Reparameterizing Challenging Distributions

Francisco J. R. Ruiz

Joint work with Christian Naesseth, Scott Linderman, Michalis Titsias, David Blei

November 22nd, 2016

Overview

- Reparameterization allows for low-variance gradient estimators
- But it is available for some distributions only
- We show how to extend reparameterization to other distributions
- Allows 1-sample Monte Carlo estimation of gradient
- Our goal: General variational inference for probabilistic models

Gradient of Expectations

- Consider the gradient

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}, \boldsymbol{\lambda}) \right]$$

  w.r.t. some parameters $\boldsymbol{\lambda}$

Gradient of Expectations

- Consider the gradient

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}, \boldsymbol{\lambda}) \right]$$

  w.r.t. some parameters $\boldsymbol{\lambda}$
- This is common in statistics/machine learning:

$$\boldsymbol{\lambda}^\star = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}, \boldsymbol{\lambda}) \right]$$

  - Optimization of loss functions
  - Reinforcement learning (policy gradient)
  - Variational inference
  - ...

Variational Inference (1/3)

$$\boldsymbol{\lambda}^\star = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}, \boldsymbol{\lambda}) \right]$$

- $\mathbf{z}$: Latent variables in a probabilistic model
- $\mathbf{x}$: Data
- $p(\mathbf{x}, \mathbf{z})$: Probabilistic model
- $q(\mathbf{z}; \boldsymbol{\lambda})$: Variational distribution

$$f(\mathbf{z}, \boldsymbol{\lambda}) = \log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\lambda})$$

Variational Inference (2/3)

$$\boldsymbol{\lambda}^\star = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\lambda})\right]$$

- Variational inference approximates the posterior
- Minimizes the KL divergence between $q(\mathbf{z}; \boldsymbol{\lambda})$ and the posterior

$$\boldsymbol{\lambda}^\star = \arg\min_{\boldsymbol{\lambda}} D_{\mathrm{KL}}(q(\mathbf{z}; \boldsymbol{\lambda}) || p(\mathbf{z} \,|\, \mathbf{x}))$$

Variational Inference (3/3)

- Optimization problem:

$$\boldsymbol{\lambda}^\star = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\lambda})\right]$$

  - Conditionally conjugate models: *coordinate ascent*
  - Non-conjugate models: score function method, reparameterization

Variational Inference (3/3)

- Optimization problem:

$$\boldsymbol{\lambda}^\star = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\lambda})\right]$$

  - Conditionally conjugate models: *coordinate ascent*
  - Non-conjugate models: score function method, reparameterization
- Score function and reparameterization are two ways to estimate

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}, \boldsymbol{\lambda})\right]$$

- For simplicity, we focus on

$$f(\mathbf{z}) = \log p(\mathbf{x}, \mathbf{z})$$

  (assume that the gradient of the entropy is tractable)

Score Function Method[1]

- Monte Carlo estimator of the gradient

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \right]$$

- Score function method:

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \right] = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z};\boldsymbol{\lambda}) \right]$$

- Stochastic optimization

[1] Paisley et al. (2012), Ranganath et al. (2014), Mnih and Gregor (2014)

Score Function Method[1]

- Monte Carlo estimator of the gradient

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \right]$$

- Score function method:

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \right] = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z};\boldsymbol{\lambda}) \right]$$

- Stochastic optimization
- Algorithm:
  1. Sample $\mathbf{z}^{(s)} \overset{\text{iid}}{\sim} q(\mathbf{z};\boldsymbol{\lambda})$
  2. Evaluate $f(\mathbf{z}^{(s)})$ and $\nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z}^{(s)};\boldsymbol{\lambda})$ for each sample s
  3. Obtain a Monte Carlo estimate of the gradient
  4. Take a gradient step for $\boldsymbol{\lambda}$

[1] Paisley et al. (2012), Ranganath et al. (2014), Mnih and Gregor (2014)

Reparameterization Trick[2] (1/2)

- Define an invertible transformation

$$\mathbf{z} = \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda}), \qquad \boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda})$$

  such that $\pi(\boldsymbol{\epsilon})$ does not depend on $\boldsymbol{\lambda}$.

[2] Salimans and Knowles (2013), Kingma and Welling (2014), Rezende et al. (2014), Titsias & Lázaro-Gredilla (2014)

Reparameterization Trick[2] (1/2)

- Define an invertible transformation

$$\mathbf{z} = \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda}), \qquad \boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda})$$

  such that $\pi(\boldsymbol{\epsilon})$ does not depend on $\boldsymbol{\lambda}$.

- Push the gradient inside the expectation

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} [f(\mathbf{z})] = \mathbb{E}_{\pi(\boldsymbol{\epsilon})} \left[\nabla_{\mathbf{z}} f(\mathbf{z}) \big|_{\mathbf{z}=\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})} \nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda}) \right]$$

[2] Salimans and Knowles (2013), Kingma and Welling (2014), Rezende et al. (2014), Titsias & Lázaro-Gredilla (2014)

Reparameterization Trick[2] (1/2)

- Define an invertible transformation

$$\mathbf{z} = \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda}), \qquad \boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda})$$

  such that $\pi(\boldsymbol{\epsilon})$ does not depend on $\boldsymbol{\lambda}$.

- Push the gradient inside the expectation

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} [f(\mathbf{z})] = \mathbb{E}_{\pi(\boldsymbol{\epsilon})} \left[\nabla_{\mathbf{z}} f(\mathbf{z}) \big|_{\mathbf{z}=\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})} \nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda}) \right]$$

- Algorithm:
  1. Sample $\boldsymbol{\epsilon}^{(s)}$ iid from $\pi(\boldsymbol{\epsilon})$
  2. Obtain $\mathbf{z}^{(s)} = \mathcal{T}(\boldsymbol{\epsilon}^{(s)}; \boldsymbol{\lambda})$
  3. Evaluate $\nabla_{\mathbf{z}} f(\mathbf{z})$ and $\nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda})$ for each sample s
  4. Obtain a Monte Carlo estimate of the gradient
  5. Take a gradient step for $\boldsymbol{\lambda}$

[2] Salimans and Knowles (2013), Kingma and Welling (2014), Rezende et al. (2014), Titsias & Lázaro-Gredilla (2014)

Reparameterization Trick (2/2)

- Simple example: $q(\mathbf{z}; \boldsymbol{\lambda})$ is a Gaussian

$$\boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda}) = \boldsymbol{\Sigma}^{-1/2}(\mathbf{z} - \boldsymbol{\mu})$$

The transformed density is $\pi(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \mathbf{1})$ (independent of $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$)

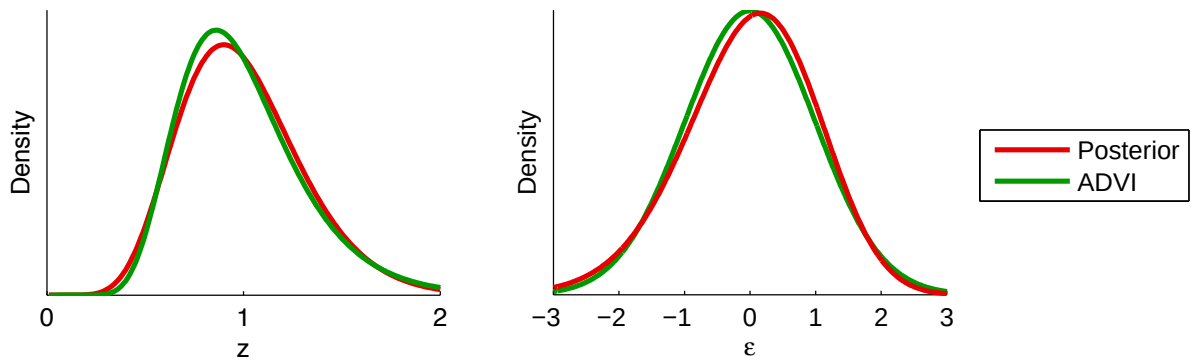
[3] Kucukelbir et al. (2015, 2016)

Reparameterization Trick (2/2)

- Simple example: $q(\mathbf{z}; \boldsymbol{\lambda})$ is a Gaussian

$$\boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda}) = \boldsymbol{\Sigma}^{-1/2}(\mathbf{z} - \boldsymbol{\mu})$$

  The transformed density is $\pi(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \mathbf{1})$ (independent of $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$)
- Used in ADVI[3]
  - Non-linear transformation
  - Gaussian distribution on the transformed space



[3] Kucukelbir et al. (2015, 2016)

Comparison

- Score function
  - \+ Any probabilistic model
  - \+ Any variational distribution (as long as we can sample)
  - − High variance: Requires additional tricks and many samples
- Reparameterization
  - − Differentiable probabilistic models (continuous $\mathbf{z}$)[4]
  - − Limited to some distributions (location-scale, inverse CDF)
  - \+ Low variance: Only 1 sample suffices in practice

[4] But see Maddison et al. (2016), Jang et al. (2016), Kusner & Hernández-Lobato (2016)

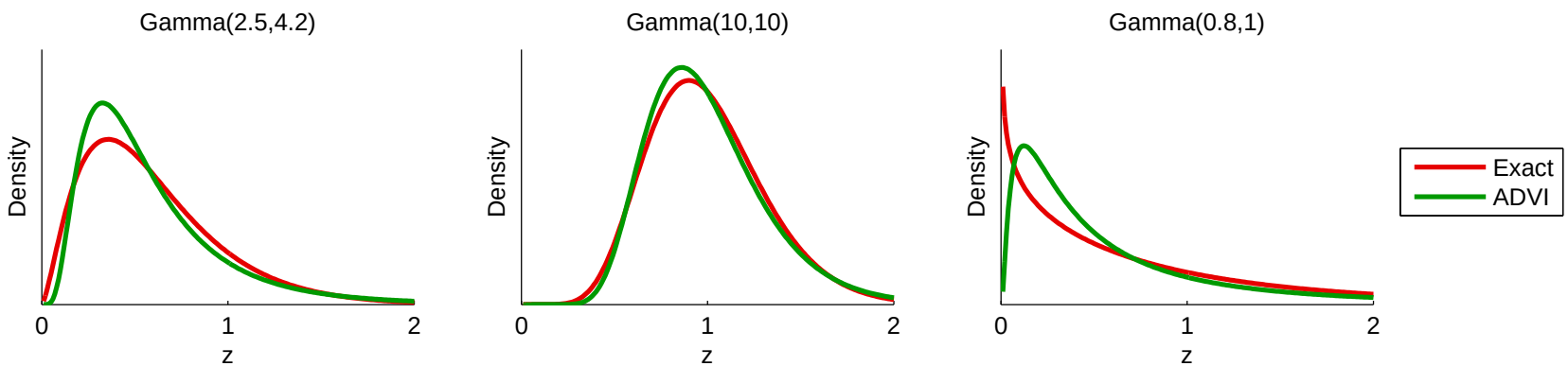
Comparison

- Score function
  - \+ Any probabilistic model
  - \+ Any variational distribution (as long as we can sample)
  - – High variance: Requires additional tricks and many samples
- Reparameterization
  - – Differentiable probabilistic models (continuous $\mathbf{z}$)[4]
  - – Limited to some distributions (location-scale, inverse CDF)
  - \+ Low variance: Only 1 sample suffices in practice
- Our contribution
  - Extend reparameterization to other distributions

[4] But see Maddison et al. (2016), Jang et al. (2016), Kusner & Hernández-Lobato (2016)

Motivation

- We may need more expressive variational families $q(\mathbf{z}; \boldsymbol{\lambda})$



ADVI struggles to approximate distributions with sparsity

Contributions

The Generalized Reparameterization Gradient

Francisco J. R. Ruiz
University of Cambridge
Columbia University

Michalis K. Titsias
Athens University of
Economics and Business

David M. Blei
Columbia University

Rejection Sampling Variational Inference

Christian A. Naesseth[⋆†‡] **Francisco J. R. Ruiz**[‡§] **Scott W. Linderman**[‡] **David M. Blei**[‡]
[†]Linköping University [‡]Columbia University [§]University of Cambridge

The Generalized Reparameterization Gradient (1/3)

- Define an invertible transformation

$$\mathbf{z} = \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda}), \qquad \boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda})$$

  but allow $\pi(\boldsymbol{\epsilon}; \boldsymbol{\lambda})$ to depend **weakly** on $\boldsymbol{\lambda}$.

- Gradient:

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})} \left[f(\mathbf{z}) \right] = \mathbf{g}^{\text{rep}} + \mathbf{g}^{\text{corr}}$$

The Generalized Reparameterization Gradient (2/3)

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})}\left[f(\mathbf{z})\right] = \mathbf{g}^{\text{rep}} + \mathbf{g}^{\text{corr}}$$

- $\mathbf{g}^{\text{rep}}$: Reparameterization gradient

$$\mathbf{g}^{\text{rep}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[\nabla_{\mathbf{z}} f(\mathbf{z})\big|_{\mathbf{z}=\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})} \nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$

- $\mathbf{g}^{\text{corr}}$: Correction term

$$\mathbf{g}^{\text{corr}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[f(\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda}))\nabla_{\boldsymbol{\lambda}} \log \pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$

The Generalized Reparameterization Gradient (3/3)

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})}\left[f(\mathbf{z})\right] = \mathbf{g}^{\text{rep}} + \mathbf{g}^{\text{corr}}$$
$$\mathbf{g}^{\text{rep}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[\nabla_{\mathbf{z}} f(\mathbf{z})\big|_{\mathbf{z}=\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})} \nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$
$$\mathbf{g}^{\text{corr}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[f(\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda}))\nabla_{\boldsymbol{\lambda}} \log \pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$

- Under a transformation such that $\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})$ does not depend on $\boldsymbol{\lambda}$:

$$\mathbf{g}^{\text{rep}} = \text{reparameterization gradient}$$
$$\mathbf{g}^{\text{corr}} = \mathbf{0}$$

- Under identity transformation:

$$\mathbf{g}^{\text{rep}} = \mathbf{0}$$
$$\mathbf{g}^{\text{corr}} = \text{score function gradient}$$

The Generalized Reparameterization Gradient (3/3)

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})}\left[f(\mathbf{z})\right] = \mathbf{g}^{\text{rep}} + \mathbf{g}^{\text{corr}}$$
$$\mathbf{g}^{\text{rep}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[\nabla_{\mathbf{z}} f(\mathbf{z})\big|_{\mathbf{z}=\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})} \nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$
$$\mathbf{g}^{\text{corr}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[f(\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda}))\nabla_{\boldsymbol{\lambda}} \log \pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$

- Under a transformation such that $\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})$ does not depend on $\boldsymbol{\lambda}$:

$$\mathbf{g}^{\text{rep}} = \text{reparameterization gradient}$$
$$\mathbf{g}^{\text{corr}} = \mathbf{0}$$

- Under identity transformation:

$$\mathbf{g}^{\text{rep}} = \mathbf{0}$$
$$\mathbf{g}^{\text{corr}} = \text{score function gradient}$$

Goal: Find a transformation that makes $\mathbf{g}^{\text{corr}}$ small

Example: Gamma distribution

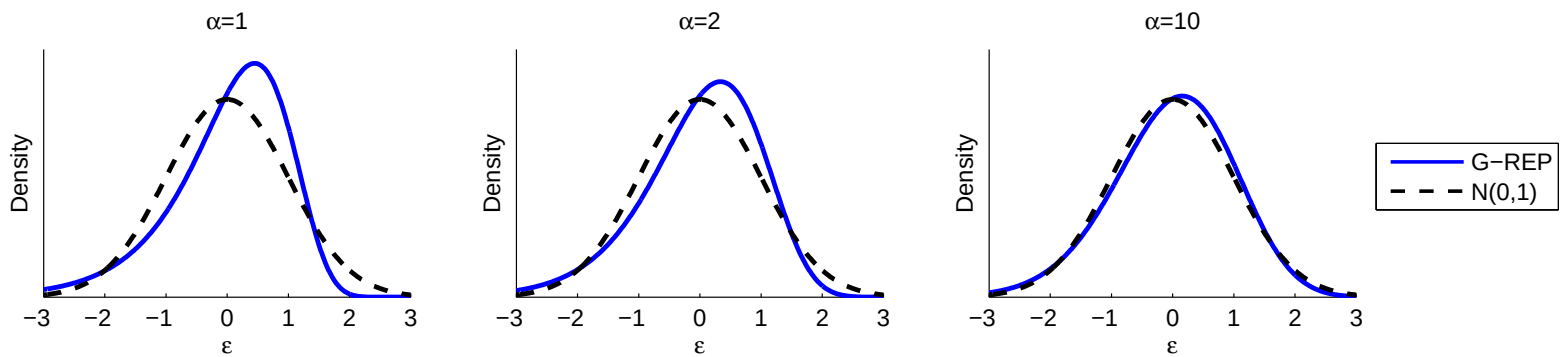
- We use transformations based on *standardization*
- Example: $q(z; \alpha, \beta) = \mathrm{Gamma}(z; \alpha, \beta)$

$$\epsilon = \mathcal{T}^{-1}(z; \alpha, \beta) = \frac{\log(z) - (\psi(\alpha) - \log(\beta))}{\sqrt{\psi_1(\alpha)}},$$

Example: Gamma distribution

- We use transformations based on *standardization*
- Example: $q(z; \alpha, \beta) = \mathrm{Gamma}(z; \alpha, \beta)$

$$\epsilon = \mathcal{T}^{-1}(z; \alpha, \beta) = \frac{\log(z) - (\psi(\alpha) - \log(\beta))}{\sqrt{\psi_1(\alpha)}},$$

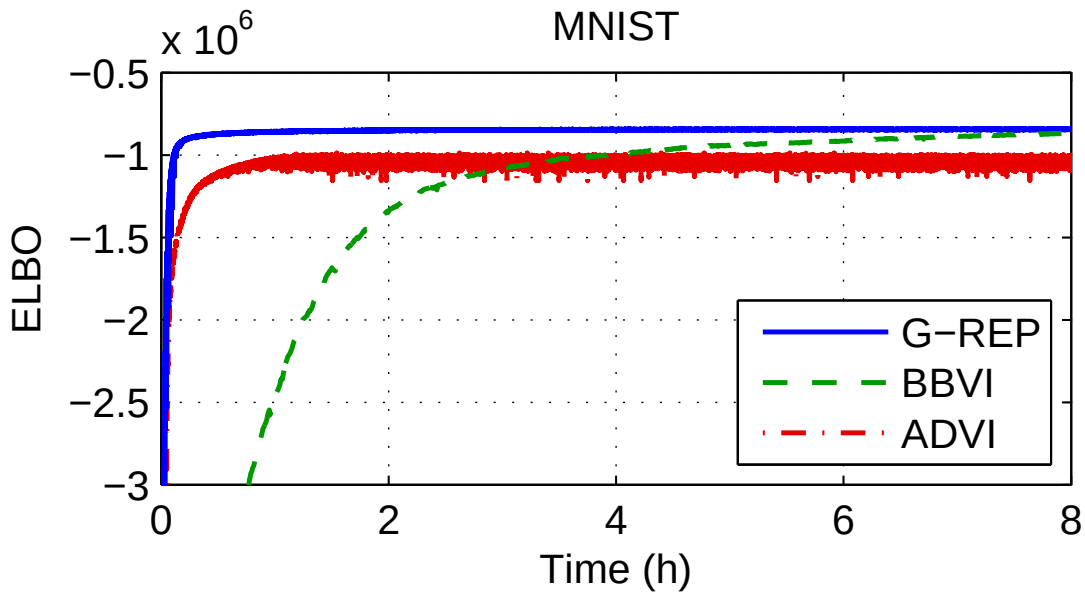


G-REP: Full Algorithm

1. Draw a single sample $\mathbf{z} \sim q(\mathbf{z}; \boldsymbol{\lambda})$
2. Obtain $\boldsymbol{\epsilon} = \mathcal{T}^{-1}(\mathbf{z}; \boldsymbol{\lambda})$
3. Estimate $\mathbf{g}^{\text{rep}}$ and $\mathbf{g}^{\text{corr}}$ (with 1 sample)
4. Take a gradient step for $\boldsymbol{\lambda}$

Results: MNIST

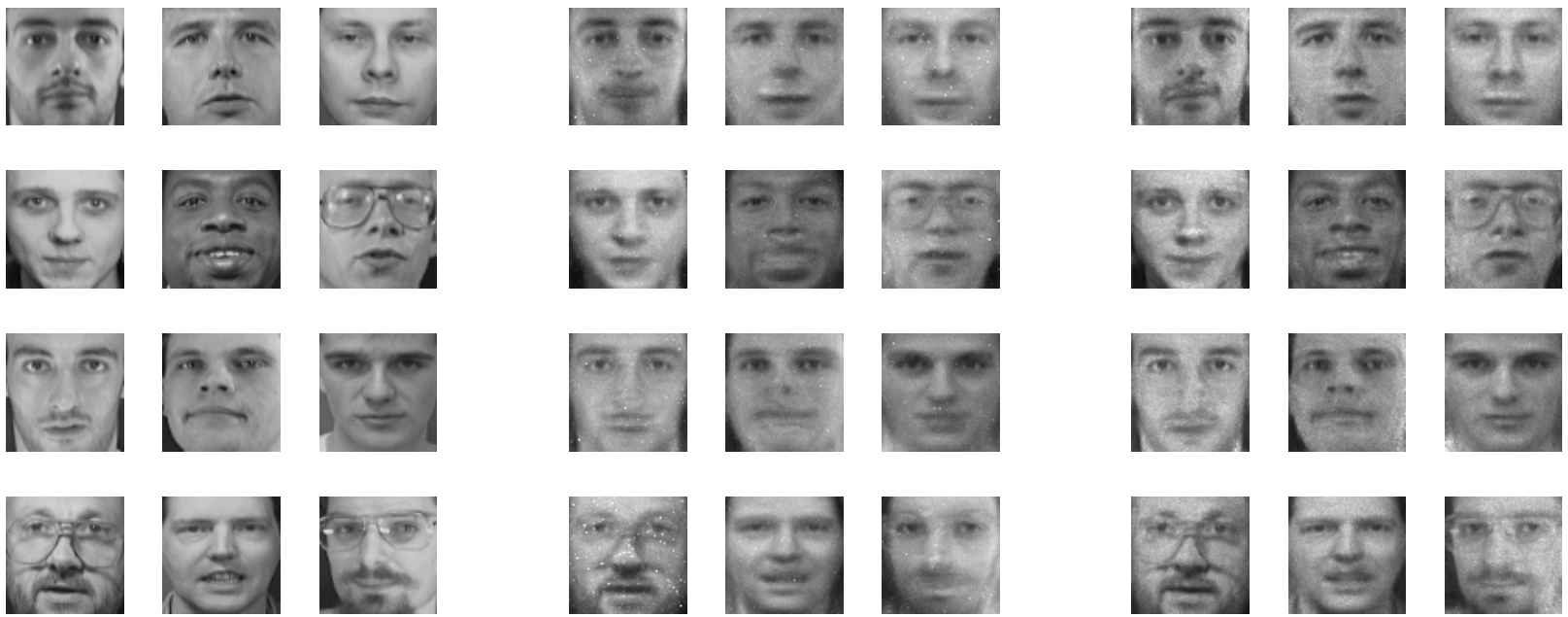
Model: Gamma-beta matrix factorization



Results: Olivetti Dataset

Model: Sparse gamma deep exponential family[5]

Ground truth | Reconstructed (ADVI) | Reconstructed (G-REP)



[5] Ranganath et al. (2015)

Rejection Sampling Variational Inference

Every random variable that we can simulate on our computers is ultimately a transformation of elementary random variables

- In theory, this should allow for reparameterization of any distribution

Rejection Sampling Variational Inference

Every random variable that we can simulate on our computers is ultimately a transformation of elementary random variables

- In theory, this should allow for reparameterization of any distribution
- Challenge: rejection sampling steps
  - We cannot push the gradient inside the integral

Reparameterized Rejection Sampling

- In standard rejection sampling, we have:
  - A target, $q(z; \boldsymbol{\lambda})$
  - A proposal, $r(z; \boldsymbol{\lambda})$
  - A uniform random variable, $u \sim \mathcal{U}(0, 1)$
  - An accept/reject step, $u < \frac{q(z;\boldsymbol{\lambda})}{M_{\boldsymbol{\lambda}} r(z;\boldsymbol{\lambda})}$

Reparameterized Rejection Sampling

- In standard rejection sampling, we have:
  - A target, $q(z; \boldsymbol{\lambda})$
  - A proposal, $r(z; \boldsymbol{\lambda})$
  - A uniform random variable, $u \sim \mathcal{U}(0, 1)$
  - An accept/reject step, $u < \frac{q(z;\boldsymbol{\lambda})}{M_{\boldsymbol{\lambda}} r(z;\boldsymbol{\lambda})}$
- In *reparameterized* rejection sampling, we have:
  - A target, $q(z; \boldsymbol{\lambda})$
  - An elementary proposal, $s(\varepsilon)$
  - A transformation, $z = \mathcal{T}(\varepsilon; \boldsymbol{\lambda})$, such that $z \sim r(z; \boldsymbol{\lambda})$
  - A uniform random variable, $u \sim \mathcal{U}(0, 1)$
  - An accept/reject step, $u < \frac{q(\mathcal{T}(\varepsilon;\boldsymbol{\lambda});\boldsymbol{\lambda})}{M_{\boldsymbol{\lambda}} r(\mathcal{T}(\varepsilon;\boldsymbol{\lambda});\boldsymbol{\lambda})}$

  The accepted sample $z = \mathcal{T}(\varepsilon; \boldsymbol{\lambda}) \sim q(z; \boldsymbol{\lambda})$

Rejection Sampling VI

Main idea:

- Integrate out the accept/reject variables u
- Consider the distribution of the *accepted* sample ε

$$\pi(\varepsilon; \boldsymbol{\lambda}) = s(\varepsilon) \frac{q(\mathcal{T}(\varepsilon; \boldsymbol{\lambda}); \boldsymbol{\lambda})}{r(\mathcal{T}(\varepsilon; \boldsymbol{\lambda}); \boldsymbol{\lambda})}$$

- Use the transformation $z = \mathcal{T}(\varepsilon; \boldsymbol{\lambda})$

Example: Gamma Distribution (1/2)

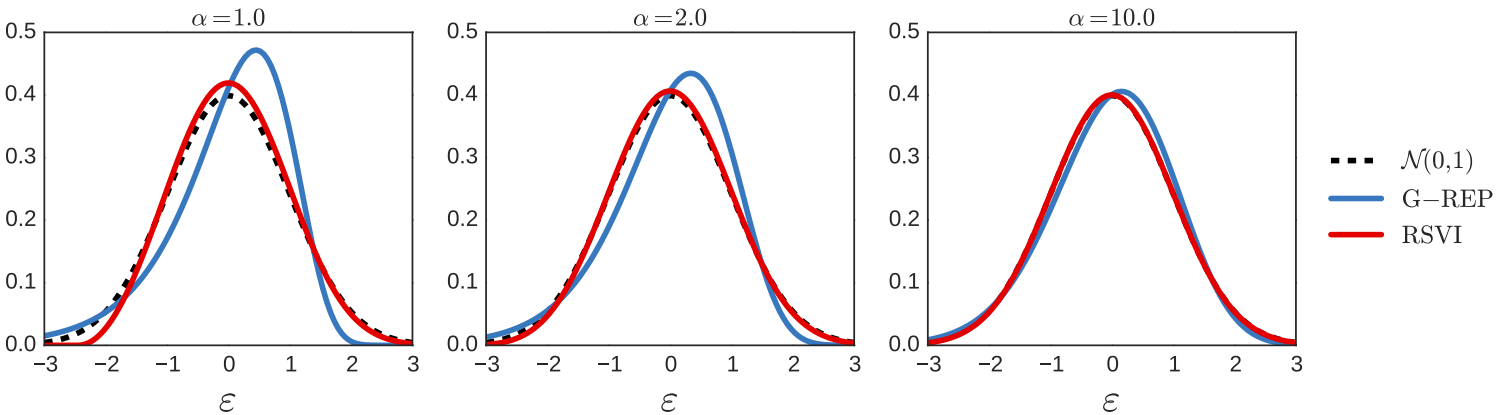
Algorithm[6] to sample from a $\mathrm{Gamma}(\alpha, 1)$ (with $\alpha \geq 1$)

1. Generate $\varepsilon \sim s(\varepsilon) = \mathcal{N}(0, 1)$
2. Transform as $z = \mathcal{T}(\varepsilon; \alpha) = \left(\alpha - \frac{1}{3}\right)\left(1 + \frac{\varepsilon}{\sqrt{9\alpha - 3}}\right)^3$
3. Generate $u \sim \mathcal{U}(0, 1)$
4. Accept or reject ε
   - 4.1 If accept, return z and ε
   - 4.2 If reject, go to Step 1 and repeat

[6] Marsaglia & Tsang (2000)

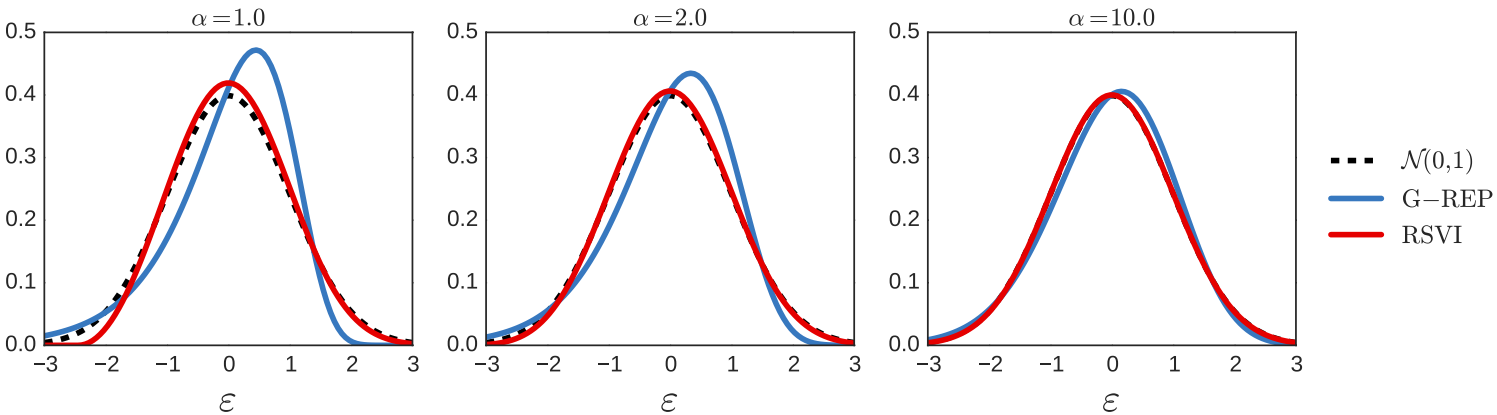
Example: Gamma Distribution (2/2)

The transformation has very high acceptance probability (> 0.95 for $\alpha = 1$)



Example: Gamma Distribution (2/2)

The transformation has very high acceptance probability (> 0.95 for $\alpha = 1$)



We can leverage 60+ years of research in rejection sampling to find good transformations!

Gradient of Rejection Sampling VI

- The form of the gradient is similar to G-REP:

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z};\boldsymbol{\lambda})}\left[f(\mathbf{z})\right] = \mathbf{g}^{\mathrm{rep}} + \mathbf{g}^{\mathrm{corr}}$$

$$\mathbf{g}^{\mathrm{rep}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[\nabla_{\mathbf{z}} f(\mathbf{z})\big|_{\mathbf{z}=\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})} \nabla_{\boldsymbol{\lambda}} \mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$

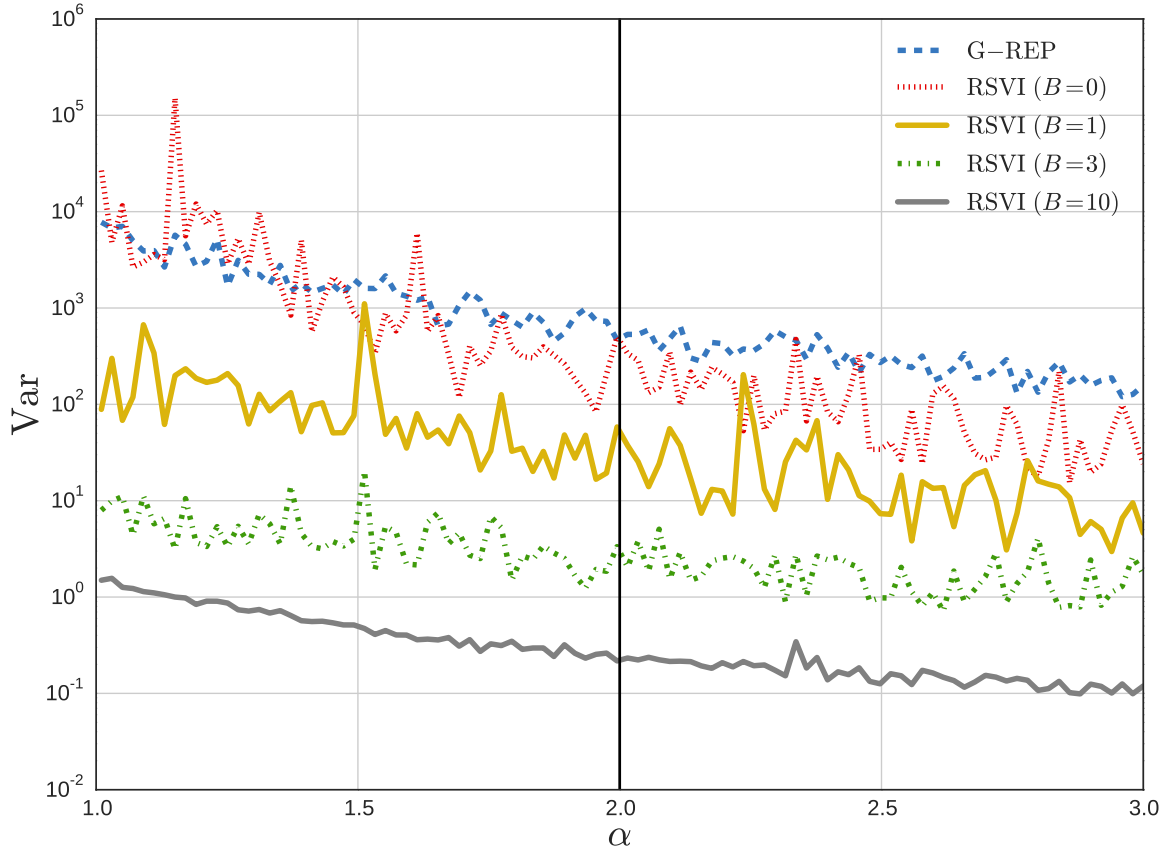
$$\mathbf{g}^{\mathrm{corr}} = \mathbb{E}_{\pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})}\left[f(\mathcal{T}(\boldsymbol{\epsilon};\boldsymbol{\lambda}))\nabla_{\boldsymbol{\lambda}} \log \pi(\boldsymbol{\epsilon};\boldsymbol{\lambda})\right]$$

RSVI: Full Algorithm

1. Run the reparameterized rejection sampling to draw $\boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon}; \boldsymbol{\lambda})$
2. Transform $\mathbf{z} = \mathcal{T}(\boldsymbol{\epsilon}; \boldsymbol{\lambda})$
3. Estimate $\mathbf{g}^{\text{rep}}$ and $\mathbf{g}^{\text{corr}}$ (with 1 sample)
4. Take a gradient step for $\boldsymbol{\lambda}$

Results: Variance of the Gradient

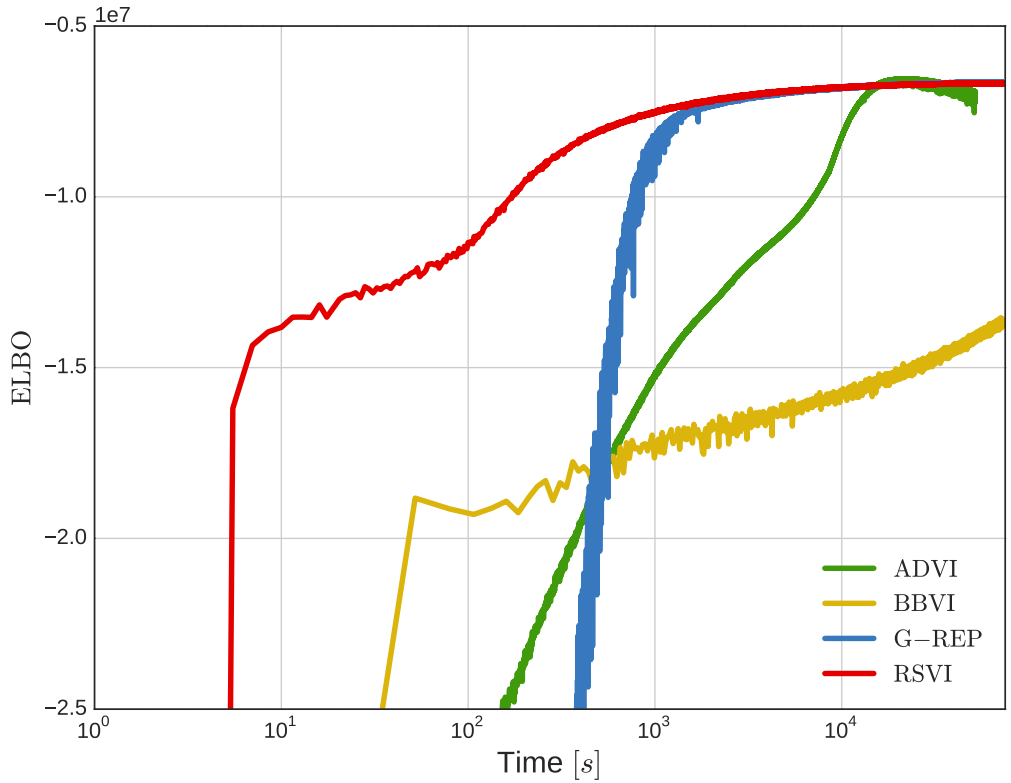
Model: Dirichlet/Multinomial with 100 components



(B denotes "shape augmentation")

Results: Olivetti Dataset

Model: Sparse gamma deep exponential family[7]



[7] Ranganath et al. (2015)

Summary: G-REP and RSVI

- Extend reparameterization trick to non-reparameterizable distributions (gamma, beta, Dirichlet, ...)
- Allow variational inference on continuous non-conjugate models
- Fast: Monte Carlo estimation with only 1 sample

Thank you for your attention!