ARM: Augment-REINFORCE-merge gradient for discrete latent variable models

Mingyuan Zhou[⋆ §]
Joint work with Mingzhang Yin[§]

[⋆]IROM Department, McCombs School of Business
[§]Department of Statistics and Data Sciences
The University of Texas at Austin

Institut de Recherche en Informatique de Toulouse
Toulouse, July 5, 2018

Joint work with

Mingzhang Yin

PhD student (since Fall 2015)
in Statistics and Data Sciences

Discrete latent variables

- Discrete latent variables are widely used in mixture models, mixed membership model, sparse factor model, variable selection, etc.
- A common task is to optimize

$$\mathcal{E}(\phi) = \int f(\boldsymbol{z}) q_\phi(\boldsymbol{z}) d\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z} \sim q_\phi(\boldsymbol{z})}[f(\boldsymbol{z})]$$

- This objective includes
  - Maximizing the marginal likelihood of a hierarchical Bayesian model
  - Maximizing the evidence lower bound (ELBO) in variational inference

Unbiased gradient estimators

Reparameterization

If $\nabla_{\boldsymbol{z}} f(\boldsymbol{z})$ is tractable to compute and $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$ can be generated via reparameterization as $\boldsymbol{z} = \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}),\ \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$, then one may apply the reparameterization trick

$$\nabla_{\boldsymbol{\phi}} \mathcal{E}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})}[f(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] = \mathbb{E}_{\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} f(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))]$$

REINFORCE (score-function estimator)

If $\mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[\nabla_{\boldsymbol{\phi}} f(\boldsymbol{z})] = 0$, using the score function $\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \nabla_{\boldsymbol{\phi}} q_{\boldsymbol{\phi}}(\boldsymbol{z}) / q_{\boldsymbol{\phi}}(\boldsymbol{z})$, one may use REINFORCE as

$$\nabla_{\boldsymbol{\phi}} \mathcal{E}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[f(\boldsymbol{z}) \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})] \approx \frac{1}{K} \sum_{k=1}^{K} f(\boldsymbol{z}^{(k)}) \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z}^{(k)})$$

Problems

However, neither estimator is problem free:

- The reparameterization trick requires $f(\boldsymbol{z})$ to be differentiable and cannot be applied to discrete $\boldsymbol{z}$
- REINFORCE suffers from high Monte Carlo estimation variance

Gradient estimation for discrete latent variables

For discrete latent variable $\boldsymbol{z}$, to compute the gradient of $\mathcal{E}(\phi) = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}[f(\boldsymbol{z})]$, existing solutions include

- Biased but low-variance gradient estimator via a continuous relaxation of discrete random variables
  - Gumbel-softmax trick (Maddison et al., 2017; Jang et al., 2017)
- Variance reduction by adding control variates (a.k.a. baselines)

$$\nabla_\phi \mathcal{E}(\phi) = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}[(f(\boldsymbol{z}) - c(\boldsymbol{z}))\nabla_\phi \log q_\phi(\boldsymbol{z})] + \mu_c$$

where $\mu_c = \nabla_\phi \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}[c(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}[c(\boldsymbol{z})\nabla_\phi \log q_\phi(\boldsymbol{z})]$ is known

- REBAR (Tucker et al., 2017)
- RELAX (Grathwohl et al., 2018)

Gradient estimation for discrete latent variables

Our ideas:

- Spike gradient:
  - Don't move unless you are pretty sure which direction to move
  - If you do move, move with large spikes
  - The temporal average of the spikes shall code the temporal evolution of the true gradient
- Variance reduction by sharing common random numbers between different expectations
- No need to construct baselines (control variates), as the function f itself will be used to construct a baseline
- Augmentation + REINFORCE + merge is how we derive such a gradient estimator

Exponential, Gumbel, and categorical random variables

- If $x_i \sim \text{Exp}(\lambda_i)$ are independent exponential random variables for $i = 1, \ldots, M$, then

$$P\left(i = \arg\min_j x_j\right) = P\left(x_i < x_j,\ \forall\, j \neq i\right) = \lambda_i \Big/ \sum_{i=1}^{M} \lambda_i \quad .$$

- Note $x \sim \text{Exp}(\lambda)$ can be reparameterized as

$$x = \epsilon/\lambda,\ \ \epsilon \sim \text{Exp}(1),$$

where $\epsilon \sim \text{Exp}(1)$ can be equivalently generated as

$$\epsilon = -\log u,\ \ u \sim \text{Uniform}(0, 1)$$

Exponential, Gumbel, and categorical random variables

- If

$$x_i \sim \text{Exp}(\lambda_i),$$

then we have

$$\begin{aligned}\arg\min_i x_i &\overset{d}{=} \arg\min_i\{-\log u_i/\lambda_i\}, \qquad u_i \overset{iid}{\sim} \text{Uniform}(0,1)\\ &= \arg\max_i\{\log\lambda_i - \log(-\log u_i)\}\\ &\overset{d}{=} \arg\max_i\{\log\lambda_i + \epsilon_i\}, \quad \epsilon_i \sim \text{Gumbel}(0,1)\end{aligned}$$

- Note if $u \sim \text{Uniform}(0,1)$, then $\epsilon = -\log(-\log u)$ follows the Gumbel$(0,1)$ distribution (Type-I extreme-value distribution)

Augmentation of categorical random variable

Denoting $\sigma(\phi) = \left(\frac{e^{\phi_1}}{\sum_{m=1}^{M} e^{\phi_m}}, \ldots, \frac{e^{\phi_M}}{\sum_{m=1}^{M} e^{\phi_m}} \right)$, categorical $z \sim \text{Discrete}(\sigma(\phi))$ can be augmented as

$$z = \arg\min_{i \in \{1,\ldots,M\}} e_i \ , \text{where } e_i \sim \text{Exp}(e^{\phi_i})$$

The objective can be rewritten with respect to M augmented exponential random variables as

$$\begin{aligned} \mathcal{E}(\phi) &= \mathbb{E}_{z \sim \text{Discrete}(\sigma(\phi))}[f(z)] \\ &= \mathbb{E}_{e_1 \sim \text{Exp}(e^{\phi_1}), \ldots, e_M \sim \text{Exp}(e^{\phi_M})}[f(\arg\min_i e_i)] \end{aligned}$$

REINFORCE in augmented space

Use REINFORCE we have

$$\begin{aligned}\nabla_{\phi_m}\mathcal{E}(\phi) &= \mathbb{E}_{e_1\sim\mathsf{Exp}(e^{\phi_1}),\ldots,e_M\sim\mathsf{Exp}(e^{\phi_M})}[f(\arg\min_i e_i)\nabla_{\phi_m}\log\mathsf{Exp}(e_m;e^{\phi_m})]\\ &= \mathbb{E}_{e_1\sim\mathsf{Exp}(e^{\phi_1}),\ldots,e_M\sim\mathsf{Exp}(e^{\phi_M})}[f(\arg\min_i e_i)(1-e_m e^{\phi_m})]\end{aligned}$$

Since the exponential random variable $x\sim\mathsf{Exp}(e^{\phi})$ can be reparameterized as $x=\epsilon e^{-\phi},\ \epsilon\sim\mathsf{Exp}(1)$, we have

$$\nabla_{\phi_m}\mathcal{E}(\phi) = \mathbb{E}_{\epsilon_1,\ldots,\epsilon_M\overset{iid}{\sim}\mathsf{Exp}(1)}[f(\arg\min_i \epsilon_i e^{-\phi_i})(1-\epsilon_m)]$$

REINFORCE in augmented space

A key observation is that we may choose M as reference category and rewrite the gradient estimator

$$\nabla_{\phi_m}\mathcal{E}(\phi) = \mathbb{E}_{\epsilon_1,\ldots,\epsilon_M \overset{iid}{\sim} \text{Exp}(1)}[f(\arg\min_i \epsilon_i e^{-\phi_i})(1-\epsilon_m)]$$

as

$$\nabla_{\phi_m}\mathcal{E}(\phi) = \mathbb{E}_{\epsilon_1,\ldots,\epsilon_M \overset{iid}{\sim} \text{Exp}(1)}[f(\arg\min_i \epsilon_{(m\leftrightharpoons M)_i} e^{-\phi_i})(1-\epsilon_M)]$$

where $(m \leftrightharpoons M)$ denotes a vector of indices constructed by swapping the m-th and M-th elements of vector $(1,\ldots,M)$, which means

$$\begin{cases} (m \leftrightharpoons M)_M = m \\ (m \leftrightharpoons M)_m = M \\ (m \leftrightharpoons M)_i = i, \qquad \text{if } i \notin \{m, M\} \end{cases}$$

Merge the gradients

As $\sigma(\phi - \phi_M \mathbf{1}_M) = \sigma(\phi)$, one may update $\tilde{\phi}_m = \phi_m - \phi_M$ for $m \leq M-1$ and set $\tilde{\phi}_M = 0$.
Denoting $\tilde{\phi} = (\tilde{\phi}_1, \ldots, \tilde{\phi}_{M-1})' = \mathbf{A}\phi$, where $\mathbf{A} = [\text{diag}(\mathbf{1}_{M-1}), -\mathbf{1}_{M-1}]$, we have

$$\begin{aligned}
\nabla_{\phi}\mathcal{E}(\phi)' =& \nabla_{\tilde{\phi}}\mathcal{E}([\tilde{\phi}', 0]')' \frac{\partial \tilde{\phi}}{\partial \phi} = \nabla_{\tilde{\phi}}\mathcal{E}([\tilde{\phi}', 0]')'\mathbf{A} \\
\nabla_{\tilde{\phi}_m}\mathcal{E}([\tilde{\phi}', 0]') =& \frac{1}{M}\sum_{j=1}^{M}(\nabla_{\phi_m}\mathcal{E}(\phi) - \nabla_{\phi_j}\mathcal{E}(\phi)) \\
=& \nabla_{\phi_m}\mathcal{E}(\phi) - \frac{1}{M}\sum_{j=1}^{M}\nabla_{\phi_j}\mathcal{E}(\phi).
\end{aligned}$$

Notice previously we have

$$\nabla_{\phi_m}\mathcal{E}(\phi) = \mathbb{E}_{\epsilon_1,\ldots,\epsilon_M \overset{iid}{\sim} \text{Exp}(1)}[f(\arg\min_i \epsilon_{(m \rightleftharpoons M)_i} e^{-\phi_i})(1-\epsilon_M)].$$

Merge the gradients

- Combining the two equations the ARM estimator is

$$\nabla_{\tilde{\phi}_m} \mathcal{E}([\tilde{\boldsymbol{\phi}}', 0]') = \mathbb{E}_{\epsilon_1,\ldots,\epsilon_M \overset{iid}{\sim} \text{Exp}(1)}[f_\Delta(\boldsymbol{\epsilon}, \boldsymbol{\phi}, m)(1 - \epsilon_M)],$$

- Common random numbers are shared to compute:

$$\begin{aligned} f_\Delta(\boldsymbol{\epsilon}, \boldsymbol{\phi}, m) &= f(\arg\min_i \epsilon_{(m \leftrightharpoons M)_i} e^{-\phi_i}) - \frac{1}{M} \sum_{j=1}^{M} f(\arg\min_i \epsilon_{(j \leftrightharpoons M)_i} e^{-\phi_i}) \\ &= \frac{1}{M} \sum_{j \neq m} \left[f(\arg\min_i \epsilon_{(m \leftrightharpoons M)_i} e^{-\phi_i}) - f(\arg\min_i \epsilon_{(j \leftrightharpoons M)_i} e^{-\phi_i}) \right] \end{aligned}$$

- Note that

$$\mathbb{E}_{\epsilon_1,\ldots,\epsilon_M \overset{iid}{\sim} \text{Exp}(1)}[f_\Delta(\boldsymbol{\epsilon}, \boldsymbol{\phi}, m)] = 0,$$

  thus there is no need to add control variates

Merge the gradients

- $\epsilon_1, \ldots, \epsilon_M \overset{iid}{\sim} \text{Exp}(1)$ is the same in distribution as

  $$\epsilon_i = \pi_i \epsilon, \quad \text{for } i = 1, \ldots, M, \text{ where } \boldsymbol{\pi} \sim \text{Dirichlet}\,(\mathbf{1}_M),\ \epsilon \sim \text{Gamma}(M, 1),$$

- $\arg\min_i \pi_{(m \leftrightharpoons M)_i} e^{-\phi_i} = \arg\min_i \epsilon \pi_{(m \leftrightharpoons M)_i} e^{-\phi_i}$
- The gradient can be re-expressed as

  $$\nabla_{\tilde{\phi}_m} \mathcal{E}(\boldsymbol{\phi}) = \nabla_{\tilde{\phi}_m} \mathcal{E}([\tilde{\boldsymbol{\phi}}', 0]') = \mathbb{E}_{\boldsymbol{\pi} \sim \text{Dirichlet}(\mathbf{1}_M)}[f_\Delta(\boldsymbol{\pi}, \boldsymbol{\phi}, m)(1 - M\pi_M)]$$

  where

  $$\begin{aligned} & f_\Delta(\boldsymbol{\pi}, \boldsymbol{\phi}, m) \\ &= f\left(\underset{i \in \{1,\ldots,M\}}{\arg\min}\ \pi_{(m \leftrightharpoons M)_i} e^{-\phi_i}\right) - \frac{1}{M} \sum_{j=1}^{M} f\left(\underset{i \in \{1,\ldots,M\}}{\arg\min}\ \pi_{(j \leftrightharpoons M)_i} e^{-\phi_i}\right) \\ &= \frac{1}{M} \sum_{j \neq m} \left[f\left(\underset{i \in \{1,\ldots,M\}}{\arg\min}\ \pi_{(m \leftrightharpoons M)_i} e^{-\phi_i}\right) - f\left(\underset{i \in \{1,\ldots,M\}}{\arg\min}\ \pi_{(j \leftrightharpoons M)_i} e^{-\phi_i}\right) \right] \end{aligned}$$

ARM gradient for binary random variable

For a binary random variable, the gradient of $\mathbb{E}_{z\sim\text{Bernoulli}(\sigma(\phi))}[f(z)]$ with respect to ϕ can be expressed as

$$\nabla_\phi \mathcal{E}(\phi) = \mathbb{E}_{u\sim\text{Uniform}(0,1)}[f_\Delta(u,\phi)(u-1/2)]$$
$$f_\Delta(u,\phi) = f\big(\mathbf{1}[u > \sigma(-\phi)]\big) - f\big(\mathbf{1}[u < \sigma(\phi)]\big).$$

Proposition

(i) $f_\Delta(u,\phi) = 0$ *with probability* $\sigma(|\phi|) - \sigma(-|\phi|)$, $f_\Delta(u,\phi) = f(1) - f(0)$ *with probability* $1 - \sigma(|\phi|)$, *and* $f_\Delta(u,\phi) = f(0) - f(1)$ *with probability* $\sigma(-|\phi|)$.

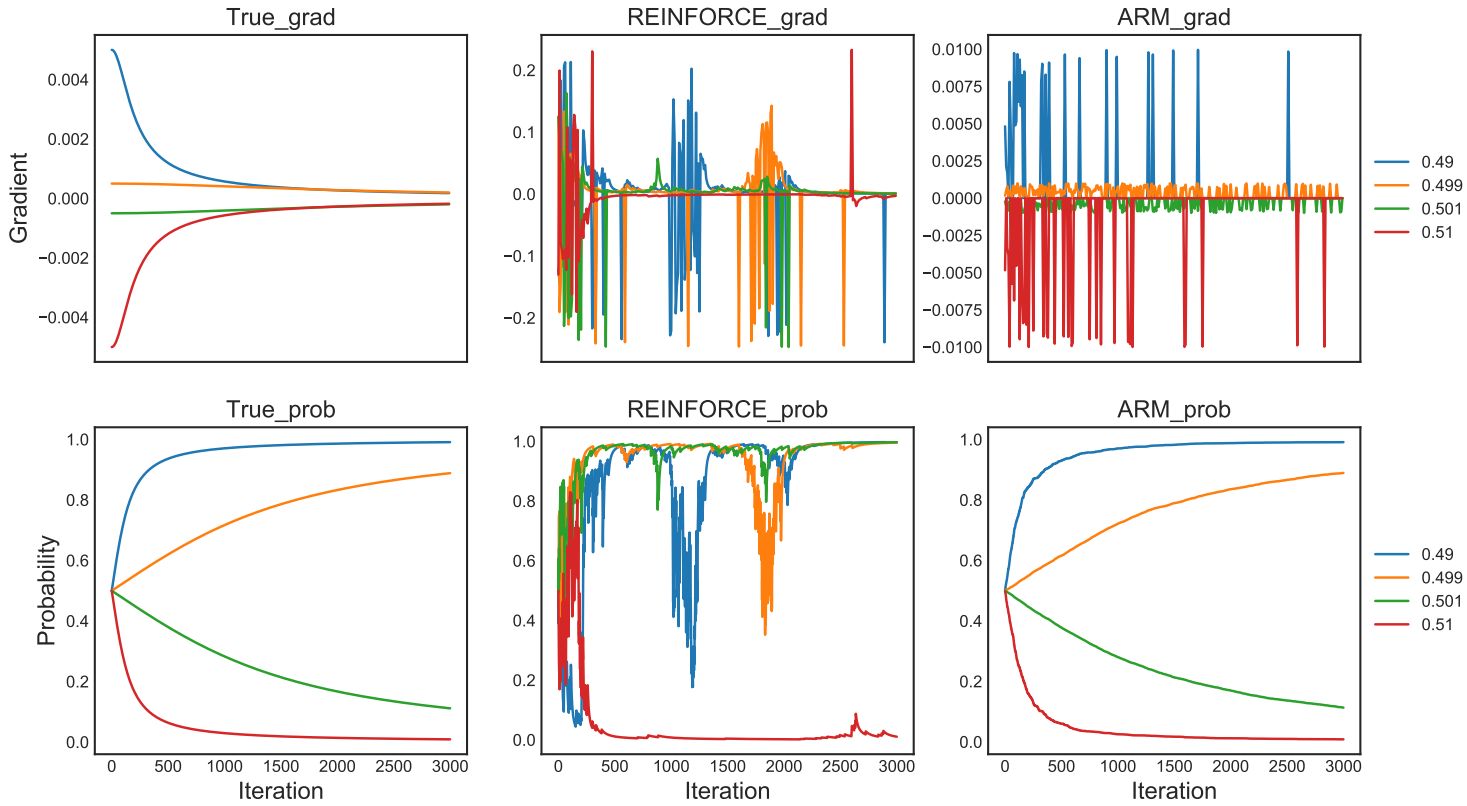
(ii) $g_{ARM}(u,\phi) = f_\Delta(u,\phi)(u-1/2)$ *is unbiased with* $\mathbb{E}_{u\sim Uniform(0,1)}[g_{ARM}(u,\phi)] = \mathbb{E}_{z\sim Bernoulli(\sigma(\phi))}[g_{REINFORCE}(z,\phi)]$.

(iii) $g_{ARM}(u,\phi)$ *reaches its largest variance at* $0.039788[f(1)-f(0)]^2$ *when* $\frac{P(f_\Delta=0)}{P(f_\Delta\neq 0)}$ *is equal to the golden ratio* $\frac{\sqrt{5}+1}{2}$.

(iv) $\frac{sup_\phi Var[g_{ARM}]}{sup_\phi Var[g_{REINFORCE}]} \leq \frac{16}{25}(1 - 2\frac{f(0)}{f(0)+f(1)})^2$ *and* $\sup_\phi Var[g_{ARM}] \leq \frac{1}{25}[f(1)-f(0)]^2$.

A simple example

Learning ϕ to maximize $\mathcal{E}(\phi) = \mathbb{E}_{z\sim\text{Bernoulli}(\sigma(\phi))}[(z - p_0)^2]$, where $p_0 \in \{0.49, 0.499, 0.501, 0.51\}$



A simple example

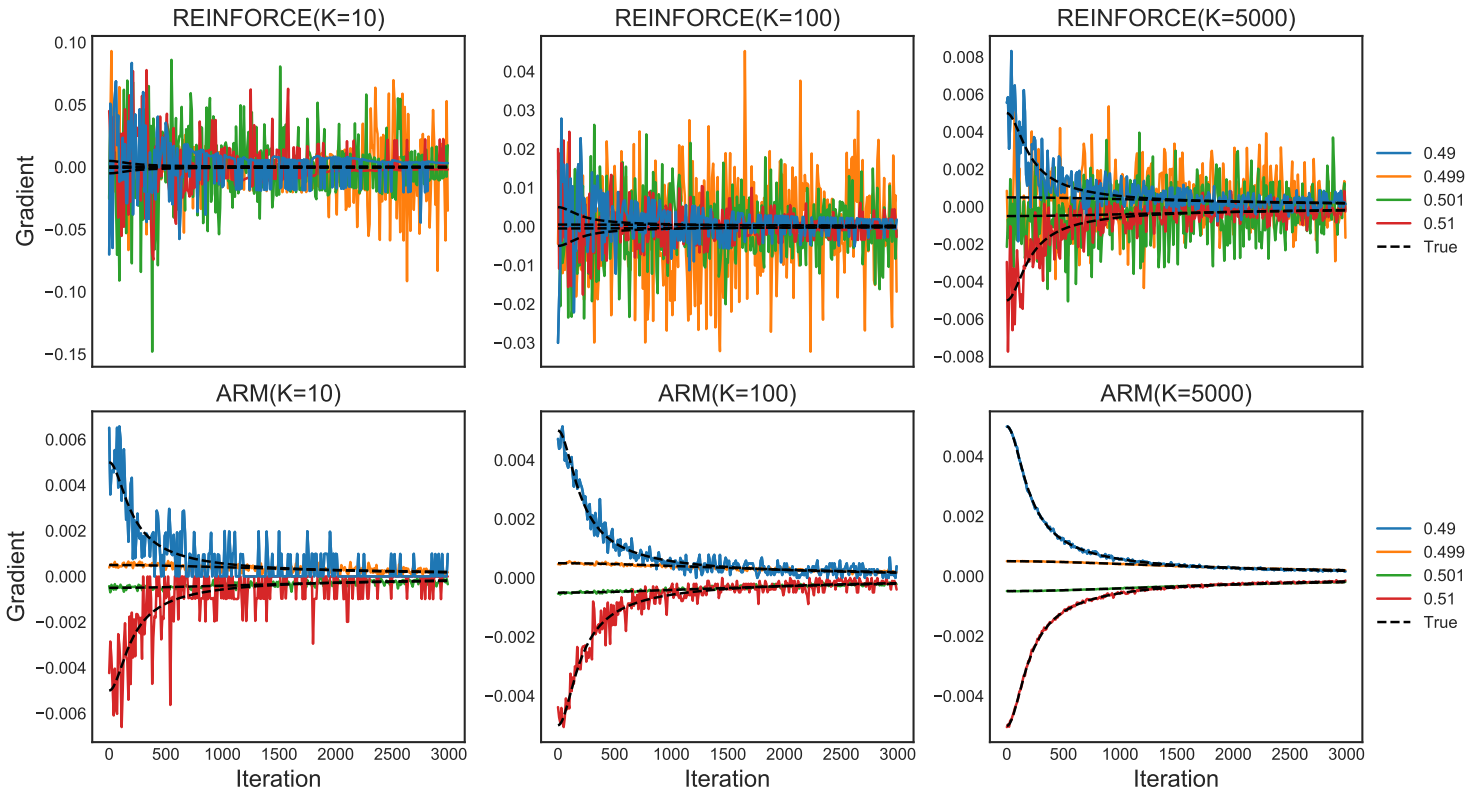


Figure: Estimation of the true gradient at each iteration using $K > 1$ Monte Carlo samples, using REINFORCE, shown in the top row, or ARM, shown in the bottom row. The ARM estimator exhibits significant lower variance given the same number of Monte Carlo samples.

Multiple discrete stochastic layers

- A latent variable model with multiple stochastic hidden layers is

$$\boldsymbol{x} \sim p_{\theta_0}(\boldsymbol{x} \mid \boldsymbol{b}_1),\ \boldsymbol{b}_1 \sim p_{\theta_1}(\boldsymbol{b}_1 \mid \boldsymbol{b}_2), \ldots, \boldsymbol{b}_T \sim p_{\theta_T}(\boldsymbol{b}_T),$$

- The joint likelihood is

$$p(\boldsymbol{x}, \boldsymbol{b}_{1:T} \mid \boldsymbol{\theta}_{0:T}) = p_{\theta_0}(\boldsymbol{x} \mid \boldsymbol{b}_1)\Big[\prod_{t=1}^{T-1} p_{\theta_t}(\boldsymbol{b}_t \mid \boldsymbol{b}_{t+1})\Big] p_{\theta_T}(\boldsymbol{b}_T).$$

VAE with multiple discrete stochastic layers

- The encoder is designed as

$$q_{\boldsymbol{w}_{1:T}}(\boldsymbol{b}_{1:T} \mid \boldsymbol{x}) = q_{\boldsymbol{w}_1}(\boldsymbol{b}_1 \mid \boldsymbol{x}) \Big[\prod_{t=1}^{T-1} q_{\boldsymbol{w}_{t+1}}(\boldsymbol{b}_{t+1} \mid \boldsymbol{b}_t) \Big]$$

- $q_{\boldsymbol{w}_t}(\boldsymbol{b}_t \mid \boldsymbol{b}_{t-1}) = \text{Bernoulli}(\boldsymbol{b}_t; \sigma(\mathcal{T}_{\boldsymbol{w}_t}(\boldsymbol{b}_{t-1})))$
- The ELBO can be expressed as

$$\mathcal{E}(\boldsymbol{w}_{1:T}) = \mathbb{E}_{\boldsymbol{b}_{1:T} \sim q_{\boldsymbol{w}_{1:T}}(\boldsymbol{b}_{1:T} \mid \boldsymbol{x})} \left[f(\boldsymbol{b}_{1:T}) \right], \text{ where}$$
$$f(\boldsymbol{b}_{1:T}) = \log p_{\boldsymbol{\theta}_0}(\boldsymbol{x} \mid \boldsymbol{b}_1) + \log p_{\boldsymbol{\theta}_{1:T}}(\boldsymbol{b}_{1:T}) - \log q_{\boldsymbol{w}_{1:T}}(\boldsymbol{b}_{1:T} \mid \boldsymbol{x}).$$

ARM gradient of the ELBO

First, to compute the gradient with respect to $\boldsymbol{w}_1$, since

$$\mathcal{E}(\boldsymbol{w}_{1:T}) = \mathbb{E}_{q(\boldsymbol{b}_1)}\mathbb{E}_{q(\boldsymbol{b}_{2:T} \mid \boldsymbol{b}_1)}[f(\boldsymbol{b}_{1:T})]$$

we have

$$\nabla_{\boldsymbol{w}_1}\mathcal{E}(\boldsymbol{w}_{1:T}) = \mathbb{E}_{\boldsymbol{u}_1 \sim \text{Uniform}(0,1)}[f_\Delta(\boldsymbol{u}_1, \mathcal{T}_{\boldsymbol{w}_1}(\boldsymbol{x}))(\boldsymbol{u}_1 - 1/2)]\nabla_{\boldsymbol{w}_1}\mathcal{T}_{\boldsymbol{w}_1}(\boldsymbol{x}),$$

where

$$\begin{aligned} f_\Delta(\boldsymbol{u}_1, \mathcal{T}_{\boldsymbol{w}_1}(\boldsymbol{x})) = &\mathbb{E}_{\boldsymbol{b}_{2:T} \sim q(\boldsymbol{b}_{2:T} \mid \boldsymbol{b}_1),\ \boldsymbol{b}_1 = \mathbf{1}[\boldsymbol{u}_1 > \sigma(-\mathcal{T}_{\boldsymbol{w}_1}(\boldsymbol{x}))]}[f(\boldsymbol{b}_{1:T})] \\ &- \mathbb{E}_{\boldsymbol{b}_{2:T} \sim q(\boldsymbol{b}_{2:T} \mid \boldsymbol{b}_1),\ \boldsymbol{b}_1 = \mathbf{1}[\boldsymbol{u}_1 < \sigma(\mathcal{T}_{\boldsymbol{w}_1}(\boldsymbol{x}))]}[f(\boldsymbol{b}_{1:T})] \end{aligned}$$

ARM gradient of the ELBO

Second, to compute the gradient with respect to $\boldsymbol{w}_t$, where $2 \leq t \leq T-1$, since

$$\mathcal{E}(\boldsymbol{w}_{1:T}) = \mathbb{E}_{q(\boldsymbol{b}_{1:t-1})}\mathbb{E}_{q(\boldsymbol{b}_t \mid \boldsymbol{b}_{t-1})}\mathbb{E}_{q(\boldsymbol{b}_{t+1:T} \mid \boldsymbol{b}_t)}[f(\boldsymbol{b}_{1:T})]$$

we have

$$\begin{aligned}&\nabla_{\boldsymbol{w}_t}\mathcal{E}(\boldsymbol{w}_{1:T})\\ &= \mathbb{E}_{q(\boldsymbol{b}_{1:t-1})}\left[\mathbb{E}_{\boldsymbol{u}_t \sim \text{Uniform}(0,1)}[f_\Delta(\boldsymbol{u}_t, \mathcal{T}_{\boldsymbol{w}_t}(\boldsymbol{b}_{t-1}), \boldsymbol{b}_{1:t-1})(\boldsymbol{u}_t - 1/2)]\nabla_{\boldsymbol{w}_t}\mathcal{T}_{\boldsymbol{w}_t}(\boldsymbol{b}_{t-1})\right],\end{aligned}$$

where

$$\begin{aligned}f_\Delta(\boldsymbol{u}_t, \mathcal{T}_{\boldsymbol{w}_t}(\boldsymbol{b}_{t-1}), \boldsymbol{b}_{1:t-1}) = &\mathbb{E}_{\boldsymbol{b}_{t+1:T} \sim q(\boldsymbol{b}_{t+1:T} \mid \boldsymbol{b}_t),\ \boldsymbol{b}_t = \mathbf{1}[\boldsymbol{u}_t > \sigma(-\mathcal{T}_{\boldsymbol{w}_t}(\boldsymbol{b}_{t-1}))])}[f(\boldsymbol{b}_{1:T})]\\ &- \mathbb{E}_{\boldsymbol{b}_{t+1:T} \sim q(\boldsymbol{b}_{t+1:T} \mid \boldsymbol{b}_t),\ \boldsymbol{b}_t = \mathbf{1}[\boldsymbol{u}_t < \sigma(\mathcal{T}_{\boldsymbol{w}_t}(\boldsymbol{b}_{t-1}))])}[f(\boldsymbol{b}_{1:T})]\end{aligned}$$

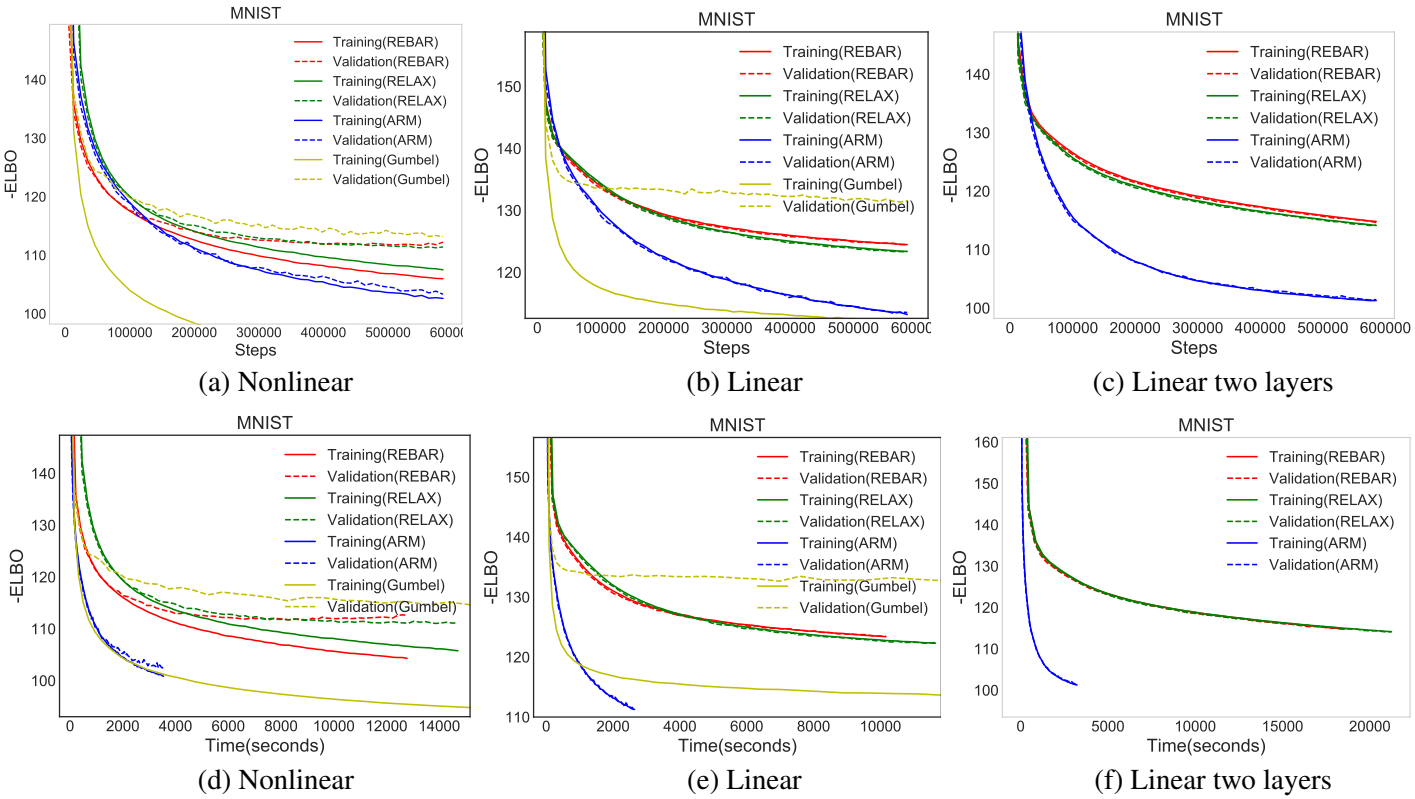
ARM gradient of the ELBO

Finally, to compute the gradient with respect to $\boldsymbol{w}_T$, we have

$$\begin{aligned}
&\nabla_{\boldsymbol{w}_T} \mathcal{E}(\boldsymbol{w}_{1:T}) \\
&= \mathbb{E}_{q(\boldsymbol{b}_{1:T-1})} \left[\mathbb{E}_{\boldsymbol{u}_T \sim \text{Uniform}(0,1)} [f_\Delta(\boldsymbol{u}_T, \mathcal{T}_{\boldsymbol{w}_T}(\boldsymbol{b}_{T-1}), \boldsymbol{b}_{1:T-1})(\boldsymbol{u}_T - 1/2)] \nabla_{\boldsymbol{w}_T} \mathcal{T}_{\boldsymbol{w}_T}(\boldsymbol{b}_{T-1}) \right]
\end{aligned}$$

$$\begin{aligned}
f_\Delta(\boldsymbol{u}_T, \mathcal{T}_{\boldsymbol{w}_T}(\boldsymbol{b}_{T-1}), \boldsymbol{b}_{1:T-1}) = f(\boldsymbol{b}_{1:T-1}, \boldsymbol{b}_T = \mathbf{1}[\boldsymbol{u}_T > \sigma(-\mathcal{T}_{\boldsymbol{w}_T}(\boldsymbol{b}_{T-1}))])) \\
- f(\boldsymbol{b}_{1:T-1}, \boldsymbol{b}_T = \mathbf{1}[\boldsymbol{u}_T < \sigma(\mathcal{T}_{\boldsymbol{w}_T}(\boldsymbol{b}_{T-1}))])
\end{aligned}$$

ARM for VAE



(a) Nonlinear (b) Linear (c) Linear two layers

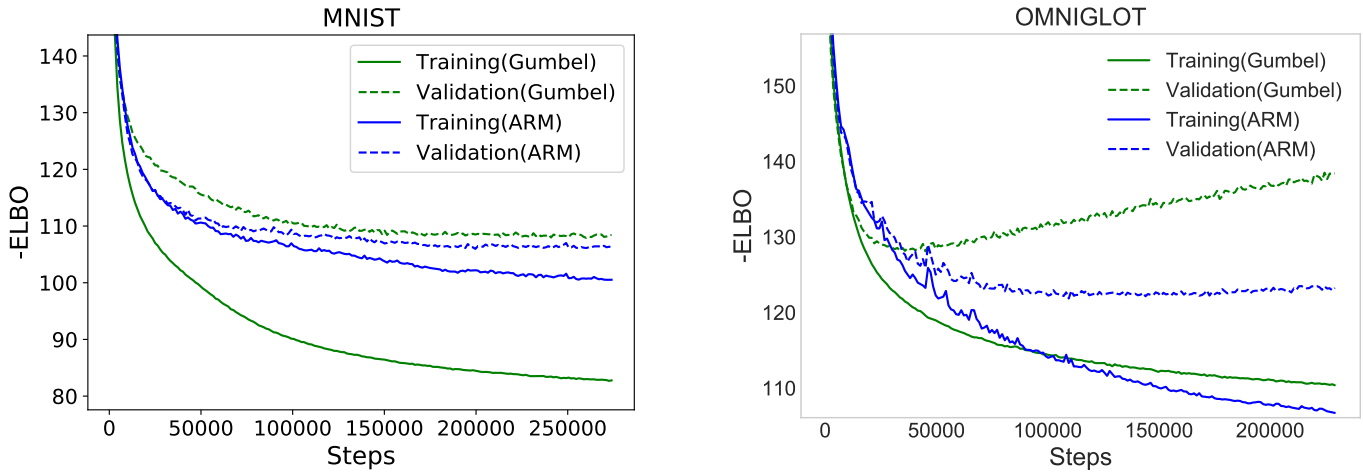
(d) Nonlinear (e) Linear (f) Linear two layers

Figure 2: Test negative ELBOs on MNIST with respect to training iterations, shown in the top row, and wall clock times on Tesla-K40 GPU, shown in the bottom row, for three differently structured Bernoulli VAEs.

ARM for VAE

Table 2: Test negative ELBOs of discrete VAEs trained with four different stochastic gradient estimators.

			ARM	RELAX	REBAR	ST Gumbel-Softmax
Bernoulli	Nonlinear	MNIST	**101.3**	110.9	111.6	112.5
		OMNIGLOT	129.5	**128.2**	128.3	140.7
	Linear	MNIST	**110.3**	122.1	123.2	129.2
		OMNIGLOT	**124.2**	124.4	124.9	129.8
	Two layers	MNIST	**98.2**	114.0	113.7	NA
		OMNIGLOT	**118.3**	119.1	118.8	NA
Categorical	Nonlinear	MNIST	**105.8**	NA	NA	107.9
		OMNIGLOT	**121.9**	NA	NA	127.6



(a) (b)

Figure: Comparison of the negative ELBOs for categorical variational auto-encoders trained by ARM and ST Gumbel-softmax on MNIST and OMNIGLOT, using the "Nonlinear" network.

Gradient estimator	ARM	ST	1/2	Annealed ST	ST Gumbel-S.	SF	MuProp
$-\log p(\boldsymbol{x}_l \mid \boldsymbol{x}_u)$	**54.8**	56.1	57.2	58.7	59.3	72.0	56.7

MLE with multiple discrete stochastic layers

- The log marginal likelihood can be expressed as

$$\log p_{\theta_{0:T}}(\boldsymbol{x}) = \log \mathbb{E}_{\boldsymbol{b}_{1:T}\sim p_{\theta_{1:T}}(\boldsymbol{b}_{1:T})}[p_{\theta_0}(\boldsymbol{x}\mid \boldsymbol{b}_1)]$$
$$\geq \mathcal{E}(\boldsymbol{\theta}_{1:T}) = \mathbb{E}_{\boldsymbol{b}_{1:T}\sim p_{\theta_{1:T}}(\boldsymbol{b}_{1:T})}[\log p_{\theta_0}(\boldsymbol{x}\mid \boldsymbol{b}_1)].$$

- For stochastic binary network

$$p_{\theta_t}(\boldsymbol{b}_t \mid \boldsymbol{b}_{t+1}) = \text{Bernoulli}(\boldsymbol{b}_t; \sigma(\mathcal{T}_{\theta_t}(\boldsymbol{b}_{t+1}))),$$

- The gradient of the lower bound can be expressed as

$$\nabla_{\theta_t}\mathcal{E}(\boldsymbol{\theta}_{1:T})$$
$$= \mathbb{E}_{p(\boldsymbol{b}_{t+1:T})}\left[\mathbb{E}_{\boldsymbol{u}_t}[f_\Delta(\boldsymbol{u}_t, \mathcal{T}_{\theta_t}(\boldsymbol{b}_{t+1}), \boldsymbol{b}_{t+1:T})(\boldsymbol{u}_t - 1/2)]\nabla_{\theta_t}\mathcal{T}_{\theta_t}(\boldsymbol{b}_{t+1})\right],$$
$$f_\Delta(\boldsymbol{u}_t, \mathcal{T}_{\theta_t}(\boldsymbol{b}_{t+1}), \boldsymbol{b}_{t+1:T}) =$$
$$\mathbb{E}_{\boldsymbol{b}_{1:t-1}\sim p(\boldsymbol{b}_{1:t-1}\mid \boldsymbol{b}_t),\ \boldsymbol{b}_t=\mathbf{1}[\boldsymbol{u}_t>\sigma(-\mathcal{T}_{\theta_t}(\boldsymbol{b}_{t+1}))]}[\log p_{\theta_0}(\boldsymbol{x}\mid \boldsymbol{b}_1)]$$
$$- \mathbb{E}_{\boldsymbol{b}_{1:t-1}\sim p(\boldsymbol{b}_{1:t-1}\mid \boldsymbol{b}_t),\ \boldsymbol{b}_t=\mathbf{1}[\boldsymbol{u}_t<\sigma(\mathcal{T}_{\theta_t}(\boldsymbol{b}_{t+1}))]}[\log p_{\theta_0}(\boldsymbol{x}\mid \boldsymbol{b}_1)].$$

MLE with multiple discrete stochastic layers

- Predict lower half of MNIST digit $\boldsymbol{x}_l$ given the upper half $\boldsymbol{x}_u$;
- Maximizing the conditional likelihood $p_{\boldsymbol{\theta}_{0:2}}(\boldsymbol{x}_l \mid \boldsymbol{x}_u)$
- Approximate $\log p_{\boldsymbol{\theta}_{0:2}}(\boldsymbol{x}_l \mid \boldsymbol{x}_u)$ with

$$\log \frac{1}{K} \sum_{k=1}^{K} \text{Bernoulli}(\boldsymbol{x}_l; \sigma(\mathcal{T}_{\boldsymbol{\theta}_0}(\boldsymbol{b}_1^{(k)})))$$

  where $\boldsymbol{b}_1^{(k)} \sim \text{Bernoulli}(\sigma(\mathcal{T}_{\boldsymbol{\theta}_1}(\boldsymbol{b}_2^{(k)})))$, $\boldsymbol{b}_2^{(k)} \sim \text{Bernoulli}(\sigma(\mathcal{T}_{\boldsymbol{\theta}_2}(\boldsymbol{x}_u)))$.
- Training with K=1 and on the test approximate negative log-likelihood $-\log p_{\boldsymbol{\theta}_{0:2}}(\boldsymbol{x}_l \mid \boldsymbol{x}_u)$ with $K = 1000$

MLE with multiple discrete stochastic layers

Table 3: For the MNIST conditional distribution estimation benchmark task, comparison of the test negative log-likelihood between various gradient estimators, with the best results in [14, 20] reported here.

Gradient estimator	ARM	ST	$1/2$	Annealed ST	ST Gumbel-S.	SF	MuProp
$-\log p(\boldsymbol{x}_l \mid \boldsymbol{x}_u)$	**54.8**	56.1	57.2	58.7	59.3	72.0	56.7

Thank you!