DSC291: Machine Learning with Few Labels

Unsupervised Learning

Zhiting Hu

Lecture 16, May 8, 2024

UC San Diego

HALICIOĞLU DATA SCIENCE INSTITUTE

Recap: EM Algorithm

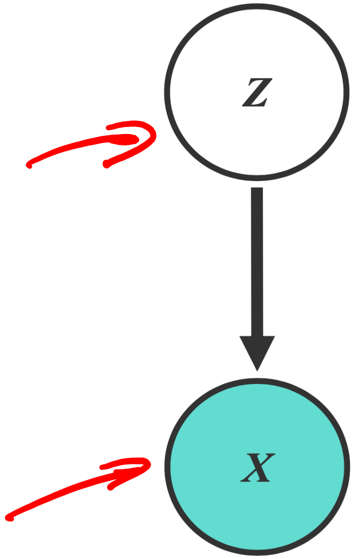
- The EM algorithm is $\underline{\text{coordinate-decent}}$ on $F(q, \theta)$
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right) = p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$
  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right) = \text{argmax}_\theta \sum_z q^{t+1}(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$

$$\underline{\ell(\theta; \boldsymbol{x})} = \underbrace{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right]}_{\text{ELBO}} + \text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

✓

$$= \underline{-F(q, \theta)} + \underline{\text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)}$$

Recap: Gaussian Mixture Models (GMMs)



- Consider a mixture of K Gaussian components:
  - Z is a latent class indicator vector:

$$p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

  - X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n \mid z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2}|\Sigma_k|^{1/2}} \exp\left\{-\tfrac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\right\}$$

  - The likelihood of a sample:

mixture component

mixture proportion

$$\begin{aligned} p(x_n|\mu,\Sigma) &= \sum_k p(z^k = 1 \mid \pi)\, p(x,\mid z^k = 1, \mu, \Sigma) \\ &= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k}\right) = \sum_k \pi_k N(x,\mid \mu_k, \Sigma_k) \end{aligned}$$

Recap: Gaussian Mixture Models (GMMs)

- E-step: computing the posterior of z_n given the current estimate of the parameters (i.e., π , μ, Σ)

Bayes' theory

$$\underline{p(z^k = 1 \mid \boldsymbol{x})} = \frac{p(z^k = 1)p(\boldsymbol{x} \mid z^k = 1)}{p(\boldsymbol{x})}$$

$$q(z|x) = p(z|x)$$

$$= \frac{p(z^k = 1)p(\boldsymbol{x} \mid z^k = 1)}{\sum_{j=1}^{K} p(z^j = 1)p(\boldsymbol{x} \mid z^j = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\boldsymbol{x} \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x} \mid \mu_j, \Sigma_j)}$$

$$:= \gamma_k$$

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

Recap: EM Algorithm

- The EM algorithm is coordinate-decent on $F(q, \theta)$
  - E-step: $\underline{q^{t+1}} = \arg\min_q F\left(q, \theta^t\right) = \underline{p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)}$
  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right) = \operatorname{argmax}_\theta \sum_z q^{t+1}(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$

$$\boxed{\begin{aligned} \ell(\theta; \boldsymbol{x}) &= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big) \\ &= -F(q, \theta) + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big) \end{aligned}}$$

- Limitation: need to be able to compute $\underline{p(\boldsymbol{z}|\boldsymbol{x}, \theta)}$, not possible for more complicated models --- solution: Variational inference

Variational Inference

Content adapted from CMU 10-708 Spring 2017

Inference

$p_\theta(x,z)$

training / learning : θ

θ

- Given a model, the goals of inference can include:
  - Computing the likelihood of observed data $p(\boldsymbol{x}^*)$
  - Computing the marginal distribution over a given subset of variables in the model $p(\boldsymbol{x}_A)$

    $p_\theta(x) = \int_z p_\theta(x,z)\,dz$
  - Computing the conditional distribution over a subsets of nodes given a disjoint subset of nodes $p(\boldsymbol{x}_A|\boldsymbol{x}_B)$

    $p_\theta(z|x)$ posterior.
  - Computing a mode of the density (for the above distributions) $\operatorname{argmax}_{\boldsymbol{x}} p(\boldsymbol{x})$
  - ....

EM: $\left\{\begin{array}{l} E:\ \theta^t \to p(z|x,\theta^t) \\ M:\ \theta^t \to \theta^{t+1} \end{array}\right.$

Variational Inference

- Observed variables $\boldsymbol{x}$, latent variables $\boldsymbol{z}$
- Variational (Bayesian) inference, a.k.a. **variational Bayes**, is most often used to approximately infer the conditional distribution over the latent variables given the observations (and parameters)
  - i.e., the **posterior distribution** over the latent variables

approximate inference

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

exact

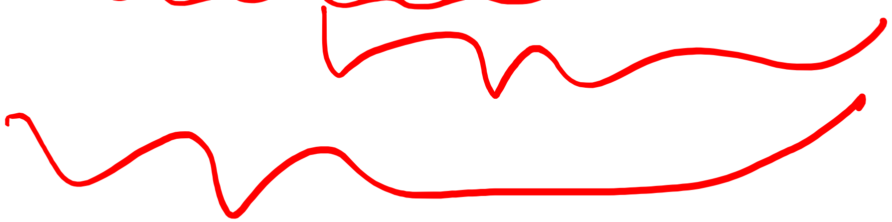
$\approx$ . . . .

Motivating Example

- Why do we often need to use an approximate inference methods (such as variational Bayes) to compute the posterior distribution?

- It's because we cannot directly compute the posterior distribution for many interesting models
  - I.e. the posterior density is in an intractable form (often involving integrals) which cannot be easily analytically solved.

- As a motivating example, we will try to compute the posterior for a (Bayesian) mixture of Gaussians.



Bayesian mixture of Gaussians

K

- The mean μ_k is treated as a (latent) random variable

$$\mu_k \sim \mathcal{N}(0, \tau^2) \text{ for } k = 1, \dots, K$$

$\{x\}$

- For each data $i = 1, \dots, n$

$$z_i \sim \mathrm{Cat}(\pi).$$

$$x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2).$$

- We have
  - observed variables $x_{1:n}$
  - latent variables $\mu_{1:k}$ and $z_{1:n}$
  - parameters $\{\tau^2, \pi, \sigma^2\}$

$$\prod_{k=1}^{K} p(\mu_k | \tau^2) \prod_{i=1}^{n} p(z_i | \pi) p(x_i | z_i, \mu_{1:K}$$

- $p(x_{1:n}, z_{1:n}, \mu_{1:k} | \tau^2, \pi, \sigma^2) = \prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i | z_i, \mu_{1:K}) , \sigma^2)$

Bayesian mixture of Gaussians

Bayes' rule

- We can write the posterior distribution as

$$p(\mu_{1:K}, z_{1:n} | x_{1:n}) = \frac{\prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i | z_i, \mu_{1:K})}{\underline{\int_{\mu_{1:K}} \sum_{z_{1:n}} \prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i | z_i, \mu_{1:K})}}$$

- The numerator can be computed for any choice of the latent variables
- The problem is the denominator (the marginal probability of the observations)
  - This integral cannot easily be computed analytically
- We need some approximation..

$$q(\mu, z | x) \cancel{=} p(\mu, z | x)$$

$$\approx$$

E :

$q^{t+1}(z|X) = \arg\min_{q} F(q, \theta^t)$

$= p(z|x)$

Variational Inference

The main idea behind variational inference:

- Choose a family of distributions over the latent variables $z_{1:m}$ with its own set of variational parameters ν, i.e.

$$q(z_{1:m}|\nu)$$

q

- Then, we find the setting of the parameters that makes our approximation q closest to the posterior distribution.
  - This is where optimization algorithms come in.

M-step

- Then we can use q with the fitted parameters in place of the posterior.
  - E.g. to form predictions about future data, or to investigate the posterior distribution over the hidden variables, find modes, etc.

“close” “distance”

q $p(z|x,\theta)$

Variational Inference

- We want to minimize the KL divergence between our approximation $q(\boldsymbol{z}|\boldsymbol{x})$ and our posterior $p(\boldsymbol{z}|\boldsymbol{x})$

$$\min_{q} \mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}))$$

s.t. q: $q(z|x)$

  - But we can't actually minimize this quantity w.r.t q because $p(\boldsymbol{z}|\boldsymbol{x})$ is unknown
- **Question:** how can we minimize the KL divergence?
  - Hint: recall what we did in EM:

ELBO

$$\underbrace{\ell(\theta;\boldsymbol{x})}_{\text{constant}} = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log\frac{p(\boldsymbol{x},\boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x},\theta)\big)$$

$$\mathrm{KL}(q(z|x)\,||\,p(z|x,\theta)) = \text{const.} - \text{ELBO}$$

Variational Inference

$$\cancel{p(z|x,\theta)} \Rightarrow \frac{p(x,z|\theta)\ \checkmark}{p(x|\theta)\ \times}$$

- We want to minimize the KL divergence between our approximation $q(\boldsymbol{z}|\boldsymbol{x})$ and our posterior $p(\boldsymbol{z}|\boldsymbol{x})$

$$\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x})\ ||\ p(\boldsymbol{z}|\boldsymbol{x}))$$

  - ○ But we can't actually minimize this quantity w.r.t q because $p(\boldsymbol{z}|\boldsymbol{x})$ is unknown

- **Question:** how can we minimize the KL divergence?

$$\ell(\theta;\boldsymbol{x}) = \boxed{\mathbb{E}_{\underline{q(\boldsymbol{z}|\boldsymbol{x})}}\left[\log \frac{\underline{p(\boldsymbol{x},\boldsymbol{z}|\theta)}\ \checkmark}{\underline{q(\boldsymbol{z}|\boldsymbol{x})}}\right]} + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x})\ ||\ \underline{p(\boldsymbol{z}|\boldsymbol{x},\theta)}\big)$$

Evidence Lower Bound (ELBO)

- The ELBO is equal to the negative KL divergence up to a constant $\ell(\theta;\boldsymbol{x})$
- We maximize the ELBO over q to find an "optimal approximation" to $\underline{p(\boldsymbol{z}|\boldsymbol{x})}$

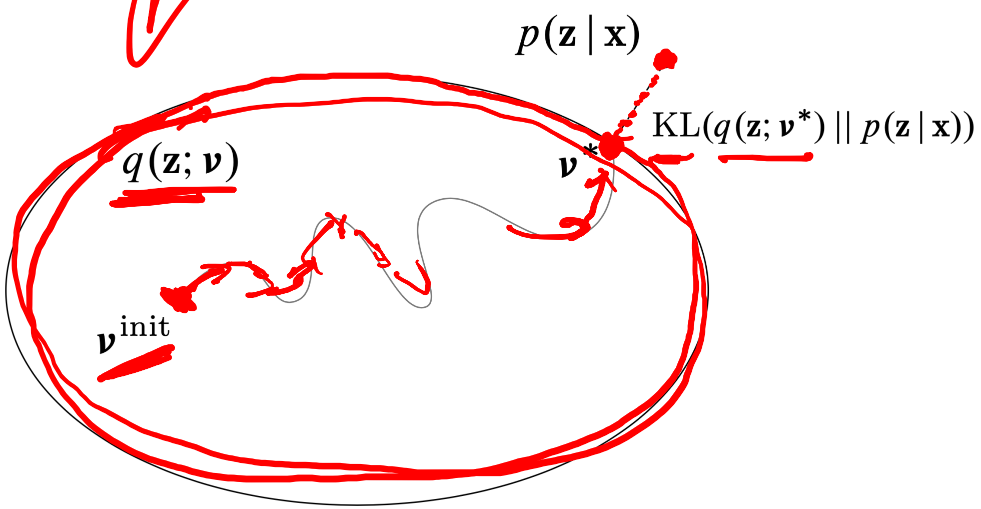
Variational Inference

- Choose a family of distributions over the latent variables $\boldsymbol{z}$ with its own set of variational parameters ν , i.e. $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})$
- We maximize the ELBO over q to find an "optimal approximation" to $p(\boldsymbol{z}|\boldsymbol{x})$

$$\text{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\left[\log\frac{p(\boldsymbol{x},\boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\right]$$

$H(q)$

$$= \text{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\log p(\boldsymbol{x},\boldsymbol{z}|\theta)] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\,\log q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})\,]$$

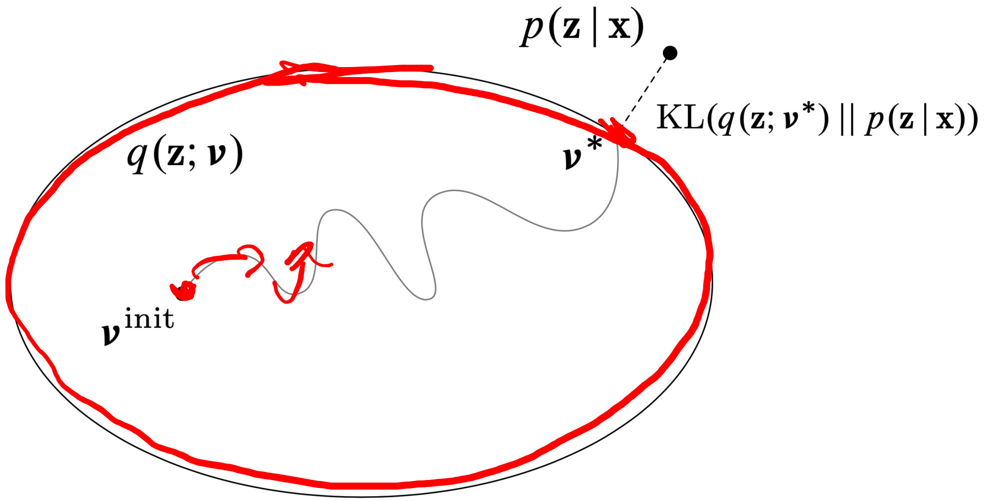

$p(\mathbf{z}\,|\,\mathbf{x})$
$\mathrm{KL}(q(\mathbf{z};\boldsymbol{\nu}^*)\,||\,p(\mathbf{z}\,|\,\mathbf{x}))$
$\boldsymbol{\nu}^*$
$q(\mathbf{z};\boldsymbol{\nu})$
$\boldsymbol{\nu}^{\text{init}}$

Variational Inference

- Choose a family of distributions over the latent variables $\boldsymbol{z}$ with its own set of variational parameters ν , i.e. $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{v})$
- We maximize the ELBO over q to find an "optimal approximation" to $p(\boldsymbol{z}|\boldsymbol{x})$

$$\text{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\right]$$

$$= \text{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\log p(\boldsymbol{x},\boldsymbol{z}|\theta)] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\log q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})]$$


$p(\mathbf{z}\,|\,\mathbf{x})$
$\mathrm{KL}(q(\mathbf{z};\boldsymbol{\nu}^*)\,||\,p(\mathbf{z}\,|\,\mathbf{x}))$
$q(\mathbf{z};\boldsymbol{\nu})$
$\boldsymbol{\nu}^*$
$\boldsymbol{\nu}^{\text{init}}$

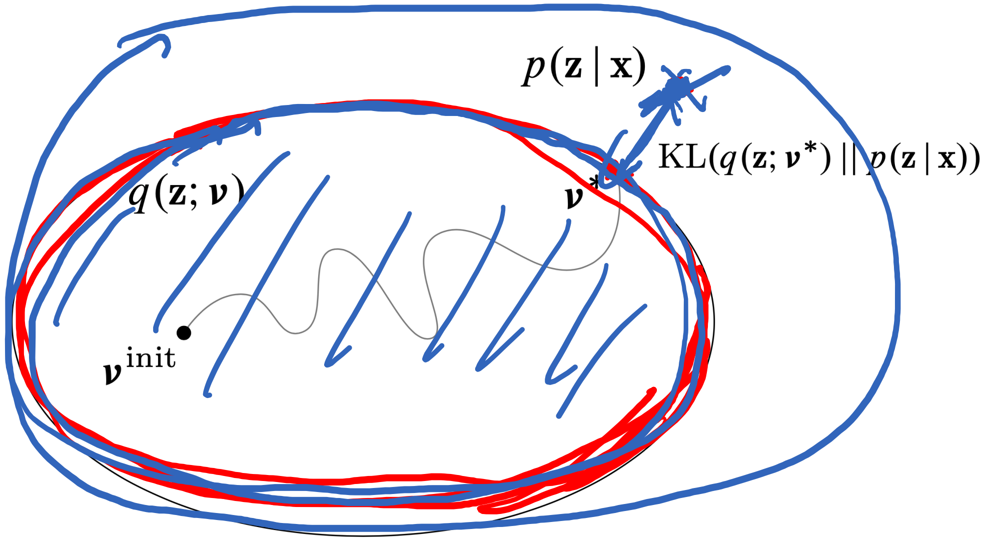
Question: How do we choose the variational family $\underline{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{v})}$?

Variational Inference

- Choose a family of distributions over the latent variables $\boldsymbol{z}$ with its own set of variational parameters ν , i.e. $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{v})$
- We maximize the ELBO over q to find an "optimal approximation" to $p(\boldsymbol{z}|\boldsymbol{x})$

$$\mathrm{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\left[\log\frac{p(\boldsymbol{x},\boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}\right]$$

$$= \mathrm{argmax}_{\nu}\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\log p(\boldsymbol{x},\boldsymbol{z}|\theta)] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})}[\ \log q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\nu})\ ]$$


$p(\mathbf{z}\,|\,\mathbf{x})$
$\mathrm{KL}(q(\mathbf{z};\boldsymbol{\nu}^*)\,||\,p(\mathbf{z}\,|\,\mathbf{x}))$
$q(\mathbf{z};\boldsymbol{\nu})$
$\boldsymbol{\nu}^*$
$\boldsymbol{\nu}^{\text{init}}$

Question: How do we choose the variational family $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{v})$?

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

2014

Example: Mean Field Variational Inference

- A popular family of variational approximations

- In this type of variational inference, we assume the variational distribution over the latent variables factorizes as

$$q(\boldsymbol{z}) = q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j)$$

  - (where we omit variational parameters for ease of notation)
  - We refer to $q(z_j)$, the variational approximation for a single latent variable, as a "local variational approximation"

- In the above expression, the variational approximation $q(z_j)$ over each latent variable z_j is independent

Example: Mean Field Variational Inference

- Typically, this approximation does not contain the true posterior (because the latent variables are dependent).
  - E.g.: in the (Bayesian) mixture of Gaussians model, all of the cluster assignments z_i for $i = 1, \ldots, n$ are dependent on each other and on the cluster locations $\mu_{1:K}$ given data.

Example: Mean Field Variational Inference

How do we optimize the ELBO in mean field variational inference?

- Typically, we use coordinate ascent optimization.
- I.e. we optimize each latent variable's variational approximation $q(z_j)$ in turn while holding the others fixed.
  - At each iteration we get an updated "local" variational approximation.
  - And we iterate through each latent variable until convergence.

Mean Field Variational Inference with Coordinate Ascent

Recap: Bayesian mixture of Gaussians

- Treat the mean μ_k as latent variables

$$\mu_k \sim \mathcal{N}(0, \tau^2) \text{ for } k = 1, \ldots, K$$

- For each data $i = 1, \ldots, n$

$$z_i \sim \mathrm{Cat}(\pi).$$

$$x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2).$$

- We have
  - observed variables $x_{1:n}$
  - latent variables $\mu_{1:k}$ and $z_{1:n}$
  - parameters $\{\tau^2, \sigma^2, \pi\}$

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 $q(z_i)$          (Variational) E-step
    ○ End for
    ○ Update the global variational distributions $q(\mu_k)$
    ○ Update the parameters $\{\tau^2, \sigma^2, \pi\}$                 → (Variational) M-step
• Until ELBO converges
```

- 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

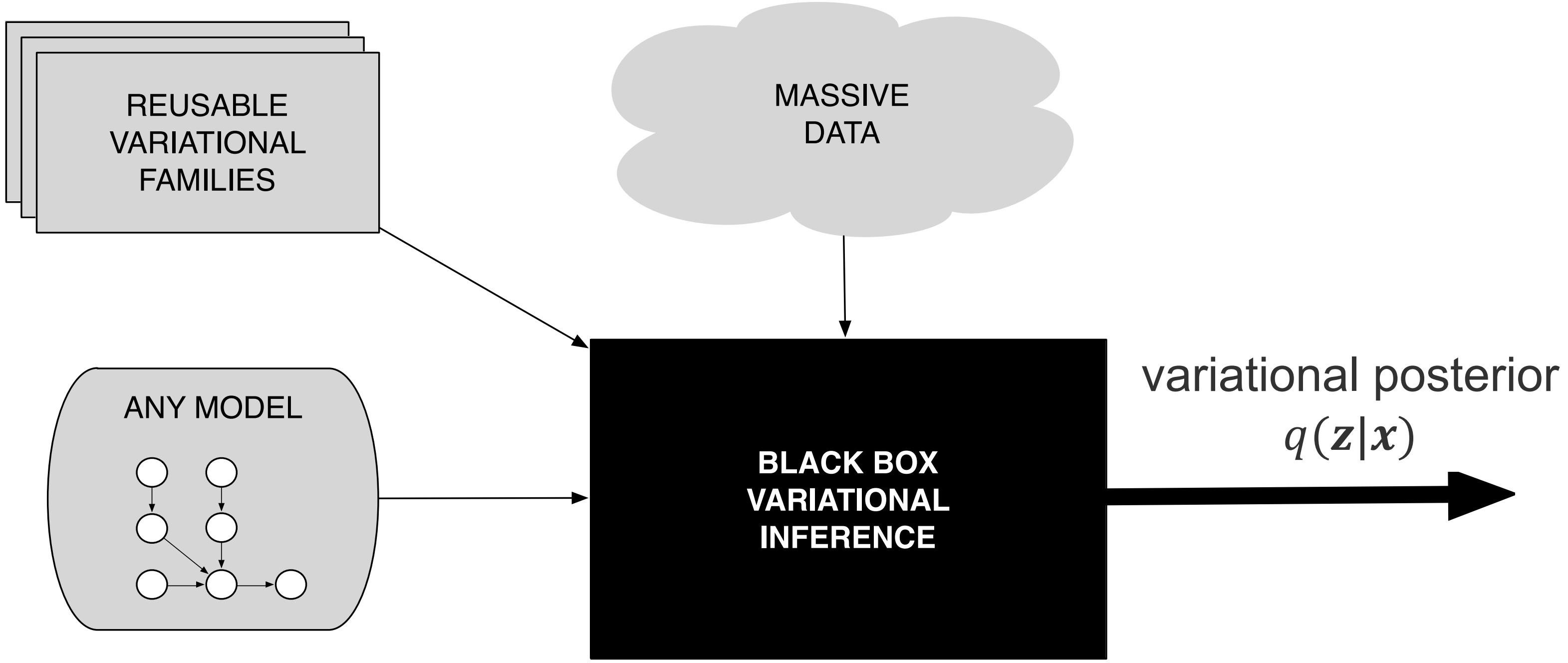
[Hoffman et al., Stochastic Variational Inference, 2013]

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)

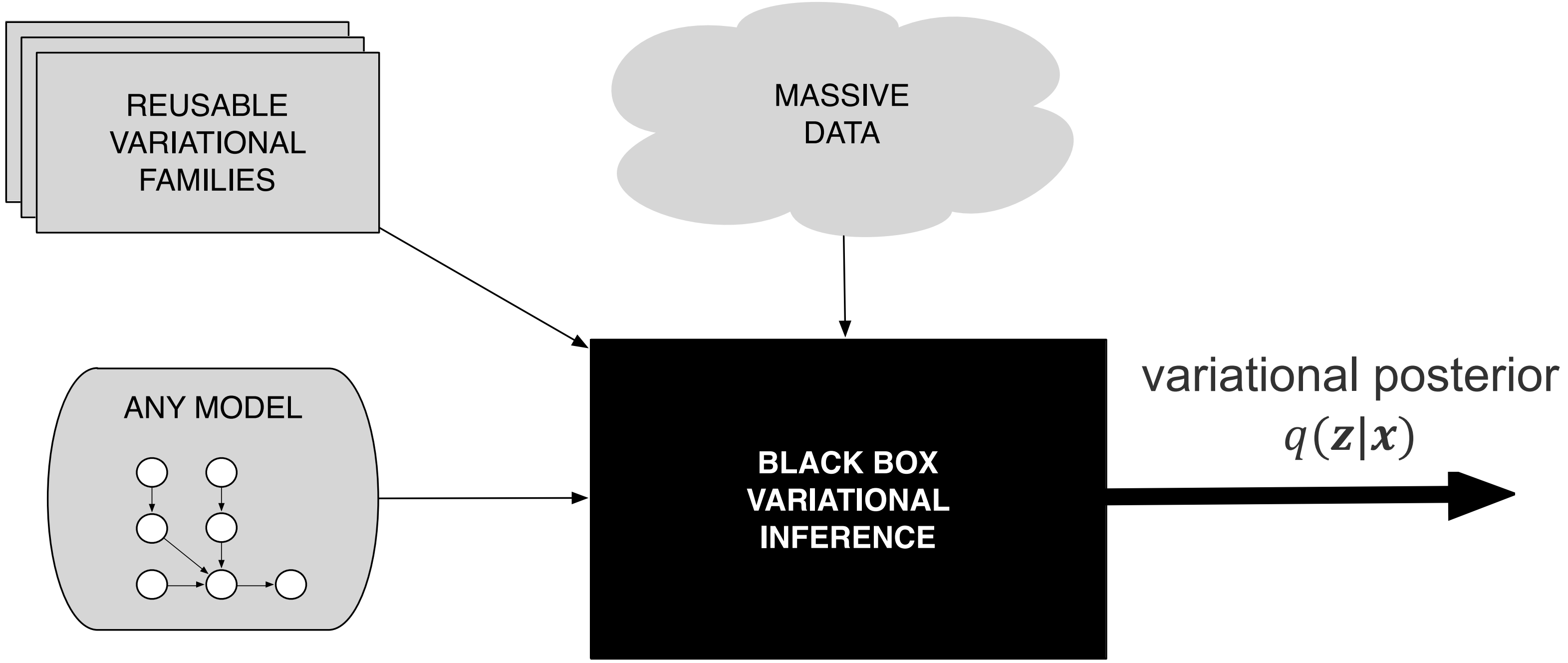


REUSABLE VARIATIONAL FAMILIES
MASSIVE DATA
ANY MODEL
BLACK BOX VARIATIONAL INFERENCE
variational posterior
$q(\boldsymbol{z}|\boldsymbol{x})$

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

(Courtesy: Blei et al., 2018)

Black-box Variational Inference (BBVI)


REUSABLE VARIATIONAL FAMILIES
MASSIVE DATA
ANY MODEL
BLACK BOX VARIATIONAL INFERENCE
variational posterior $q(\boldsymbol{z}|\boldsymbol{x})$

- Sample from $q(.)$
- Form noisy gradients (without model-specific computation)
- Use stochastic optimization

(Courtesy: Blei et al., 2018)

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
- ELBO:

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

- 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 $q_\lambda(\boldsymbol{z})$

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

- Computing exact gradients w.r.t. the parameters λ is often unfeasible
- 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)

Computing Gradients of Expectations w/ score function

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

- Log-derivative trick: $\nabla_\lambda q_\lambda = q_\lambda \nabla_\lambda \log q_\lambda$
- Gradient w.r.t. λ:

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

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

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

- Pros: generally applicable to any distribution $q(z|\lambda)$
- Cons: empirically has high variance → slow convergence
  - To reduce variance: Rao-Blackwellization, control variates, importance sampling, ...

Computing Gradients of Expectations w/ reparametrization trick

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

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

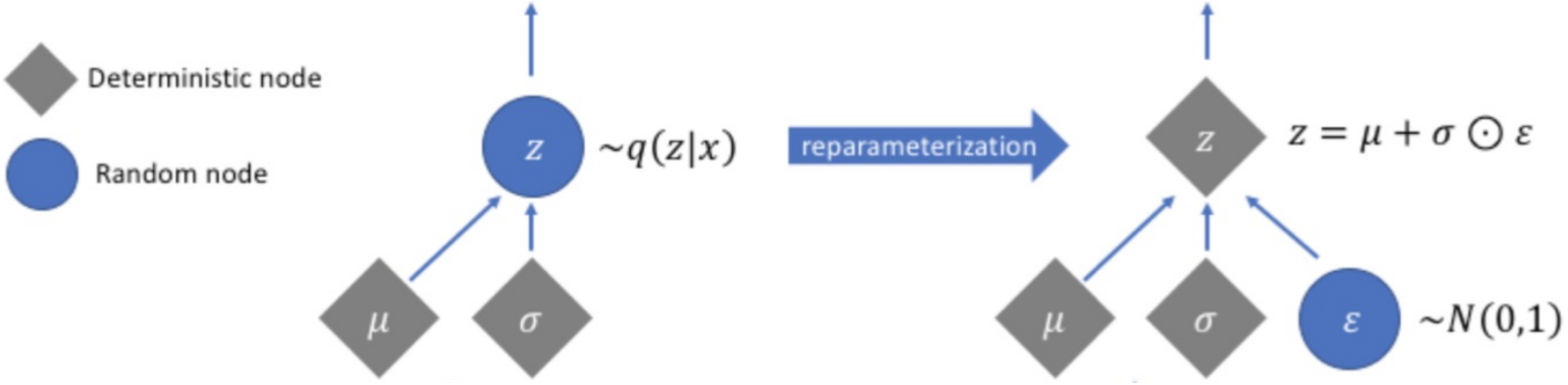
$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda)]$$

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

Reparameterization trick

- Reparametrizing Gaussian distribution

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



[Courtesy: Tansey, 2016]

Reparameterization trick

- Reparametrizing Gaussian distribution

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

- Other reparameterizable distributions:
  - 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

$$\begin{array}{l} \epsilon \sim Uniform(\epsilon) \\ z = F^{-1}(\epsilon) \end{array} \Leftrightarrow \quad z \sim q(z)$$

[Courtesy: Tansey, 2016]

Computing Gradients of Expectations: Summary

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

- Score gradient

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

  - 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:

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

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

[Ranganath et al.,14]

BBVI with the score gradient

- ELBO:

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

- Gradient w.r.t. λ (using the log-derivative trick)

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

- 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:

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

- Gradient w.r.t. λ

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

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

Questions?