# Relaxations to Discrete Latent Variables

Philip Schulz and Wilker Aziz

VI Tutorial @ Host

Site

1. Reparameterised Gradient

2. Biased Gradient Estimates for Discrete Variables

# Change of density

For Gaussians

$$z \sim \mathcal{N}(\mu, \sigma^2) \qquad \frac{z - \mu}{\sigma^2} \sim \mathcal{N}(0, 1)$$

# Change of density

For Gaussians

$$z \sim \mathcal{N}(\underbrace{\mu, \sigma^2}_{\lambda}) \qquad \underbrace{\frac{z-\mu}{\sigma^2}}_{\epsilon = t^{-1}(z,\lambda)} \sim \mathcal{N}(0,1)$$

More generally

$$f_{Z|\lambda}(z) = s(\underbrace{t^{-1}(z,\lambda)}_{\epsilon})|\det J_{t^{-1}}(z,\lambda)|$$

# Change of density

For Gaussians

$$z \sim \mathcal{N}(\underbrace{\mu, \sigma^2}_{\lambda}) \qquad \underbrace{\frac{z - \mu}{\sigma^2}}_{\epsilon = t^{-1}(z, \lambda)} \sim \mathcal{N}(0, 1)$$

More generally

$$f_{Z|\lambda}(z) = s(\underbrace{t^{-1}(z, \lambda)}_{\epsilon})|\det J_{t^{-1}}(z, \lambda)|$$

$$\mathbb{E}_{f_{Z|\lambda}(z)}[\psi(z)] =$$

# Change of density

For Gaussians

$$z \sim \mathcal{N}(\underbrace{\mu, \sigma^2}_{\lambda}) \qquad \underbrace{\frac{z-\mu}{\sigma^2}}_{\epsilon = t^{-1}(z,\lambda)} \sim \mathcal{N}(0,1)$$

More generally

$$f_{Z|\lambda}(z) = s(\underbrace{t^{-1}(z,\lambda)}_{\epsilon})|\det J_{t^{-1}}(z,\lambda)|$$

$$\mathbb{E}_{f_{Z|\lambda}(z)}\left[\psi(z)\right] = \underbrace{\mathbb{E}_{s(\epsilon)}\left[\psi(t(\epsilon,\lambda))\right]}_{\text{check class on ADVI}}$$

# Reparameterised gradient

$$\frac{\partial}{\partial \lambda} \mathbb{E}_{f_{Z|\lambda}(z)} \left[ \psi(z) \right] = \mathbb{E}_{s(\epsilon)} \left[ \frac{\partial}{\partial \lambda} \psi(t(\epsilon, \lambda)) \right]$$

# Reparameterised gradient

$$\frac{\partial}{\partial \lambda} \mathbb{E}_{f_{Z|\lambda}(z)} [\psi(z)] = \mathbb{E}_{s(\epsilon)} \left[ \frac{\partial}{\partial \lambda} \psi(t(\epsilon, \lambda)) \right]$$
$$= \mathbb{E}_{s(\epsilon)} \left[ \frac{\partial}{\partial z} \psi(z) \frac{\partial}{\partial \lambda} t(\epsilon, \lambda) \right]$$

Easy to MC estimate!

# Comparing gradient estimators

Reparameterised gradient | Score function estimator

$$\mathbb{E}_{s(\epsilon)}\left[\underbrace{\frac{\partial}{\partial z}\psi(z)\frac{\partial}{\partial \lambda}t(\epsilon,\lambda)}_{\hat{g}_{\text{rep}}}\right] = \mathbb{E}_{f_\lambda(z)}\left[\underbrace{\psi(z)\frac{\partial}{\partial \lambda}f_{Z|\lambda}(z)}_{\hat{g}_{\text{sfe}}}\right]$$

- $\hat{g}_{\text{sfe}}$ is typically cursed with variance
- but is $\hat{g}_{\text{rep}}$ available in general?

# Comparing gradient estimators

Reparameterised gradient | Score function estimator

$$\mathbb{E}_{s(\epsilon)}\left[\underbrace{\frac{\partial}{\partial z}\psi(z)\frac{\partial}{\partial\lambda}t(\epsilon,\lambda)}_{\hat{g}_{\text{rep}}}\right] = \mathbb{E}_{f_\lambda(z)}\left[\underbrace{\psi(z)\frac{\partial}{\partial\lambda}f_{Z|\lambda}(z)}_{\hat{g}_{\text{sfe}}}\right]$$

- $\hat{g}_{\text{sfe}}$ is typically cursed with variance
- but is $\hat{g}_{\text{rep}}$ available in general?
  in particular, is it available for discrete variables?

# A general reparameterisation

What transformation will always absorb the parameters of a density $f_{Z|\lambda}(z)$?

# A general reparameterisation

What transformation will always absorb the parameters of a density $f_{Z|\lambda}(z)$?

$$\underbrace{F_{Z|\lambda}(z)}_{\text{cdf}} \sim \mathcal{U}(\underbrace{0, 1}_{\text{fixed}})$$

# A general reparameterisation

What transformation will always absorb the parameters of a density $f_{Z|\lambda}(z)$?

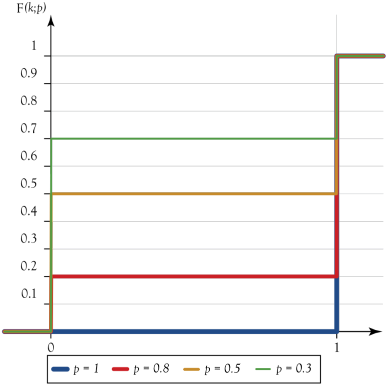
$$\underbrace{F_{Z|\lambda}(z)}_{\text{cdf}} \sim \mathcal{U}(\underbrace{0, 1}_{\text{fixed}})$$

So, if I know the inverse cdf,

$$\begin{aligned} \epsilon &\sim \mathcal{U}(0, 1) \\ F^{-1}_{Z|\lambda}(\epsilon) &\sim Z \end{aligned}$$

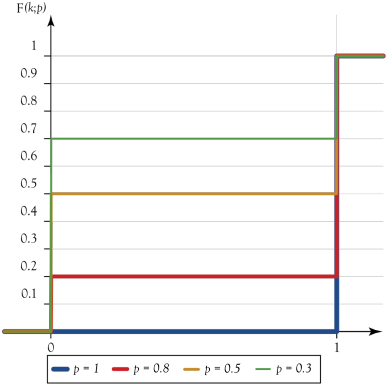
do I have access to $\hat{g}_{\text{rep}}$?

# Let's reparameterise a Bernoulli




$$Z \sim \text{Bernoulli}(p)$$

# Let's reparameterise a Bernoulli

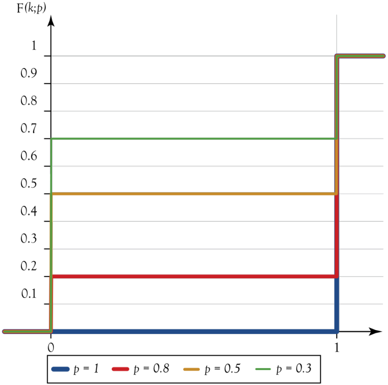



$$Z \sim \text{Bernoulli}(p)$$

$$F^{-1}_{Z|p}(\epsilon) = \begin{cases} 1 & \text{if } \epsilon < p \\ 0 & \text{otherwise} \end{cases}$$

$$= \mathbb{1}_{(0,p)}(\epsilon)$$

# Let’s reparameterise a Bernoulli



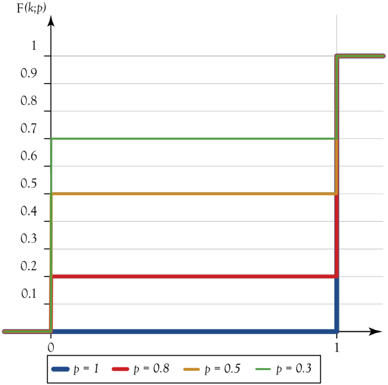

$$Z \sim \text{Bernoulli}(p)$$

$$F^{-1}_{Z|p}(\epsilon) = \begin{cases} 1 & \text{if } \epsilon < p \\ 0 & \text{otherwise} \end{cases}$$

$$= \mathbb{1}_{(0,p)}(\epsilon)$$

How about $\frac{\partial}{\partial p} F^{-1}_{Z|p}(\epsilon)$?

# Let's reparameterise a Bernoulli




$$Z \sim \text{Bernoulli}(p)$$

$$F^{-1}_{Z|p}(\epsilon) = \begin{cases} 1 & \text{if } \epsilon < p \\ 0 & \text{otherwise} \end{cases}$$

$$= \mathbb{1}_{(0,p)}(\epsilon)$$

How about $\frac{\partial}{\partial p} F^{-1}_{Z|p}(\epsilon)$? Mostly 0, sometimes undefined!

# Discrete case

Discrete variables do not admit a differentiable reparameterisation. The cells of the Jacobian are either 0 or undefined :/

# Discrete case

Discrete variables do not admit a differentiable reparameterisation. The cells of the Jacobian are either 0 or undefined :/

The score function estimator is fully general, but very noisy.

# Discrete case

Discrete variables do not admit a differentiable reparameterisation. The cells of the Jacobian are either 0 or undefined :/

The score function estimator is fully general, but very noisy.

Ask the deep learning literature for help ;D

# Discrete case

Discrete variables do not admit a differentiable reparameterisation. The cells of the Jacobian are either 0 or undefined :/

The score function estimator is fully general, but very noisy.

Ask the deep learning literature for help ;D
Fake a Jacobian!

# Straight-Through Estimator (STE)

In lack of a Jacobian, use the identity

$$J_t(\epsilon, \lambda) = \mathrm{diag}(\mathbf{1})$$

STE is a biased gradient estimator that works in some cases, but unfortunately there are no general recipes.

# Bernoulli-STE

Consider a VAE where $q(z|x) = \mathrm{Bern}(z|g(x;\lambda))$.

# Bernoulli-STE

Consider a VAE where $q(z|x) = \text{Bern}(z|g(x;\lambda))$.
We sample $z$ via a reparameterisation that absorbs $\lambda$:

$$\epsilon \sim \mathcal{U}(0,1) \qquad p = g(x;\lambda) \qquad z = \underbrace{\mathbb{1}_{(0,p)}(\epsilon)}_{t(\epsilon,\lambda)}$$

# Bernoulli-STE

Consider a VAE where $q(z|x) = \text{Bern}(z|g(x;\lambda))$.
We sample $z$ via a reparameterisation that absorbs $\lambda$:

$$\epsilon \sim \mathcal{U}(0,1) \qquad p = g(x;\lambda) \qquad z = \underbrace{\mathbb{1}_{(0,p)}(\epsilon)}_{t(\epsilon,\lambda)}$$

A gradient estimate of the ELBO involves computing:

$$\frac{\partial}{\partial\lambda}\log p(x|z = t(\epsilon,\lambda)) = \frac{\partial}{\partial z}\log p(x|z)\frac{\partial}{\partial\lambda}t(\epsilon,\lambda)$$

# Bernoulli-STE

Consider a VAE where $q(z|x) = \mathrm{Bern}(z|g(x;\lambda))$.
We sample $z$ via a reparameterisation that absorbs $\lambda$:

$$\epsilon \sim \mathcal{U}(0,1) \qquad p = g(x;\lambda) \qquad z = \underbrace{\mathbb{1}_{(0,p)}(\epsilon)}_{t(\epsilon,\lambda)}$$

A gradient estimate of the ELBO involves computing:

$$\frac{\partial}{\partial \lambda} \log p(x|z = t(\epsilon,\lambda)) = \frac{\partial}{\partial z} \log p(x|z) \frac{\partial}{\partial \lambda} t(\epsilon,\lambda)$$

and we use our *pseudo gradient*

$$\frac{\partial}{\partial \lambda} t(\epsilon,\lambda) = \frac{\partial}{\partial \lambda} g(x;\lambda) \underbrace{\frac{\partial}{\partial p}\mathbb{1}_{(0,p)}(\epsilon)}_{\stackrel{\text{def}}{=} 1}$$

# Concrete (Gumbel-Softmax) Distribution

We can sample from a Categorical distribution via

$$\epsilon_k \sim \mathrm{Gumbel}(0, 1)$$

$$\underbrace{\arg\max_k \ \{\lambda_k + \epsilon_k\}_{k=1}^K}_{z=t(\epsilon,\lambda)} \sim \mathrm{Cat}(\mathrm{softmax}(\lambda))$$

# Concrete (Gumbel-Softmax) Distribution

We can sample from a Categorical distribution via

$$\epsilon_k \sim \mathrm{Gumbel}(0, 1)$$

$$\underbrace{\arg\max_k \ \{\lambda_k + \epsilon_k\}_{k=1}^K}_{z = t(\epsilon, \lambda)} \sim \mathrm{Cat}(\mathrm{softmax}(\lambda))$$

The problem is that $t(\epsilon, \lambda)$ is not differentiable, but note

$$\mathrm{onehot}(z) \approx \mathrm{softmax}\left(\frac{\lambda + \epsilon}{\tau}\right) \qquad \text{as } \tau \to 0$$

and now the transformation is differentiable, but the outcome is dense. For sparsity, use (biased) STE.

# Summary

- The inverse cdf is a general reparameterisation procedure
- In the discrete case, its inverse is piecewise constant
- Relaxations of Categorical variables are based on the idea of relaxing the one-hot representation of the outcome
- Dense relaxations are mapped to sparse (one-hot) representations via a discontinuity which is ignored in backpropagation (STE).

# Literature I

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continous relaxation of discrete random variables. In *ICLR*, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

# Literature II

Jason Tyler Rolfe. Discrete variational autoencoders. In *ICLR*, 2017.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through $l\_0$ regularization. In *ICLR*, 2018.

Joost Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *ACL*, July 2019.