$

- Notation:

What can go wrong?

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

- Optimization by stochastic gradient descent:

$$\theta_{n+1} \leftarrow \theta_n - 0.01 \times \widehat{(\nabla_\theta \mathrm{KL}_\theta)}|_{\theta=\theta_n}.$$

Issues in Stochastic Variational Inference

- Typical optimization objective:

$$\boxed{\text{Issue 1: Undefined } \mathrm{KL}_\theta}$$

$$\operatorname{argmin}_\theta \underbrace{\mathrm{KL}_\theta}_{\triangleq\, \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}$$

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z)\,||\,p(z|x)].$$

- Optimization by stochastic gradient descent:

$$\theta_{n+1} \leftarrow \theta_n - 0.01 \times \widehat{(\nabla_\theta \mathrm{KL}_\theta)|_{\theta=\theta_n}}.$$

Issues in Stochastic Variational Inference

- Typical optimization objective:

Issue 1: Undefined KL_θ

$$\operatorname{argmin}_\theta \underbrace{\mathrm{KL}_\theta}_{\triangleq\, \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}$$

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

- Optimization by stochastic gradient descent:

$$\theta_{n+1} \leftarrow \theta_n - 0.01 \times \widehat{(\nabla_\theta \mathrm{KL}_\theta)|_{\theta=\theta_n}}.$$

Issue 2: Undefined $\nabla_\theta \mathrm{KL}_\theta$

Issues in Stochastic Variational Inference

- Typical optimization objective:

$$\operatorname{argmin}_\theta \underbrace{\mathrm{KL}_\theta}_{\triangleq \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]}$$

Issue 1: Undefined KL_θ

- Notation:

$$\mathrm{KL}_\theta = \mathrm{KL}[q_\theta(z) \,||\, p(z|x)].$$

- Optimization by stochastic gradient descent:

Issue 3: Wrong estimate

$$\theta_{n+1} \leftarrow \theta_n - 0.01 \times \widehat{(\nabla_\theta \mathrm{KL}_\theta)|_{\theta=\theta_n}}.$$

Issue 2: Undefined $\nabla_\theta \mathrm{KL}_\theta$

Issue 1: Undefined KL_θ

$$\begin{aligned}\mathrm{KL}_\theta &= \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right] \\ &= \int dz\, \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)}\right)\end{aligned}$$

KL_θ could be undefined for two reasons.

Issue 1: Undefined KL_θ

$$\mathrm{KL}_\theta = \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right]$$

$$= \int dz \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)}\right)$$

KL_θ could be undefined for two reasons.

(a) Undefined integrand: $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

Issue 1: Undefined KL_θ

$$\begin{aligned}\mathrm{KL}_\theta &= \mathbb{E}_{q_\theta(z)}\left[\log \frac{q_\theta(z)}{p(z|x)}\right] \\ &= \int dz\, \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)}\right)\end{aligned}$$

KL_θ could be undefined for two reasons.

(a) Undefined integrand: $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

(b) Undefined integral: $\int dz\, (\cdots)$ is not integrable.

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2
  ...
  sigma = pyro.sample("sigma", Uniform(0., 10.))
  ...
  pyro.sample("obs", Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
  ...
  sigma = pyro.sample("sigma", Normal(θ, 0.05))
```

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2
   ...
   sigma = pyro.sample(“sigma”, Uniform(0., 10.))
   ...
   pyro.sample(“obs”, Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
   ...
   sigma = pyro.sample(“sigma”, Normal(θ, 0.05))
```

- KL_θ is undefined.

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2
  ...
  sigma = pyro.sample("sigma", Uniform(0., 10.))
  ...
  pyro.sample("obs", Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
  ...
  sigma = pyro.sample("sigma", Normal(0, 0.05))
```

- KL_{θ} is undefined.

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2
  ...
  sigma = pyro.sample("sigma", Uniform(0., 10.))
  ...
  pyro.sample("ob
```

```
def qθ(): # guide
  ...
  sigma = pyro.sample("sigma", Normal(θ, 0.05))
```

$q_\theta(z) \neq 0$ and $p(z|x) = 0$ for $z < 0$.

- KL_θ is undefined. Reason: (a) undefined integrand.

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2
  ...
  sigma = pyro.sample("sigma", Uniform(0., 10.))
  ...
  pyro.sample("obs", Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
  ...
  sigma = pyro.sample("sigma", Normal(θ, 0.05))
```

- KL_θ is undefined. Reason: (a) undefined integrand.
- [Q] How to fix model?

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2'
  ...                                  Normal(5., 5.)
  sigma = pyro.sample("sigma", Uniform(0., 10.))
  ...                                  abs(sigma)
  pyro.sample("obs", Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
  ...
  sigma = pyro.sample("sigma", Normal(θ, 0.05))
```

- [Q] How to fix model?

Issue 1: Example

- Example 2: Bayesian regression (from Pyro webpage).

```
def p(): # model_eg2'
  ...                                    Normal(5., 5.)
  sigma = pyro.sample("sigma", Uniform(0., 10.))
  ...                                  abs(sigma)
  pyro.sample("obs", Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
  ...
  sigma = pyro.sample("sigma", Normal(θ, 0.05))
```

- KL_θ is still undefined.
- [Q] How to fix model?

Issue 1: Example

- Example 2: Bayesian regression (fr

$$\int_{-\infty}^{\infty} \frac{1}{|z|} \times \mathcal{N}(z;\theta,0.05)\, dz = \infty$$

```
def p(): # model_eg2’
  ...
  sigma = pyro.sample(“sigma”, Uniform(0., 10.))
  ...                                   abs(sigma)
  pyro.sample(“obs”, Normal(..., sigma), obs=...)
```

```
def qθ(): # guide_eg2
  ...
  sigma = pyro.sample(“sigma”, Normal(θ, 0.05))
```

- KL_θ is still undefined. Reason: (b) undefined integral.
- [Q] How to fix model?

Issue 2: Undefined $\nabla_\theta \mathrm{KL}_\theta$

KL_θ could be non-differentiable w.r.t. θ.

Issue 2: Undefined $\nabla_\theta \mathrm{KL}_\theta$

KL_θ could be non-differentiable w.r.t. θ.

```
def p(): # model_eg1
  z = pyro.sample(“z”, Normal(0., 5.))
  if (z > 0):  pyro.sample(“x”, Normal( 1., 1.), obs=0.)
  else:        pyro.sample(“x”, Normal(-2., 1.), obs=0.)
```

```
def qθ(): # guide_eg1’
  θ = pyro.param(“θ”, 0.)
  z = pyro.sample(“z”, Normal(0, 1.))
                     Uniform(θ-1.,θ+1.)
```

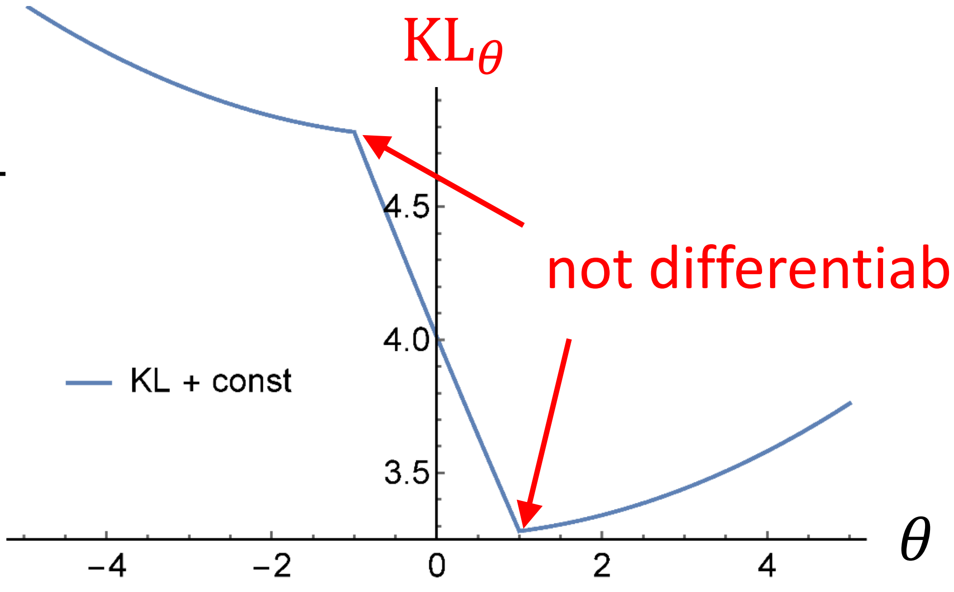
Issue 2: Undefined $\nabla_\theta \mathrm{KL}_\theta$

KL_θ could be non-differentiable w.r.t. θ.

```
def p(): # model_eg1
  z = pyro.sample("z", Normal(0., 5.))
  if (z > 0):  pyro.sample("x", Normal( 1., 1.), obs=0.)
  else:        pyro.sample("x", Normal(-2., 1.), obs=0.)
```

```
def qθ(): # guide_eg1'
  θ = pyro.param("θ", 0.)
  z = pyro.sample("z", Normal(0, 1.))
```

```
Uniform(θ-1.,θ+1.)
```



Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Score Estimator:

$$\widehat{\nabla_\theta \mathrm{KL}_\theta} = \nabla_\theta \log q_\theta(z') \times \log \frac{q_\theta(z')}{p(z', x)}$$

where z' is sampled from q_θ.

Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Score Estimator:

$$\widehat{\nabla_\theta \mathrm{KL}_\theta} = \nabla_\theta \log q_\theta(z') \times \log \frac{q_\theta(z')}{p(z', x)}$$

where z' is sampled from q_θ.

- Theorem:

$$\nabla_\theta \mathrm{KL}_\theta = \mathbb{E}_{z'}[\widehat{\nabla_\theta \mathrm{KL}_\theta}]$$

if some requirements are met.

Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Score Estimator:

$$\widehat{\nabla_\theta \mathrm{KL}_\theta} = \nabla_\theta \log q_\theta(z') \times \log \frac{q_\theta(z')}{p(z',x)}$$

where z' is

$\widehat{\nabla_\theta \mathrm{KL}_\theta}$ could be an incorrect estimator if the requirements are not met.

- Theorem:

$$\nabla_\theta \mathrm{KL}_\theta \neq \mathbb{E}_{z'}[\widehat{\nabla_\theta \mathrm{KL}_\theta}]$$

if some requirements are met.

Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Proof of Theorem:

$$\nabla_\theta \mathrm{KL}_\theta = \nabla_\theta \int \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)} \right) dz$$

$$= \int \nabla_\theta \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)} \right) dz$$

$$\ldots$$

$$= \mathbb{E}_{z'} \left[\nabla_\theta \log q_\theta(z') \times \log \frac{q_\theta(z')}{p(z',x)} \right]$$

$$= \mathbb{E}_{z'} [\widehat{\nabla_\theta \mathrm{KL}_\theta}].$$

Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Proof of Theorem:

$$\nabla_\theta \mathrm{KL}_\theta = \nabla_\theta \int \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)}\right) dz$$

$$= \int \nabla_\theta \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)}\right) dz$$

This might fail.

$$\cdots$$

$$= \mathbb{E}_{z'}\left[\nabla_\theta \log q_\theta(z') \times \log \frac{q_\theta(z')}{p(z',x)}\right]$$

$$= \mathbb{E}_{z'}[\widehat{\nabla_\theta \mathrm{KL}_\theta}].$$

Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Proof of Theorem:

$$\nabla_\theta \mathrm{KL}_\theta = \nabla_\theta \int \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)} \right) dz$$

$$= \int \nabla_\theta \left(q_\theta(z) \log \frac{q_\theta(z)}{p(z|x)} \right) dz$$

This might fail.

$\cdots$

$$\times \log \frac{q_\theta(z')}{p(z',x)}]$$

$$\nabla_\theta \int_{D(\theta)} (\cdots) dz \neq \int_{D(\theta)} \nabla_\theta (\cdots) dz$$

Issue 3: Wrong Estimate of $\nabla_\theta \mathrm{KL}_\theta$

- Proof of Theorem:

$$\nabla_\theta \mathrm{KL}_\theta = \nabla_\theta \int \left(q_\theta(z)\log\frac{q_\theta(z)}{p(z|x)}\right)dz$$

$$= \int \nabla_\theta \left(q_\theta(z)\log\frac{q_\theta(z)}{p(z|x)}\right)dz$$

This might fail...

$$\nabla_\theta \int_{D(\theta)} (\cdots)dz \neq \int_{D(\theta)} \nabla_\theta(\cdots)dz$$

$$\times \log\frac{q_\theta(z')}{p(z',x)}\Big]$$

How to check that such issues do not happen?

How to Verify Issues?

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

   (b) $\int(\cdots)dz$ is not integrable.

2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   - $\nabla_\theta \int (\cdots)dz \neq \int \nabla_\theta(\cdots)dz$.

How to Verify Issues?

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

   (b) $\int(\cdots)dz$ is not integrable.

2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   - $\nabla_\theta \int (\cdots)dz \neq \int \nabla_\theta(\cdots)dz$.

How to Verify Issues?

E.g., $\mathcal{N}(z; \_, \_) > 0$ for all $z \in \mathbb{R}$.

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

   (b) $\int(\cdots)dz$ is not integrable.

2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   - $\nabla_\theta \int (\cdots)dz \neq \int \nabla_\theta(\cdots)dz$.

How to Verify Issues?

E.g., $\mathcal{N}(z; \_, \_) > 0$ for all $z \in \mathbb{R}$.

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

   (b) $\int (\cdots) dz$ is not integrable.

   $\Leftrightarrow$ familiar type of static analysis problem

2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   - $\nabla_\theta \int (\cdots) dz \neq \int \nabla_\theta (\cdots) dz$.

How to Verify Issues?

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

   (b) $\int (\cdots) dz$ is not integrable.

2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   - $\nabla_\theta \int (\cdots) dz \neq \int \nabla_\theta (\cdots) dz.$

How to Verify Issues?

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$ for some z.

   (b) $\int (\cdots) dz$ is not integrable.

2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   - $\nabla_\theta \int (\cdots) dz \neq \int \nabla_\theta (\cdots) dz$.

$\not\Leftrightarrow$ familiar type of static analysis problems

How to Verify Issues?

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$

   $\neg$ (b) $\int (\cdots) dz$ is not integrable.

$\neg$ 2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

   $\neg$ • $\nabla_\theta \int (\cdots) dz \neq \int \nabla_\theta (\cdots) dz$.

Sufficient conditions about

- differentiability of simpler func's,
- boundedness of simpler func's.

familiar type of
static analysis problems

How to Verify Issues?

1. Undefined KL_θ.

    (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$

    $\neg$ (b) $\int (\cdots) dz$ is not integrable.

$\neg$ 2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$.

    $\neg$ • $\nabla_\theta \int (\cdots) dz \neq \int \nabla_\theta (\cdots) dz$.

Sufficient conditions about

- differentiability of simpler func's,
- boundedness of simpler func's.

$\Swarrow$ $\Updownarrow$

familiar type of
static analysis problems

How to Verify Issues?

Sufficient conditions about

- differentiability of simpler func's,
- boundedness of simpler func's.

1. Undefined KL_θ.

   (a) $q_\theta(z) \neq 0$ and $p(z|x) = 0$

   ¬ (b) $\int (\cdots) dz$ is not integrable.

¬ 2. Undefined $\nabla_\theta \mathrm{KL}_\theta$.

3. Wrong estimate of $\nabla_\theta \mathrm{KL}_\theta$

familiar type of
static analysis problems

Density semantics is used for formalization.

Our automatic verifier.

Our Automatic Verifier

- Analyzes models/guides written in Pyro.
- Verifies that issue 1a (below) does not happen:

$$q_\theta(z) \neq 0 \text{ and } p(z|x) = 0 \text{ for some } z.$$

Our Automatic Verifier

- Analyzes models/guides written in Pyro.

- Verifies that issue 1a (below) does not happen:

$$q_\theta(z) \neq 0 \text{ and } p(z|x) = 0 \text{ for some } z.$$

- Handles many (but not all) features of Python/PyTorch/Pyro (e.g., tensor broadcasting).

github.com/wonyeol/static-analysis-for-support-match

Experimental Evaluation

- Analyzed 8 representative examples (from Pyro webpage).

			Total #				Total dimension		
Name	Corresponding probabilistic model	LoC	`for`	`plate`	`sample`	`score`	`sample`	`score`	θ
`br`	Bayesian regression	27	0	1	10	1	10	170	9
`csis`	Inference compilation	31	0	0	2	2	2	2	480
`lda`	Amortised latent Dirichlet allocation	76	0	5	8	1	21008	64000	121400
`vae`	Variational autoencoder (VAE)	91	0	2	2	1	25600	200704	353600
`sgdef`	Deep exponential family	94	0	8	12	1	231280	1310720	231280
`dmm`	Deep Markov model	246	3	2	2	1	640000	281600	594000
`ssvae`	Semi-supervised VAE	349	0	2	4	1	24000	156800	844000
`air`	Attend-infer-repeat	410	2	2	6	1	20736	160000	6040859

Experimental Evaluation

Name	Valid?	Time
`br`	x	0.006
`csis`	o	0.007
`lda`	x	0.014
`vae`	o	0.005
`sgdef`	o	0.070
`dmm`	o	0.536
`ssvae`	o	0.013
`air`	o	4.093

Experimental Evaluation

Example 2 → br

Name	Valid?	Time
`br`	x	0.006
`csis`	o	0.007
`lda`	x	0.014
`vae`	o	0.005
`sgdef`	o	0.070
`dmm`	o	0.536
`ssvae`	o	0.013
`air`	o	4.093

p uses `Uniform`.
q_θ uses `Normal`.

Experimental Evaluation

Example 2 → br

Name	Valid?	Time
br	x	0.006
csis	o	0.007
lda	x	0.014
vae	o	0.005
sgdef	o	0.070
dmm	o	0.536
ssvae	o	0.013
air	o	4.093

p uses Uniform.
q_θ uses Normal.

p uses Dirichlet.
q_θ uses Delta.

Experimental Evaluation

Example 2 → br

Name	Valid?	Time
br	x	0.006
csis	o	0.007
lda	x	0.014
vae	o	0.005
sgdef	o	0.070
dmm	o	0.536
		0.013
		4.093

p uses Uniform.
q_θ uses Normal.

p uses Dirichlet.
q_θ uses Delta.

Performs different inference algorithm, using variational inference engine.

Key Messages

- ML algorithms often make nontrivial assumptions implicitly.
  They could be violated sometimes; need to be checked carefully.

Key Messages

- ML algorithms often make nontrivial assumptions implicitly.
  They could be violated sometimes; need to be checked carefully.

- This gives opportunities for PL research.
  E.g., how to check these assumptions automatically?

Key Messages

- ML algorithms often make nontrivial assumptions implicitly.
  They could be violated sometimes; need to be checked carefully.

Thank you!

- This gives opportunities for PL research.
  E.g., how to check these assumptions automatically?

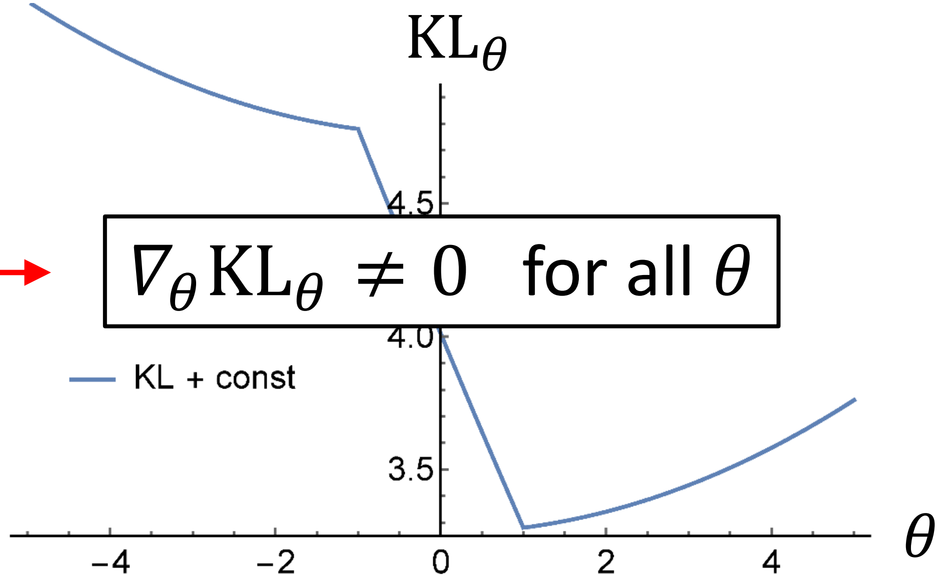
Issue 3: Example

- Example 1’:

```
def p(): # model_eg1
  z = pyro.sample(“z”, Normal(0., 5.))
  if (z > 0):   pyro.sample(“x”, Normal( 1., 1.), obs=0.)
  else:         pyro.sample(“x”, Normal(-2., 1.), obs=0.)
```

```
def qθ(): # guide_eg1’
  θ = pyro.param(“θ”, 0.)
  z = pyro.sample(“z”, Uniform(θ-1., θ+1.))
```

$\widehat{\nabla_\theta \mathrm{KL}_\theta} = 0$ for all θ, z' ⟷ $\nabla_\theta \mathrm{KL}_\theta \neq 0$ for all θ



Issue 3: Example

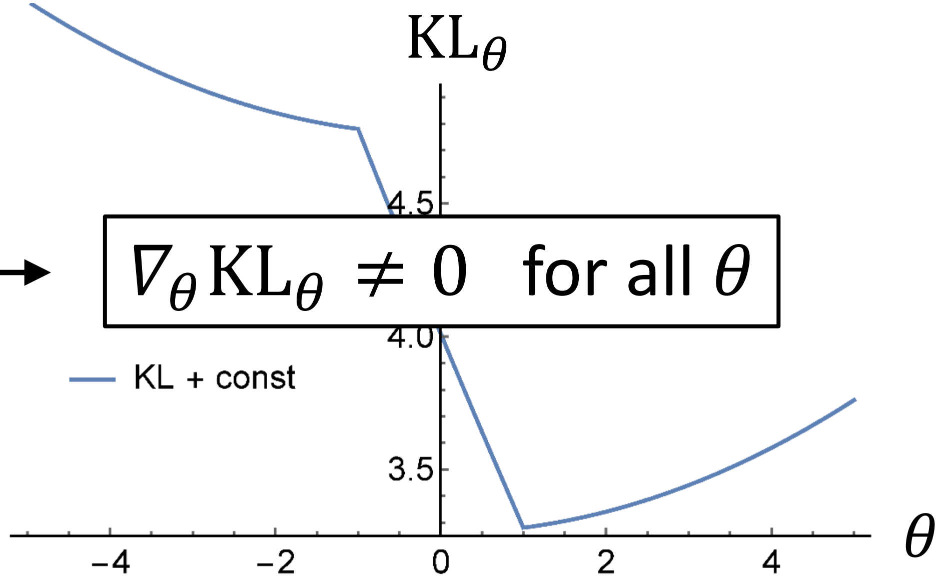
- Example 1’:

```
def p(): # model_eg1
  z = pyro.sample(“z”, Normal(0., 5.))
  if (z > 0):  pyro.sample(“x”, Normal( 1., 1.), obs=0.)
  else:        pyro.sample(“x”, Normal(-2., 1.), obs=0.)
```

```
1’
’, θ.)
”, Uniform(θ-1., θ+1.))
```

$$\nabla_\theta \int_{\theta-1}^{\theta+1} (\cdots) dz \neq \int_{\theta-1}^{\theta+1} \nabla_\theta (\cdots) dz$$

$$\boxed{\widehat{\nabla_\theta \mathrm{KL}_\theta} = 0 \quad \text{for all } \theta, z'} \longleftrightarrow \boxed{\nabla_\theta \mathrm{KL}_\theta \neq 0 \quad \text{for all } \theta}$$



Sufficient Conditions

- Assume: $p(z, x)$ and $q_\theta(z)$ use only normal distributions.

  $\mu,\ \sigma$: mean, standard deviation in $p(z, x)$.

  μ', σ': mean, standard deviation in $q_\theta(z)$.

- Theorem: The following ensure non-occurrence of the issues.
  - $|\mu(z)| \leq \exp(f(|z|))$ for some affine f.
    $\exp(g(|z|)) \leq \sigma(z) \leq \exp(h(|z|))$ for some affine g, h.
  - μ', σ' are continuously differentiable w.r.t. θ.

Sufficient Conditions

1b. $\int (\cdots) dz$ is not integrable.

2. $\int (\cdots) dz$ is not differentiabl

3. $\nabla_\theta \int (\cdots) dz \neq \int \nabla_\theta (\cdots) dz$.

normal distributions.

$$\int_{-\infty}^{\infty} \exp(f(z)) \times \mathcal{N}(z; \cdots)\, dz < \infty$$

for any affine f.

- Theorem: The following ensure non-occurrence of the issues.
  - $|\mu(z)| \leq \exp(f(|z|))$ for some affine f.
    $\exp(g(|z|)) \leq \sigma(z) \leq \exp(h(|z|))$ for some affine g, h.
  - μ', σ' are continuously differentiable w.r.t. θ.

Sufficient Conditions

1b. $\int(\cdots)dz$ is not integrabl

well-behaved function of θ, z

2. $\int(\cdots)dz$ is not differentiable.

$p(z,x)$.

3. $\nabla_\theta \int(\cdots)dz \neq \int \nabla_\theta(\cdots)dz$.

$q_\theta(z)$.

- Theorem: The following ensure non-occurrence of the issues.
  - $|\mu(z)| \leq \exp(f(|z|))$ for some affine f.

    $\exp(g(|z|)) \leq \sigma(z) \leq \exp(h(|z|))$ for some affine g, h.
  - μ', σ' are continuously differentiable w.r.t. θ.