DSC291: Machine Learning with Few Labels

Unsupervised Learning

Zhiting Hu

Lecture 18, May 15, 2024

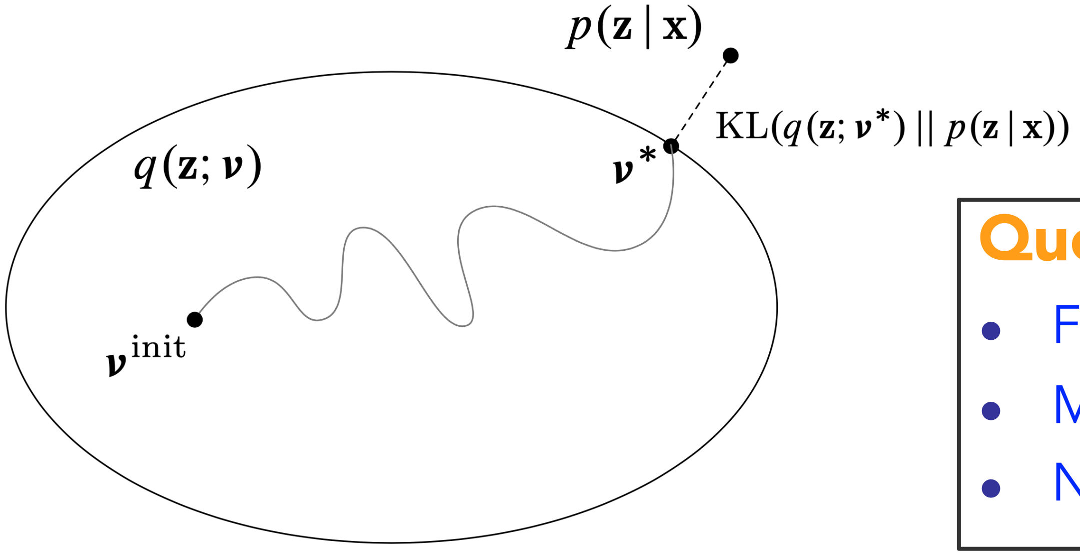
UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Recap: EM and Variational Inference

- The EM algorithm:
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$

$$= p(\boldsymbol{z}|\boldsymbol{x}, \theta^t) = \frac{p(\boldsymbol{z}, \boldsymbol{x}|\theta^t)}{\sum_z p(\boldsymbol{z}, \boldsymbol{x}|\theta^t)}$$

Intractable when model $p(\boldsymbol{z}, \boldsymbol{x}|\theta)$ is complex



Approximate $p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$:

- find a **tractable** $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\nu}^*)$ that is closest to $p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$

$$q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\nu}^*) = \min_{\boldsymbol{\nu}} \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\nu}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)\big)$$

$$= \min_{\boldsymbol{\nu}} F(q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\nu}), \theta^t) + const.$$

Question: What forms of $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v})$ shall we choose?

- Factorized distribution -> mean field VI
- Mixture of Gaussian distribution -> black-box VI
- Neural-based distribution -> Variational Autoencoders

Mean Field Variational Inference with Coordinate Ascent

Recap: Bayesian mixture of Gaussians

Assume mean-field $q(\mu_{1:K}, z_{1:n}) = \prod_k q(\mu_k) \prod_i q(z_i)$

- Initialize the global variational distributions $q(\mu_k)$ and parameters $\{\tau^2, \sigma^2, \pi\}$
- Repeat:
  - For each data example $i \in \{1,2,\dots,D\}$
    - Update the local variational distribution $q(z_i)$
  - End for
  - Update the global variational distributions $q(\mu_k)$
  - Update the parameters $\{\tau^2, \sigma^2, \pi\}$
- Until ELBO converges

(Variational) E-step

(Variational) M-step

- What if we have millions of data examples? This could be very slow.

Stochastic VI

Recap: Bayesian mixture of Gaussians

Assume mean-field $q(\mu_{1:K}, z_{1:n}) = \prod_k q(\mu_k) \prod_i q(z_i)$

- Initialize the global variational distributions $q(\mu_k)$ and parameters $\{\tau^2, \sigma^2, \pi\}$
- Repeat:
  - Sample a data example $i \in \{1,2,\dots,D\}$
  - Update the local variational distribution $q(z_i)$
  - Update the global variational distributions $q(\mu_k)$ with **natural gradient ascent**
  - Update the parameters $\{\tau^2, \sigma^2, \pi\}$
- Until ELBO converges

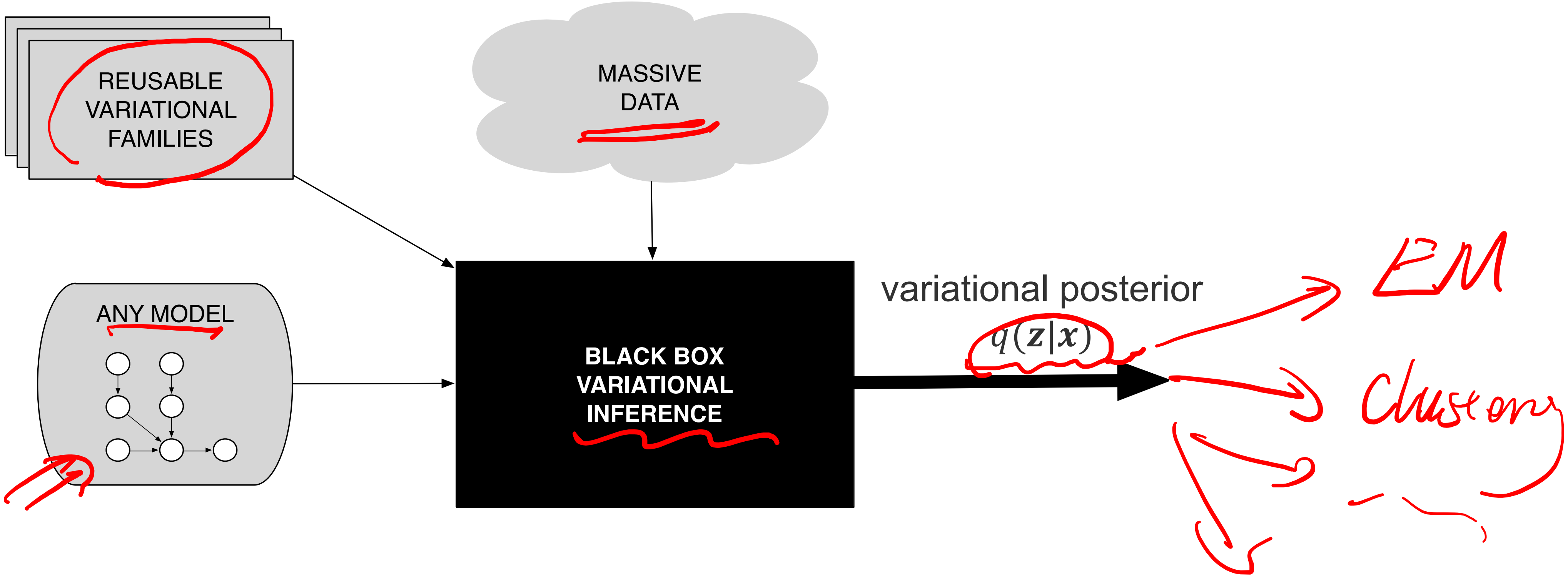
[Hoffman et al., Stochastic Variational Inference, 2013]

Black-box Variational Inference

Black-box Variational Inference (BBVI)

- We have derived variational inference specific for Bayesian Gaussian (mixture) models
- There are innumerable models
- Can we have a solution that does not entail model-specific work?

Black-box Variational Inference (BBVI)



- Easily use variational inference with **any model**
- **No mathematical work** beyond specifying the model
- Perform inference with **massive data** Stochastic VI

(Courtesy: Blei et al., 2018)

$q(z|x,\lambda)$

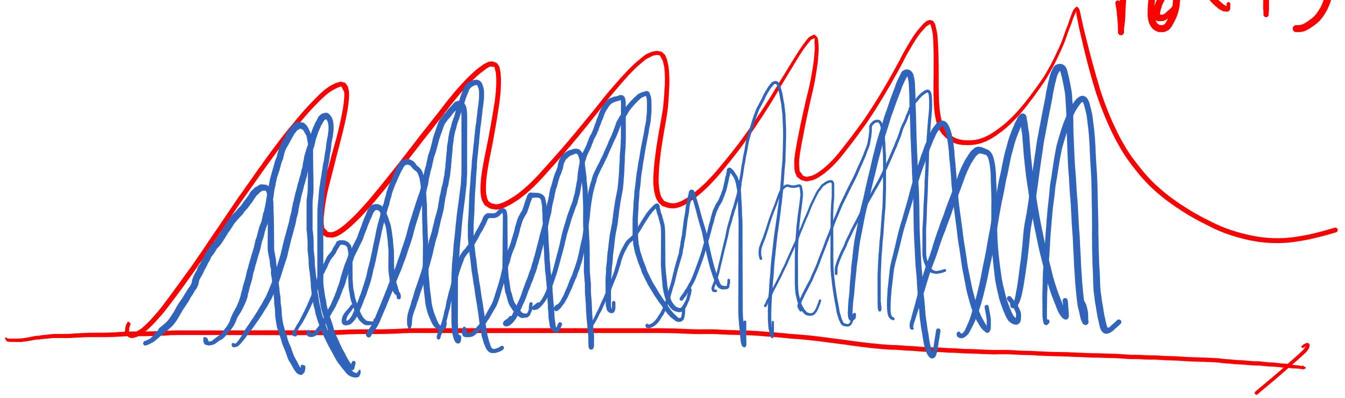
$p_\theta(z|x)$

Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters λ, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - **Question:** what other "universal approximator" can we use?

$KL(q||p) \leq \varepsilon$

$p_\theta(z|x)$



Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters λ, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - **Question:** what other "universal approximator" can we use? Deep neural networks → VAE

- ELBO to be maximized:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log p(\boldsymbol{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log q(\mathbf{z}|\lambda)]$$

$= -F(q, \theta)$

GD

- Want to compute the gradient w.r.t variational parameters λ

[Ranganath et al.,14]

The General Problem: Computing Gradients of Expectations

- When the objective function $\mathcal{L}$ is defined as an expectation of a (differentiable) test function $f_\lambda(\boldsymbol{z})$ w.r.t. a probability distribution $q_\lambda(\boldsymbol{z})$

$$\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$$

$E_{q_\lambda(z)}[g(z)]$

- Computing exact gradients w.r.t. the parameters λ is often infeasible
- Need stochastic gradient estimates
  - The score function estimator (a.k.a log-derivative trick, REINFORCE)
  - The reparameterization trick (a.k.a the pathwise gradient estimator)

RL.

score-based GM
diffusion models.

VAE

GAN

Computing Gradients of Expectations w/ score function

$$\nabla_\lambda \log q_\lambda = \frac{1}{q_\lambda} \nabla_\lambda q_\lambda$$

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(z)}[f_\lambda(z)]$
- Log-derivative trick: $\nabla_\lambda q_\lambda = q_\lambda \nabla_\lambda \log q_\lambda$
- **Question:** show that the gradient of $\mathcal{L}$ w.r.t. λ is:

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(z)}[f_\lambda(z)\nabla_\lambda \log q_\lambda(z) + \nabla_\lambda f_\lambda(z)]$$

$$\nabla_\lambda \mathbb{E}_{q_\lambda}[f_\lambda] = \nabla_\lambda \int q_\lambda f_\lambda = \int_\lambda \nabla_\lambda q_\lambda \cdot f_\lambda + q_\lambda \cdot \nabla_\lambda f_\lambda$$

$$= \int_\lambda q_\lambda \cdot \nabla_\lambda \log q_\lambda \cdot f_\lambda + q_\lambda \cdot \nabla_\lambda f_\lambda$$

$$= \mathbb{E}_{q_\lambda}[\nabla_\lambda \log q_\lambda f_\lambda + \nabla_\lambda f_\lambda]$$

Computing Gradients of Expectations w/ score function

score-based GM

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$

$a = E_{p(y)}[g(y)]$

- Log-derivative trick: $\nabla_\lambda q_\lambda = q_\lambda \nabla_\lambda \log q_\lambda$
- Gradient of $\mathcal{L}$ w.r.t. λ:

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \nabla_\lambda \log q_\lambda(\boldsymbol{z}) + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

$y_i \sim p(y)$

  - score function: the gradient of the log of a probability distribution

$a \approx \frac{1}{n} \sum_i g(y_i)$

- Monte Carlo estimation of the expectation:
  - Compute noisy unbiased gradients with Monte Carlo samples from q_λ

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} f_\lambda(\boldsymbol{z}_s) \nabla_\lambda \log q_\lambda(\boldsymbol{z}_s) + \nabla_\lambda f_\lambda(\boldsymbol{z}_s) \qquad \text{where } \boldsymbol{z}_s \sim q_\lambda(\boldsymbol{z})$$

Computing Gradients of Expectations w/ score function

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$

- Log-derivative trick: $\nabla_\lambda q_\lambda = q_\lambda \nabla_\lambda \log q_\lambda$
- Gradient of $\mathcal{L}$ w.r.t. λ:

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \underline{\nabla_\lambda \log q_\lambda(\boldsymbol{z})} + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - score function: the gradient of the log of a probability distribution
- Monte Carlo estimation of the expectation:
  - Compute noisy unbiased gradients with Monte Carlo samples from q_λ

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} f_\lambda(\boldsymbol{z}_s) \nabla_\lambda \log q_\lambda(\boldsymbol{z}_s) + \nabla_\lambda f_\lambda(\boldsymbol{z}_s) \qquad \text{where } \boldsymbol{z}_s \sim q_\lambda(\boldsymbol{z})$$

- Pros: generally applicable to any distribution $q(z|\lambda)$
- Cons: empirically has high variance → slow convergence

Computing Gradients of Expectations w/ reparametrization trick

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$ $\mathbb{E}_{z \sim q_\lambda(z)}[f_\lambda(z)]$

- Assume that we can express the distribution $q_\lambda(\boldsymbol{z})$ with a transformation

$$\begin{aligned} \epsilon &\sim s(\epsilon) \\ z &= t(\epsilon, \lambda) \end{aligned} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

  - E.g.,

$$\begin{aligned} \epsilon &\sim Normal(0,1) \\ z &= \epsilon\sigma + \mu \end{aligned} \quad \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Reparameterization gradient:

$$\mathcal{L} = \mathbb{E}_{\epsilon \sim s(\epsilon)}[f_\lambda(\boldsymbol{z}(\boldsymbol{\epsilon}, \lambda))]$$

  - **Question:** what's the gradient of $\mathcal{L}$ w.r.t. λ ?

Computing Gradients of Expectations w/ reparametrization trick

$\nabla_\lambda \mathcal{L} \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_z f_\lambda(z) \nabla_\lambda t(\epsilon_i, \lambda)$

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
- Assume that we can express the distribution $q_\lambda(\boldsymbol{z})$ with a transformation

$$\begin{array}{l} \epsilon \sim s(\epsilon) \\ z = t(\epsilon, \lambda) \end{array} \Leftrightarrow \quad z \sim q(z|\lambda)$$

  - E.g.,

$$\begin{array}{l} \epsilon \sim Normal(0,1) \\ z = \epsilon\sigma + \mu \end{array} \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\epsilon \sim s(\epsilon)}[\nabla_\lambda f_\lambda(z_\lambda)]$

$= \mathbb{E}_{\epsilon \sim s(\epsilon)}[\nabla_z f_\lambda \nabla_\lambda z_\lambda]$

- Reparameterization gradient:

$$\mathcal{L} = \mathbb{E}_{\epsilon \sim s(\epsilon)}[f_\lambda(\boldsymbol{z}(\boldsymbol{\epsilon}, \lambda))]$$

  - **Question:** what's the gradient of $\mathcal{L}$ w.r.t. λ ?

$\epsilon_i \sim s(\epsilon)$

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\epsilon \sim s(\epsilon)}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda)] \quad ✓$$

Computing Gradients of Expectations w/ reparametrization trick

- Loss: $\mathcal{L} = \mathbb{E}_{q_{\lambda}(\boldsymbol{z})}[f_{\lambda}(\boldsymbol{z})]$

- Assume that we can express the distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z})$ with a transformation

$$\begin{array}{l} \epsilon \sim s(\epsilon) \\ z = t(\epsilon, \lambda) \end{array} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

  - E.g.,

$$\begin{array}{l} \epsilon \sim Normal(0, 1) \\ z = \epsilon\sigma + \mu \end{array} \quad \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Reparameterization gradient

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim s(\boldsymbol{\epsilon})}[f_{\lambda}(\boldsymbol{z}(\boldsymbol{\epsilon}, \lambda))]$$

$$\nabla_{\lambda}\mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_{\lambda}(\boldsymbol{z})\, \nabla_{\lambda} t(\epsilon, \lambda)]$$

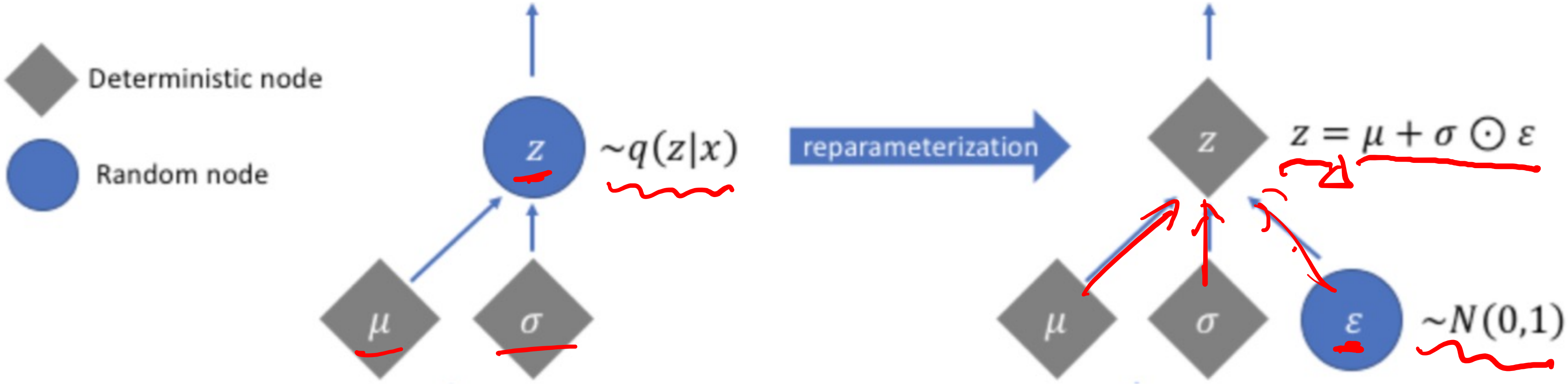
VAE → 2014

- Pros: empirically, lower variance of the gradient estimate
- Cons: Not all distributions can be reparameterized

Reparameterization trick

- Reparametrizing Gaussian distribution

$$\begin{aligned} \epsilon &\sim Normal(0,1) \\ z &= \epsilon\sigma + \mu \end{aligned} \iff z \sim Normal(\mu, \sigma^2)$$



[Courtesy: Tansey, 2016]

Reparameterization trick

- Reparametrizing Gaussian distribution

$$\begin{aligned} \epsilon &\sim Normal(0,1) \\ z &= \epsilon\sigma + \mu \end{aligned} \quad \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Other reparameterizable distributions:
  $$\begin{aligned} \epsilon &\sim Uniform(\epsilon) \\ z &= F^{-1}(\epsilon) \end{aligned} \quad \Leftrightarrow \quad z \sim q(z)$$
  - Tractable inverse CDF F^{-1}:
    - Exponential, Cauchy, Logistic, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel, Erlang
  - Location-scale:
    - Laplace, Elliptical, Student's t, Logistic, Uniform, Triangular, Gaussian
  - Composition:
    - Log-Normal (exponentiated normal) Gamma (sum of exponentials) Dirichlet (sum of Gammas) Beta, Chi-Squared, F

multinomial distribution

[Courtesy: Tansey, 2016]

Computing Gradients of Expectations: Summary

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
- Score gradient

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \nabla_\lambda \log q_\lambda(\boldsymbol{z}) + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - Pros: generally applicable to any distribution $q(z|\lambda)$
  - Cons: empirically has high variance → slow convergence

- Reparameterization gradient

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda)]$$

  - Pros: empirically, lower variance of the gradient estimate
  - Cons: Not all distributions can be reparameterized

Recall: Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters λ, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - Deep neural networks

$$\mathcal{L}(\lambda) \triangleq \mathrm{E}_{q_\lambda(z)}[\log p(x, z) - \log q(z)]$$

- ELBO to be maximized:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Want to compute the gradient w.r.t variational parameters λ

[Ranganath et al.,14]

BBVI with the score gradient

$$\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$$
$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})\nabla_\lambda \log q_\lambda(\boldsymbol{z}) + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- **Question:** what's the score gradient w.r.t. λ ?

$$\nabla_\lambda \mathcal{L} = \mathrm{E}_q[\nabla_\lambda \log q(z|\lambda)(\log p(x,z) - \log q(z|\lambda))]$$

- Compute noisy unbiased gradients of the ELBO with Monte Carlo samples from the variational distribution

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda \log q(z_s|\lambda)(\log p(x,z_s) - \log q(z_s|\lambda)),$$

$$\text{where } z_s \sim q(z|\lambda).$$

[Ranganath et al.,14]

BBVI with the reparameterization gradient

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- **Question:** what's the reparamerization gradient w.r.t. λ ?

$$\begin{array}{l} \epsilon \sim s(\epsilon) \\ z = t(\epsilon, \lambda) \end{array} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

$$\nabla_\lambda \mathcal{L} = \mathrm{E}_{\epsilon \sim s(\epsilon)}[\nabla_z [\log p(x, z) - \log q(z)] \nabla_\lambda t(\epsilon, \lambda)]$$

$$\boxed{\begin{array}{c} \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})] \\ \nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z}) \nabla_\lambda t(\epsilon, \lambda)] \end{array}}$$

Questions?